AMD

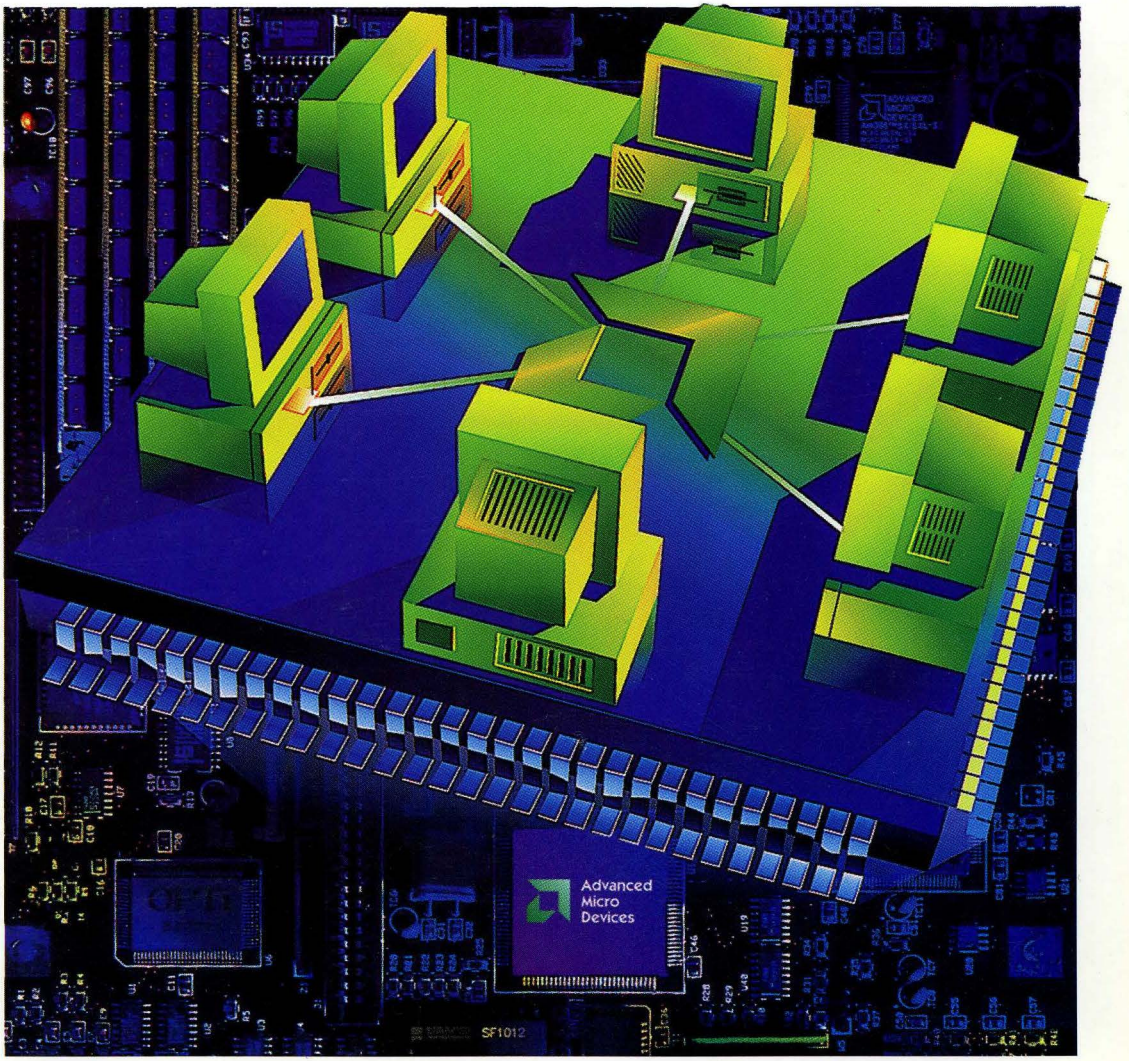
Ethernet/IEEE 802.3 Family

1994

Ethernet/IEEE 802.3 Family

1994 World Network Data Book/Handbook

Advanced
Micro
Devices



Ethernet/IEEE 802.3 Family

World Network Data Book/Handbook

1994

ADVANCED MICRO DEVICES

Trademarks

TABLE OF CONTENTS

INTRODUCTION

Advanced Micro Devices was the first company in the industry to offer a complete 802.3/Ethernet chip set in 1985. Today, AMD is a leading supplier of integrated circuits to the local area network and the wide area marketplace. Our total portfolio includes products for 802.3/Ethernet, Fiber Distributed Data Interface (FDDI) and ISDN applications. This rich mix of products reflects AMD's commitment to your needs and insures leadership in this exciting marketplace.

This handbook includes a complete offering of solutions for the systems architect/designer of 802.3/Ethernet local area network (LAN) applications.

AMD Value Proposition

AMD provides products that accelerate your products' time-to-market.

Our products are supported with software and board level solutions to accelerate the design cycle. A great emphasis is placed in standards compliance, interoperability testing and systems verification of our integrated circuits.

Many of AMD's products result from joint development programs with premier networking systems corporations. This ensures optimal product definition and system verification. Examples of joint development efforts include Digital Equipment Corporation, Hewlett-Packard Corporation, 3COM Corporation, and SynOptics Communications, Inc.

Ethernet Media Access Controllers

AMD is the leading supplier of 802.3/Ethernet ICs. This market includes engineering workstations, personal computer platforms and embedded applications. AMD offers solutions for both 16-bit and 32-bit microprocessor busses. The industry's most widely designed Ethernet controller, the NMOS Am7990 LANCE, defined the industry preferred architecture for efficient software interface in high-performance applications. From the LANCE, the Am79C90 C-LANCE was developed, answering the need for Ethernet Controllers in low power applications. The Am79C940 Media Access Controller for Ethernet (MACE) is a 16-bit controller with a superior modular architecture and versatile system interface that allow it to be configured as a stand-alone device or as a connectivity cell incorporated into a larger, integrated system. The MACE is specifically designed to address applications where multiple I/O peripherals are present, and a centralized or system specific DMA is required.

Complementing the controller offering are a Manchester encoder/decoder (Am7992B) and several physical layer devices for either thick coax Ethernet/IEEE 802.3 (l0BASE5), thin coax Cheapernet/IEEE 802.3 (10BASE2), or twisted pair Ethernet/IEEE 802.3 (10BASE-T).

Medium Attachment Units

For medium attachment units (MAUs), also known as stand-alone transceivers, AMD offers three products. The original Ethernet/802.3 10BASE5 and 10BASE2 transceiver, the Am7996, is a proven industry solution used extensively in all markets. An evaluation board, the Am7996EVAL-HW, facilitates rapid design and production of Am7996 based MAUs.

The Am79C98 Twisted Pair Ethernet Transceiver (TPEX) and Am79C100 Twisted Pair Ethernet Transceiver Plus (TPEX+), are AMD's offerings for the 802.3 10BASE-T market. The Am79C98 and Am79C100 are highly Integrated devices that allow for a very cost effective LAN system implementation using 10BASE-T medium attachment units.

Highly Integrated Single-Chip Controllers

Since 1992, AMD has been offering a complete family of single-chip 802.3/Ethernet controllers that integrate a complete 802.3/Ethernet node into a single VLSI device. The PCnet™ family of highly-integrated bus-mastering controllers are binary code compatible. This family of 16-bit and 32-bit devices have glueless interfaces to some of the most popular platforms, including ISA (Am79C960/61), EISA (Am79C960/61), VL (Am79C965), and PCI (Am79C970), that help system designers to reduce board space requirements. In addition, AMD is the first to offer a single-chip 802.3/Ethernetcontroller that conforms to Microsoft's Plug and Play Specifications for ISA (Am79C961). These controllers' unique architecture and features also makes them well suited for internetworking systems and network peripherals.

Multiport Repeaters

Multiport repeaters, hubs and concentrators have been used in the industry for many years in coaxal cable networks. With the emergence of 10BASE-T and its structured cabling system or physical star configuration, the multiport repeater has become an essential part of a local area network. Without a l0BASE-T multiport repeater there is no 10BASE-T network.

The principal value of a 10BASE-T local area network is that it allows the network manager to build, reconfigure and maintain a larger and reliable network with a low cost of ownership. To improve reliability, reduce system cost, and allow for effective LAN management in a 10BASE-T multiport repeater implementation, AMD has introduced the Am79C981 Integrated Multiport Repeater Plus (IMR+). The Am79C987 Hardware Implemented Management Information Base™ (HIMIB™) provides repeater management functions, complying to all options detailed in the Layer Management for 10 Mb/s Baseband Repeaters (IEEE 802.3k) Standard. The HIMIB™ device is designed to be used in conjunction with AMD's IMR+ device. These devices allow the system designer to easily develop reliable, maintainable 10BASE-T multiport repeaters of various complexity and functionality. The ISA-HUB-KT is designed to serve as a repeater application example as well as an evaluation vehicle for the AMD IMR+ (Am79C981) and HIMIB™ (Am79C987) devices.

Evaluation Platforms

AMD also has design evaluation vehicles to assist system designers gain a detailed and thorough understanding of the inner working of our Ethernet chips. Evaluation kits are available for media access controllers, transceivers, multiport repeaters, and highly-integrated single chip controllers.

SECTION

1 PRODUCT DATA SHEETS

PRELIMINARY

Am79C90

CMOS Local Area Network Controller for Ethernet (C-LANCE)

Advanced Micro Devices

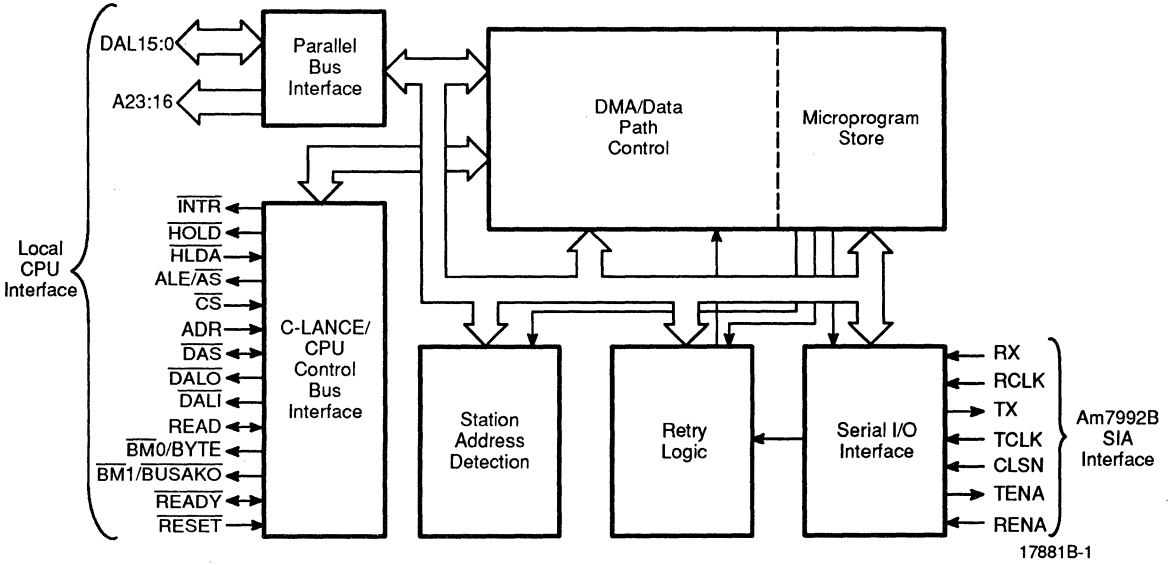
DISTINCTIVE CHARACTERISTICS

- **Compatible with Ethernet and IEEE 802.3 10BASE 5 Type A, and 10BASE 2 Type B, "Cheapernet," 10BASE-T**
- **Easily interfaced with 80x86, 680x0, Am29000®, Z8000™, LSI-II™ microprocessors**
- **On-board DMA and buffer management, 64-byte Receive, 48-byte Transmit FIFOs**
- **24-bit wide linear addressing (Bus Master Mode)**
- **Network and packet error reporting**
- **Back-to-back packet reception with as little as 0.5 μs interframe spacing**
- **Diagnostic Routines**
  - Internal/external loop back
  - CRC logic check
  - Time domain reflectometer
- **Low power consumption for power sensitive applications**
- **Completely software and hardware compatible to AMD's LANCE device (Am7990) (see Appendix B)**

GENERAL DESCRIPTION

The Am79C90 CMOS Local Area Network Controller for Ethernet (C-LANCE) is a 48-pin VLSI device designed to greatly simplify interfacing a microcomputer or minicomputer to an IEEE 802.3/Ethernet Local Area Network. The C-LANCE, in conjunction with the Am7992B Serial Interface Adapter (SIA), Am7996 or Am79C98 Transceiver, and closely coupled local memory and microprocessor, is intended to provide the user with a complete interface module for an Ethernet network. The Am79C90 is designed using a scalable CMOS technology and is compatible with a variety of microprocessors. On-board DMA, advanced buffer management, and extensive error reporting and diagnostics facilitate design and improve system performance.

BLOCK DIAGRAM



17881B-1

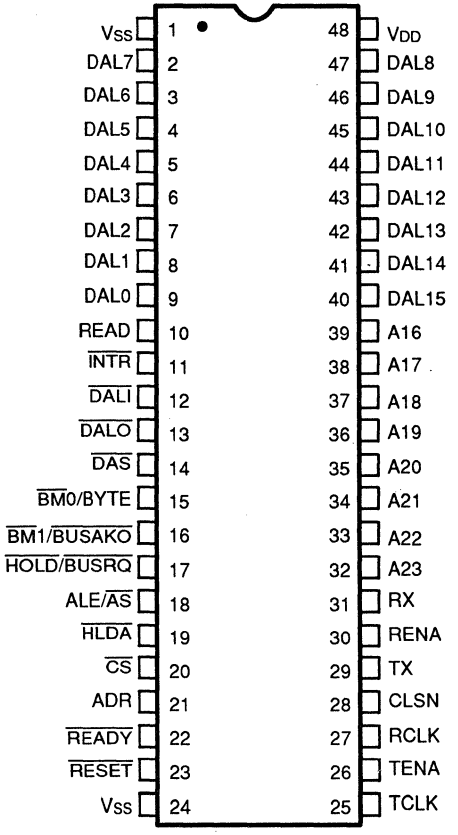
Publication# 17881 Rev. B Amendment/0
Issue Date: May 1994

RELATED AMD PRODUCTS

Part No.	Description
Am7996	IEEE 802.3/Ethernet/Cheapernet Tap Transceiver
Am79C100	Twisted-Pair Ethernet Transceiver Plus (TPEX+)
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA Single-Chip Ethernet Controller (with Microsoft® Plug n' Play support)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller (for 386DX, 486 and VL buses)
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems
Am79C98	Twisted-Pair Ethernet Transceiver (TPEX)
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)

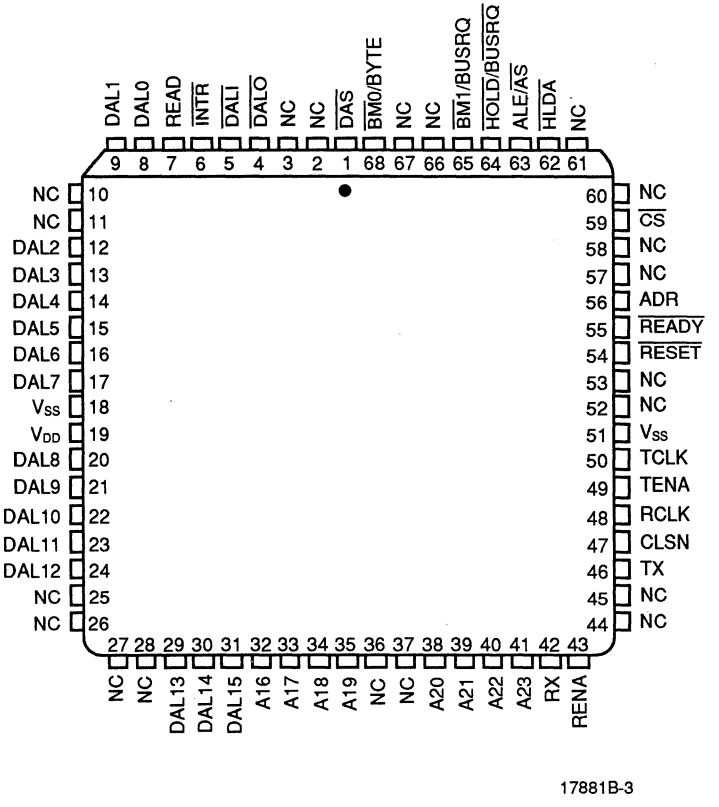
CONNECTION DIAGRAMS

DIP



17881B-2

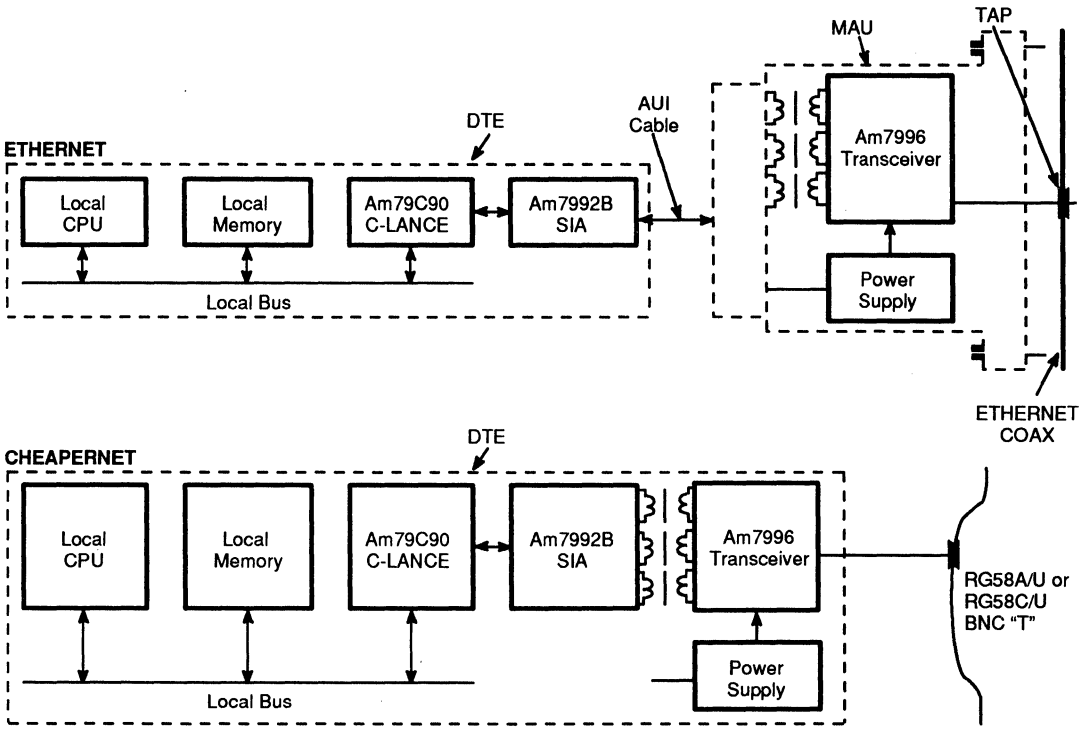
PLCC



17881B-3

Note:
Pin 1 is marked for orientation.

TYPICAL ETHERNET/CHEAPERNET NODE



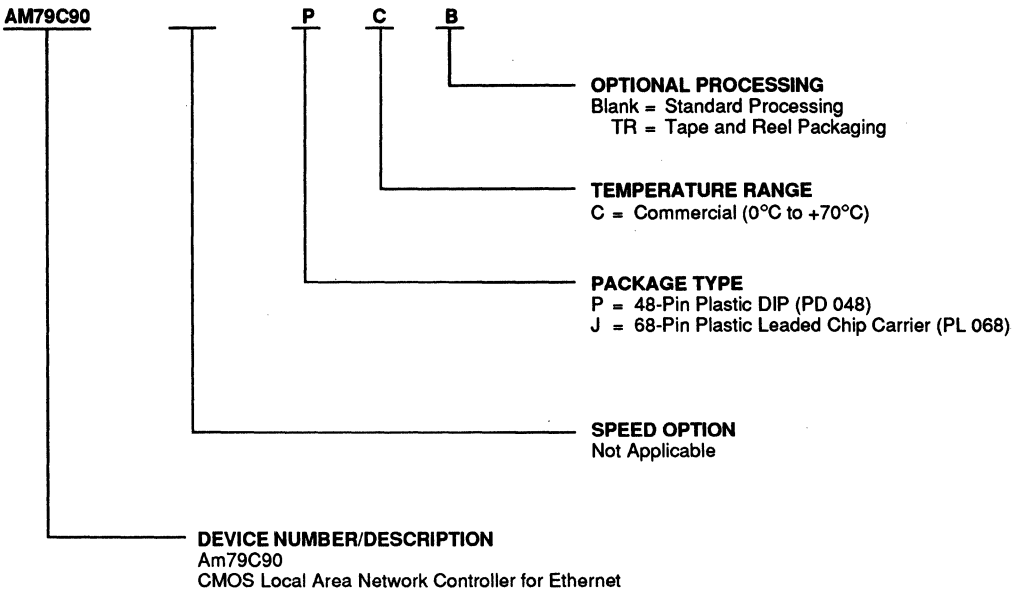
AUI – Attachment Unit Interface
DTE – Data Terminal Equipment
MAU – Medium Attachment Unit

17881B-4

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM79C90	PC, JC, JCTR

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

PIN DESCRIPTION

A16 – A23
High Order Address Bus (Output Three State)

Additional address bits to access a 24-bit address. These lines are driven as a Bus Master only.

ADR
Register Address Port Select (Input)

When the C-LANCE is a slave, ADR indicates which of the two register ports is selected. ADR LOW selects register data port; ADR HIGH selects register address port. ADR must be valid throughout the data portion of the bus cycle and is only used by the C-LANCE when $\overline{CS}$ is LOW.

ALE/$\overline{AS}$
Address Latch Enable (Output, Three-State)

Used to demultiplex the DAL lines and define the address portion of the bus cycle. This I/O pin is programmable through bit (01) of CSR3.

As ALE (CSR3 (01), ACON = 0), the signal transitions from a HIGH to a LOW during the address portion of the transfer and remains LOW during the data portion. ALE can be used by a Slave device to control a latch on the bus address lines. When ALE is HIGH, the latch is open, and when ALE goes LOW, the latch Is closed.

As $\overline{AS}$ (CSR3 (01), ACON = 1), the signal pulses LOW during the address portion of the bus transaction. The LOW-to-HIGH transition of $\overline{AS}$ can be used by a Slave device to strobe the address into a register.

The C-LANCE drives the ALE/$\overline{AS}$ line only as a Bus Master.

$\overline{BM0}$/BYTE, $\overline{BM1}$/$\overline{BUSAKO}$
(Output, Three-State)

The two pins are programmable through bit (00) of CSR3

$\overline{BM0}$, $\overline{BM1}$ – If CSR3 (00) BCON = 0
PIN 15 = $\overline{BM0}$ (Output Three-state) (48-Pin DIPs)
PIN 16 = $\overline{BM1}$ (Output Three-state) (48-Pin DIPs)

$\overline{BM0}$, $\overline{BM1}$ (Byte Mask). This indicates the byte(s) on the DAL are to be read or written during this bus transaction. The C-LANCE drives these lines only as a Bus Master. It ignores the Byte Mask lines when it is a Bus Slave and assumes word transfers.

Byte selection using Byte Mask is done as described by the following table:

$\overline{BM1}$	$\overline{BM0}$	Selection
LOW	LOW	Whole Word
LOW	HIGH	Upper Byte
HIGH	LOW	Lower Byte
HIGH	HIGH	None

BYTE, $\overline{BUSAKO}$ – If CSR3 (00) BCON = 1
PIN 15 = BYTE (Output Three-state) (48-Pin DIPs)
PIN 16 = $\overline{BUSAKO}$ (Output) (48-Pin DIPs)

Byte selection may also be done using the BYTE line and DAL00 line, latched during the address portion of the bus cycle. The C-LANCE drives BYTE only as a Bus Master and ignores it when a Bus Slave selection is done (similar to $\overline{BM0}$, $\overline{BM1}$). Byte selection is done as outlined in the following table:

BYTE	DAL00	Selection
LOW	LOW	Whole Word
LOW	HIGH	Illegal Condition
HIGH	LOW	Lower Byte
HIGH	HIGH	Upper Byte

$\overline{BUSAKO}$ is a bus request daisy chain output. If the chip is not requesting the bus and it receives $\overline{HLDA}$, $\overline{BUSAKO}$ will be driven LOW. If the C-LANCE is requesting the bus when it receives $\overline{HLDA}$, $\overline{BUSAKO}$ will remain HIGH.

Byte Swapping

In order to be compatible with the variety of 16-bit microprocessors available to the designer, the C-LANCE may be programmed to swap the position of the upper and lower order bytes on data involved in transfers with the internal FIFOs.

Byte swapping is done when BSWP = 1. The most significant byte of the word in this case will appear on DAL lines 7–0 and the least significant byte on DAL lines 15–8.

When BYTE = H (indicating a byte transfer) the table indicates on which part of the 16-bit data bus the actual data will appear.

Whenever byte swap is activated, the only data that is swapped is data traveling to and from the Transmit/Receive FIFOs.

Signal Line	Mode Bits	
	BSWP = 0 and BCON = 1	BSWP = 1 and BCON = 1
BYTE = L and DAL00 = L	Word	Word
BYTE = L and DAL00 = H	Illegal	Illegal
BYTE = H and DAL00 = H	Upper Byte	Lower Byte
BYTE = H and DAL00 = L	Lower Byte	Upper Byte

CLSN
Collision (Input)

A logical input that indicates that a collision is occurring on the channel.

$\overline{CS}$
Chip Select (Input)

Indicates, when asserted, that the C-LANCE is the slave device of the data transfer. $\overline{CS}$ must be valid throughout the data portion of the bus cycle. $\overline{CS}$ must not be asserted when $\overline{HLDA}$ is LOW.

DAL00 – DAL15
Data/Address Lines (Input/Output, Three-State)

The time multiplexed Address/Data bus. During the address portion of a memory transfer, DAL00 – DAL15 contains the lower 16 bits of the memory address. The upper 8 bits of address are contained in A16 – A23.

During the data portion of a memory transfer, DAL00 – DAL15 contains the read or write data, depending on the type of transfer.

The C-LANCE drives these lines as a Bus Master and as a Bus Slave.

$\overline{DALI}$
Data/Address Line In (Output, Three-State)

An external bus transceiver control line. $\overline{DALI}$ is asserted when the C-LANCE reads from the DAL lines. It will be LOW during the data portion of a READ transfer and remain HIGH for the entire transfer if it is a WRITE. $\overline{DALI}$ is driven only when C-LANCE is a Bus Master.

$\overline{DALO}$
Data/Address Line Out (Output, Three-State)

An external bus transceiver control line. $\overline{DALO}$ is asserted when the C-LANCE drives the DAL lines. $\overline{DALO}$ will be LOW only during the address portion if the transfer is a READ. It will be LOW for the entire transfer if the transfer is a WRITE. $\overline{DALO}$ is driven only when C-LANCE is a Bus Master.

$\overline{DAS}$
Data Strobe (Input/Output, Three-State)

Defines the data portion of the bus transaction. $\overline{DAS}$ is high during the address portion of a bus transaction and low during the data portion. The LOW-to-HIGH transition can be used by a Slave device to strobe bus data into a register. $\overline{DAS}$ is driven only as a Bus Master.

$\overline{HLDA}$
Bus Hold Acknowledge (Input)

A response to $\overline{HOLD}$. When $\overline{HLDA}$ is LOW in response to the chip's assertion of $\overline{HOLD}$, the chip is the Bus Master.

During bus master operation the C-LANCE waits for $\overline{HLDA}$ to be deasserted HIGH before reasserting $\overline{HOLD}$ LOW. This insures proper bus handshake under all situations.

$\overline{HOLD}$/$\overline{BUSRQ}$
Bus Hold Request (Output, Open Drain)

Asserted by the C-LANCE when it requires access to memory. $\overline{HOLD}$ is held LOW for the entire ensuing bus transaction. The function of this pin is programmed through bit (00) of CSR3. Bit (00) of CSR3 is cleared when $\overline{RESET}$ is asserted.

When CSR3 (00) BCON= 0

PIN 17 = $\overline{HOLD}$
(Output Open Drain and input sense) (48-Pin DIPs)

When CSR3 (00) BCON = 1

PIN 17 = $\overline{BUSRQ}$ (I/O Sense, Open Drain) (48-Pin DIPs)

If the C-LANCE wants to use the bus, it looks at $\overline{HOLD}$/$\overline{BUSRQ}$; if it is HIGH the C-LANCE can pull it LOW and request the bus. If it is already LOW, the C-LANCE waits for it to go inactive-HIGH before requesting the bus.

$\overline{INTR}$
Interrupt (Output, Open Drain)

An attention signal that indicates, when active, that one or more of the following CSR0 status flags is set: BABL, MERR, MISS, RINT, TINT or IDON. $\overline{INTR}$ is enabled by bit 06 of CSR0 (INEA = 1). $\overline{INTR}$ remains asserted until the source of Interrupt is removed.

RCLK
Receive Clock (Input)

A 10 MHz square wave synchronized to the Receive data and only active while receiving an Input Bit Stream.

READ
(Input/Output, Three-State)

Indicates the type of operation to be performed in the current bus cycle. This signal is an output when the C-LANCE is a Bus Master.

High – Data is taken off the DAL lines by the C-LANCE.

Low – Data is placed on the DAL lines by the C-LANCE.

The signal is an input when the C-LANCE is a Bus Slave.

High – Data is placed on the DAL lines by the C-LANCE.

Low – Data is taken off the DAL lines by the C-LANCE.

$\overline{\text{READY}}$
(Input/Output, Open Drain)

When the C-LANCE is a Bus Master, $\overline{\text{READY}}$ is an asynchronous acknowledgment from the bus memory that it will accept data in a WRITE cycle or that it has put data on the DAL lines in a READ cycle.

As a Bus Slave, the C-LANCE asserts $\overline{\text{READY}}$ when it has put data on the DAL lines during a READ cycle or is about to take data off the DAL lines during a write cycle. $\overline{\text{READY}}$ is a response to $\overline{\text{DAS}}$ and will return High after $\overline{\text{DAS}}$ has gone High. $\overline{\text{READY}}$ is an input when the C-LANCE is a Bus Master and an output when the C-LANCE is a Bus Slave.

RENA
Receive Enable (Input)

A logical input that indicates the presence of carrier on the channel.

$\overline{\text{RESET}}$
Reset (Input)

Reset causes the C-LANCE to cease operation, clear its internal logic, force all three-state buffers to the high impedance state, and enter an idle state with the stop bit of CSR0 set. It is recommended that a 3.3 kΩ pullup resistor be connected to this pin.

RX
Receive (Input)

Receive Input Bit Stream.

TCLK
Transmit Clock (Input)

10 MHz clock.

TENA
Transmit Enable (Output)

Transmit Output Bit Stream enable. When asserted, it enables valid transmit output (TX).

TX
Transmit (Output)

Transmit Output Bit Stream.

V_{DD}
Power Supply Pin +5 V ±5%

It is recommended that 0.1 µF and 10 µF decoupling capacitors be used between V_{DD} and V_{SS}.

V_{SS}
Ground

Pin 1 and 24 (48-Pin DIPs) should be connected together externally, as close to the chip as possible.

FUNCTIONAL DESCRIPTION

The parallel interface of the CMOS Local Area Network Controller for Ethernet (C-LANCE) has been designed to be "friendly" or easy to interface to a variety of popular microprocessors. These microprocessors include the Am29000, 80x86, 680x0, Z8000 and LSI-11. The C-LANCE has a 24-bit wide linear address space when it is in the Bus Master Mode. A programmable mode of operation allows byte addressing in one of two ways: a Byte/Word control signal compatible with the 80x86 and Z8000 or an Upper Data Strobe and Lower Data Strobe signal compatible with microprocessors such as the 68000. A programmable polarity on the Address Strobe signal eliminates the need for external logic. The C-LANCE interfaces with both multiplexed and de-multiplexed data busses and features control signals for address/data bus transceivers. The C-LANCE is pin-for-pin compatible with AMD's LANCE device (Am7990). Please refer to Appendix B for a complete comparison between the C-LANCE and LANCE devices.

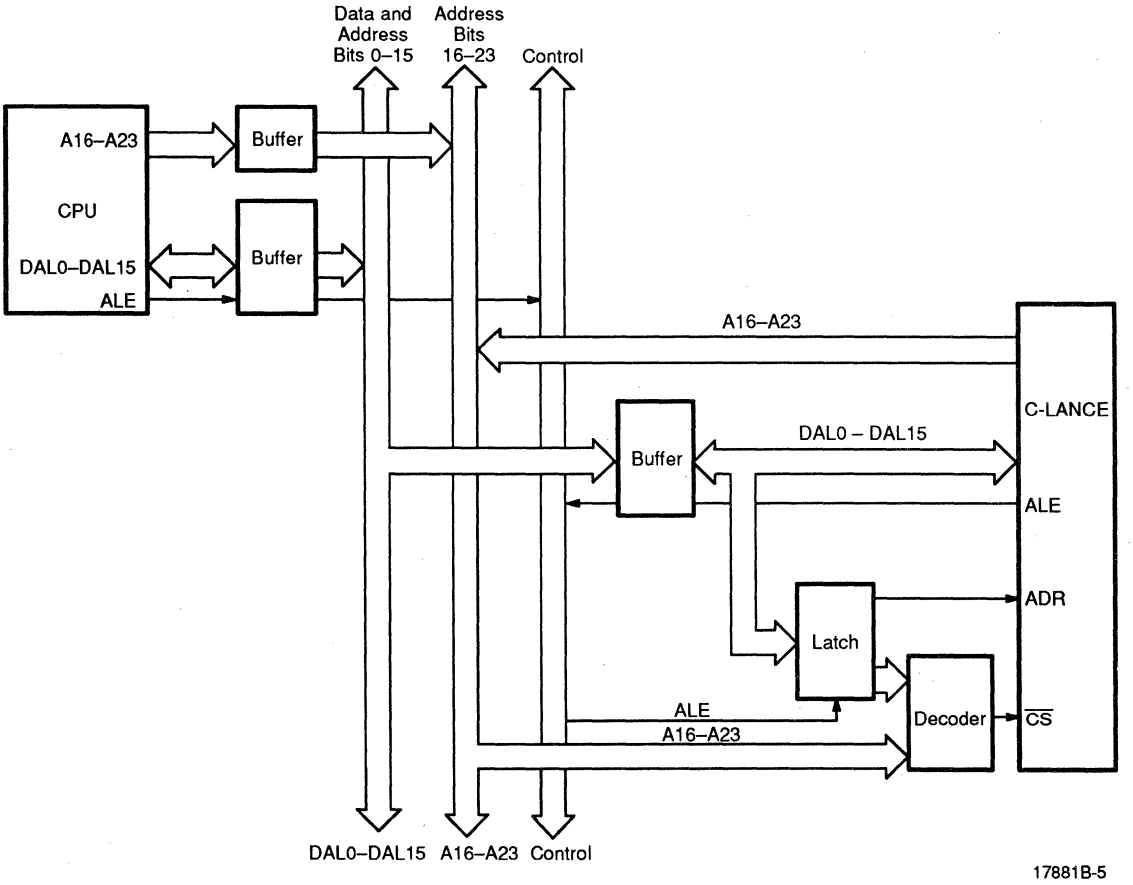


Figure 1. C-LANCE/CPU Interfacing Multiplexed Bus

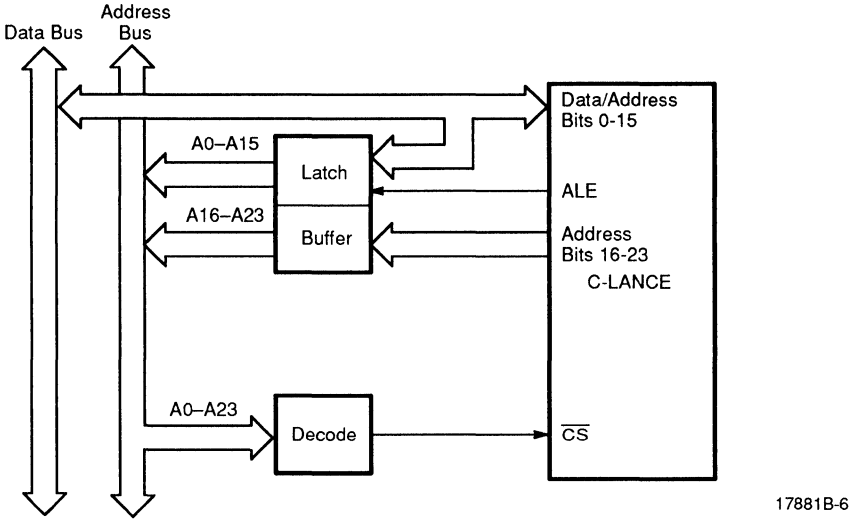


Figure 2. C-LANCE/CPU Interfacing Demultiplexed Bus

During initialization, the CPU loads the starting address of the initialization block into two internal control registers. The C-LANCE has four internal control and status registers (CSR0, 1, 2, 3) which are used for various functions, such as the loading of the initialization block address, and programming different modes and status conditions. The host processor communicates with the C-LANCE during the initialization phase, for demand transmission, and periodically to read the status bits following interrupts. All other transfers to and from the memory are automatically handled as DMA.

Interrupts to the microprocessor are generated by the C-LANCE upon:

- completion of its initialization routine
- the reception of a packet
- the transmission of a packet
- transmitter timeout error
- a missed packet
- memory error

The cause of the interrupt is ascertained by reading CSR0. Bit (06) of CSR0, (INEA), enables or disables interrupts to the microprocessor. In systems where polling is used in place of interrupts, bit (07) of CSR0, (INTR), indicates an interrupt condition.

The basic operation of the C-LANCE consists of two distinct modes: transmit and receive. In the transmit mode, the C-LANCE chip directly accesses data (in a transmit buffer) in memory. It prefaces the data with a preamble, start frame delimiter (SFD), and calculates and appends a 32-bit CRC. On transmission, the first byte of data loads into the 48-byte Transmit FIFO; the C-LANCE then begins to transmit preamble while simultaneously loading the rest of the packet into Transmit FIFO for transmission.

In the receive mode, packets are sent via the Am7992B SIA to the C-LANCE. The packets are loaded into the 64-byte Receive FIFO for preparation of automatic downloading into buffer memory. A CRC is calculated and compared with the CRC appended to the data packet. If the calculated CRC does not agree with the packet CRC, an error bit is set.

Addressing

Packets can be received using three different destination addressing schemes: physical, logical and promiscuous.

The first type is a full comparison of the 48-bit destination address in the packet with the node address that was programmed into the C-LANCE during an initialization cycle. There are two types of logical addresses. One is group type mask where the 48-bit address in the packet is put through a hash filter to map the 48-bit physical addresses into 1 of 64 logical groups. If any of these 64 groups have been preselected as the logical address, then the 48-bit address is stored in main memory. At this time, a look up is performed by the host computer comparing the 48-bit incoming address with the pre-stored 48-bit logical address. This mode can be useful if sending packets to all of a particular type of device simultaneously (i.e., send a packet to all file servers or all printer servers). Additional details on logical addressing can be found in the INITIALIZATION section

under "Logical Address Filter." The second logical address is a broadcast address where all nodes on the network receive the packet. The last receive mode of operation is referred to as "promiscuous mode" in which a node will accept all packets on the medium regardless of their destination address.

Collision Detection and Implementation

The Ethernet and IEEE 802.3 CSMA/CD network access algorithms are implemented completely within the C-LANCE. In addition to listening for a clear medium before transmitting, Ethernet handles collisions in a predetermined way. Should two transmitters attempt to seize the medium at the same time, they will collide and the data on the medium will be garbled. The transmitting nodes listen while they transmit, detect the collision, then continue to transmit for a predetermined length of time to "jam" the network and ensure that all nodes have recognized the collision. The transmitting nodes then delay a random amount of time according to the Ethernet "truncated binary backoff" algorithm in order that the colliding nodes do not try to repeatedly access the network at the same time. The C-LANCE also offers a selectable Modified Backoff Algorithm for better performance on busy networks. Up to 16 attempts to access the network are made by the C-LANCE before reporting an error due to excessive collisions.

Error Reporting and Diagnostics

Extensive error reporting is provided by the C-LANCE. Error conditions reported relate either to the network as a whole or to individual data packets. Network-related errors are recorded as flags in the CSRs and are examined by the CPU following interrupt. Packet-related errors are written into descriptor entries corresponding to the packet.

System errors include:

- Babbling Transmitter
  - — Transmitter attempting to transmit more than 1518 bytes, excluding preamble and start frame delimiter
- Collision
  - — Collision detection circuitry nonfunctional
- Missed Packet
  - — Insufficient buffer space
- Memory timeout
  - — Memory response failure

Packet-related errors:

- CRC
  - — Invalid data
- Framing
  - — Packet did not end on a byte boundary

  Overflow/Underflow
  - — Indicates abnormal latency in servicing a DMA request
- Buffer
  - — Insufficient buffer space available

The C-LANCE performs several diagnostic routines which enhance the reliability and integrity of the system. These include a CRC check and two loop back modes (internal/external). Errors may be introduced into the system to check error detection logic. A Time Domain Reflectometer is incorporated into the C-LANCE to aid system designers in locating faults in the Ethernet physical medium. Shorts and opens manifest themselves in reflections which are sensed by the TDR.

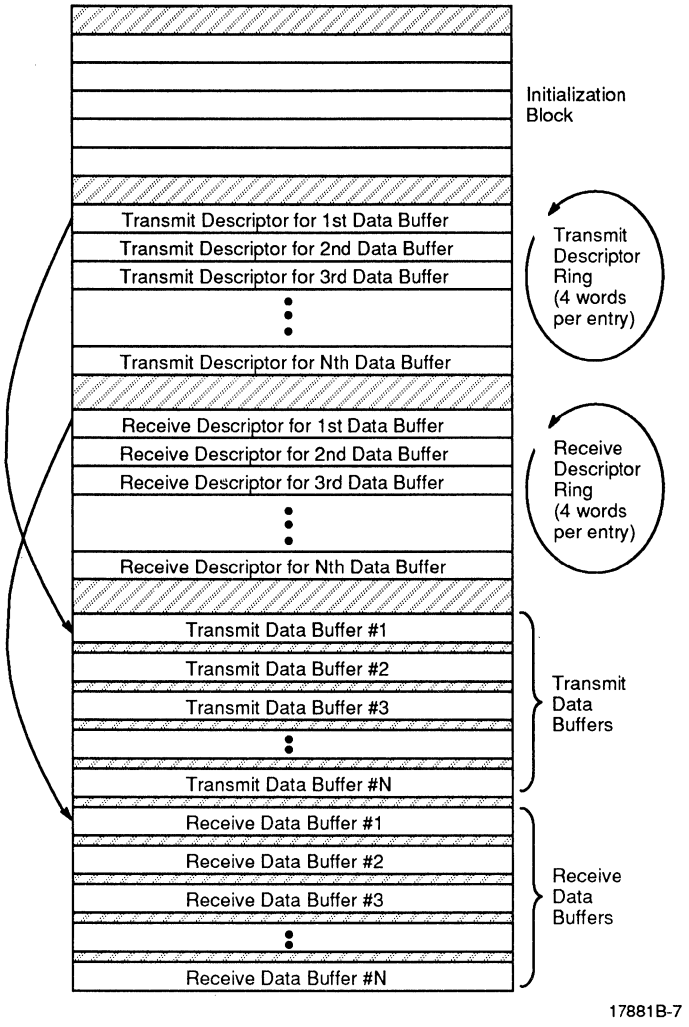


Figure 2-1. C-LANCE/Processor Memory Interface

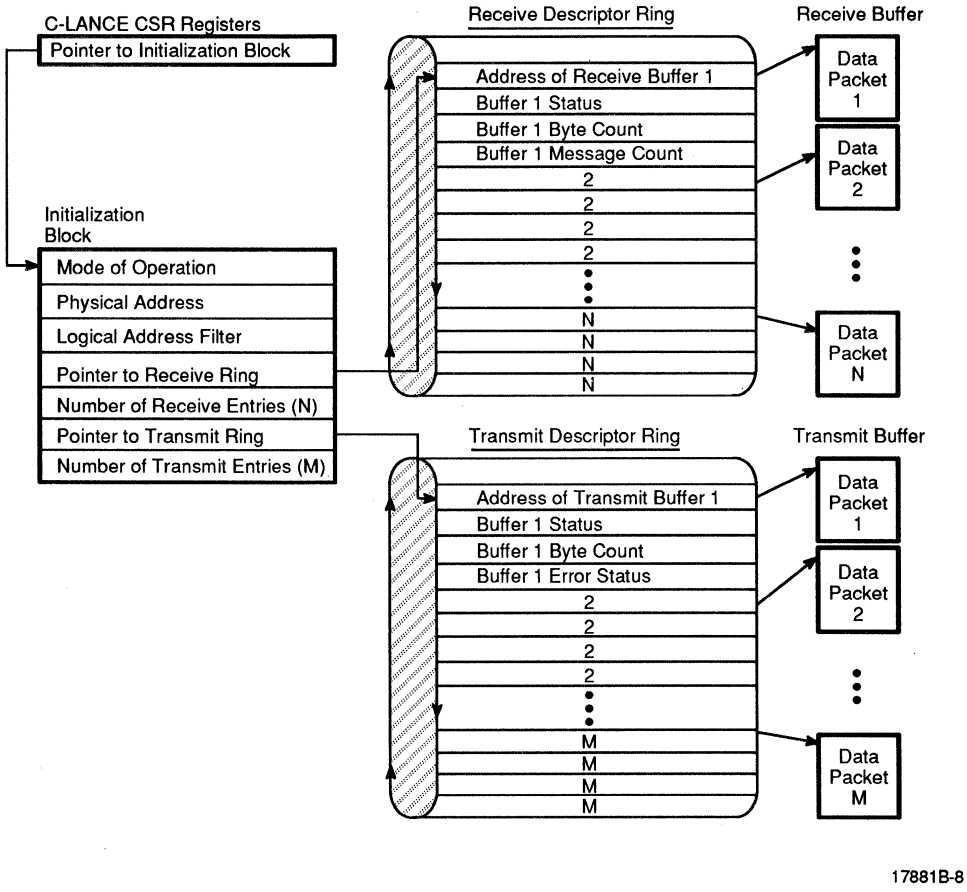


Figure 2-2. C-LANCE Memory Management

Buffer Management

A key feature of the C-LANCE and its on-board DMA channel is the flexibility and speed of communication between the C-LANCE and the host microprocessor through common memory locations. The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings as shown in Figures 2-1 and 2-2. There are separate descriptor rings to describe transmit and receive operations. Up to 128 tasks may be queued up on a descriptor ring awaiting execution by the C-LANCE. Each entry in a descriptor ring holds a pointer to a data memory buffer and an entry for the length of the data buffer. Data buffers can be chained or cascaded to handle a long packet in multiple data buffer areas. The C-LANCE searches the descriptor rings in a "lookahead" manner to determine the next empty buffer in order to chain buffers together or to handle back-to-back packets. As each buffer is filled, the "own" bit is reset, allowing the host processor to process the data in the buffer.

C-LANCE Interface

CSR bits such as ACON, BCON and BSWP are used for programming the pin functions used for different interfacing schemes. For example, ACON is used to program the polarity of the Address Strobe signal (ALE/$\overline{AS}$).

BCON is used for programming the pins, for handling either the BYTE/$\overline{WORD}$ method for addressing word organized, byte addressable memories where the BYTE signal is decoded along with the least significant address bit to determine upper or lower byte, or an explicit scheme in which two signals labeled as BYTE MASK ($\overline{BM0}$ and $\overline{BM1}$) indicate which byte is addressed. When

the BYTE scheme is chosen, the $\overline{BM1}$ pin can be used for performing the function $\overline{BUSAKO}$.

BCON is also used to program pins for different DMA modes. In a daisy chain DMA scheme, 3 signals are used ($\overline{BUSRQ}$, $\overline{HLDA}$, $\overline{BUSAKO}$). In systems using a DMA controller for arbitration, only $\overline{HOLD}$ and $\overline{HLDA}$ are used.

C-LANCE in Bus Slave Mode

The C-LANCE enters the Bus Slave Mode whenever $\overline{CS}$ becomes active. This mode must be entered whenever writing or reading the four status control registers (CSR0, CSR1, CSR2, and CSR3) and the Register Address Pointer (RAP). RAP and CSR0 may be read or written to at anytime, but the C-LANCE must be stopped (by setting the stop bit in CSR0) for CSR1, CSR2, and CSR3 access.

Read Sequence (Slave Mode)

At the beginning of a read cycle, $\overline{CS}$, READ, and $\overline{DAS}$ are asserted. ADR must be valid at this time. (If ADR is a "1," the contents of RAP are placed on the DAL lines. Otherwise the contents of the CSR register addressed by RAP are placed on the DAL lines.) After the data on the DAL lines become valid, the C-LANCE asserts $\overline{READY}$, $\overline{CS}$, READ, $\overline{DAS}$, and ADR must remain stable throughout the cycle. Refer to Figure 3.

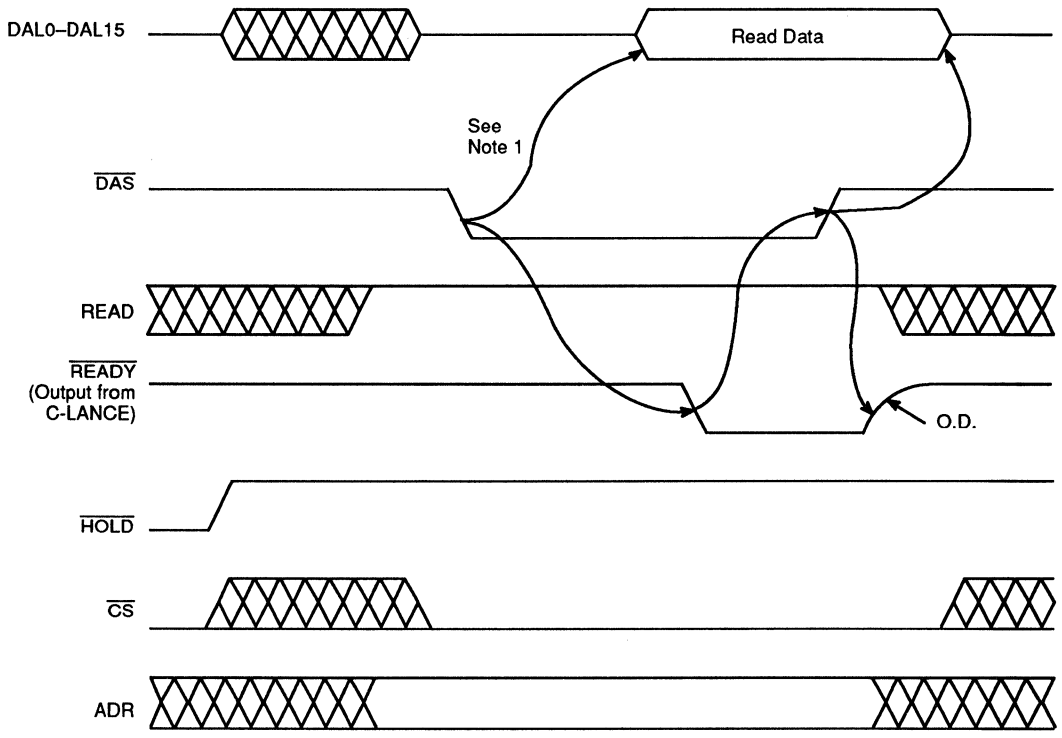
Write Sequence (Slave Mode)

This cycle is similar to the read cycle, except that during this cycle, READ is not asserted (READ is LOW). The DAL buffers are tristated which configures these lines as inputs. The assertion of $\overline{READY}$ by C-LANCE indicates to the memory device that the data on the DAL lines have been stored by C-LANCE in its appropriate CSR register. $\overline{CS}$, READ, $\overline{DAS}$, ADR and DAL 15:00 must remain stable throughout the write cycle. Refer to Figure 4.

Note: *Setting the STOP bit in the C-LANCE will generate a C-LANCE reset, which will cause all bus control output signals (including $\overline{READY}$) to float. To guarantee slave write timing when the STOP bit is being set in CSR0, the C-LANCE will latch the STOP bit and will wait for the slave cycle to complete before resetting itself and floating the output signals.*

C-LANCE in Bus Master Mode

All data transfers from the C-LANCE in the bus Master mode are timed by ALE, $\overline{DAS}$, and $\overline{READY}$. The automatic adjustment of the C-LANCE cycle by the $\overline{READY}$ signal allows synchronization with variable cycle time memory due either to memory refresh or to dual port access. Transfers are a minimum of 600 ns in length except for the first transfer of a bus mastership period in which the minimum is 700 ns. Transfers can be increased in 100 ns increments.



Note:

1. *There are two types of delays which depend on which internal register is accessed.*
   Type 1 refers to access of CSR0, CSR3 and RAP.
   Type 2 refers to access of CSR1 and CSR2 which are longer than Type 1 delay.

Figure 3. Bus Slave Read Timing

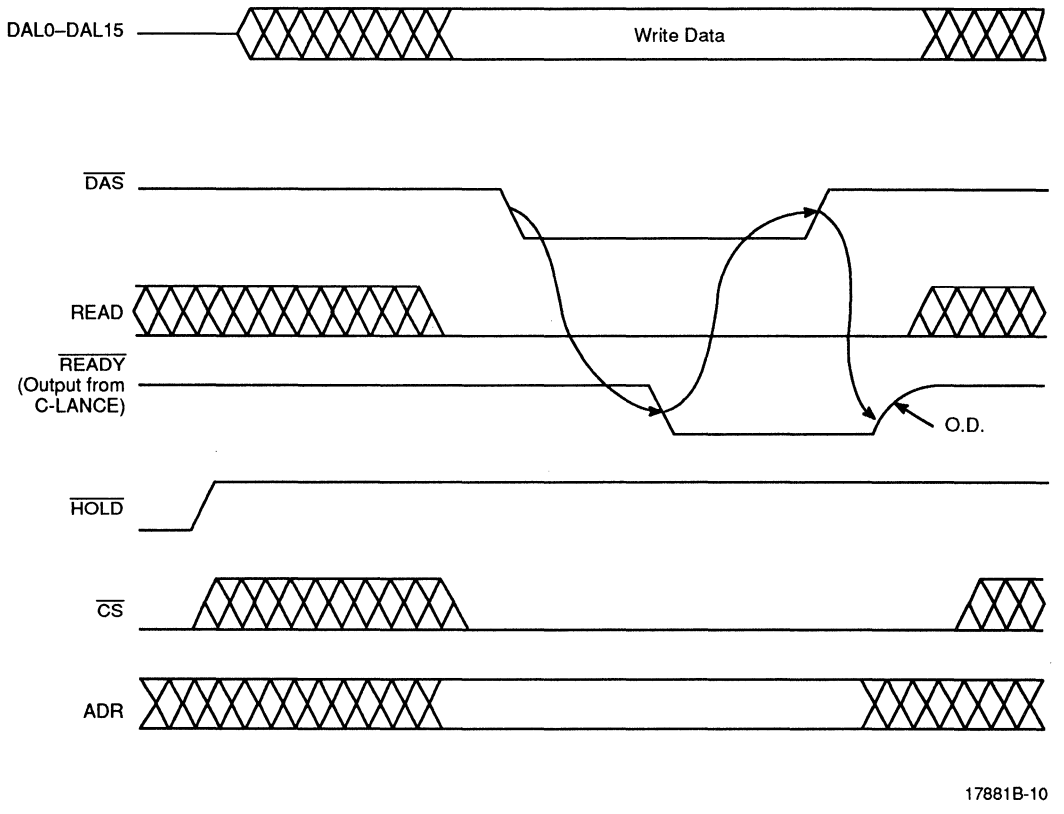


Figure 4. Bus Slave Write Timing

Read Sequence (Master Mode)

A read cycle is begun by placing a valid address on DAL00 – DAL15 and A16 – A23. The BYTE MASK signals are asserted to indicate a word, upper byte or lower byte memory reference. READ indicates the type of cycle. ALE or $\overline{AS}$ is pulsed, and the trailing edge of either can be used to latch addresses. DAL00 – DAL15 go into a 3-state mode, and $\overline{DAS}$ falls LOW to signal the beginning of the memory access. The memory responds by placing $\overline{READY}$ LOW to indicate that the DAL lines have valid data. The C-LANCE then latches memory data on the rising edge of $\overline{DAS}$, which in turn ends the memory cycle and $\overline{READY}$ returns HIGH. Refer to Figure 5-1.

The bus transceiver controls, $\overline{DALI}$ and $\overline{DALO}$, are used to control the bus transceivers. $\overline{DALI}$ directs data toward the C-LANCE, and $\overline{DALO}$ directs data or addresses away from the C-LANCE. During a read cycle, $\overline{DALO}$ goes inactive before $\overline{DALI}$ becomes active to avoid "spiking" of the bus transceivers.

Write Sequence (Master Mode)

The write cycle is similar to the read cycle except that the DAL00 – DAL15 lines change from containing addresses to data after either ALE or $\overline{AS}$ goes inactive. After data is valid on the bus, $\overline{DAS}$ goes active. Data to memory is held valid after $\overline{DAS}$ goes inactive. Refer to Figure 5-2.

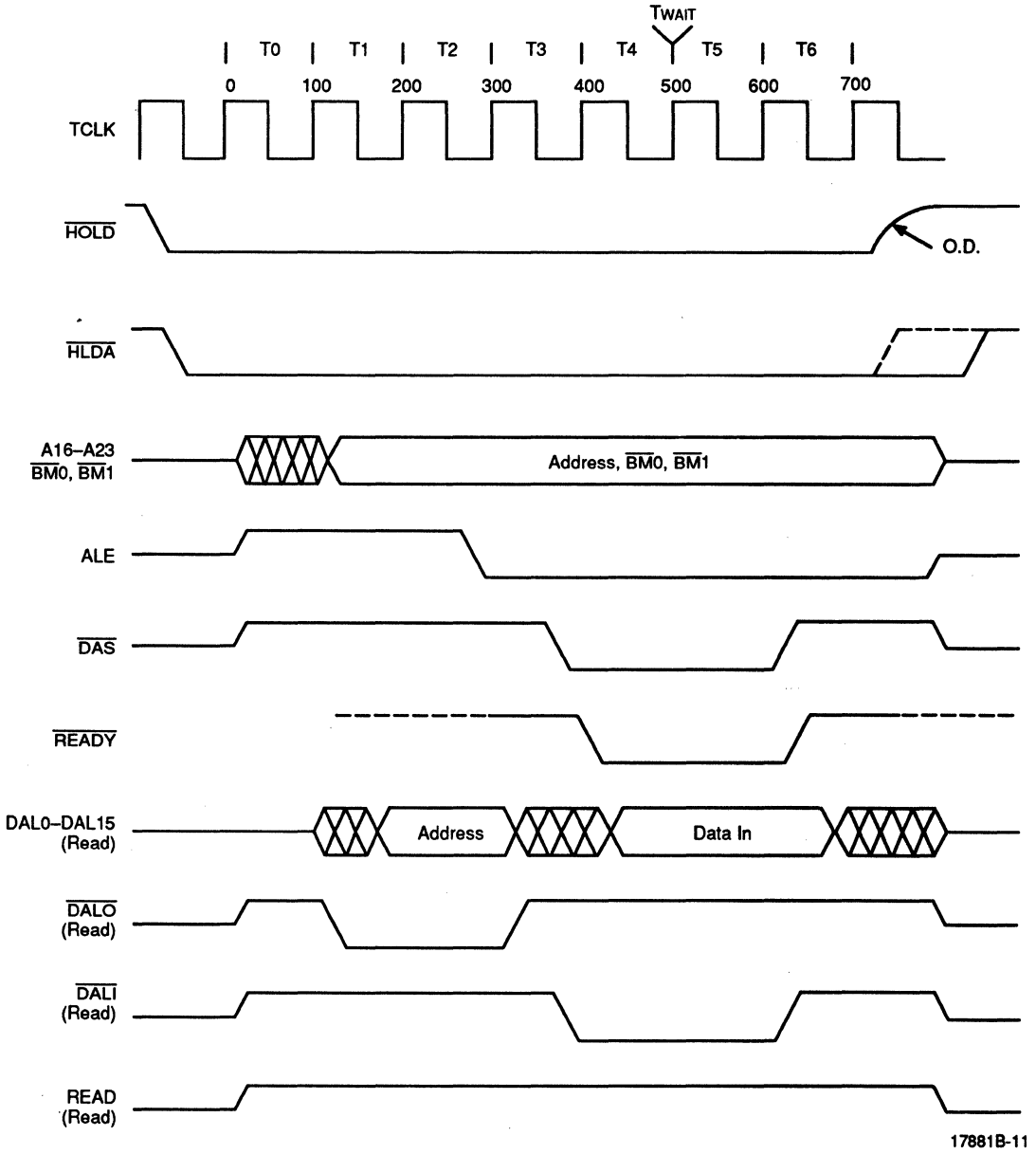


Figure 5-1. Bus Master Read Timing (Single DMA Cycle)

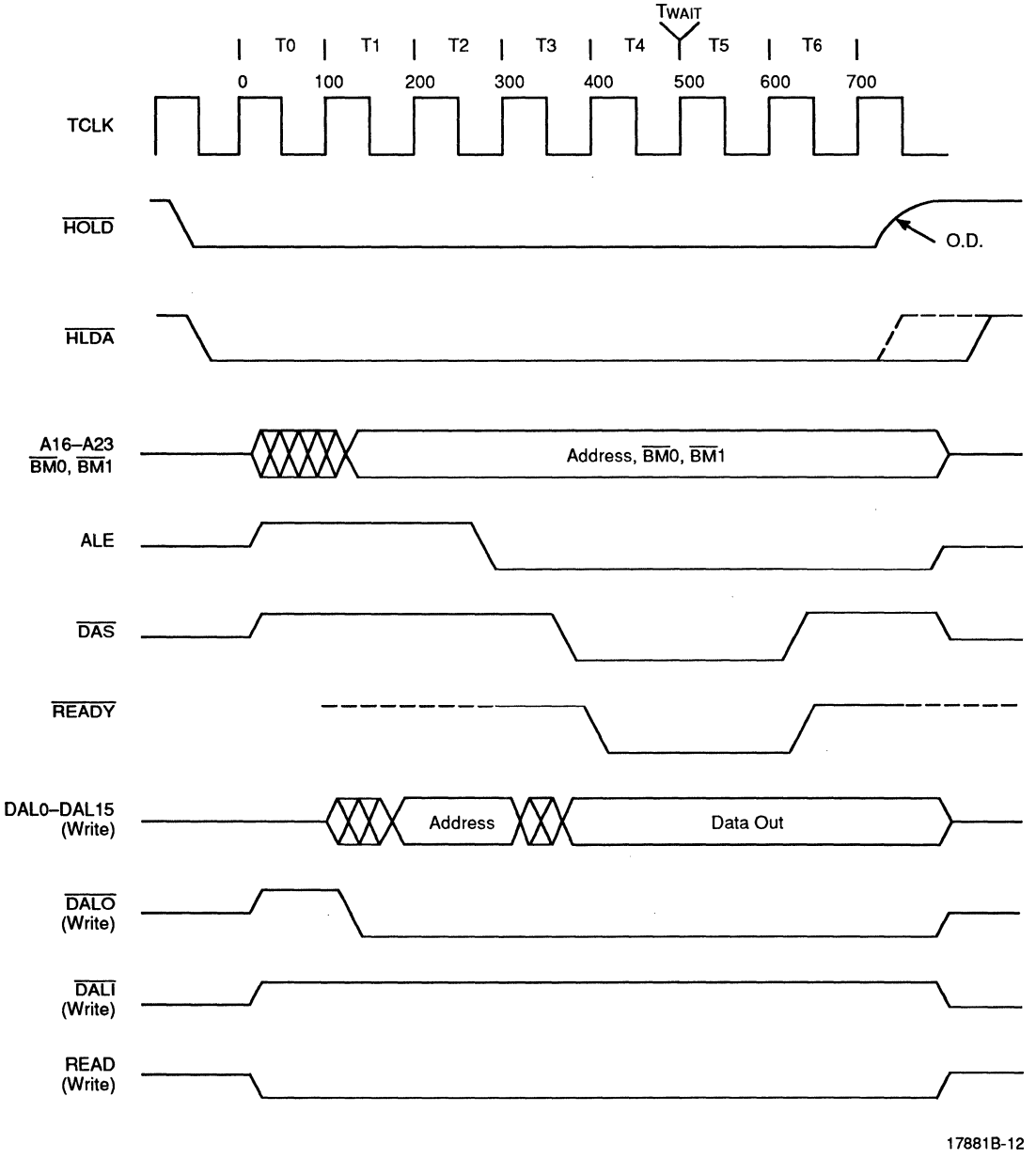


Figure 5-2. Bus Master Write Timing (Single DMA Cycle)

Differences Between Ethernet Versions 1 and 2

a. Version 2 specifies that the collision detect of the transceiver must be activated during the inter-packet gap time.

b. Version 2 specifies some network management functions, such as reporting the occurrence of collisions, retries and deferrals.

c. Version 2 specifies that when transmission is terminated, the differential transmit lines are driven to 0 volt differentially (half step).

Differences Between IEEE 802.3 and Ethernet

a. IEEE 802.3 specifies a 2-byte length field rather than a type field. The length field (802.3) describes the actual amount of data in the frame.

b. IEEE 802.3 allows the use of a PAD field in the data section of a frame, while Ethernet specifies the minimum packet size at 64 bytes. The use of a PAD allows the user to send and receive packets which have less than 46 bytes of data.

A list of significant differences between Ethernet and IEEE 802.3 at the physical layer include the following:

	IEEE 802.3	**Ethernet**
End of Transmission State	Half Step	Full Step (Rev 1) or Half Step (Rev 2)
Common Mode Voltage	±5.5 V	0 – +5 V
Common Mode Current	Less than 1 mA	1.6 mA ±40%
Receive±, Collision±		
Input Threshold	±160 mV	±175 mV
Fault Protection	16 V	0 V

PROGRAMMING

This section defines the Control and Status Registers and the memory data structures required to program the Am79C90 (C-LANCE).

Programming the Am79C90 (C-LANCE)

The Am79C90 (C-LANCE) is designed to operate in an environment that includes close coupling with local memory and microprocessor (HOST). The Am79C90 C-LANCE is programmed by a combination of registers and data structures resident within the C-LANCE and memory registers. There are four Control and Status Registers (CSRs) within the C-LANCE which are programmed by the HOST device. Once enabled, the C-LANCE has the ability to access memory locations to acquire additional operating parameters.

The Am79C90 has the ability to do independent buffer management as well as transfer data packets to and from the Ethernet. There are three memory structures accessed by the Chip:

- Initialization Block—12 words in contiguous memory starting on a word boundary. It also contains the operating parameters necessary for device operation. The initialization block is comprised of:
  - — Mode of Operation
  - — Physical Address
  - — Logical Address Mask
  - — Location to Receive and Transmit Descriptor Rings
  - — Number of Entries in Receive and Transmit Descriptor Rings
- Receive and Transmit Descriptor Rings—Two ring structures, one for incoming and outgoing packets. Each entry in the rings is 4 words long and each entry must start on a quadword boundary. The Descriptor Rings are comprised of:
  - — The address of a data buffer
  - — The length of that data buffer
  - — Status information associated with the buffer
- Data Buffers—Contiguous portions of memory reserved for packet buffering. Data buffers may begin on arbitrary byte boundaries.

In general, the programming sequence of the C-LANCE may be summarized as:

- Program the C-LANCE's CSRs by a host device to locate an initialization block in memory. The byte control, byte address, and address latch enable modes are also defined here.
- The C-LANCE loads itself with the information contained within the initialization block.
- The C-LANCE accesses the descriptor rings for packet handling.

CONTROL AND STATUS REGISTERS

There are four Control and Status Registers (CSRs) on the chip. The CSRs are accessed through two bus addressable ports, an address port (RAP) and a data port (RDP).

Accessing the Control and Status Registers

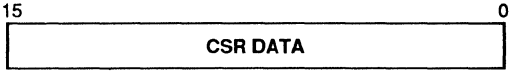
The CSRs are read (or written) in a two step operation. The address of the CSR to be accessed is written into the RAP during a bus slave transaction. During a subsequent bus slave transaction, the data being read from (or written into) the RDP is read from (or written into) the CSR selected in the RAP.

Once written, the address in RAP remains unchanged until rewritten.

To distinguish the data port from the address port, a discrete input pin is provided.

ADR Input Pin	Port
L	Register Data Port (RDP)
H	Register Address Port (RAP)

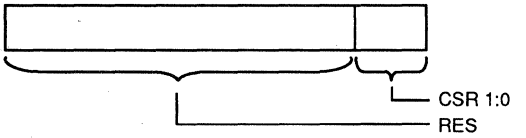
Register Data Port (RDP)



17881B-13

Bit	Name	Description
15:00	CSR Data	Writing data into RDP writes the data into the CSR selected in RAP. Reading the data from the RDP reads the data from the CSR selected in RAP. CSR1, CSR2 and CSR3 are accessible only when the STOP bit of CSR0 is set. If the STOP bit is not set while attempting to access CSR1, CSR2 or CSR3, the C-LANCE will return READY, but a READ operation will return undefined data. WRITE operation is ignored.

Register Address Port (RAP)



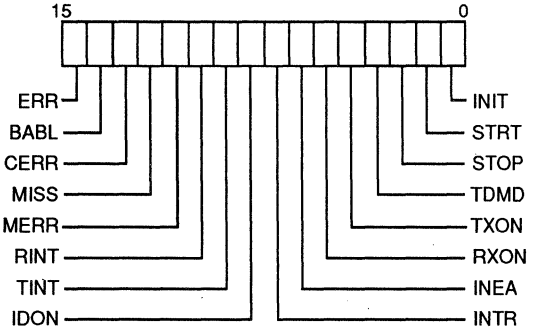
17881B-14

Bit	Name	Description
15:02	RES	Reserved. Read as zeroes. Write as zeroes.
01:00	CSR(1:0)	CSR address select. READ/WRITE. Selects the CSR to be accessed through the RDP. RAP is cleared by Bus RESET.

CSR(1 :0)	CSR
00	CSR0
01	CSR1
10	CSR2
11	CSR3

Control and Status Register Definition

Control and Status Register 0 (CSR0)



The C-LANCE updates CSR_0 by logical "ORing" the previous and present value of CSR_0.

17881B-15

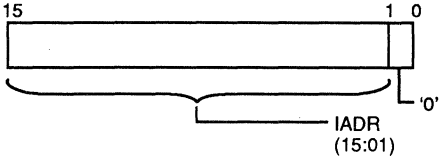
Bit	Name	Description
15	ERR	ERROR summary is set by the "ORing" of BABL, CERR, MISS and MERR. ERR remains set as long as any of the error flags are true. ERR is read only; writing it has no effect. It is cleared by Bus RESET, setting the STOP bit, or clearing the individual error flags.
14	BABL	BABBLE is a transmitter timeout error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum length packet. BABL is a flag which indicates excessive length in the transmit buffer. It will be set after 1519 bytes have been transmitted, excluding preamble and start frame delimiter; the C-LANCE will continue to transmit until the whole packet is transmitted or until there is a failure before the whole packet is transmitted. When BABL error occurs, an interrupt will be generated if INEA = 1. BABL is READ/CLEAR ONLY and is set by the C-LANCE, and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by RESET or by setting the STOP bit.
13	CERR	COLLISION ERROR indicates that the collision input to the C-LANCE was not asserted during the transmission, nor within 4.0 µs after the transmit completed. The collision after transmission is a transceiver test feature. This function is also known as heartbeat or SQE (Signal Quality Error) test. CERR is READ/CLEAR ONLY and is set by the C-LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by RESET or by setting the STOP bit. CERR error will not cause an interrupt to occur (INTR = 0).
12	MISS	MISSED PACKET is set when the receiver loses a packet because it does not own any receive buffer, indicating loss of data. FIFO overflow is not reported because there is no receive ring entry in which to write status. When MISS is set, an interrupt will be generated if INEA = 1. MISS is READ/CLEAR ONLY, and is set by the C-LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by RESET or by setting the STOP bit.

Bit	Name	Description
11	MERR	MEMORY ERROR is set when the C-LANCE is the Bus Master and has not received $\overline{\text{READY}}$ within 25.6 µs after asserting the address on the DAL lines. When a Memory Error is detected, the receiver and transmitter are turned off (CSR0, TXON = 0, RXON = 0) and an interrupt is generated if INEA = 1. MERR is READ/CLEAR ONLY, and is set by the C-LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
10	RINT	RECEIVER INTERRUPT is set when the C-LANCE updates an entry in the Receive Descriptor Ring for the last buffer received or reception is stopped due to a failure. When RINT is set, an interrupt is generated if INEA = 1. RINT is READ/CLEAR ONLY, and is set by the C-LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
09	TINT	TRANSMITTER INTERRUPT is set when the C-LANCE updates an entry in the transmit descriptor ring for the last buffer sent or transmission is stopped due to a failure. When TINT is set, an interrupt is generated if INEA = 1. TINT is READ/CLEAR ONLY and is set by the C-LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
08	IDON	INITIALIZATION DONE indicates that the C-LANCE has completed the initialization procedure started by setting the INIT bit. When IDON is set, the C-LANCE has read the Initialization Block from memory and stored the new parameters. When IDON is set, an interrupt is generated if INEA = 1. IDON is READ/CLEAR ONLY, and is set by the C-LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
07	INTR	INTERRUPT FLAG is set by the "ORing" of BABL, MISS, MERR, RINT, TINT and IDON. If INEA = 1 and INTR = 1, the $\overline{\text{INTR}}$ pin will be LOW. INTR is READ ONLY; writing this bit has no effect. INTR is cleared by RESET, by setting the STOP bit, or by clearing the condition causing the interrupt.
06	INEA	INTERRUPT ENABLE allows the $\overline{\text{INTR}}$ pin to be driven LOW when the Interrupt Flag is set. If INEA = 1 and INTR = 1, the $\overline{\text{INTR}}$ pin will be Low. If INEA = 0, the $\overline{\text{INTR}}$ pin will be HIGH, regardless of the state of the Interrupt Flag. INEA is READ/WRITE and cleared by RESET or by setting the STOP bit. INEA can be set at any time, regardless of the state of the STOP bit. (reference Appendix B).
05	RXON	RECEIVER ON indicates that the receiver is enabled. RXON is set when STRT is set if DRX = 0 in the MODE register in the initialization block and the initialization block has been read by the C-LANCE by setting the INIT bit. RXON is cleared when IDON is set from setting the INIT bit and DRX = 1 in the MODE register, or a memory error (MERR) has occurred. RXON is READ ONLY; writing this bit has no effect. RXON is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
04	TXON	TRANSMITTER ON indicates that the transmitter is enabled. TXON is set when STRT is set if DTX = 0 in the MODE register in the initialization block and the INIT bit has been set. TXON is cleared when IDON is set and DTX = 1 in the MODE register, or an error, such as MERR, UFLO or BUFF, has occurred during transmission. TXON is READ ONLY; writing this bit has no effect. TXON is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.

Bit	Name	Description
03	TDMD	TRANSMIT DEMAND, when set, causes the C-LANCE to access the Transmit Descriptor Ring without waiting for the polltime interval to elapse. TDMD need not be set to transmit a packet; it merely hastens the C-LANCE's response to a Transmit Descriptor Ring entry insertion by the host. TDMD is WRITE WITH ONE ONLY and is cleared by the microcode after it is used. It may read as a "1" for a short time after it is written because the microcode may have been busy when TDMD was set. It is also cleared by RESET or by setting the STOP bit. Writing a "0" in this bit has no effect.
02	STOP	STOP disables the C-LANCE from all external activity when set and clears the internal logic. Setting STOP is the equivalent of asserting RESET. The C-LANCE remains inactive and STOP remains set until the STRT or INIT bit is set. If STRT, INIT and STOP are all set together, STOP will override the other bits and only STOP will be set. STOP is READ/WRITE WITH ONE ONLY and set by RESET. Writing a "0" to this bit has no effect. STOP is cleared by setting either INIT or STRT. CSR3 must be reloaded when the STOP bit is set.
01	STRT	START enables the C-LANCE to send and receive packets, perform direct memory access, and do buffer management. The STOP bit must be set prior to setting the STRT bit. Setting STRT clears the STOP bit. STRT is READ/WRITE and is set with one only. Writing a "0" into this bit has no effect. STRT is cleared by RESET or by setting the STOP bit.
00	INIT	INITIALIZE, when set, causes the C-LANCE to begin the initialization procedure and access the Initialization Block. The STOP bit must be set prior to setting the INIT bit. Setting INIT clears the STOP bit. INIT is READ/WRITE WITH "1" ONLY. Writing a "0" into this bit has no effect. INIT is cleared by RESET or by setting the STOP bit. The C-LANCE latches CSR0 during a slave read; therefore, the CSR0 status bits are guaranteed to be stable for the duration of the CSR0 access.

Control and Status Register 1 (CSR1)

READ/WRITE: Accessible only when the STOP bit of CSR0 is a ONE and RAP = 01. The C-LANCE preserves the contents of CSR1 after STOP.

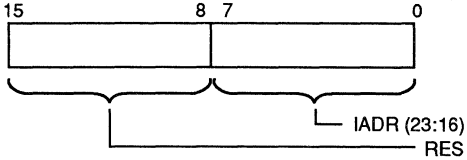


17881B-16

Bit	Name	Description
15:01	IADR	The low order 15 bits of the address of the first word (lowest address) in the Initialization Block.
00		Must be zero.

Control and Status Register 2 (CSR2)

READ/WRITE: Accessible only when the STOP bit of CSR0 is a ONE and RAP = 10. The C-LANCE preserves the contents of CSR2 after STOP.



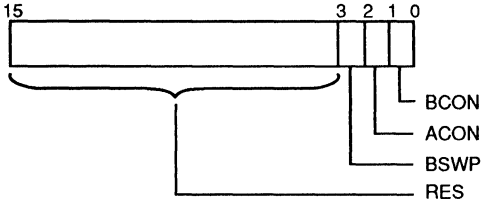
17881B-17

Bit	Name	Description
15:08	RES	Reserved. Read as zeroes. Write as zeroes.
07:00	IADR	The high order 8 bits of the address of the first word (lowest address) in the initialization Block.

Control and Status Register 3 (CSR3)

CSR3 allows redefinition of the Bus Master interface.

READ/WRITE: Accessible only when the STOP bit of CSR0 is ONE and RAP = 11. CSR3 is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit in CSR0.



17881B-18

Bit	Name	Description
15:03	RES	Reserved. Read as zeroes. Write as zeroes.
02	BSWP	BYTE SWAP allows the chip to operate in systems that consider bits (15:08) of data to be pointed at an even address and bits (07:00) to be pointed at an odd address. When BSWP = 1, the C-LANCE will swap the high and low bytes on DMA data transfers between the Receive FIFO and bus memory. Only data from the Receive FIFO transfers is swapped; the Initialization Block data and the Descriptor Ring entries are NOT swapped. BSWP is READ/WRITE and cleared by $\overline{\text{RESET}}$ or by setting the STOP bit in CSR0.
01	ACON	ALE CONTROL defines the assertive state of ALE when the C-LANCE is a Bus Master. ACON is READ/WRITE and cleared by $\overline{\text{RESET}}$ and by setting the STOP bit in CSR0. **ACON** — **ALE** 0 — Asserted HIGH 1 — Asserted LOW
00	BCON	BYTE CONTROL redefines the Byte Mask and Hold I/O pins. BCON is READ/WRITE and cleared by $\overline{\text{RESET}}$ or by setting the STOP bit in CSR0. **BCON** — **Pin 16** — **Pin 15** — **Pin 17** 0 — $\overline{\text{BM1}}$ — $\overline{\text{BM0}}$ — $\overline{\text{HOLD}}$ 1 — $\overline{\text{BUSAKO}}$ — BYTE — $\overline{\text{BUSRQ}}$

All data transfers from the C-LANCE in the Bus Master mode are in words. However, the C-LANCE can handle odd address boundaries and/or packets with an odd number of bytes.

Initialization

Initialization Block

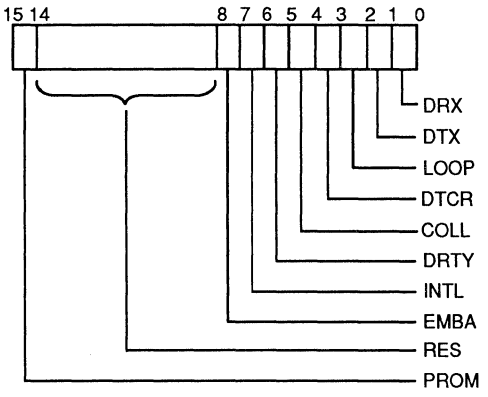
Chip initialization includes the reading of the initialization block in memory to obtain the operating parameters. The following is a definition of the Initialization Block.

The Initialization Block is read by the C-LANCE when the INIT bit in CSR0 is set. The INIT bit should be set before or concurrent with the STRT bit to insure proper parameter initialization and chip operation. After the C-LANCE has read the Initialization Block, IDON is set in CSR0 and an interrupt is generated if INEA = 1.

Higher Address	TLEN–TDR (23:16)	IADR +22
	TDRA (15:00)	IADR +20
	RLEN–RDRA (23:16)	IADR +18
	RDRA (15:00)	IADR +16
	LADRF (63:48)	IADR +14
	LADRF (47:32)	IADR +12
	LADRF (31:16)	IADR +10
	LADRF (15:00)	IADR +08
	PADR (47:32)	IADR +06
	PADR (31:16)	IADR +04
	PADR (15:00)	IADR +02
Base Address of Block	MODE	IADR +00

Mode

The Mode Register allows alteration of the C-LANCE's operating parameters. Normal operation is with the Mode Register clear.



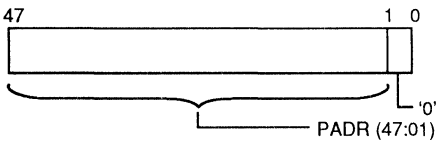
17881B-19

Bit	Name	Description
15	PROM	PROMISCUOUS mode. When PROM = 1, all incoming packets are accepted.
14:08	RES	RESERVED. Read as zeroes. Write as zeroes.
07	EMBA	Enable Modified Back-off Algorithm. When set (EMBA=1), enables the modified backoff algorithm. EMBA is cleared by activation of the RESET pin or setting the STOP bit.
06	INTL	INTERNAL LOOPBACK is used with the LOOP bit to determine where the loopback is to be done. Internal loopback allows the chip to receive its own transmitted packet. Since this represents full duplex operation, the packet size is limited to 8–32 bytes. Internal loopback in the C-LANCE is operational when the packets are addressed to the node itself. The C-LANCE will not receive any packets externally when it is in internal loopback mode. EXTERNAL LOOPBACK allows the C-LANCE to transmit a packet through the SIA transceiver cable out to the Ethernet medium. It is used to determine the operability of all circuitry and connections between the C-LANCE and the physical medium. Multicast addressing in external loopback is valid only when DTCR = 1 (user needs to append the 4 bytes CRC). In external loopback, the C-LANCE also receives packets from other nodes. The FIFOs READ/WRITE pointers may misalign in the C-LANCE under heavy traffic. The packet could then be corrupted or not received. Therefore, the external loopback execution may need to be repeated. See specific discussion under "Loopback" in later section. INTL is only valid if LOOP = 1; otherwise, it is ignored.
05	DRTY	DISABLE RETRY. When DRTY = 1, the C-LANCE will attempt only one transmission of a packet. If there is a collision on the first transmission attempt, a Retry Error (RTRY) will be reported in Transmit Message Descriptor 3 (TMD3).

LOOP	INTL	LOOPBACK
0	X	No loopback, normal
1	0	External
1	1	Internal

Bit	Name	Description
04	COLL	FORCE COLLISION. This bit allows the collision logic to be tested. The C-LANCE must be in internal loopback mode for COLL to be valid. If COLL = 1, a collision will be forced during the subsequent transmission attempt. This will result in 16 total transmission attempts with a retry error reported in TMD3.
03	DTCR	DISABLE TRANSMIT CRC. When DTCR = 0, the transmitter will generate and append a CRC to the transmitted packet. When DTCR = 1, the CRC logic is allocated to the receiver and no CRC is generated and sent with the transmitted packet. The ADD_FCS bit (bit 13, TMD1) can be used to override a DTCR=1 setting on a per packet basis. During loopback, DTCR = 0 will cause a CRC to be generated on the transmitted packet, but no CRC check will be done by the receiver since the CRC logic is shared and cannot generate and check CRC at the same time. The generated CRC will be written into memory with the data and can be checked by the host software. If DTCR = 1 during loopback, the host software must append a CRC value to the transmit data. The receiver will check the CRC on the received data and report any errors.
02	LOOP	LOOPBACK allows the C-LANCE to operate in full duplex mode for test purposes. The packet size is limited to 8–32 bytes.The received packet can be up to 36 bytes (32 + 4 bytes CRC) when DTCR = 0. During loopback, the runt packet filter is disabled because the maximum packet is forced to be smaller than the minimum size Ethernet packet (64 bytes). LOOP = 1 allows simultaneous transmission and reception for a message constrained to fit within the Transmit FIFO. The C-LANCE waits until the entire message is in the Transmit FIFO before serial transmission begins. The incoming data stream fills the Receive FIFO. Moving the received message out of the Receive FIFO to memory does not begin until reception has ceased.

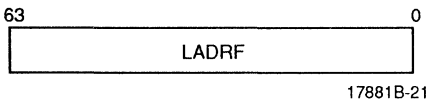
Bit	Name	Description
		In loopback mode, transmit data chaining is not possible. Receive data chaining is possible if receive buffers are 32 bytes long to allow time for lookahead.
01	DTX	DISABLE THE TRANSMITTER causes the C-LANCE to not access the Transmitter Descriptor Ring, and therefore, no transmissions are attempted. DTX = 1 will clear the TXON bit in CSR0 when initialization is complete.
00	DRX	DISABLE THE RECEIVER causes the C-LANCE to reject all incoming packets and not access the Receive Descriptor Ring. DRX = 1 will clear the RXON bit in the CSR0 when initialization is complete.



17881B-20

47:00	PADR	PHYSICAL ADDRESS is the unique 48-bit physical address assigned to the C-LANCE. PADR (0) must be zero.

Logical Address Filter



17881B-21

63:00	LADRF	The 64-bit mask used by the C-LANCE to accept logical addresses.

The purpose of logical (or group or multicast) addresses is to allow a group of nodes in a network to receive the same message. Each node can maintain a list of multicast addresses that it will respond to. The logical address filter mechanism in the C-LANCE is a hardware aide that reduces the average amount of host computer time required to determine whether or not an incoming packet with a multicast destination address should be accepted.

The logical address filter hardware is an implementation of a hash code searching technique commonly used by software programmers. If the multicast bit of the destination address of an incoming packet is set, the hardware maps this address into one of 64 categories which correspond to 64 bits in the Logical Address Filter Register. The hardware then accepts or rejects the packet depending on the state of the bit in the Logical Address Filter Register which corresponds to the selected category. For example, if the address maps into category 24, and bit 24 of the logical address filter register is set, the packet is accepted.

A node can be made a member of several groups by setting the appropriate bits in the logical address filter register.

The details of the hardware mapping algorithm are as follows:

If the first bit of an incoming address is a "1" [PADR (0) =1], the address is deemed logical and is passed through the logical address filter.

The logical address filter is a 64-bit mask composed of four sixteen-bit registers, LADRF (63:00) in the initialization block, that is used to accept incoming Logical Addresses. The incoming address is sent through the CRC circuit. After all 48 bits of the address have gone through the CRC circuit, the high order 6 bits of the resultant CRC (32-bit CRC) are strobed into a register. This register is used to select one of the 64-bit positions in the Logical Address Filter. If the selected filter bit is a "1," the address is accepted and the packet will be put in memory. The logical address filter only assures that there is a possibility that the incoming logical address belongs to the node. To determine if it belongs to the node, the incoming logical address that is stored in main memory is compared by software to the list of logical addresses to be accepted by this node.

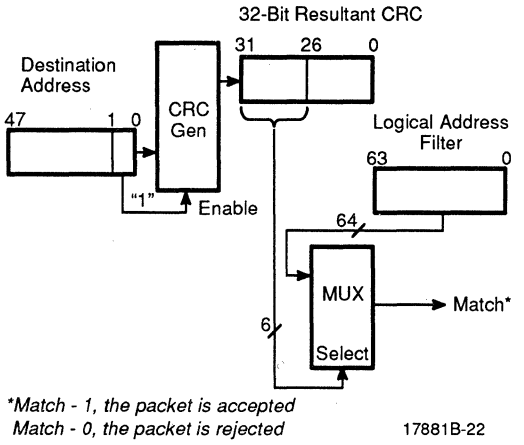
The task of mapping a logical address to one of 64-bit positions requires a simple computer program (see Appendix A) which uses the same CRC algorithm (used in C-LANCE and defined per Ethernet) to calculate the HASH (see Figure 7).

Driver software that manages a list of multicast addresses can work as follows. First the multicast address list and the logical address filter must be initialized. Some sort of management function such as the driver initialization routine passes to the driver a list of addresses. For each address in the list the driver uses a subroutine similar to the one listed in the appendix to set the appropriate bit in a software copy of the logical address filter register. When the complete list of addresses has been processed, the register is loaded.

Later, when a packet is received, the driver first looks at the Individual/Group bit of the destination address of the packet to find out whether or not this is a multicast address. If it is, the driver must search the multicast address list to see if this address is in the list. If it is not in the list, the packet is discarded.

The Broadcast address, which consists of all ones is a special multicast address. Packets addressed to the broadcast address must be received by all nodes. Since broadcast packets are usually more common than other multicast packets, the broadcast address should be the first address in the multicast address list.

The Broadcast address does not go through the Logical Address Filter and is always enabled. If the Logical Address Filter is loaded with all zeroes, all incoming logical addresses except broadcast will be rejected. The multicast addressing in external loopback is operational only when DTCR in the mode register is set to 1.

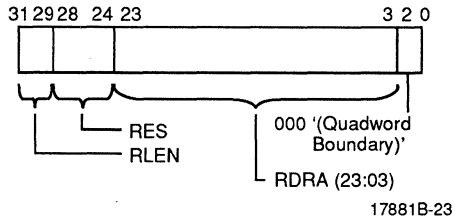


Match - 1, the packet is accepted
Match - 0, the packet is rejected

17881B-22

Figure 7. Logical Address Filter Operation

Receive Descriptor Ring Pointer



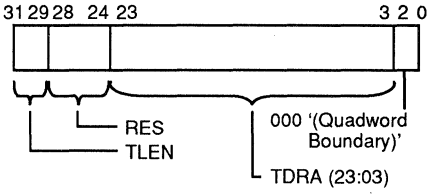
17881B-23

Bit	Name	Description
31:29	RLEN	RECEIVE RING LENGTH is the number of entries in the receive ring expressed as a power of two.

RLEN	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

28:24	RES	RESERVED. Read as zeroes. Write as zeroes.
23:03	RDRA	RECEIVE DESCRIPTOR RING ADDRESS is the base address (lowest address) of the Receive Descriptor Ring.
02:00		MUST BE ZEROES. These bits are RDRA (02:00) and must be zeroes because the Receive Ring is aligned on a quadword boundary.

Transmit Descriptor Ring Pointer



17881B-24

31:29	TLEN	TRANSMIT RING LENGTH is the number of entries in the Transmit Ring expressed as a power of two.

TLEN	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

28:24	RES	RESERVED. Read as zeroes. Write as zeroes.
23:03	TDRA	TRANSMIT DESCRIPTOR RING ADDRESS is the base address (lowest address) of the Transmit Descriptor Ring.
02:00		MUST BE ZEROES. These bits are TDRA (02:00) and must be zeroes because the Transmit Ring is aligned on a quadword boundary.

Buffer Management

Buffer Management is accomplished through message descriptors organized in ring structures in memory. Each message descriptor entry is four words long. There are two rings allocated for the device: a Receive ring and a Transmit ring. The device is capable of polling each ring for buffers to either empty or fill with packets to or from the channel. The device is also capable of entering status information in the descriptor entry. C-LANCE polling is limited to looking one ahead of the descriptor entry the C-LANCE is currently working with.

The location of the descriptor rings and their length are found in the initialization block, accessed during the initialization procedure by the C-LANCE. Writing a "ONE" into the STRT bit of CSR0 will cause the C-LANCE to start accessing the descriptor rings and enable it to send and receive packets.

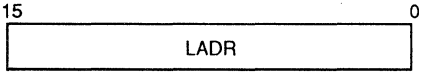
The C-LANCE communicates with a HOST device through the ring structures in memory. Each entry in the ring is either owned by the C-LANCE or the HOST. There is an ownership bit (OWN) in the message descriptor entry. Mutual exclusion is accomplished by a protocol which states that each device can only relinquish ownership of the descriptor entry to the other device; it can never take ownership, and no device can change the state of any field in any entry after it has relinquished ownership.

Descriptor Ring

Each descriptor in a ring in memory is a 4-word entry. The following is the format of the receive and the transmit descriptors.

Receive Message Descriptor Entry

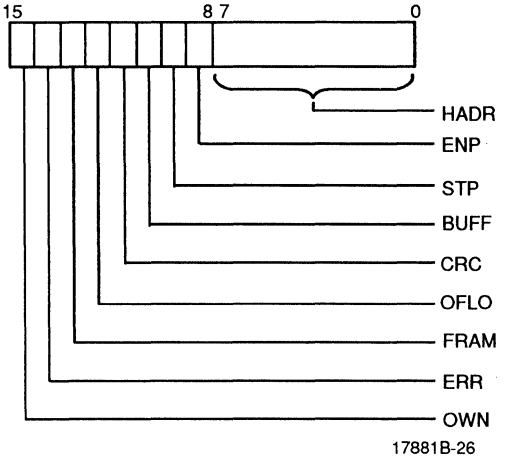
Receive Message Descriptor 0 (RMD0)



17881B-25

Bit	Name	Description
15:00	LADR	The LOW ORDER 16 address bits of the buffer pointed to by this descriptor. LADR is written by the host and is not changed by the C-LANCE.

Receive Message Descriptor 1 (RMD1)

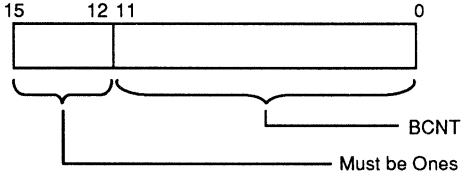


17881B-26

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = 0) or by the C-LANCE (OWN = 1). The C-LANCE clears the OWN bit after filling the buffer pointed to by the descriptor entry. The host sets the OWN bit after emptying the buffer. Once the C-LANCE or host has relinquished ownership of a buffer, it must not change any field in the four words that comprise the descriptor entry.
14	ERR	ERROR summary is the OR of FRAM, OFLO, CRC or BUFF.
13	FRAM	FRAMING ERROR indicates that the incoming packet contained a non-integer multiple of eight bits and there was a CRC error. If there was not a CRC error on the incoming packet, then FRAM will not be set even if there was a non-integer multiple of eight bits in the packet. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not.

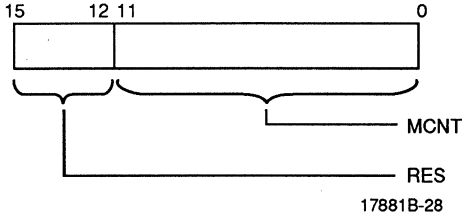
Bit	Name	Description
12	OFLO	OVERFLOW error indicates that the receiver has lost all or part of the incoming packet due to an inability to store the packet in a memory buffer before the internal Receive FIFO overflowed. OFLO is valid only when ENP is not set.
11	CRC	CRC indicates that the receiver has detected a CRC error on the incoming packet. CRC is valid only when ENP is set and OFLO is not.
10	BUFF	BUFFER ERROR is set any time the C-LANCE does not own the next buffer while data chaining a received packet. This can occur in either of two ways: 1) the OWN bit of the next buffer is zero, or 2) the Receive FIFO overflow occurred before the C-LANCE has performed a lookahead poll of the next receive descriptor. If a Buffer Error occurs, an Overflow Error may also occur internally in the Receive FIFO, but will not be reported in the descriptor status entry unless both BUFF and OFLO errors occur at the same time.
09	STP	START OF PACKET indicates that this is the first buffer used by the C-LANCE for this packet. It is used for data chaining buffers.
08	ENP	END OF PACKET indicates that this is the last buffer used by the C-LANCE for this packet. It is used for data chaining buffers. If both STP and ENP are set, the packet fits into one buffer and there is no data chaining.
07:00	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and unchanged by the C-LANCE.

Receive Message Descriptor 2 (RMD2)



15:12		MUST BE ONES. This field is written by the host and is not changed by the C-LANCE.
11:00	BCNT	BUFFER BYTE COUNT is the length of the buffer pointed to by this descriptor, expressed as a two's complement number. This field is written by the host and is not changed by the C-LANCE. Minimum buffer size is 64 bytes for the first buffer of packet.

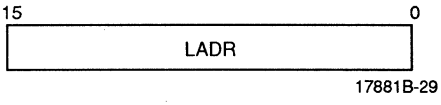
Receive Message Descriptor 3 (RMD3)



15:12	RES	RESERVED. Read as zeroes. Write as zeroes.
11:00	MCNT	MESSAGE BYTE COUNT is the length in bytes of the received message. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the chip and cleared by the host.

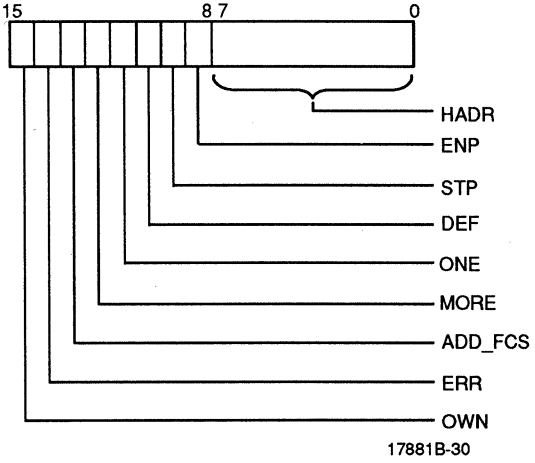
Transmit Message Descriptor Entry

Transmit Message Descriptor 0 (TMD0)



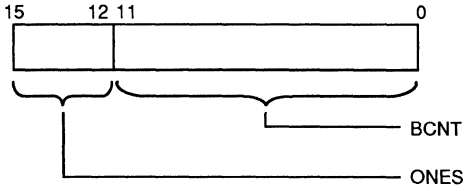
Bit	Name	Description
15:00	LADR	The LOW ORDER 16 address bits of the buffer pointed to by this descriptor. LADR is written by the host and is not changed by the C-LANCE.

Transmit Message Descriptor 1 (TMD1)



Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = O) or by the C-LANCE (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by this descriptor. The C-LANCE clears the OWN bit after transmitting the contents of the buffer. Neither the host nor the C-LANCE may alter a descriptor entry after it has relinquished ownership.
14	ERR	ERROR summary is the "OR" of LCOL, LCAR, UFLO or RTRY.
13	ADD_FCS	Setting ADD_FCS=1, instructs the controller to append a CRC to this transmitted frame, regardless of the setting of the DTCR bit (bit 3 in the Mode Register). The ADD_FCS bit allows the controller to be configured to append CRC on a per packet basis, when DTCR=1. ADD_FCS is only valid when STP=1.
12	MORE	MORE indicates that more than one retry was needed to transmit a packet.
11	ONE	ONE indicates that exactly one retry was needed to transmit a packet. The ONE flag is not valid when LCOL is set.
10	DEF	DEFERRED indicates that the C-LANCE had to defer while trying to transmit a packet. This condition occurs if the channel is busy when the C-LANCE is ready to transmit.
09	STP	START OF PACKET indicates that this is the first buffer to be used by the C-LANCE for this packet. It is used for data chaining buffers. STP is set by the host and is not changed by the C-LANCE. The STP bit must be set in the first buffer of the packet, or the C-LANCE will skip over this descriptor and poll the next descriptor(s) until the OWN and STP bits are set.
08	ENP	END OF PACKET indicates that this is the last buffer to be used by the C-LANCE for this packet. It is used for data chaining buffers. If both STP and ENP are set, the packet fits into one buffer and there is no data chaining. ENP is set by the host and is not changed by the C-LANCE.
07:00	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and is not changed by the C-LANCE.

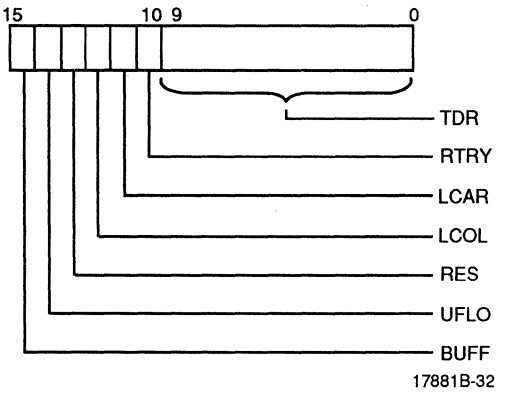
Transmit Message Descriptor 2 (TMD2)



17881B-31

Bit	Name	Description
15:12	ONES	Must be ones. This field is set by the host and is not changed by the C-LANCE.
11:00	BCNT	BUFFER BYTE COUNT is the usable length in bytes of the buffer pointed to by this descriptor expressed as a negative two's complement number. This is the number of bytes from this buffer that will be transmitted by the C-LANCE. This field is written by the host and is not changed by the C-LANCE. The first buffer of a packet has to be at least 100 bytes minimum when data chaining and 64 byte (DTCR = 1) or 60 bytes (DCTR = 0) when not data chaining.

Transmit Message Descriptor 3 (TMD3)



17881B-32

Bit	Name	Description
15	BUFF	BUFFER ERROR is set by the C-LANCE during transmission when the C-LANCE does not find the ENP flag in the current buffer and does not own the next buffer. This can occur in either of two ways: either the OWN bit of the next buffer is zero, or Transmit FIFO underflow occurred before the C-LANCE has performed a lookahead poll of the next transmit descriptor. BUFF is set by the C-LANCE and cleared by the host. BUFF error will turn off the transmitter (CSR0, TXON = 0). If a Buffer Error occurs, an Underflow Error will also occur. BUFF error is not valid when LCOL or RTRY error is set during TX data chaining.
14	UFLO	UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that the Transmit FIFO has emptied before the end of the packet was reached. Upon UFLO error, transmitter is turned off (CSR0, TXON = 0).
13	RES	RESERVED bit. The C-LANCE will write this bit with a "0."
12	LCOL	LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The C-LANCE does not retry on late collisions.
11	LCAR	LOSS OF CARRIER is set when the carrier input (RENA) to the C-LANCE goes false during a C-LANCE-initiated transmission. The C-LANCE does not retry upon loss of carrier. It will continue to transmit the whole packet until done. LCAR is not valid in INTERNAL LOOPBACK MODE.
10	RTRY	RETRY ERROR indicates that the transmitter has failed in 16 attempts to successfully transmit a message due to repeated collisions on the medium. If DRTY = 1 in the MODE register, RTRY will set after 1 failed transmission attempt.
09:00	TDR	TIME DOMAIN REFLECTOMETRY reflects the state of an internal C-LANCE counter that counts from the start of a transmission to the occurrence of a collision. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the C-LANCE and is valid only if RTRY is set.

Ring Access Mechanism in the C-LANCE

Once the C-LANCE is initialized through the initialization block and started, the CPU and the C-LANCE communicate via transmit and receive rings, for packet transmission and reception.

There are 2 sets of RAM locations (four 16-bit register per set, corresponding to the 4 entries in each descriptor) in the C-LANCE. The first set points to the current buffer, and they are the working registers which are used for transferring the data for the packet. The second set contains the pointers to the next buffer in the ring which the C-LANCE obtained from the lookahead operation.

There are three types of ring access in the C-LANCE. The first type is when the C-LANCE polls the rings to own a buffer. The second type is when the buffers are data chained. The C-LANCE does a lookahead operation between the time that it is transferring data to/from the Transmit/Receive FIFOs; this lookahead is done only once. The third type is when the C-LANCE tries to own the next descriptor in the ring when it clears the OWN bit for the current buffer.

Transmit Ring Buffer Management

When there is no Ethernet activity, the C-LANCE will automatically poll the transmit ring in the memory once it has started (CSR0, STRT = 1). This polling occurs every 1.6 ms, (CSR0 TDMD bit = 0) and consists of reading the status word of the transmit descriptor, TMD1, until the C-LANCE owns the descriptor. The C-LANCE will read TMD0 and TMD2 to get the rest of the buffer address and the buffer byte count when it owns the descriptor. Each of these memory reads is done separately with a new arbitration cycle for each transfer.

If the transmit buffers are data chained (current buffer ENP = 0), the C-LANCE will look ahead to the next descriptor in the ring while transferring the current buffer into the Transmit FIFO (see Figure 8-1). The C-LANCE does this lookahead only once. If it does not own the next transmit Descriptor Table Entry (DTE) (2nd TX ring

for this packet) it will transmit the current buffer and update the status of current Ring with the BUFF and UFLO error bits set. If the C-LANCE owns the 2nd DTE, it will also read the buffer address and the buffer byte count of this entry. Once the C-LANCE has finished emptying the current buffer, it clears the OWN bit for this buffer, and immediately starts loading the Transmit FIFO from the next (2nd) buffer. Between DMA bursts, starting from the 2nd buffer, the C-LANCE does a lookahead again to check if it owns the next (3rd) buffer. This activity goes on until the last transmit DTE indicates the end of the packet (TMD1, ENP = 1). Once the last part of the packet has been transmitted out from the Transmit FIFO to the medium, the C-LANCE will update the status in TMD1, TMD3 (TMD3 is updated only when there is an error) and will relinquish the last buffer to the CPU. The C-LANCE tries to own the next buffer (first buffer of the next packet), immediately after it relinquishes the last buffer of the current packet. This guarantees the back-to-back transmission of the packets. If the C-LANCE does not own the next buffer, it then polls the TX ring every 1.6 ms.

When an error occurs before all of the buffers get transmitted, the status, TMD3 , is updated in the current DTE, own bit is cleared in TMD1, and TINT bit is set in CSR0 which causes an interrupt if INEA = 1. The C-LANCE will then skip over the rest of the descriptors for this packet (clears the OWN bit and sets the TINT bit in CSR0) until it finds a buffer with both the STP and OWN bit being set (this indicates the first buffer for the next packet).

When the transmit buffers are not data chained (current descriptor's ENP = 1), the C-LANCE will not perform any lookahead operation. It will transmit the current buffer, update the TMD3 if any error, and then update the status and clear the OWN bit in TMD1 . The C-LANCE will then immediately check the next descriptor in the ring to see if it owns it. If it does, the C-LANCE will also read the rest of the entries from the descriptor table. If the C-LANCE does not own it, it will poll the ring once every 1.6 ms until it owns it. User may set the TDMD bit in CSR0 when it has relinquished a buffer to the C-LANCE. This will force the C-LANCE to check the OWN bit at this buffer without waiting for the polling time to elapse.

Receive Ring Buffer Management

Receive Ring access is similar to the transmit ring access. Once the receiver is enabled, the C-LANCE will always try to have a receive buffer available, should there be a packet addressed to this node for reception. Therefore, when the C-LANCE is idle, it will poll the receive ring entry once every 1.6 ms, until it owns the current receive DTE. Once the C-LANCE owns the buffer, it will read RMD0 and RMD2 to get the rest of buffer address and buffer byte count. When a packet arrives from the physical medium, after the Address Recognition Logic accepts the packet, the C-LANCE will immediately poll the Receiver Ring once for a buffer. If it still does not own the buffer, it will set the MISS error in CSR0 and will not poll the receive ring until the packet ends.

Assuming the C-LANCE owns a receive buffer when the packet arrives, it will perform a lookahead operation on the next DTE between periods when it is dumping the received data from the Receive FIFO to the first receive buffer in case the current buffer requires data chaining. When the C-LANCE owns the buffer, the lookahead operation consists of three separate single word DMA reads: RMD1, RMD0, and RMD2. When the C-LANCE does not own the next buffer, the lookahead operation consists of only one single DMA read, RMD1. Either lookahead operation is done only once. Following the lookahead operation, whether C-LANCE owns the next buffer or not, the C-LANCE will transfer the data from Receive FIFO to the first receive buffer for this packet in burst mode (8 word transfer per one DMA cycle arbitration).

If the packet being received requires data chaining, and the C-LANCE does not own the second DTE, the C-LANCE will update the current buffer status, RMD1, with the BUFF and/or OFLO error bits set. If the C-LANCE does own the next buffer (second DTE) from previous lookahead, the C-LANCE will relinquish the current buffer and start filling up the second buffer for this packet. Between the time that the C-LANCE is transferring data from the Receive FIFO to the second buffer, it does a lookahead operation again to see if it owns the next (third) buffer. If the C-LANCE does own the third DTE, it will also read RMD0, and RMD2 to get the rest of buffer pointer address and buffer byte count.

This activity continues on until the C-LANCE recognizes the end of the packet (physical medium is idle); it then updates the current buffer status with the end of packet bit (ENP) set. The C-LANCE will also update the message byte count (RMD3) with the total number of bytes received for this packet in the current buffer (the last buffer for this packet).

The dual FIFOs in the C-LANCE are utilized by the internal microcode to guarantee that continuous receive activity does not prevent the servicing of pending transmit packets. The microcode includes a single transmit descriptor poll operation at the beginning of buffer DMA operations for an incoming receive packet. This single transmit descriptor poll is performed only once during the receive microcode routine for each packet that is received. If the OWN bit in the transmit descriptor is set, burst transfers to the Transmit FIFO are interleaved with burst transfers from the Receive FIFO. By interleaving the transmit buffer transfers with the receive buffer transfers, the beginning of the transmit packet is preloaded in the Transmit FIFO, ready to be transmitted immediately following the end of the receive packet on the wire.

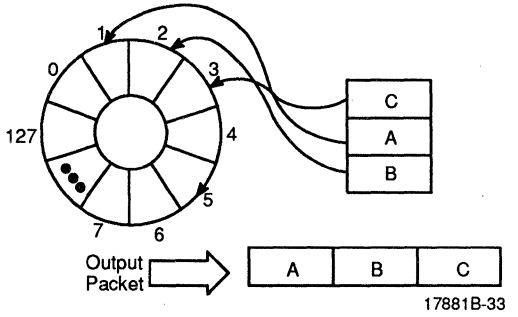
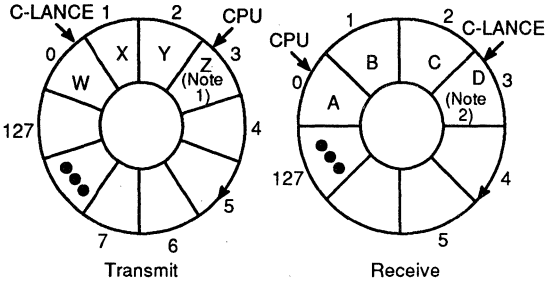


Figure 8-1. Data Chaining (Transmit)



Notes:

1. *W, X, Y, Z are the packets queued for transmission.*
2. *A, B, C, D are the packets received by the C-LANCE.*

Figure 8-2. Buffer Management Descriptor Rings

C-LANCE DMA Transfer (Bus Master Mode)

There are two types of DMA Transfers with the C-LANCE:

- Burst mode DMA
- Single word DMA

Burst Mode DMA

Burst DMA is used for Transmission or Reception of the Packets, (Read/Write from/to Memory).

The Burst Transfers are 8 consecutive word reads (transmit) or writes (receive) that are done in a single bus arbitration cycle. In other words, once the C-LANCE receives the bus acknowledge, ($\overline{HLDA}$ = LOW), it will do 8 word transfers (8 DMA cycle, min. at 600 ns per cycle) without releasing the bus request signal ($\overline{HOLD}$ = LOW). If there are more than 16 bytes empty in the Transmit FIFO, in transmit mode, or at least 16 bytes of data, in the Receive FIFO in receive mode, when the C-LANCE releases the bus ($\overline{HOLD}$ deasserted), the C-LANCE will request the bus again within 700 ns ($\overline{HOLD}$ dwell time). Burst DMAs are always 8 transfer cycles unless there are fewer than 8 words left to be transferred to/from the Transmit/Receive FIFO, or if there are fewer than 8 words left to be transferred to/from the RX/TX buffer. Transmit DMAs may be shorter than 8 words if a collision is detected during the DMA.

Single Word DMA Transfer

The C-LANCE initiates single word DMA transfers to access the transmit and receive rings or the initialization block. The C-LANCE will not initiate any burst DMA transfers while reading the initialization block. The C-LANCE will not initiate any burst DMA transfers between the time that it discovers ownership of a descriptor and the time that it reads the buffer pointer and buffer byte count entries of that descriptor.

FIFO Operation

The dual FIFOs provide temporary buffer storage for data being transferred between the parallel bus I/O pins and serial I/O pins. The capacity of the Transmit FIFO is 48 bytes and the Receive FIFO is 64 bytes.

Transmit

Data is loaded into the Transmit FIFO under internal microprogram control. The Transmit FIFO has to have more than 16 bytes empty before the C-LANCE requests the bus ($\overline{HOLD}$ is asserted). The C-LANCE will start sending the preamble (if the line is idle) as soon as the first byte is loaded to the Transmit FIFO from memory.

Receive

Data is loaded into the Receive FIFO from the serial input shift register during reception. Data leaves the Receive FIFO under microprogram control. The C-LANCE microcode will wait until there are at least 16 bytes of data in the Receive FIFO before initiating a DMA burst transfer. Preamble and Start Frame Delimiter (SFD) are not loaded into the Receive FIFO.

FIFOs – Memory Byte Alignment

Memory buffers may begin and end on arbitrary byte boundaries. Parallel data is byte aligned between the Transmit or Receive FIFO and DAL lines (DAL0–DAL15). Byte alignment can be reversed by setting the Byte Swap (BSWP) bit in CSR3.

TRANSMISSION – WORD READ FROM EVEN MEMORY ADDRESS

BSWP=0: FIFO BYTE n gets DAL <07:00>
FIFO BYTE n + 1 gets DAL <15:08>

BSWP=1: FIFO BYTE n gets DAL <15:08>
FIFO BYTE n + 1 gets DAL <07:00>

TRANSMISSION – BYTE READ FROM EVEN MEMORY ADDRESS

BSWP=0:	FIFO BYTE n	gets DAL <07:00>
	–don't care	gets DAL <15:08>
BSWP=1:	FIFO BYTE n	gets DAL <15:08>
	–don't care	gets DAL <07:00>

TRANSMISSION – BYTE READ FROM ODD MEMORY ADDRESS

BSWP=0:	FIFO BYTE n	gets DAL <15:08>
	–don't care	gets DAL <07:00>
BSWP=1:	FIFO BYTE n	gets DAL <07:00>
	–don't care	gets DAL <15:08>

RECEPTION – WORD WRITE TO EVEN MEMORY ADDRESS

BSWP=0:	DAL <07:00>	gets FIFO BYTE n
	DAL <15:08>	gets FIFO BYTE n + 1
BSWP=1:	DAL <15:08>	gets FIFO BYTE n
	DAL <07:00>	gets FIFO BYTE n + 1

RECEPTION – BYTE WRITE TO EVEN MEMORY ADDRESS

BSWP=0:	DAL <07:00>	gets FIFO BYTE n
	DAL <15:08>	–undefined
BSWP=1:	DAL <15:08>	gets FIFO BYTE n
	DAL <07:00>	–undefined

RECEPTION – BYTE WRITE TO ODD MEMORY ADDRESS

BSWP=0:	DAL <07:00>	–undefined
	DAL <15:08>	gets FIFO BYTE n
BSWP=1:	DAL <15:08>	–undefined
	DAL <07:00>	gets FIFO BYTE n

The C-LANCE Recovery and Reinitialization

The transmitter and receiver section of the C-LANCE are turned on via the initialization block (MODE REG: DRX, DTX bits). The state of the transmitter and the receiver are monitored through the CSR0 register (RXON, TXON bits). The C-LANCE must be reinitialized if the transmitter and/or the receiver has not been turned on during the original initialization, and later it is desired to have them turned on. When either the transmitter or receiver shuts off because an error (MERR, UFLO, TX BUFF error), it is necessary to reinitialize the C-LANCE to turn the transmitter and/or receiver back on again. The user should rearrange the descriptors in the transmit or receive ring prior to reinitialization. This is necessary since the transmit and receive descriptor pointers are reset to the beginning of the ring upon initialization.

To reinitialize the C-LANCE, the user must first stop the C-LANCE by setting the stop bit in CSR0. The user needs to reprogram CSR3 because its contents get cleared when the stop bit gets set (CSR3 reprogramming is not needed when default values of BCON, ACON, and BSWP are used; BCON, ACON, and BSWP default values are 0, 0, and 0 respectively). Only then the user may set the INIT bit in CSR0.

It is recommended that the C-LANCE not be re-started, once it has been stopped (STOP = 1 in CSR0), by setting the STRT bit in CSR0 without reinitialization. Restarting the C-LANCE in this way puts the C-LANCE in operation in accordance with the parameters set up in the mode register, but the contents of the descriptor pointers in the C-LANCE will not be guaranteed.

Frame Formatting

The C-LANCE performs the encapsulation/decapsulation function of the data link layer (second layer of ISO model) as follows:

Transmit

In transmit mode, the user must supply the destination address, source address, and Type Field (or Length Field) as a part of data in transmit data buffer memory. The C-LANCE will append the preamble, SFD, and CRC (FCS) to the frame as is shown in Figures 9-1 and 9-2.

Receive

In receive mode, the C-LANCE strips off the preamble and SFD and transfers the rest of the frame, including the CRC bytes (4 bytes), to the memory. The C-LANCE will discard packets with less than 64 bytes (runt packet) and will reuse the receive buffer for the next packet. This is the only case where the packet is discarded after the packet has been transferred to the receive buffer. A runt packet is normally the result of a collision.

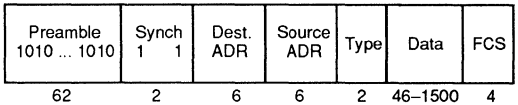


Figure 9-1. Ethernet Frame Format

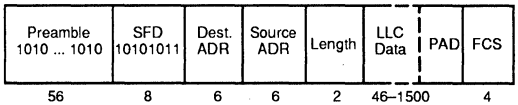


Figure 9-2. IEEE 802.3 MAC Frame Format

Framing Error (Dribbling Bits)

The C-LANCE can handle up to 7 dribbling bits when a received packet terminates; the input to the C-LANCE, RCLK, stops following the deassertion of RENA. During the reception, the CRC is generated on every serial bit (including the dribbling bits) coming from the medium,

and the CRC gets sampled internally on every byte boundary. The framing error is reported to the user as follows:

- If the number of the dribbling bits is 1 to 7 bits and there is no CRC error, then there is no Framing error (FRAM = 0).
- If the number of the dribbling bits is less than 8 and there is a CRC error, then there is also a Framing error (FRAM = 1).
- If the number of the dribbling bits = 0, then there is no Framing error. There may or may not be a CRC error.

Interframe Spacing (IFS)

The C-LANCE implements the two-part deferral algorithm following both receive and transmit activity, as specified as an option in the IEEE 802.3 Standard (ISO/IEC 8802-3 1990). With two-part deferral, the interframe spacing, which begins immediately after the negation of RENA, is divided into two parts, IFS1 and IFS2. If RENA is asserted during IFS1, the interframe spacing counter is continually reset until RENA is deasserted (any pending transmissions will defer to the incoming receive traffic and the incoming frame may be received by the C-LANCE). Once the interframe spacing counter reaches IFS2, the counter proceeds, regardless of the state of RENA. When IFS2 expires, the C-LANCE may begin transmitting a frame if there is one pending.

In the C-LANCE, IFS1 is 6.0 μs and IFS2 is 3.6 μs, making the minimum possible interframe spacing 9.6 μs. The 9.6 μs minimum interframe spacing complies with IEEE 802.3 specifications.

Following each frame transmission, the C-LANCE blinds itself from any receive activity for the first 4.1 μs of the interframe spacing. The C-LANCE begins looking for the 011 start frame delimiter pattern after 800ns (8 bit times) of preamble has passed. Hence, if RENA is asserted during the first 4.1 μs of the interframe spacing, there must be at least 8 bits of preamble left following the end of the 4.1 μs window in order for the frame to be received correctly.

Following each frame reception, the C-LANCE blinds itself from any receive activity for the first 0.5 μs of the interframe spacing.

Collision Detection and Collision JAM

Collisions are detected by monitoring the CLSN pin. If CLSN becomes asserted during a frame transmission, TENA will remain asserted for at least 32 (but not more than 40) additional bit times (including CLSN synchronization). This additional transmission after collision is referred to as COLLISION JAM. If collision occurs during the transmission of the preamble, the C-LANCE continues to send the preamble, and sends the JAM pattern following the preamble. If collision occurs after the preamble, the C-LANCE will send the JAM pattern following the transmission of the current byte. The JAM pattern is any pattern except the CRC bytes.

Receive Based Collision

If CLSN becomes asserted during the reception of a packet, this reception is immediately terminated. Depending on the timing of COLLISION DETECTION, one of the following will occur. A collision that occurs within 6 byte times of the detection of the SFD (4.8 μs) will result in the packet being rejected because of an address mismatch; the Receive FIFO write pointer will be reset. A collision that occurs within 64 byte times (51.2 μs) will result in the packet being rejected since it is a runt packet. A collision that occurs after 64 byte times (late collision) will result in a truncated packet being written to the memory buffer with the CRC error bit most likely being set in the Status Word of the Receive Ring. Late collision error is not reported in receive mode.

Transmit Based Collision

When a transmission attempt has been terminated due to the assertion of CLSN, (a collision that occurs within 64 byte times), the C-LANCE will attempt to retry transmission 15 more times. The scheduling of the retransmissions is determined by a controlled randomized process called "truncated binary exponential backoff." Upon the negation of the COLLISION JAM interval, the C-LANCE calculates a delay before retransmitting. The delay is an integral multiple of the SLOT TIME. The SLOT TIME is 512 bit times. The number of SLOT TIMES to delay before the nth retransmission is chosen as a uniformly distributed random integer in the range: $0 \leq r \leq 2^k$ where k = min (n, 10).

When the Modified Backoff Algorithm is enabled (EMBA), the backoff time may be longer than the minimum time specified above. Specifically, the backoff count will be suspended whenever a carrier is detected on the network. The backoff count will resume when the carrier drops. This behavior has the effect of making the backoff interval equal to the SUM of an integral number of SLOT TIMES plus the total duration of the carrier on the network during the backoff interval.

If all 16 attempts fail, the C-LANCE sets the RTRY bit in the current Transmit Message Descriptor 3, TMD3, in memory, gives up ownership (sets the own bit to zero) for this packet, and processes the next packet in transmit ring for transmission. If there is a late collision (collision occurring after 64 byte times), the C-LANCE will not attempt to transmit this packet again; it will terminate the transmission, note the LCOL error in TMD3, and transmit the next packet in the ring.

Collision—Microcode Interaction

The microprogram uses the time provided by COLLISION JAM, INTERPACKET DELAY, and the backoff interval to restore the address and byte counts internally and starts loading the Transmit FIFO in anticipation of retransmission. It is important that C-LANCE be ready to transmit when the backoff interval elapses to utilize the channel properly.

If, during the backoff interval, RENA and CLSN are never asserted (no wire activity), the C-LANCE does not re-poll the OWN bit and does not re-read the buffer address and byte count in the transmit descriptor before reloading the transmit data and retransmitting the transmit packet. However, if RENA or CLSN are asserted during the backoff interval, the C-LANCE must re-poll the OWN bit and re-read the buffer address and byte count in the transmit descriptor before starting the DMA access of the transmit buffer and performing the retry. Note that the re-polling of the transmit descriptor could be preceeded by receive DMA operations if an incoming packet arrives during the backoff interval and an address match is detected or when the C-LANCE is in promiscuous mode.

Time Domain Reflectometry

The C-LANCE contains a time domain reflectometry counter. The TDR counter is ten bits wide. It counts at a 10 MHz rate. It is cleared by the microprogram and counts upon the assertion of RENA during transmission. Counting ceases if CLSN becomes true, or RENA goes inactive. The counter does not wrap around. Once all ONEs are reached in the counter, the counter value is held until cleared. The value in the TDR is written into memory following the transmission of the packet. TDR is used to determine the location of suspected cable faults.

Heartbeat

During the interpacket gap time following the negation of TENA, the CLSN input is asserted by some transceivers as a self-test. If the CLSN input is not asserted within 4 µs following the completion of transmission, then the C-LANCE will set the CERR bit in CSR0. CERR error will not cause an interrupt to occur (INTR = 0).

Cyclic Redundancy Check (CRC)

The C-LANCE utilizes the 32-bit CRC function as described in the IEEE 802.3 standard section 3.2.8 to generate the Frame Check Sequence (FCS) field. The C-LANCE requirements for the CRC logic are the following:

- TRANSMISSION – MODE <02> LOOP = 0, MODE <03> DTCR = 0. The C-LANCE calculates the CRC from the first bit following the SFD to the last bit of the data field. The CRC value inverted is appended onto the transmission in one unbroken bit stream.
- RECEPTION – MODE <02> LOOP = 0. The C-LANCE performs a check on the input bit stream from the first bit following the SFD to the last bit in the frame. The C-LANCE continually samples the state of the CRC check on framed byte boundaries, and, when the incoming bit stream stops, the last sample determines the state of the CRC error. Framing error (FRAM) is not reported if there is no CRC error.
- LOOPBACK – MODE <02> LOOP =1, MODE <03> DTRC = 0. The C-LANCE generates and appends the CRC value to the outgoing bit stream as in Transmission but does not perform the CRC check of the incoming bit stream.
- LOOPBACK – MODE <02> LOOP = 1 MODE <03> DTRC = 1. C-LANCE performs the CRC check on the incoming bit stream as in Reception, but does not generate or append the CRC value to the outgoing bit stream during transmission.

Loopback

The normal operation of the C-LANCE is as a half-duplex device. However, to provide an on-line operational test of the C-LANCE, a pseudo-full duplex mode is provided. In this mode simultaneous transmission and reception of a loopback packet are enabled with the following constraints:

- The packet length must be no longer than 32 bytes, and no shorter than 8 bytes, exclusive of the CRC.
- Serial transmission does not begin until the Transmit FIFO contains the entire output packet.
- Moving the input packet from the Receive FIFO to the memory does not begin until the serial input bit stream terminates.
- CRC may be generated and appended to the output serial bit stream or may be checked on the input serial bit stream. CRC may not be used for both transmission and reception simultaneously.
- In internal loopback, the packets should be addressed to the node itself.
- In external loopback, multicast addressing can be used only when DTCR = 1 is in the mode register. In this case, the user needs to append the CRC bytes.

Loopback is controlled by bits <06, 03, 02> INTL, DTCR, and LOOP of the MODE register.

Serial Transmission

Serial transmission consists of sending an unbroken bit stream from the TX output pin consisting of:

- Preamble/SFD: 56 alternating ONES and ZEROES terminating with the SFD byte (10101011).
- Data: The serialized bit stream from the Transmit FIFO Shifted out with LSB first.
- CRC: The inverted 32-bit polynomial calculated from the data, address, and type field. CRC is not transmitted if:
  - —Transmission of the data field is truncated for any reason.
  - — CLSN becomes asserted any time during transmission.
  - — MODE <03> DTCR = 1 in a normal or loopback transmission mode, and ADD_FCS=0 in the transmit descriptor.

The Transmission is indicated at the output pin by the assertion of TENA with the first bit of the preamble and the negation of TENA after the last transmitted bit.

The C-LANCE starts transmitting the preamble when the following are satisfied:

- There is at least one byte of data to be transmitted in the Transmit FIFO.
- The interpacket delay has elapsed.
- The backoff interval has elapsed, if doing a retransmission.

Serial Reception

Serial reception consists of receiving an unbroken bit stream on the RX input pin consisting of:

- Preamble/SFD: Two ONES occurring a minimum of 8 bit times after the assertion of RENA.
- Destination Address: The 48 bits (6 bytes) following the SFD.
- Data: The serial bit stream following the Destination Address. The last 4 complete bytes of data are the CRC. The Destination Address and the data are framed into bytes and enter the Receive FIFO. Source Address and Length field are part of the data which are transparent to the C-LANCE.

Reception is indicated at the input pin by the assertion of RENA and the presence of clock on RCLK while TENA is inactive. The C-LANCE does not sample the received data until about 800 ns after RENA goes high.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C

Ambient Temperature with Power Applied ..... −25°C to +125°C

Supply Voltages to Ground Potential Continuous ..... −0.3 V to +6 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A) ..... 0°C to +70°C

Supply Voltage (V_{DD}) ..... +4.75 V to +5.25 V

V_{SS} ..... 0 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Commercial			Unit
			Min	Typ	Max	
V_{IL}	Input LOW Voltage				0.8	V
V_{IH}	Input HIGH Voltage		2			V
V_{OL}	Output LOW Voltage	I_{OL} = 3.2 mA			0.5	V
V_{OH}	Output HIGH Voltage	I_{OH} = −0.4 mA	2.4			V
I_{IL}	Input Leakage	V_{IN} = 0.4 V to V_{CC}			±10	µA
I_{DD}*	Power Supply Current				50	mA

**I_{DD} is measured while running a functional pattern with spec. value I_{OH} and I_{OL} load applied.*

CAPACITANCE** (T_A = 25°C; V_{DD} = 0)

Parameter Symbol	Parameter Description	Test Conditions	Min	Typ	Max	Unit
C_{IN}	Input Pin Capacitance	f = 1 MHz			10	pF
C_{OUT}	Output Pin Capacitance	f = 1 MHz			15	pF
C_{IO}	I/O Pin Capacitance	f = 1 MHz			20	pF

***Parameters are not tested.*

SWITCHING CHARACTERISTICS

No.	Parameter Symbol	Parameter Description	Test Conditions	Min	Typ	Max	Unit
1	t_{TCT}	TCLK Period		99		101	ns
2	t_{TCL}	TCLK LOW Time		45		55	ns
3	t_{TCH}	TCLK HIGH Time		45		55	ns
4	t_{TCR}	Rise Time of TCLK	(Note 3)			8	ns
5	t_{TCF}	Fall Time of TCLK	(Note 3)			8	ns
6	t_{TEP}	TENA Propagation Delay After the Rising Edge of TCLK				60	ns
7	t_{TEH}	TENA Hold Time After the Rising Edge of TCLK		5			ns
8	t_{TDP}	TX Data Propagation Delay After the Rising Edge of TCLK				60	ns
9	t_{TDH}	TX Data Hold Time After the Rising Edge of TCLK		5			ns
10	t_{RCT}	RCLK Period	(Note 3)	85		118	ns
11	t_{RCH}	RCLK HIGH Time	(Note 2)	38			ns
12	t_{RCL}	RCLK LOW Time	(Note 2)	38			ns
13	t_{RCR}	Rise Time of RCLK	(Note 3)			8	ns
14	t_{RCF}	Fall Time of RCLK	(Note 3)			8	ns
15	t_{RDR}	RX Data Rise Time	(Note 3)			8	ns
16	t_{RDF}	RX Data Fall Time	(Note 3)			8	ns
17	t_{RDH}	RX Data Hold Time (RCLK to RX Data Change)	(Note 2)	5			ns
18	t_{RDS}	RX Data Setup Time (RX Data Stable to the Rising Edge of RCLK)	(Note 2)	35			ns
19	t_{DPL}	RENA LOW Time		$1t_{TCT} + 20$			ns
20	t_{CPH}	CLSN HIGH Time		80			ns
21	t_{DOFF}	Bus Master Driver Disable After Rising Edge of $\overline{HOLD}$				50	ns
22	t_{DON}	Bus Master Driver Enable After Falling Edge of $\overline{HLDA}$		50		$2t_{TCT} + 50$	ns
23	t_{HHA}	Delay to Falling Edge of $\overline{HLDA}$ from Falling Edge of $\overline{HOLD}$ (Bus Master)		0			ns
24	t_{RW}	$\overline{RESET}$ Pulse Width LOW	(Note 7)	$2t_{TCT}$			ns
25	t_{CYCLE}	Read/Write, Address/Data Cycle Time	(Note 1)	$6t_{TCT}$			ns
26	t_{XAS}	Address Setup Time to the Falling Edge of ALE		75			ns
27	t_{XAH}	Address Hold Time After the Rising Edge of $\overline{DAS}$		35			ns
28	t_{AS}	Address Setup Time to the Falling Edge of ALE		75			ns
29	t_{AH}	Address Hold Time After the Falling Edge of ALE		35			ns
30	t_{RDAS}	Data Setup Time to the Rising Edge of $\overline{DAS}$ (Bus Master Read)		40			ns

SWITCHING CHARACTERISTICS (continued)

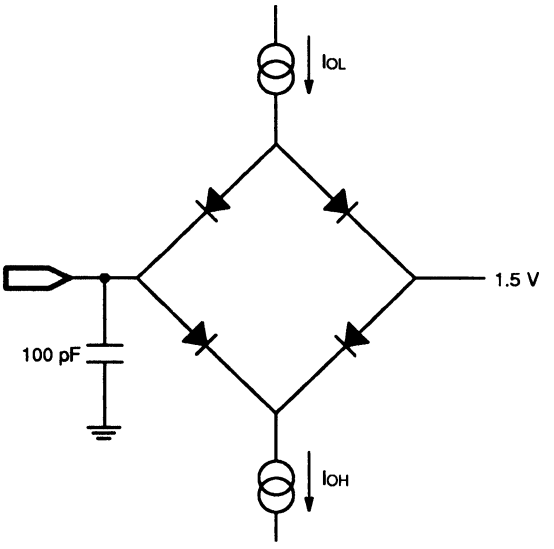
No.	Parameter Symbol	Parameter Description	Test Conditions	Min	Typ	Max	Unit
31	t_{RDAH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master Read)		0			ns
32	t_{DDAS}	Data Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Master Write)		10			ns
33	t_{WDS}	Data Setup Time to the Rising Edge of $\overline{DAS}$ (Bus Master Write)		200			ns
34	t_{WDH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master Write)		35			ns
35	t_{SD01}	Data Driver Delay After the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CSR0, CSR3, RAP) (Note 6)		$4t_{TCT}$		ns
36	t_{SD02}	Data Driver Delay After the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CSR1, 2) (Note 6)		$12t_{TCT}$		ns
37	t_{SRDH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave Read)		0		55	ns
38	t_{SWDH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave Write)		0			ns
39	t_{SWDS}	Data Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave Write)		0			ns
40	t_{ALEW}	ALE Width HIGH		120			ns
41	t_{DALE}	Delay from Rising Edge of $\overline{DAS}$ to the Rising Edge of ALE		70			ns
42	t_{DSW}	$\overline{DAS}$ Width LOW		200			ns
43	t_{ADAS}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{DAS}$		80		130	ns
44	t_{RIDF}	Delay from the Rising of $\overline{DALO}$ to the Falling Edge of $\overline{DAS}$ (Bus Master Read)		15			ns
45	t_{RDYS}	Delay from the Falling Edge of $\overline{READY}$ to the Rising Edge of $\overline{DAS}$		65		250	ns
46	t_{ROIF}	Delay from the Rising Edge of $\overline{DALO}$ to the Falling Edge of $\overline{DALI}$ (Bus Master Read)		15			ns
47	t_{RIS}	$\overline{DALI}$ Setup Time to the Rising Edge of $\overline{DAS}$ (Bus Master)		135			ns
48	t_{RIH}	$\overline{DALI}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master Read)		0			ns
49	t_{RIOF}	Delay from the Rising Edge of $\overline{DALI}$ to the Falling Edge of $\overline{DALO}$ (Bus Master Read)		55			ns
50	t_{OS}	$\overline{DALO}$ and READ Setup Time to the Falling Edge of ALE (Bus Master Write and Read)		110			ns
51	t_{ROH}	$\overline{DALO}$ Hold Time After the Falling Edge of ALE (Bus Master Read)		35			ns
52	t_{WDSI}	Delay from the Rising Edge of $\overline{DAS}$ to the Rising Edge of $\overline{DALO}$ (Bus Master Write)		35			ns
53	t_{CSH}	$\overline{CS}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0			ns
54	t_{CSS}	$\overline{CS}$ Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave)		0			ns

SWITCHING CHARACTERISTICS (continued)

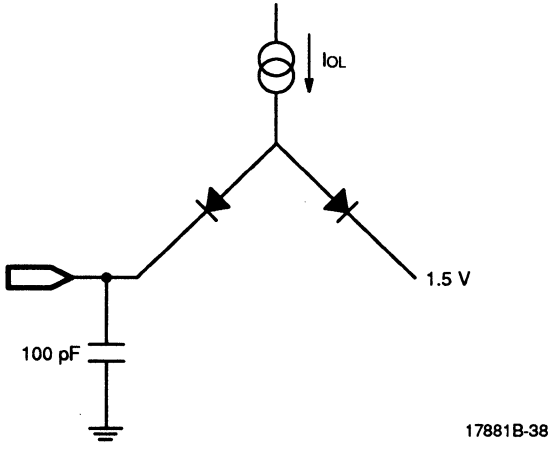
No.	Parameter Symbol	Parameter Description	Test Conditions	Min	Typ	Max	Unit
55	t_{SAH}	ADR Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0			ns
56	t_{SAS}	ADR Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave)		0			ns
57	t_{ARYD}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{READY}$ to insure a Minimum Bus Cycle Time (600 ns)	(Note 5)			80	ns
58	t_{SRDS}	Data Setup Time to the Falling Edge of $\overline{READY}$ (Bus Slave Read)		75			ns
59	t_{RDYH}	$\overline{READY}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master)		0			ns
60	t_{SR01}	$\overline{READY}$ Driver Turn On After the Falling Edge of $\overline{DAS}$ (Bus Slave)	(CSR0, CSR3, RAP) (Notes 4, 6)		$6t_{TCT}$		ns
61	t_{SR02}	$\overline{READY}$ Driver Turn On After the Falling Edge of $\overline{DAS}$ (Bus Slave)	(CSR1, 2) (Note 6)		$14t_{TCT}$		ns
62	t_{SRYH}	$\overline{READY}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0		35	ns
63	t_{SRH}	$\overline{READ}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0			ns
64	t_{SRS}	$\overline{READ}$ Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave)		0			ns
65	t_{CHL}	TCLK Rising Edge to $\overline{HOLD}$ LOW or High Delay				95	ns
66	t_{CAV}	TCLK to Address Valid				100	ns
67	t_{CCA}	TCLK Rising Edge to Control Signals Active				75	ns
68	t_{CALE}	TCLK Falling Edge to ALE LOW				90	ns
69	t_{CDL}	TCLK Falling Edge to $\overline{DAS}$ Falling Edge				90	ns
70	t_{RCS}	Ready Setup Time to TCLK Falling Edge	(Note 5)	0			ns
71	t_{CDH}	TCLK Rising Edge to $\overline{DAS}$ HIGH				90	ns
72	t_{HCS}	$\overline{HLDA}$ Setup to TCLK Falling Edge		0			ns
73	t_{RENH}	RENA Hold Time After the Rising Edge of RCLK		0			ns
74	t_{CSR}	$\overline{CS}$ recovery time between deassertion of $\overline{CS}$ or $\overline{HOLD}$ and assertion of $\overline{CS}$		$t_{TCT}+60$			ns

Notes:

1. *Not shown in the timing diagrams, specifies the minimum bus cycle for a single DMA data transfer. Tested by functional data pattern.*
2. *Applicable parameters associated with Receive circuit are tested at t_{RCT} (RCLK Period) = 100 ns, t_{TCT} = 100 ns (TCLK Period).*
3. *Not tested.*
4. *CSR0 write access time (t_{SR01}) when STOP bit is being set can be as long as $12t_{TCT}$.*
5. *It is guaranteed that no wait states will be added by the C-LANCE if either parameter #57 or #70 is met.*
6. *Parameter is for design reference only.*
7. *Reset must be asserted for at least two rising and two falling edges of TCLK for the device to be reset. If reset is deasserted before TCLK starts, the device behavior is undefined.*



A. Normal and Three-State Outputs



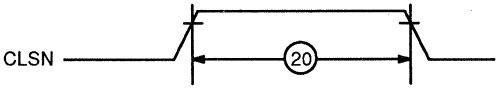
B. Open-Drain Outputs ($\overline{\text{INTR}}$, $\overline{\text{HOLD}}$/$\overline{\text{BUSRQ}}$, $\overline{\text{READY}}$)

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	Must Be Steady	Will Be Steady
	May Change from H to L	Will Be Changing from H to L
	May Change from L to H	Will Be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off" State

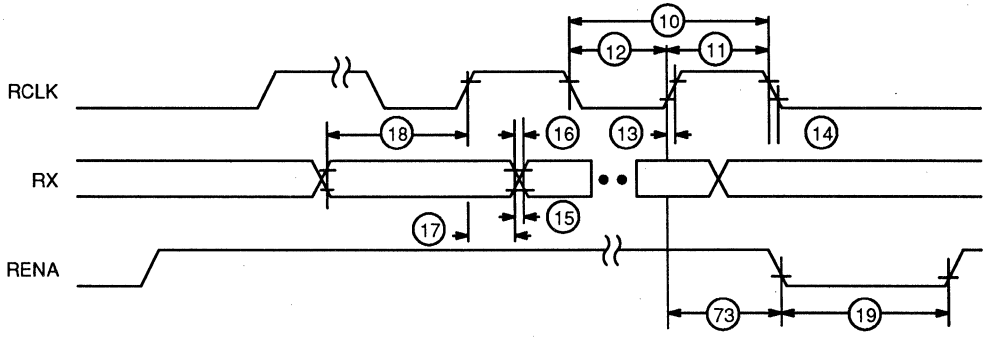
KS000010

SWITCHING WAVEFORMS (Note 1)



Serial Link Timing (Collision)

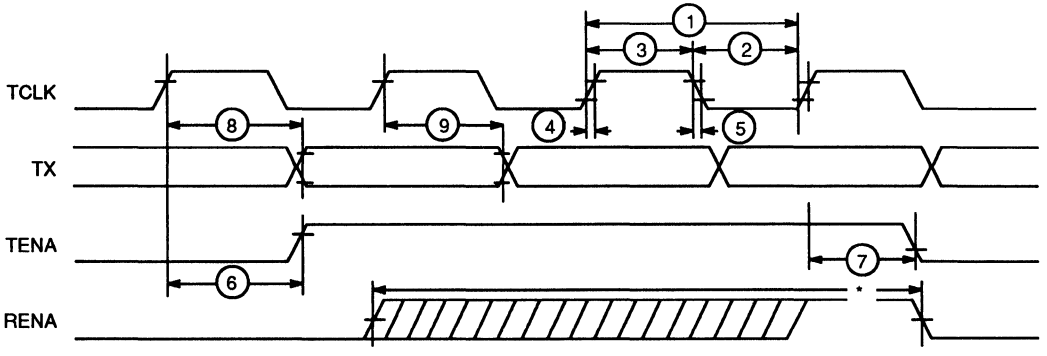
17881B-39



17881B-40

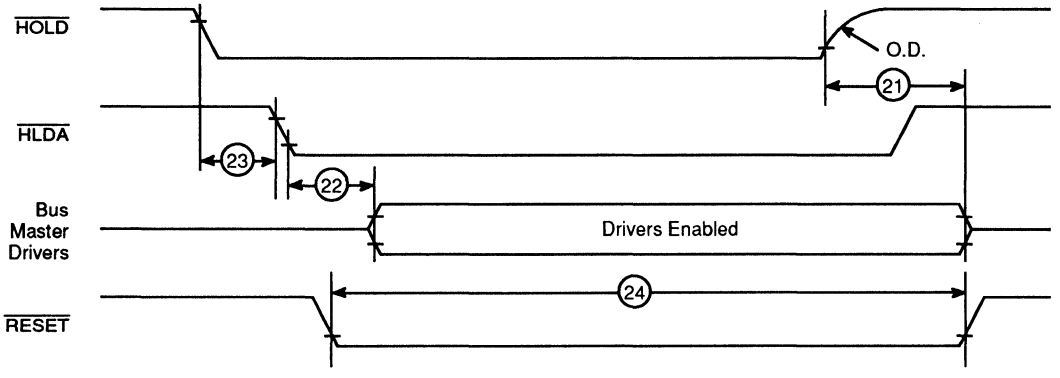
Serial Link Timing (Receive)

SWITCHING WAVEFORMS



**During transmit, RENA input must be asserted (HIGH) and remain active-HIGH before TENA goes inactive (LOW). If RENA is deasserted before TENA is deasserted, LCAR will be reported in TMD3 after the transmission is completed by the C-LANCE.*

Serial Link Timing (Transmit)

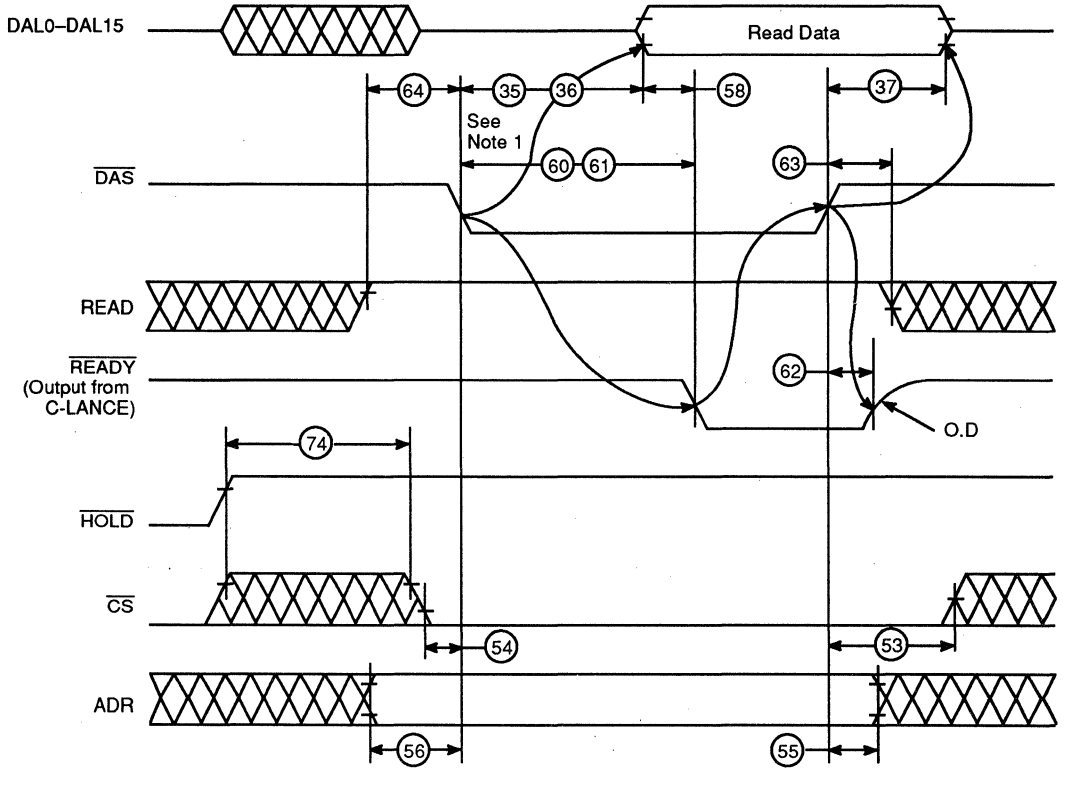


Note:

1. *$\overline{RESET}$ is an asynchronous input to the C-LANCE and is not part of the Bus Acquisition timing. When $\overline{RESET}$ is asserted, the C-LANCE becomes a Bus Slave.*

Bus Acquisition Timing

SWITCHING WAVEFORMS



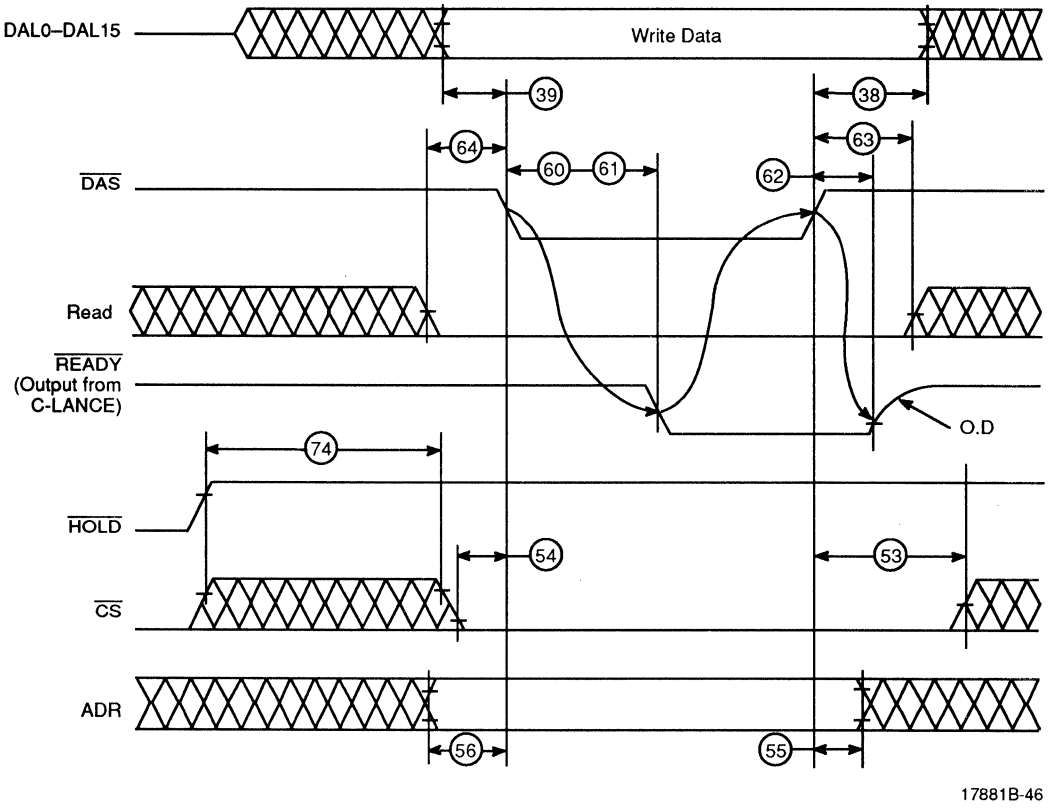
17881B-45

Note:

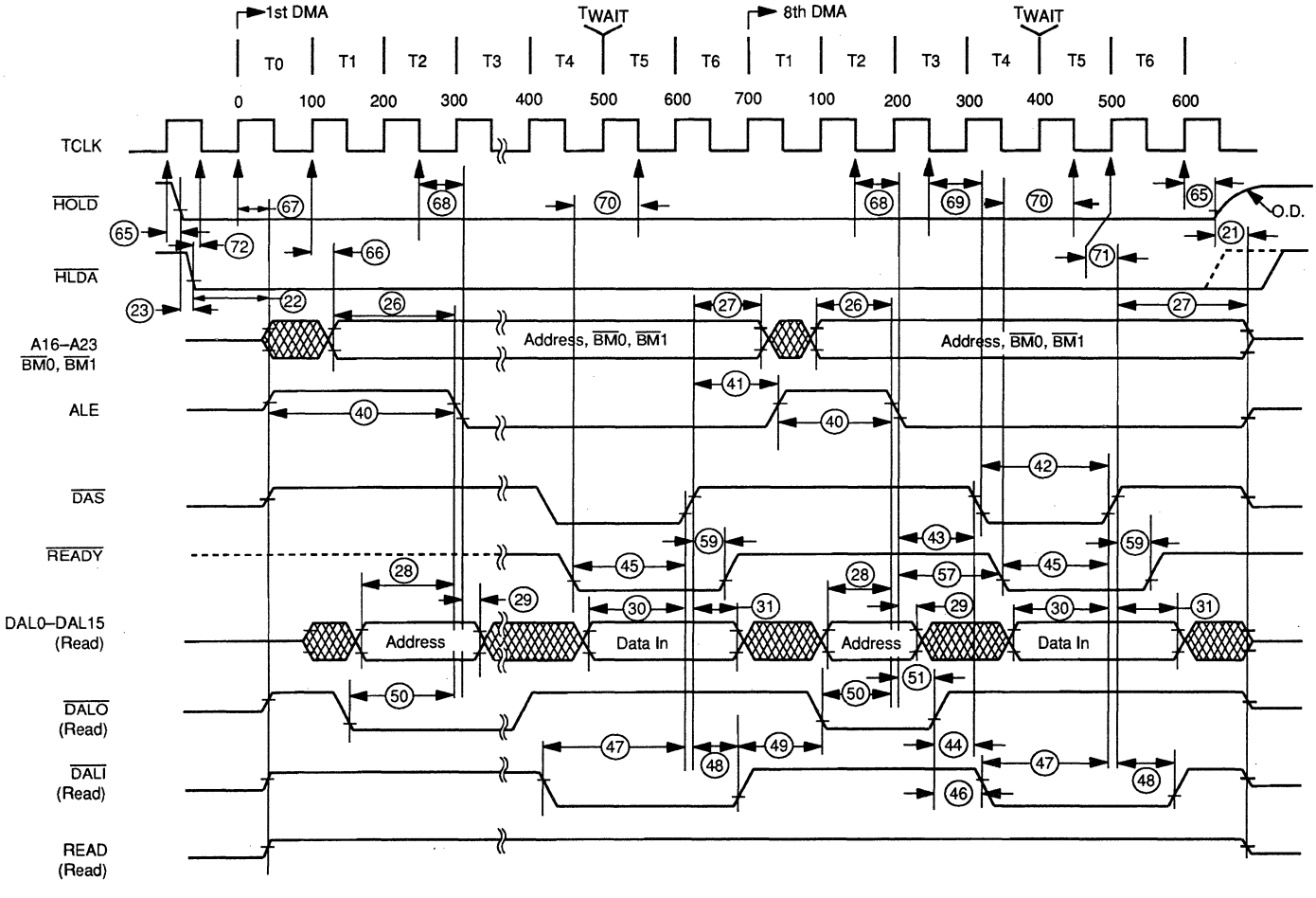
1. *There are two types of delays which depend on which internal register is accessed.*
   Type 1 refers to access of CSR0 CSR3 and RAP.
   Type 2 refers to access of CSR1 and CSR2 which are longer than Type 1 delay.

Bus Slave Read Timing

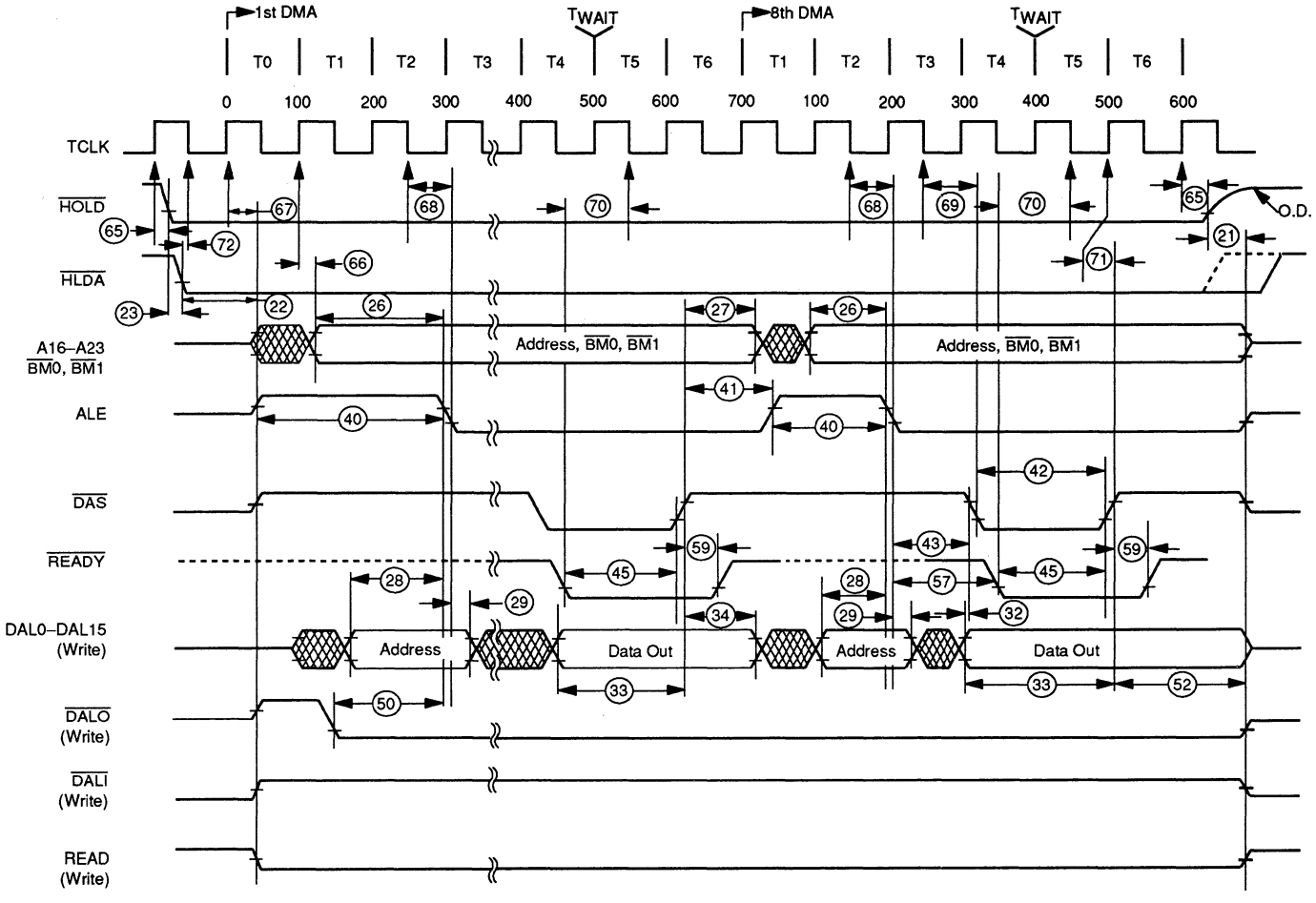
SWITCHING WAVEFORMS



Bus Slave Write Timing



Bus Master Read Timing (Burst DMA)



Bus Master Write Timing (Burst DMA)

APPENDIX A

Hash Filter Generation Programs for Logical Addressing

80x86 computer program example to generate the hash filter, for multicast addressing in the C-LANCE.

```
6                       ;       SUBROUTINE TO SET A BIT IN THE HASH FILTER FROM A
7                       ;       GIVEN ETHERNET LOGICAL ADDRESS
8                       ;       ON ENTRY SI POINTS TO THE LOGICAL ADDRESS WITH LSB FIRST
9                       ;                DI POINTS TO THE HASH FILTER WITH LSB FIRST
10                      ;       ON RETURN SI POINTS TO THE BYTE AFTER THE LOGICAL ADDRESS
11                      ;           ALL OTHER REGISTERS ARE UNMODIFIED
12                      ;
13                              PUBLIC SETHASH
14                              ASSUME CS:CSE61
15                      ;
16  = 1DB6              POLYL   EQU     1DB6H       ;CRC POLYNOMINAL TERMS
17  = 04C1              POLYH   EQU     04C1H
18                      ;
19  0000                CSE61   SEGMENT PUBLIC 'CODE'
20                      ;
21  0000                SETHASH PROC    NEAR
22  0000 50                     PUSH    AX          ;SAVE ALL REGISTERS
23  0001 53                     PUSH    BX
24  0002 51                     PUSH    CX
25  0003 52                     PUSH    DX
26  0004 55                     PUSH    BP
27                      ;
28  0005 B8 FFFF                MOV     AX,0FFFFH   ;AX,DX =CRC ACCUMULATOR
29  0008 BA FFFF                MOV     DX,0FFFFH   ;PRESET CRC ACCUMULATOR TO ALL 1'S
30  000B B5 03                  MOV     CH,3        ;CH =WORD COUNTER
31                      ;
32  000D 8B 2C          SETH10: MOV     BP,[S1]     ;GET A WORD OF ADDRESS
33  000F 83 C6 02               ADD     S1,2        ;POINT TO NEXT ADDRESS
34  0012 B1 10                  MOV     CL,16       ;CL=BIT COUNTER
35                      ;
36  0014 8B DA          SETH20: MOV     BX,DX       ;GET HIGH WORD OF CRC
37  0016 D1 C3                  ROL     BX,1        ;PUT CRC31 TO LSB
38  0018 33 DD                  XOR     BX,BP       ;COMBINE CRC31 WITH INCOMING BIT
39  001A D1 E0                  SAL     AX,1        ;LEFT SHIFT CRC ACCUMULATOR
40  001C D1 D2                  RCL     DX,1
41  001E 81 E3 0001             AND     BX,0001H    ;BX=CONTROL BIT
42  0022 74 07                  JZ      SETH30      ;DO NOT XOR IF CONTROL BIT = 0
43                      ;
44                      ;       PERFORM XOR OPERATION WHEN CONTROL BIT= 1
```

```
45                          ;
46   0024 35 1D 86          XOR     AX,POLYL
47   0027 81 F2 04C1        XOR     DX,POLYH
48                          ;
49   002B 0B C3     SETH30: OR      AX,BX       ;PUT CONTROL BIT IN CRC0
50   002D D1 CD             ROR     BP,1        ;ROTATE ADDRESS WORD
51   002F FE C9             DEC     CL          ;DECREMENT BIT COUNTER
52   0031 75 E1             JNZ     SETH20
53   0033 FE CD             DEC     CH          ;DECREMENT WORD COUNTER
54   0035 75 D6             JNZ     SETH10
55                          ;       FORMATION OF CRC COMPLETE, AL CONTAINS THE REVERSED HASH
56                          ;       CODE
58   0037 B9 000A           MOV     CX,10
49   003A D0 E0     SETH40: SAL     AL,1        ;REVERSE THE ORDER OF BITS IN AL
60   003C D0 DC             RCR     AH,1        ;AND PUT IT IN AH
61   003E E2 FA             LOOP    SETH40
62
63                          ;       AH NOW CONTAINS THE HASH CODE
64                          ;
65   0040 8A DC             MOV     BL,AH       ;BL = HASH CODE, BH IS ALREADY ZERO
66   0042 B1 03             MOV     CL,3        ;DIVIDE HASH CODE BY 8
67   0044 D2 EB             SHR     BL,CL       ;TO GET TO THE CORRECT BYTE
68   0046 B0 01             MOV     AL,01H      ;PRESET FILTER BIT
69   0048 80 E45 07         AND     AH,7H       ;EXTRACT BIT COUNT
70   004B 8A CC             MOV     CL,AH
71   004D D2 E0             SHL     AL,CL       ;SHIFT BIT TO CORRECT POSITION
72   004F 08 01             OR      [DI + BX],AL ;SET IN HASH FILTER
73   0051 5D                POP     BP
74   0052 5A                POP     DX
75   0053 59                POP     CX
76   0054 5B                POP     BX
77   0055 58                POP     AX
78   0056 C3                RET
79                          ;
80   0057           SETHASH ENDP
81                          ;
82   0057           CSEG1   ENDS
83                          ;
84                          END
```

Program example in BASIC to generate the hash filter, for multicast addressing, in the C-LANCE.

```
100  REM
110  REM PROGRAM TO GENERATE A HASH NUMBER GIVEN AN ETHERNET ADDRESS
120  REM
130  DEFINT A-Z
140  DIM A(47): REM ETHERNET ADDRESS. 48 BITS.
150  DIM A$(6): REM INPUT FROM KEYBOARD
160  DIM C(32): REM CRC REGISTER-32 BITS
```

```
170 PRINT "ENTER ETHERNET ADDRESS AS 6 HEXADECIMAL NUMBERS SEPARATED "
180 PRINT "BY BLANKS. EACH NUMBER REPRESENTS ONE BYTE. THE LEAST "
190 PRINT "SIGNIFICANT BIT OF THE FIRST BYTE IS THE FIRST BIT TRANSMITTED."
200 PRINT ""
210 PRINT "ENTER ETHERNET ADDRESS";
220 INPUT A$(0), A$(1), A$(2), A$(3), A$(4), A$(5)
240 REM
250 REM UNPACK ETHERNET ADDRESS INTO ADDRESS ARRAY
260 REM
270 M=0
280 FOR I = 0 TO 47: A(I) = 0: NEXT I
290 FOR I = 0 TO 5
300 IF LEN(A$(I)) = 1 THEN A$(I) = "0" + A$(I)
310 A$(I) = UCASE$(A$(I))
320 FOR N = 2 TO 1 STEP –1
330 Y$ = MID$(A$(I), N, 1)
340 IF Y$ = "0" THEN 510
350 IF Y$ = "1" THEN A(M) = 1: GOTO 510
360 IF Y$ = "2" THEN A(M + 1) = 1: GOTO 510
370 IF Y$ = "3" THEN A(M + 1) = 1: A(M) = 1: GOTO 510
380 IF Y$ = "4" THEN A(M + 2) = 1: GOTO 510
390 IF Y$ = "5" THEN A(M + 2) = 1: A(M) = 1: GOTO 510
400 IF Y$ = "6" THEN A(M + 2) = 1: A(M + 1) = 1: GOTO 510
410 IF Y$ = "7" THEN A(M + 2) = 1: A(M + 1) = 1: A(M) = 1: GOTO 510
420 A(M + 3) = 1
430 IF Y$ = "8" THEN 510
440 IF Y$ = "9" THEN A(M) = 1: GOTO 510
450 IF Y$ = "A" THEN A(M + 1) = 1: GOTO 510
460 IF Y$ = "B" THEN A(M + 1) = 1: A(M) = 1: GOTO 510
470 IF Y$ = "C" THEN A(M + 2) = 1: GOTO 510
480 IF Y$ = "D" THEN A(M + 2) = 1: A(M) = 1: GOTO 510
490 IF Y$ = "E" THEN A(M + 2) = 1: A(M + 1) = 1: GOTO 510
500 IF Y$ = "F" THEN A(M + 2) = 1: A(M + 1) = 1: A(M) = 1
510 M=M+4
520 NEXT N
530 NEXT I
540 REM
550 REM PERFORM CRC ALGORITHM ON ARRAY A(0–47)
560 REM
570 FOR I = 0 TO 31: C(I) = 1: NEXT I
580 FOR N = 0 TO 47
590 REM SHIFT CRC REGISTER BY 1
600 FOR I = 32 TO 1 STEP –1: C(I) = C(I–1): NEXT I
610 C(0) = 0
620 T = C(32) XOR A(N): REM T = CONTROL BIT
630 IF T = 0 THEN 700: REM JUMP IF CONTROL BIT=0
```

```
640   C(1) = C(1) XOR 1: C(2) = C(2) XOR 1: C(4) = C(4) XOR 1
650   C(5) = C(5) XOR 1: C(7) = C(7) XOR 1: C(8) = C(8) XOR 1
660   C(10) = C(10) XOR 1:·C(11) = C(11) XOR 1: C(12) = C(12) XOR 1
670   C(16) = C(16) XOR 1: C(22) = C(22) XOR 1: C(23) = C(23) XOR 1
680   C(26) = C(26) XOR 1
690   C(0) = 1
700   NEXT N
710   REM
720   REM CRC COMPUTATION COMPLETE, EXTRACT HASH NUMBER FROM C(0) TO C(5)
730   REM
740   HH=32*C(0)+16*C(1)+8*C(2)+4*C(3)+2*C(4)+C(5)
750   PRINT "THE HASH NUMBER FOR ";
760   PRINT A$(0); " "; A$(1); " "; A$(2); " "; A$(3); " "; A$(4); " "; A$(5);
770   PRINT "IS"; HH
780   GOTO 210
```

Program example in C to generate the hash filter, for multicast addressing in the C-LANCE.

```
/*****************************************************************
*  hash.c     Rev 0.1
*  Generate a logical address filter value from a list of
*  Ethernet multicast addresses.
*
*  Input:
*     User is prompted to enter an Ethernet address in
*     Ethernet hex format:  First octet entered is the first
*          octet to appear on the line.  LSB of most
*          significant octet is the first bit on the line.
*          Octets are separated by blanks.
*     After results are printed, user is prompted for
*     another address.
*
*     (Note that the first octet transmitted is stored in
*     the C-LANCE as the least significant byte of the Physical
*     Address Register.)
*  Output:
*     After each address is entered, the program prints the
*     hash code for the last address and the cumulative
*     address filter function.  The filter function is
*     printed as 8 hex bytes, least significant byte first.
*****************************************************************/
#include <stdio.h>
void updateCRC (int bit);
   int adr[6],     /* Ethernet address */
       ladrf[8],   /* Logical address filter */
       CRC[33],    /* CRC register, 1 word/bit + extra control bit */
       poly[] =    /* CRC polynomial.  poly[n] = coefficient of
                      the x**n term of the CRC generator polynomial. */
         {1,1,1,0, 1,1,0,1,
          1,0,1,1, 1,0,0,0,
          1,0,0,0, 0,0,1,1,
          0,0,1,0, 0,0,0,0
         };
void main()
```

```
{
    int k,i, byte; /* temporary array indices */
    int hashcode;  /* the object of this program */
    char buf[80];  /* holds input characters */


  for (i=0;i<8;i++) ladrf[i] = 0; /* clear log. adr. filter */


  printf ("Enter Ethernet addresses as 6 octets separated by blanks.\n");
  printf ("Each octet is one or two hex characters. The first octet \n");
  printf ("entered is the first octet to be transmitted. The LSB of \n");
  printf ("the first octet is the first bit transmitted. After each \n");
  printf ("address is entered, the Logical Address Filter contents \n");
  printf ("are displayed, least significant byte first, with the \n");
  printf ("appropriate bits set for all addresses entered so far.\n");
  printf ("      To exit press the <Enter> key.\n\n");
 while (1)
 {
 loop:
   printf ("\nEnter address: ");

   /* If 1st character = CR, quit, otherwise read address. */
   gets (buf);
   if ( buf[0] == '\0') break;
   if (sscanf (buf, "%x %x %x %x %x %x",
       &adr[0], &adr[1], &adr[2],&adr[3],&adr[4],&adr[5])
     != 6)
     { printf
       ("Address must contain 6 octets separated by blanks.\n");
       goto loop;
     }
   if ((adr[0] & 1) == 0)
     { printf ("First octet of multicast address ");
       printf ("must be an odd number.\n");
       goto loop;
     }

   /* Initialize CRC */
   for (i=0; i<32; i++) CRC[i] = 1;

   /* Process each bit of the address in the order of transmission.*/

   for (byte=0; byte<6; byte++)
     for (i=0; i<8; i++)
       updateCRC ((adr[byte] >> i) & 1);

   /* The hash code is the 6 least significant bits of the CRC
     in reverse order: CRC[0] = hash[5], CRC[1] = hash[4], etc.
   */

   hashcode = 0;
   for (i=0; i<6; i++) hashcode = (hashcode << 1) + CRC[i];

   /* Bits 3-5 of hashcode point to byte in address filter.
     Bits 0-2 point to bit within that byte.  */

   byte = hashcode >> 3;
```

```
      ladrf[byte] |= (1 << (hashcode & 7));
      printf ("hashcode = %d (decimal)  ladrf[0:63] = ", hashcode);
      for (i=0; i<8; i++)
        printf ("%02X ", ladrf[i]);
      printf (" (LSB first)\n");
    }
}

void updateCRC (int bit)
{
    int j;

    /* shift CRC and control bit (CRC[32]) */
    for (j=32; j>0; j—) CRC[j] = CRC[j–1];
    CRC[0] = 0;

    /* If bit XOR (control bit) = 1, set CRC = CRC XOR polynomial. */
    if (bit ^ CRC[32])
      for (j=0; j<32; j++) CRC[j] ^= poly[j];
}
```

Table A-1 "Mapping of Logical Address to Filter Mask" can be used to find a multicast address that maps into a particular address filter bit. For example, address BB 00 00 00 00 00 maps into bit 15. Therefore, any node that has bit 15 set in its logical address filter register will receive all packets addressed to BB 00 00 00 00 00. The table also shows that bit 15 is located in bit 7 of byte 1 of the Logical Address Filter Register.

Addresses in this table are shown in the standard Ethernet format. The leftmost byte is the first byte to appear on the network with the least significant bit appearing first.

Table A-1. Mapping of Logical Address to Filter Mask

Byte Pos	Bit Pos	LAF Bit	Destination Address Accepted	Byte Pos	Bit Pos	LAF Bit	Destination Address Accepted
0	0	0	85 00 00 00 00 00	4	0	32	21 00 00 00 00 00
0	1	1	A5 00 00 00 00 00	4	1	33	01 00 00 00 00 00
0	2	2	E5 00 00 00 00 00	4	2	34	41 00 00 00 00 00
0	3	3	C5 00 00 00 00 00	4	3	35	71 00 00 00 00 00
0	4	4	45 00 00 00 00 00	4	4	36	E1 00 00 00 00 00
0	5	5	65 00 00 00 00 00	4	5	37	C1 00 00 00 00 00
0	6	6	25 00 00 00 00 00	4	6	38	81 00 00 00 00 00
0	7	7	05 00 00 00 00 00	4	7	39	A1 00 00 00 00 00
1	0	8	2B 00 00 00 00 00	5	0	40	8F 00 00 00 00 00
1	1	9	0B 00 00 00 00 00	5	1	41	BF 00 00 00 00 00
1	2	10	4B 00 00 00 00 00	5	2	42	EF 00 00 00 00 00
1	3	11	6B 00 00 00 00 00	5	3	43	CF 00 00 00 00 00
1	4	12	EB 00 00 00 00 00	5	4	44	4F 00 00 00 00 00
1	5	13	CB 00 00 00 00 00	5	5	45	6F 00 00 00 00 00
1	6	14	8B 00 00 00 00 00	5	6	46	2F 00 00 00 00 00
1	7	15	BB 00 00 00 00 00	5	7	47	0F 00 00 00 00 00
2	0	16	C7 00 00 00 00 00	6	0	48	63 00 00 00 00 00
2	1	17	E7 00 00 00 00 00	6	1	49	43 00 00 00 00 00
2	2	18	A7 00 00 00 00 00	6	2	50	03 00 00 00 00 00
2	3	19	87 00 00 00 00 00	6	3	51	23 00 00 00 00 00
2	4	20	07 00 00 00 00 00	6	4	52	A3 00 00 00 00 00
2	5	21	27 00 00 00 00 00	6	5	53	83 00 00 00 00 00
2	6	22	67 00 00 00 00 00	6	6	54	C3 00 00 00 00 00
2	7	23	47 00 00 00 00 00	6	7	55	E3 00 00 00 00 00
3	0	24	69 00 00 00 00 00	7	0	56	CD 00 00 00 00 00
3	1	25	49 00 00 00 00 00	7	1	57	ED 00 00 00 00 00
3	2	26	09 00 00 00 00 00	7	2	58	AD 00 00 00 00 00
3	3	27	29 00 00 00 00 00	7	3	59	8D 00 00 00 00 00
3	4	28	A9 00 00 00 00 00	7	4	60	0D 00 00 00 00 00
3	5	29	89 00 00 00 00 00	7	5	61	2D 00 00 00 00 00
3	6	30	C9 00 00 00 00 00	7	6	62	6D 00 00 00 00 00
3	7	31	E9 00 00 00 00 00	7	7	63	4D 00 00 00 00 00

APPENDIX B

Comparison Between C-LANCE (Am79C90) and LANCE (Am7990) Devices

OVERVIEW

The Am79C90 C-LANCE device is a pin-for-pin equivalent for the Am7990 LANCE device. Using an advanced 0.8-micron CMOS process, the C-LANCE device consumes less power than the LANCE device, which is implemented in an outdated NMOS process. In addition to the inherent advantages provided by the advanced CMOS process, the C-LANCE device includes several functional enhancements over the LANCE device.

The C-LANCE device is available in both 48-pin plastic DIP and 68-pin PLCC packages. These packages are socket compatible with the LANCE packages.

This document provides a comparison of the C-LANCE and LANCE devices. Table 1 provides a summary of the comparison between the two devices. The remainder of the document gives details on each item listed in Table B-1.

Table B-1. Comparison Summary of the C-LANCE and LANCE Devices

	Description	Am79C90 C-LANCE	Am7990 LANCE
1	Process/Power Consumption	0.8-micron CS-21S CMOS process $I_{CC} \leq 50$ mA	NS-8B NMOS process $I_{CC} \leq 270$ mA
2	FIFOs	Dual FIFOs: 48-byte TX, 64-byte RX	Single FIFO: 48-byte TX/RX
3	Transmit Lockout Due to Receive	Will not occur with dual FIFOs and enhanced microcode.	May occur in high receive rate situations with "less than optimal" bus latencies.
4	Per-Packet FCS	Transmit descriptor bit is used to allow per packet addition of CRC when DTCR is set in the MODE register.	No per packet CRC control provided.
5	Backoff Algorithm	Selectable Modified Backoff Algorithm or standard backoff algorithm.	Only standard backoff algorithm available.
6	TX Descriptor Zero Buffer Byte Count Capability	Allows TX buffer byte count of zero.	No capability for TX buffer byte count of zero.
7	Interframe Spacing (IFS) Behavior	a) Implements 2-part deferral after transmit b) Part 1 of two part deferral after receive is 6 µs c) Heartbeat window = 4 µs d) Receive blind time after receive less than 500 ns	a) One-part deferral after transmit b) Part 1 of 2-part deferral after receive is 4.1 µs c) Heartbeat window = 2 µs d) Receive blind time after receive = 4.1 µs
8	"Heartbeat OK" (no CERR) Definition	Heartbeat OK if collision is asserted at any time from the beginning of the transmission to the end of the heartbeat window.	Heartbeat OK if collision is asserted during the heartbeat window.
9	Receive Lockup	Will not occur.	May occur when bus latency is large.
10	ALE Behavior	ALE may be driven HIGH at end of bus mastership when ACON is set to 0. When ACON is set to 1, ALE is not driven LOW at end of bus mastership period.	ALE may be driven LOW at end of bus mastership when ACON is set to 1. When ACON is set to 0, ALE is not driven HIGH at end of bus mastership period.
11	External Loopback on a Live Network	No problems.	May receive invalid loopback failure indications.
12	Software Reset (STOP bit) Handling	a) STOP bit in CSR0 is latched. When STOP is set, the slave cycle is allowed to complete before the C-LANCE resets. b) CSR1 and CSR2 contents are preserved when the STOP bit is set to one.	a) STOP bit in CSR0 not latched and will reset the device immediately when written. b) CSR1 and CSR2 are not preserved when the STOP bit is set to one.
13	CSR0 Slave Read Data Stability	CSR0 latched during slave reads to guarantee timing on DAL lines.	CSR0 not latched during slave read cycles (could give timing violations on DAL lines).
14	INEA bit behavior	INEA bit can be set in CSR0 at any time, regardless of the state of the STOP bit.	INEA cannot be set in CSR0 while the STOP bit is set.
15	Effect of setting the STOP bit on CSR0 bits.	Setting the STOP bit in CSR0 when the STOP bit is already set does not effect any of the other bits in CSR0 (they are not cleared).	Setting the STOP bit in CSR0 causes all of the other bits in CSR0 to clear, regardless of the previous state of the STOP bit.
16	AC Specification Changes	#06 (t_{TEP}) maximum = 60 ns #08 (t_{TDP}) maximum = 60 ns #18 (t_{RDS}) minimum = 35 ns #30 (t_{RDAS}) minimum = 40 ns #45 (t_{RDYS}) minimum = 65 ns	#06 (t_{TEP}) maximum = 70 ns #08 (t_{TDP}) maximum = 70 ns #18 (t_{RDS}) minimum = 40 ns #30 (t_{RDAS}) minimum = 50 ns #45 (t_{RDYS}) minimum = 75 ns
17	Burn-In Option	The burn-in option for the C-LANCE is no longer available.	
18	RX Descriptor Zero Buffer Byte Count Handling	Unpredictable results when the RX Descriptor Buffer Byte Count is set to zero.	Interprets a BCNT field setting of zero in a receive descriptor as a 4096 byte buffer.

Detailed Description of Enhancements

1. Process/Power Consumption

By using an advanced 0.8 micron CMOS process, the I_{CC} specification for the C-LANCE device is reduced to 50 mA maximum, compared to the 270 mA maximum I_{CC} specification for the LANCE device.

2. FIFOs

The C-LANCE device incorporates a dual FIFO (48 bytes Transmit, 64 bytes Receive) architecture to help it compete for bandwidth on busy networks. The LANCE device's single 48-byte FIFO architecture and its associated microcode has problems transmitting packets out on busy networks. This problem is known as the "Transmit Lockout due to Receive" problem. It occurs when minimum or near minimum IFS traffic is continually received by the LANCE device and bus latency is not "good" ("good" = latency < approximately 3 µs). In this situation, the LANCE device's microcode and bus interface is locked servicing receive packets, and is not able to poll the pending transmit descriptor (until the receive traffic stops or does not pass address match).

The C-LANCE device addresses this problem by including dual FIFOs and microcode that is modified to take advantage of the dual FIFOs. The microcode is changed so that a transmit descriptor poll operation occurs sometime early (exact time depends on bus latencies and whether the receive buffer was owned before the receive packet arrived) in the receive DMA operations for each packet. If the OWN bit in the TX descriptor is found set, transmit FIFO loading DMA is interleaved with the receive FIFO emptying DMA for the packet being received. The transmit packet is then ready to be transmitted immediately following the end of the receive packet on the wire. The dual FIFOs and microcode changes eliminate the possibility of transmit activity being locked out due to high receive activity.

Interleaving the transmit DMA activity with receive DMA activity at the beginning of a reception has the effect of increasing the bus latency for receive DMA operations. To ensure that the C-LANCE device can tolerate the same bus latency as the LANCE device, the receive FIFO in the C-LANCE device is increased to 64 bytes. The transmit FIFO in the C-LANCE device holds 48 bytes.

3. Transmit Lockout Due to Receive

As discussed in item 2, the dual FIFO architecture and modified microcode implemented in the C-LANCE device eliminates the possibility of Transmit Lockout Due to Receive from occurring.

4. Per-Packet FCS

In the LANCE device, addition of the Frame Check Sequence (FCS or CRC) to each transmit packet is controlled on a per-initialization basis. In other words, when the DTCR (Disable Transmit CRC) bit is set in the mode register at initialization, the only way that packets can subsequently be transmitted with an FCS attached is by re-initializing the device with the DTCR bit cleared.

The C-LANCE device provides the capability to override the DTCR setting on a per-packet basis. If DTCR was set in the mode register at initialization, the ADD_FCS bit in the transmit descriptor can be used to append FCS to transmitted packets on a per-packet basis, overriding the DTCR setting. If DTCR is cleared in the mode register, the ADD_FCS bit is a "don't care."

The ADD_FCS bit is located in bit 13 of TMD1 in the C-LANCE device. This bit is RESERVED in the LANCE device. Table 2 below summarizes the operation of the ADD_FCS bit. Note that the ADD_FCS bit is only meaningful in the first descriptor of a transmit buffer chain (STP=1).

This feature should be compatible with existing implementations. Non-bridge nodes normally run with FCS enabled (DTCR cleared). Bridges run with FCS disabled. **It is assumed that existing software in these applications do not set bit 13 of TMD1, which was previously RESERVED.**

The ADD_FCS bit is also implemented as bit 13 of TMD1 in the PCnet™-ISA (Am79C960) and operates identically to the way in which it operates in the C-LANCE device.

As a side note, this feature can be used by software to identify the C-LANCE device from LANCE device. The LANCE device writes bit 13 of TMD1 to zero when updating transmit status in the transmit descriptor. The C-LANCE device will write this bit with the value read, so if it is set to one it will be returned as a one.

Table B-2. ADD_FCS Bit Operation

DTCR in Mode Reg.	STP	ADD_FCS	FCS Added?
0	X	X	Yes
1	0	X	N/A
1	1	0	No
1	1	1	Yes

5. Backoff Algorithm

A selectable Modified Backoff Algorithm is provided in the C-LANCE device which can improve throughput in busy networks. Bit 7 of the Mode register (EMBA bit) is used to enable the Modified Backoff Algorithm. This bit is RESERVED in the LANCE device.

With the Modified Backoff Algorithm, counting of the IFS interval is suspended when receive carrier sense is detected. The count resumes when receive carrier sense goes away. This algorithm increases throughput in large networks with heavy traffic (many collisions). It can be considered as an "adaptive" backoff algorithm. This mode should only be used in network segments in which all nodes are using this mode. Otherwise, the nodes that are using it will be at a disadvantage to those that are not.

Note that this mode does not conflict with IEEE requirements for compliance. The IEEE 802.3 specification specifies only the minimum amount of time for the backoff interval. This leaves open the possibility of backing off more than the minimum, which is precisely how the Modified Backoff Algorithm works.

The Modified Backoff Algorithm is included as an option in the MACE™ (Am79C940) and PCnet-ISA (Am79C960) devices.

6. TX Descriptor Zero Buffer Byte Count Capability

The 12-bit BCNT field in the transmit descriptor of the LANCE and C-LANCE devices is loaded with the 2's complement of the number of bytes that must be transmitted from the buffer. With the 2's complement representation, a simple incrementer is used in the chip to count through the byte count as bytes are being read from the transmit buffer. When the 2's complement number reaches all 0's, the count has expired. The LANCE device does not check for the all 0's case when the BCNT field is first loaded from the descriptor. Hence, the all 0's case is interpreted by the LANCE device as a buffer count of 4096 (2^{12}), preventing zero length TX buffers in the LANCE device. In addition, the LANCE device ignores the upper 4 bits in TMD2, which are adjacent to the BCNT field. These bits are indicated as "must be ones" in the LANCE data sheet.

The C-LANCE device actually uses all 16 bits in TMD2 as the BCNT field. Compatibility with the LANCE device is preserved as long as the upper 4 bits in TMD2 are 1's, as specified in the LANCE data sheet. The C-LANCE device checks for the case where all 16 bits in TMD2 are zero before starting any transmit DMA from the buffer. If all 16 bits are zero, a zero length buffer is assumed, and the C-LANCE device immediately clears the OWN in the descriptor without starting any transmit activity on the network. Note that since all 16 bits are checked, compatibility with the LANCE device is preserved for non-Ethernet-compliant implementations which may use buffer lengths of 4096 bytes.

Zero Transmit Buffer Byte Count Capability is included in the PCnet-ISA device.

7. Interframe Spacing Behavior

a) 2-Part Deferral After Transmit: 2-part deferral after receive has always been an option in the IEEE 802.3 specification. However, 2-part deferral after transmit was recently added as an option in the 802.3 specification by the IEEE committee. With 2-part deferral, the IFS is divided into two parts, IFS1 and IFS2. If there is activity on the wire during IFS1, the IFS counter is reset until the wire is clear again. The IFS counter is not reset once it enters IFS2. When the IFS counter expires the chip will begin to transmit if it has anything to send.

The specification's wording for 2-part deferral after transmit is identical to the way that 2-part deferral after receive has been worded all along. That is, the specification specifies that part 1 of the two parts can be anywhere from 0 to 2/3 of the IFS (9.6 µs). If part 1 = 0 (perfectly legal), it is equivalent to not implementing 2-part deferral at all. Hence, the LANCE device, which implements 2-part deferral after receive but not after transmit, complies with IEEE specifications. However, implementation of 2-part deferral after both transmit and receive eliminates a possible scenario where packets cannot be received (due to very small or 0 IFS) but there is no indication of this fact through a collision indication at the transmitter. Hence, although this scenario is very rare, the C-LANCE device implements 2-part deferral after transmit in addition to after receive.

b) The IEEE 802.3 specifications state that part 1 of 2-part deferral can be anywhere from 0 to 2/3 of the IFS (9.6 µs). The LANCE device only implements 2-part deferral after receive, with part 1 = 4.1 µs and part 2 = 5.5 µs (compliant). The C-LANCE device implements 2-part deferral after both transmit and receive with part 1 = 6.0 µs and

part 2 = 3.6 µs. Since the receiver is blinded following a transmit for 4.0 µs (see below), part 1 of 2-part deferral after a transmit had to be extended beyond 4.1 µs or else part 1 would effectively only be from 4.0 µs to 4.1 µs during the IFS. Hence, in the C-LANCE device, part 1 of 2-part deferral after transmit was set at 6.0 µs and the same value was used for part 1 following a receive.

c) IEEE 802.3 specifications state that the Signal Quality Error (SQE) test window should be at least 4.0 µs and no more than 8.0 µs. The LANCE device implements a 2 µs window which is not compliant with this specification. This generally turns out to be a non-issue because 802.3 also specifies that the MAU must generate the collision signal within 0.6 µs to 1.6 µs after the end of the transmit packet, which is typically early enough for the LANCE device to detect it, even with its non-compliant 2 µs window. However, to comply with IEEE standards, the C-LANCE device implements an SQE test window of 4 µs.

d) IEEE specifications require that receive be blinded following transmit for 4 µs to prevent the controller from responding to any trash that may be generated by the MAU when it generates the SQE test signal. However, IEEE specifications do not state that the receiver should be blinded following a receive. The LANCE device implements a 4.1 µs blinding time following receive, violating IEEE specifications. This was erroneously implemented in the LANCE device since it was thought to be a moot issue under the assumption that there should be no valid data on the wire within 4.1 µs of the end of a receive anyway. However, since 2-part deferral after transmit and receive are both optional, as mentioned in 7a), there are rare situations where legal packets may arrive with an IFS of less than 4.1 µs. To better handle this situation, the C-LANCE device reduces the blind time following a receive to less than 500 ns. The blind time allows time to store and then clear the status that was generated by the ending reception.

8. "Heartbeat OK" (No CERR) Definition

The heartbeat test or Signal Quality Error (SQE) test is performed to verify the ability of the AUI to pass the collision (SQE) indication to the DTE. The LANCE and C-LANCE devices indicate a heartbeat test failure by setting the CERR bit in CSR0 (bit 13).

At the conclusion of each transmission, the DTE opens a time window during which it expects to see a collision indication. In the LANCE device, this window begins immediately following when TENA de-asserts and ends 2.0µs after when RENA de-asserts. The heartbeat signal is expected by the LANCE device even if the packet being transmitted suffers a collision. This implementation violates IEEE requirements in three ways:

1) IEEE 802.3 specifications state that the heartbeat window should begin when the input becomes idle (RENA de-asserts), not when the output becomes idle (TENA de-asserts).
2) If a collision occurs, the IEEE 802.3 specifications indicate that the DTE should not look for the SQE test signal.
3) As mentioned in 7c), the window should end no earlier than 4.0 µs after when RENA de-asserts.

The C-LANCE device implements the heartbeat test in full compliance with IEEE specifications. In the C-LANCE device, the heartbeat window begins when RENA de-asserts and ends 4 µs later. In addition, the C-LANCE device does not look for the heartbeat signal whenever the packet being transmitted suffers a collision.

The PCnet-ISA and MACE devices use the same heartbeat OK definition as the C-LANCE device.

Details on the LANCE device's violations of IEEE Specifications: The consequences of the violations of the standard by the LANCE device are very little in practice. Item 1 (window begins when TENA de-asserts not RENA) actually prevents the LANCE from being penalized by Item 2 (Heartbeat expected following a collision). That is, if the LANCE device did not violate Item 1 and started its window when RENA de-asserted instead of TENA, then the LANCE device could get false CERR indications when a packet it is transmitting suffers a collision. This can happen as follows. In the event of a collision, the network may remain active for a while after one node stops transmitting its JAM sequence (other nodes involved in the collision may still have their JAM on the wire). At a node that ends its JAM sequence relatively early, the heartbeat signal can overlap with the collision or the end of the collision fragment since the MAU times the heartbeat signal generation from when the controller stops transmitting. If this node uses a LANCE device as its controller, the LANCE device will see this heartbeat signal only because of the violation given in Item 1. If the LANCE device started its window when RENA de-asserted instead of TENA, it would miss the heartbeat signal, since the heartbeat passes by while the collision is still on the wire. This would give false CERR indications. Hence, the violation of Item 1 in the LANCE device is not a problem. In fact, it makes the violation of Item 2 generally a non-issue.

Although the violation of Item 1 masks the violation of Item 2 as just described, the violation of Item 2 (heartbeat still expected by the LANCE device when collision occurs) can still lead to false CERR indications when the LANCE device is used with a non-802.3-compliant MAU. The IEEE 802.3 specifications state that the MAU is to generate the SQE test signal after every transmit, even when the transmit suffers a collision. However, some MAUs on the market have been found to not

comply with this requirement. When operating with a non-compliant MAU that does not generate the heartbeat signal after a collided transmission, the LANCE device can give false CERR indications.

As mentioned in 7c), Item 3 is generally a non-issue.

9. Receive Lockup

The LANCE device has an errata where the receiver locks up when the system bus latency is very high. This errata is fixed in the C-LANCE device.

10. ALE Behavior

The LANCE device may drive the ALE pin LOW at the end of each bus mastership period when ACON=1 (ALE/AS active low – AS mode). When the bus mastership period ends, the ALE pin is tri-stated hence, if ALE is pulled HIGH by external logic, a glitch on ALE results. The glitch occurs about when the LANCE device is releasing the bus by bringing HOLD high. The C-LANCE device incorporates redesigned ALE logic to prevent this glitch from occurring.

However, in the C-LANCE, when ACON=0 (active high ALE), ALE is driven high before it is tri-stated at the end of every bus mastership period. In the LANCE, when ACON=0 (active high ALE), ALE is not driven high before it is tri-stated at the end of every bus mastership period.

This difference will not cause any problems in designs which set ACON=1 (AS; active low ALE). It could cause problems in designs in which ACON=0. The ALE signal is intended to provide a strobe signal for an external address latch. The rising edge, coupled with a subsequent falling edge that will occur if the pin is externally pulled down, will cause an invalid address to be strobed into the external address latch. However, since this occurs at the end of the bus mastership period, and further master cycles are not performed by the C-LANCE subsequent to the invalid address being strobed (until the next bus mastership period), the invalid address generally has no effect. A design could have problems with this if external logic is continuously decoding the latched address and taking some action on it even though the C-LANCE is not executing any master cycles.

11. External Loopback on a Live Network

The LANCE device has an errata which causes loopback failures when external loopback is run on a live network. This errata is fixed in the C-LANCE device.

12. Software Reset (STOP Bit) Handling

a) Latching of the STOP bit: In the LANCE device, writing the STOP bit in CSR0 causes all bus signals to immediately float. With $\overline{\text{READY}}$ pulled up externally ($\overline{\text{READY}}$ is open drain), this causes $\overline{\text{READY}}$ to de-assert prematurely during the slave cycle. If $\overline{\text{DAS}}$ and $\overline{\text{CS}}$ remain active, the LANCE device can erroneously start another slave cycle. The C-LANCE device latches the STOP bit and, when it is set, allows the slave cycle in progress to complete before resetting the part.

b) Preservation of CSR1 and CSR2: The LANCE device does not preserve the contents of CSR1 and CSR2 during the initialization process. Hence, when the STOP bit is set, the contents of CSR1 and CSR2 are not the same as they were before initialization and they must be rewritten before re-initializing. This is not really a problem in the LANCE device but it can add extra instructions to software. The C-LANCE device removes this software burden by preserving the contents of CSR1 and CSR2 during initialization so that when the STOP bit is set, they do not have to be reloaded before re-initializing. Note, however, that if the default values of CSR3 (defaults for BCON, ACON, and BSWP are 0, 0, and 0, respectively) are not used, CSR3 must still be reloaded after setting the STOP bit in the C-LANCE device since CSR3 is cleared when the STOP bit is set.

13. CSR0 Slave Read Data Stability

In the LANCE device, the status bit latches in CSR0 may change at any time, as governed by the occurrence of the external events which they monitor. Hence, the ERR, BABL, CERR, MISS, IDON, and INTR bits in CSR0 may change during a slave read cycle in which they are being accessed. This can cause timing violations on the DAL lines. In the C-LANCE device, CSR0 is latched in a shadow register during a read so that timing on the DAL lines is guaranteed.

14. INEA Behavior

With the C-LANCE device, INEA bit can be set in CSR0 at any time, regardless of the state of the STOP bit. This actually removes a restriction that was present in the LANCE device in which the INEA bit in CSR0 could be not be set while STOP bit was set.

This difference between the two devices does not effect normal device operation but could disrupt diagnostic code written for the LANCE device.

15. Effect of Setting the STOP bit on CSR0 Bits

In the LANCE device, CSR0 is reset when the STOP bit in CSR0 is set. **This reset happens even if the STOP bit was already set.** When the reset occurs, all of the other bits in CSR0 are cleared. In the C-LANCE, CSR0 is reset when the STOP bit is set in CSR0 **only if the STOP bit was not already set**.

This difference between the two devices does not effect normal device operation but could disrupt diagnostic code written for the LANCE device.

16. AC Specification Changes

The following differences in AC specification exist between the C-LANCE and the LANCE.

	C-LANCE	LANCE
#06 (t_{TEP}) maximum	60 ns	70 ns
#08 (t_{TDP}) maximum	60 ns	70 ns
#18 (t_{RDS}) minimum	35 ns	40 ns
#30 (t_{RDAS}) minimum	40 ns	50 ns
#45 (t_{RDYS}) minimum	65 ns	75 ns

17. Elimination of Burn-In Option

The burn-in option for the C-LANCE is no longer available. Thus, the ordering part number Am79C90PCB is no longer valid (see page 4 of the C-LANCE data sheet).

18. RX Descriptor Zero Buffer Byte Count Handling

The 12-bit BCNT field in the receive descriptor of the LANCE and C-LANCE devices is loaded with the 2's complement of the number of bytes allocated to the associated receive buffer. In the LANCE device, when all 0's are written to the BCNT field in a receive descriptor, a buffer length of 4096 (2^{12}) bytes is assumed. In the C-LANCE device the case of all 0's in the receive descriptor may produce unpredictable results.

This difference should not cause problems in 802.3 compliant networks, because 802.3 has a maximum packet length specification of 1518 bytes.

FINAL

Am79C940

Media Access Controller for Ethernet (MACE™)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Integrated Controller with 10BASE-T transceiver and AUI port**
- **Supports IEEE 802.3/ANSI 8802-3 and Ethernet standards**
- **84-pin PLCC and 100-pin PQFP Packages**
- **80-pin Thin Quad Flat Pack (TQFP) package available for space critical applications such as PCMCIA**
- **Modular architecture allows easy tuning to specific applications**
- **High speed, 16-bit synchronous host system interface with 2 or 3 cycles/transfer**
- **Individual transmit (136 byte) and receive (128 byte) FIFOs provide increase of system latency and support the following features:**
  - Automatic retransmission with no FIFO reload
  - Automatic receive stripping and transmit padding (individually programmable)
  - Automatic runt packet rejection
  - Automatic deletion of collision frames
  - Automatic retransmission with no FIFO reload
- **Direct slave access to all on board configuration/status registers and transmit/receive FIFOs**
- **Direct FIFO read/write access for simple interface to DMA controllers or I/O processors**
- **Arbitrary byte alignment and little/big endian memory interface supported**
- **Internal/external loopback capabilities**
- **External Address Detection Interface (EADI™) for external hardware address filtering in bridge/router applications**
- **JTAG Boundary Scan (IEEE 1149.1) test access port interface for board level production test**
- **Integrated Manchester Encoder/Decoder**
- **Digital Attachment Interface (DAI™) allows by-passing of differential Attachment Unit Interface (AUI)**
- **Supports the following types of network interface:**
  - AUI to external 10BASE2, 10BASE5 or 10BASE-F MAU
  - DAI port to external 10BASE2, 10BASE5, 10BASE-T, 10BASE-F MAU
  - General Purpose Serial Interface (GPSI) to external encoding/decoding scheme
  - Internal 10BASE-T transceiver with automatic selection of 10BASE-T or AUI port
- **Sleep mode allows reduced power consumption for critical battery powered applications**
- **1 MHz – 25 MHz system clock speed**

GENERAL DESCRIPTION

The Media Access Controller for Ethernet (MACE) chip is a CMOS VLSI device designed to provide flexibility in customized LAN design. The MACE device is specifically designed to address applications where multiple I/O peripherals are present, and a centralized or system specific DMA is required. The high speed, 16-bit synchronous system interface is optimized for an external DMA or I/O processor system, and is similar to many existing peripheral devices, such as SCSI and serial link controllers.

The MACE device is a slave register based peripheral. All transfers to and from the system are performed using simple memory or I/O read and write commands. In conjunction with a user defined DMA engine, the MACE chip provides an IEEE 802.3 interface tailored to a specific application. Its superior modular architecture and versatile system interface allow the MACE device to be configured as a stand-alone device or as a connectivity cell incorporated into a larger, integrated system.

The MACE device provides a complete Ethernet node solution with an integrated 10BASE-T transceiver, and supports up to 25-MHz system clocks. The MACE device embodies the Media Access Control (MAC) and Physical Signaling (PLS) sub-layers of the IEEE 802.3 standard, and provides an IEEE defined Attachment Unit Interface (AUI) for coupling to an external Medium Attachment Unit (MAU). The MACE device is compliant with 10BASE2, 10BASE5, 10BASE-T, and 10BASE-F transceivers.

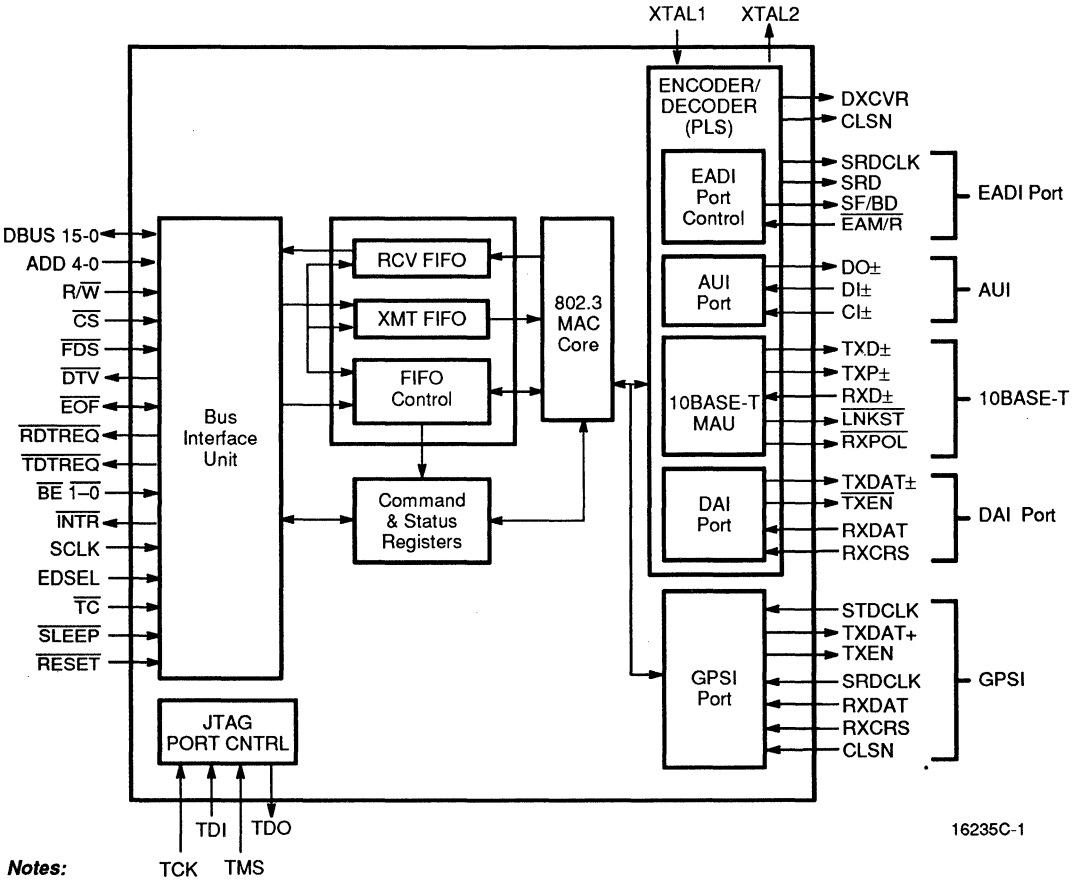
Publication#16235 Rev. C Amendment/0
Issue Date: June 1994

Additional features also enhance over-all system design. The individual transmit and receive FIFOs optimize system overhead, providing substantial latency during packet transmission and reception, and minimizing intervention during normal network error recovery. The integrated Manchester encoder/decoder eliminates the need for an external Serial Interface Adapter (SIA) in the node system. If support for an external encoding/decoding scheme is desired, the General Purpose Serial Interface (GPSI) allows direct access to/from the MAC. In addition, the Digital Attachment Interface (DAI), which is a simplified electrical attachment specification, allows implementation of MAUs that do not require DC isolation between the MAU and DTE. The DAI port can also be used to indicate transmit, receive, or collision status by connecting LEDs to the port. The MACE device also provides an External Address Detection Interface (EADI) to allow external hardware address filtering in internetworking applications.

The Am79C940 MACE chip is offered in a Plastic Leadless Chip Carrier (84-pin PLCC), a Plastic Quad Flat Package (100-pin PQFP), and a Thin Quad Flat Package (TQFP 80-pin). There are several small functional and physical differences between the 80-pin TQFP and the 84-pin PLCC and 100-pin PQFP configurations.

Because of the smaller number of pins in the TQFP configuration versus the PLCC configuration, four pins are not bonded out. Though the die is identical in all three package configurations, the removal of these four pins does cause some functionality differences between the TQFP and the PLCC and PQFP configurations. Depending on the application, the removal of these pins will or will not have an effect.

BLOCK DIAGRAM



Notes:

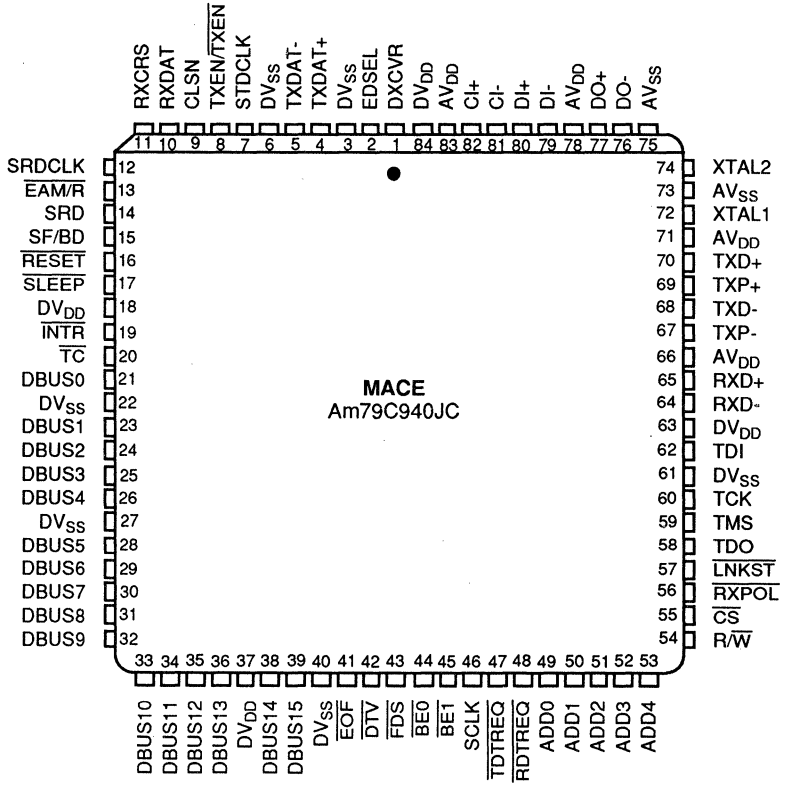
1. *Only one of the network ports AUI, 10BASE-T, DAI port or GPSI can be active at any time. Some shared signals are active regardless of which network port is active, and some are reconfigured.*
2. *The EADI port is active at all times.*

RELATED PRODUCTS

Part No.	Description
Am7996	IEEE 802.3/Ethernet/Cheapernet Transceiver
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted Pair Ethernet Transceiver Plus (TPEX+)
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)
Am79C961	PCnet-ISA+ Single-Chip Ethernet Controller for ISA (with Microsoft® Plug n' Play Support)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems

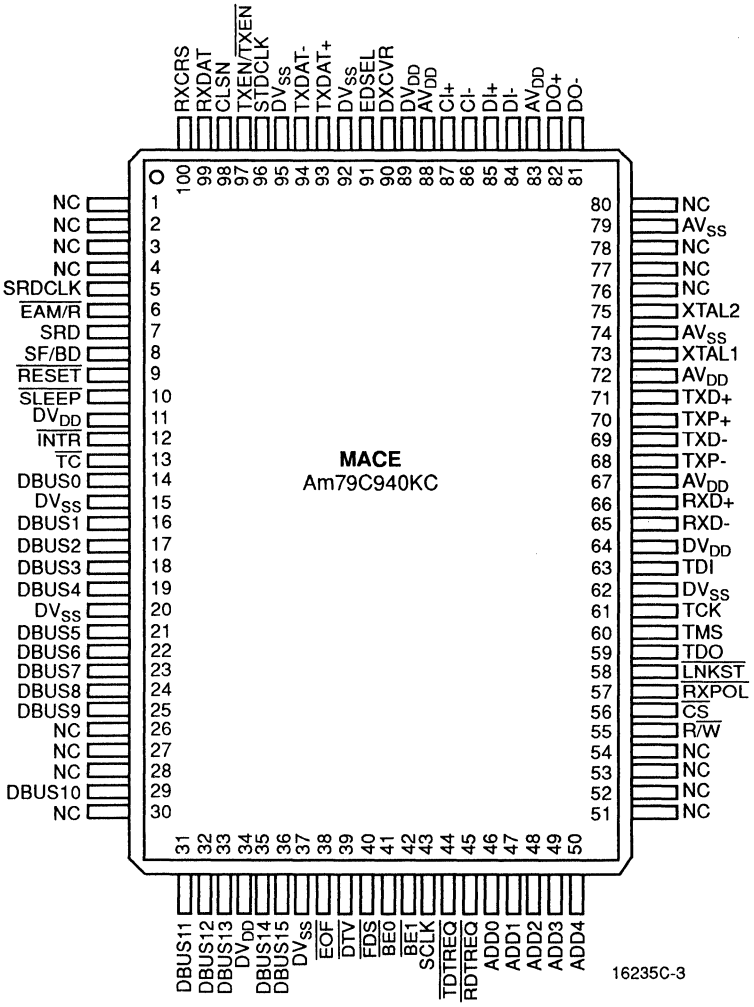
TABLE OF CONTENTS

CONNECTION DIAGRAMS
PL 084
PLCC Package

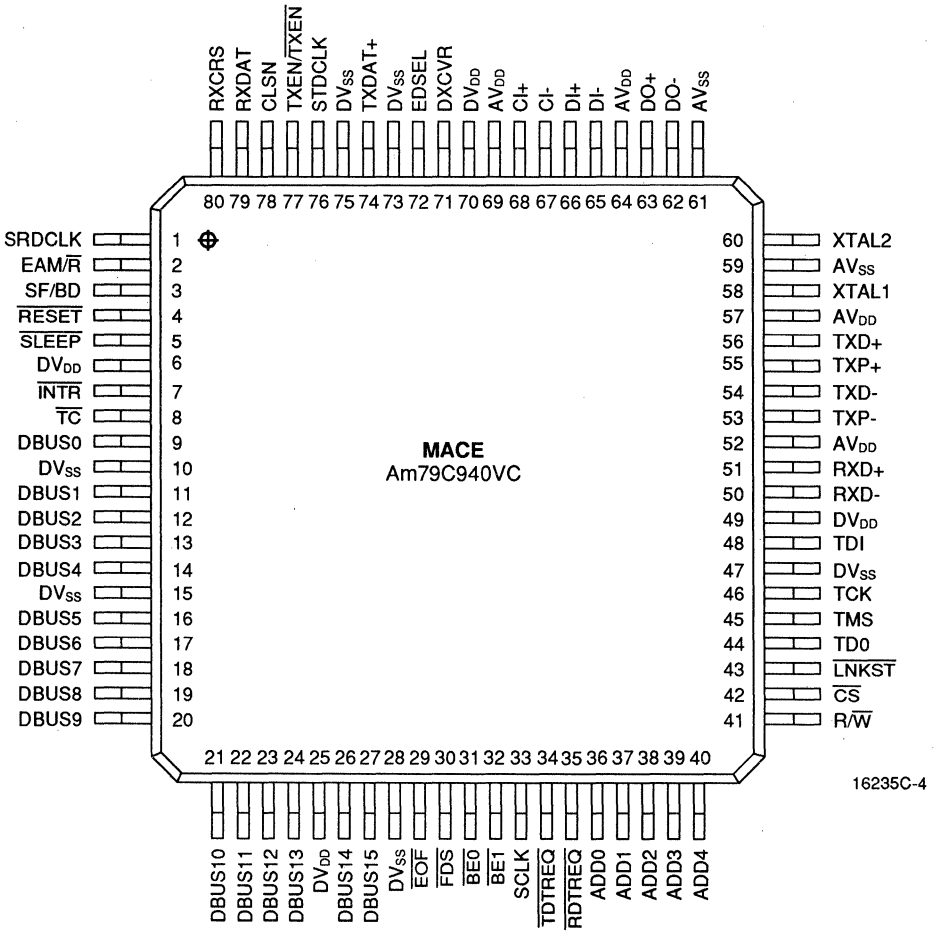


16235C-2

CONNECTION DIAGRAMS
PQR 100
PQFP Package



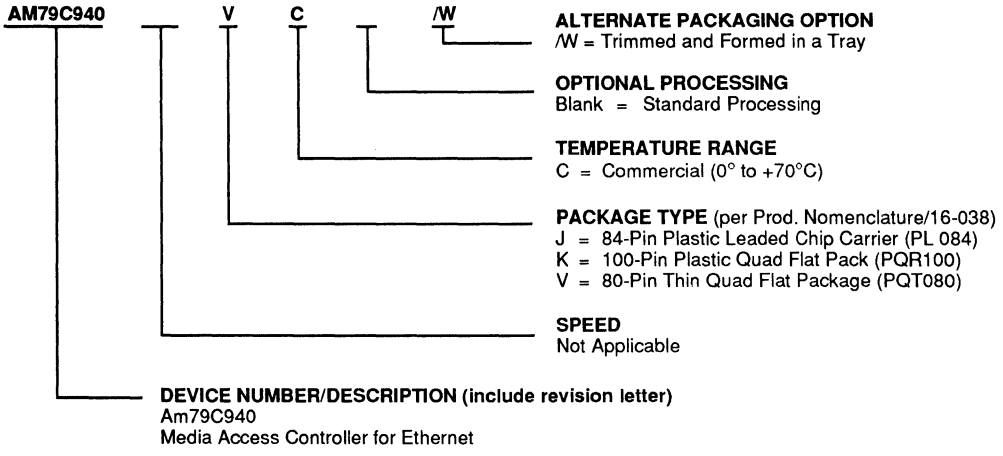
CONNECTION DIAGRAMS
PQT 080
TQFP Package



Note: *Four pin functions available on the PLCC and PQFP packages are not available with the TQFP package. (See page 27 "Pin Functions not available with the 80-pin TQFP Package").*

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM79C940	JC, KC, KC/W, VC, VC/W

Valid Combinations

The Valid Combinations table lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

PIN/PACKAGE SUMMARY

PLCC Pin #	Pin Name	Pin Function
1	DXCVR	Disable Transceiver
2	EDSEL	Edge Select
3	DVSS	Digital Ground
4	TXDAT+	Transmit Data +
5	TXDAT–	Transmit Data –
6	DVSS	Digital Ground
7	STDCLK	Serial Transmit Data Clock
8	TXEN/$\overline{\text{TXEN}}$	Transmit Enable
9	CLSN	Collision
10	RXDAT	Receive Data
11	RXCRS	Receive Carrier Sense
12	SRDCLK	Serial Receive Data Clock
13	$\overline{\text{EAM/R}}$	External Address Match/Reject
14	SRD	Serial Receive Data
15	SF/BD	Start Frame/Byte Delimiter
16	$\overline{\text{RESET}}$	Reset
17	$\overline{\text{SLEEP}}$	Sleep Mode
18	DVDD	Digital Power
19	$\overline{\text{INTR}}$	Interrupt
20	$\overline{\text{TC}}$	Timing Control
21	DBUS0	Data Bus0
22	DVSS	Digital Ground
23	DBUS1	Data Bus1
24	DBUS2	Data Bus2
25	DBUS3	Data Bus3
26	DBUS4	Data Bus4
27	DVSS	Digital Ground
28	DBUS5	Data Bus5
29	DBUS6	Data Bus6
30	DBUS7	Data Bus7
31	DBUS8	Data Bus8
32	DBUS9	Data Bus9
33	DBUS10	Data Bus10
34	DBUS11	Data Bus11
35	DBUS12	Data Bus12
36	DBUS13	Data Bus13
37	DVDD	Digital Power
38	DBUS14	Data Bus14
39	DBUS15	Data Bus15
40	DVSS	Digital Ground
41	$\overline{\text{EOF}}$	End Of Frame
42	$\overline{\text{DTV}}$	Data Transfer Valid

PIN/PACKAGE SUMMARY (continued)

PLCC Pin #	Pin Name	Pin Function
43	$\overline{FDS}$	FIFO Data Strobe
44	$\overline{BE0}$	Byte Enable0
45	$\overline{BE1}$	Byte Enable1
46	SCLK	System Clock
47	$\overline{TDTREQ}$	Transmit Data Transfer Request
48	$\overline{RDTREQ}$	Receive Data Transfer Request
49	ADD0	Address0
50	ADD1	Address1
51	ADD2	Address2
52	ADD3	Address3
53	ADD4	Address4
54	R/$\overline{W}$	Read/Write
55	$\overline{CS}$	Chip Select
56	$\overline{RXPOL}$	Receive Polarity
57	$\overline{LNKST}$	Link Status
58	TDO	Test Data Out
59	TMS	Test Mode Select
60	TCK	Test Clock
61	DV_{SS}	Digital Ground
62	TDI	Test Data Input
63	DV_{DD}	Digital Power
64	RXD–	Receive Data–
65	RXD+	Receive Data+
66	AV_{DD}	Analog Power
67	TXP–	Transmit Pre-distortion
68	TXD–	Transmit Data–
69	TXP+	Transmit Pre–distortion+
70	TXD+	Transmit Data+
71	AV_{DD}	Analog Power
72	XTAL1	Crystal Input
73	AV_{SS}	Analog Ground
74	XTAL2	Crystal Output
75	AV_{SS}	Analog Ground
76	DO–	Data Out–
77	DO+	Data Out+
78	AV_{DD}	Analog Power
79	DI–	Data In–
80	DI+	Data In+
81	CI–	Control In–
82	CI+	Control In+
83	AV_{DD}	Analog Power
84	DV_{DD}	Digital Power

PIN/PACKAGE SUMMARY (continued)

PQFP Pin #	Pin Name	Pin Function
1	NC	No Connect
2	NC	No Connect
3	NC	No Connect
4	NC	No Connect
5	SRDCLK	Serial Receive Data Clock
6	$\overline{\text{EAM/R}}$	External Address Match/Reject
7	SRD	Serial Receive Data
8	SF/BD	Start Frame/Byte Delimiter
9	$\overline{\text{RESET}}$	Reset
10	$\overline{\text{SLEEP}}$	Sleep Mode
11	DVDD	Digital Power
12	$\overline{\text{INTR}}$	Interrupt
13	$\overline{\text{TC}}$	Timing Control
14	DBUS0	Data Bus0
15	DVSS	Digital Ground
16	DBUS1	Data Bus1
17	DBUS2	Data Bus2
18	DBUS3	Data Bus3
19	DBUS4	Data Bus4
20	DVSS	Digital Ground
21	DBUS5	Data Bus5
22	DBUS6	Data Bus6
23	DBUS7	Data Bus7
24	DBUS8	Data Bus8
25	DBUS9	Data Bus9
26	NC	No Connect
27	NC	No Connect
28	NC	No Connect
29	DBUS10	Data Bus10
30	NC	No Connect
31	DBUS11	Data Bus11
32	DBUS12	Data Bus12
33	DBUS13	Data Bus13
34	DVDD	Digital Power
35	DBUS14	Data Bus14
36	DBUS15	Data Bus15
37	DVSS	Digital Ground
38	$\overline{\text{EOF}}$	End Of Frame
39	$\overline{\text{DTV}}$	Data Transfer Valid
40	$\overline{\text{FDS}}$	FIFO Data Strobe
41	$\overline{\text{BE0}}$	Byte Enable0
42	$\overline{\text{BE1}}$	Byte Enable1

PIN/PACKAGE SUMMARY (continued)

PQFP Pin #	Pin Name	Pin Function
43	SCLK	System Clock
44	$\overline{\text{TDTREQ}}$	Transmit Data Transfer Request
45	$\overline{\text{RDTREQ}}$	Receive Data Transfer Request
46	ADD0	Address0
47	ADD1	Address1
48	ADD2	Address2
49	ADD3	Address3
50	ADD4	Address4
51	NC	No Connect
52	NC	No Connect
53	NC	No Connect
54	NC	No Connect
55	R/$\overline{\text{W}}$	Read/Write
56	$\overline{\text{CS}}$	Chip Select
57	$\overline{\text{RXPOL}}$	Receive Polarity
58	$\overline{\text{LNKST}}$	Link Status
59	TDO	Test Data Out
60	TMS	Test Mode Select
61	TCK	Test Clock
62	DV_{SS}	Digital Ground
63	TDI	Test Data Input
64	DV_{DD}	Digital Power
65	RXD–	Receive Data–
66	RXD+	Receive Data+
67	AV_{DD}	Analog Power
68	TXP–	Transmit Pre-distortion–
69	TXD–	Transmit Data–
70	TXP+	Transmit Pre-distortion+
71	TXD+	Transmit Data+
72	AV_{DD}	Analog Power
73	XTAL1	Crystal Input
74	AV_{SS}	Analog Ground
75	XTAL2	Crystal Output
76	NC	No Connect
77	NC	No Connect
78	NC	No Connect
79	AV_{SS}	Analog Ground
80	NC	No Connect
81	DO–	Data Out–
82	DO+	Data Out+
83	AV_{DD}	Analog Power
84	DI–	Data In–
85	DI+	Data In+

PIN/PACKAGE SUMMARY (continued)

PQFP Pin #	Pin Name	Pin Function
86	CI–	Control In–
87	CI+	Control In+
88	AV_{DD}	Analog Power
89	DV_{DD}	Digital Power
90	DXCVR	Disable Transceiver
91	EDSEL	Edge Select
92	DV_{SS}	Digital Ground
93	TXDAT+	Transmit Data +
94	TXDAT–	Transmit Data –
95	DV_{SS}	Digital Ground
96	STDCLK	Serial Transmit Data Clock
97	TXEN/$\overline{\text{TXEN}}$	Transmit Enable
98	CLSN	Collision
99	RXDAT	Receive Data
100	RXCRS	Receive Carrier Sense

PIN/PACKAGE SUMMARY (continued)

TQFP Pin Number	Pin Name	Pin Function	TQFP Pin Number	Pin Name	Pin Function
1	SRDCLK	Serial Receive Data Clock	41	R/$\overline{W}$	Read/Write
2	EAM/$\overline{R}$	External Address Match/Reject	42	$\overline{CS}$	Chip/Select
3	SF/BD	Start Frame/Byte Delimiter	43	$\overline{LNKST}$	Link Status
4	$\overline{RESET}$	Reset	44	TDO	Test Data Out
5	$\overline{SLEEP}$	Sleep Mode	45	TMS	Test Mode Select
6	DVDD	Digital Power	46	TCK	Test Clock
7	$\overline{INTR}$	Interrupt	47	DVSS	Digital Ground
8	$\overline{TC}$	Timing Control	48	TDI	Test Data Input
9	DBUS0	Data Bus0	49	DVDD	Digital Power
10	DVSS	Digital Ground	50	RXD–	Receive Data–
11	DBUS1	Data Bus1	51	RXD+	Receive Data+
12	DBUS2	Data Bus2	52	AVDD	Analog Power
13	DBUS3	Data Bus3	53	TXP–	Transmit Pre-distortion–
14	DBUS4	Data Bus4	54	TXD–	Transmit Data–
15	DVSS	Digital Ground	55	TXP+	Transmit Pre-distortion+
16	DBUS5	Data Bus5	56	TXD+	Transmit Data+
17	DBUS6	Data Bus6	57	AVDD	Analog Power
18	DBUS7	Data Bus7	58	XTAL1	Crystal Output
19	DBUS8	Data Bus8	59	AVSS	Analog Ground
20	DBUS9	Data Bus9	60	XTAL2	Crystal Output
21	DBUS10	Data Bus10	61	AVSS	Analog Ground
22	DBUS11	Data Bus11	62	DO–	Data Out–
23	DBUS12	Data Bus12	63	DO+	Data Out+
24	DBUS13	Data Bus13	64	AVDD	Analog Power
25	DVDD	Digital Power	65	DI–	Data In–
26	DBUS14	Data Bus14	66	DI+	Data Out+
27	DBUS15	Data Bus15	67	CI–	Control In–
28	DVSS	Digital Ground	68	CI+	Control In+
29	$\overline{EOF}$	End of Frame	69	AVDD	Analog Power
30	$\overline{FDS}$	FIFO Data Strobe	70	DVDD	Digital Power
31	$\overline{BE0}$	Byte Enable0	71	DXCVR	Disable Transceiver
32	$\overline{BE1}$	Byte Enable1	72	EDSEL	Edge Select
33	SCLK	System Clock	73	DVSS	Digital Ground
34	$\overline{TDTREQ}$	Transmit Data Transfer Request	74	TXDAT+	Transmit Data+
35	$\overline{RDTREQ}$	Receive Data Transfer Request	75	DVSS	Digital Ground
36	ADD0	Address0	76	STDCLK	Serial Transmit Data Clock
37	ADD1	Address1	77	TXEN/$\overline{TXEN}$	Transmit Enable
38	ADD2	Address2	78	CLSN	Collision
39	ADD3	Address3	79	RXDAT	Receive Data
40	ADD4	Address4	80	RXCRS	Receive Carrier Sense

PIN SUMMARY

Pin Name	Pin Function	Type	Active	Comment
Attachment Unit Interface (AUI)				
DO+/DO–	Data Out	O		Pseudo-ECL
DI+/DI–	Data In	I		Pseudo-ECL
CI+/CI–	Control In	I		Pseudo-ECL
RXCRS	Receive Carrier Sense	I/O	High	TTL output. Input in DAI, GPSI port
TXEN	Transmit Enable	O	High	TTL. $\overline{\text{TXEN}}$ in DAI port
CLSN	Collision	I/O	High	TTL output. Input in GPSI
DXCVR	Disable Transceiver	O	Low	TTL low
STDCLK	Serial Transmit Data Clock	I/O		Output. Input in GPSI
SRDCLK	Serial Receive Data Clock	I/O		Output. Input in GPSI
Digital Attachment Interface (DAI)				
TXDAT+	Transmit Data +	O	High	TTL. See also GPSI
TXDAT–	Transmit Data –	O	Low	TTL
$\overline{\text{TXEN}}$	Transmit Enable	O	Low	TTL. See TXEN in GPSI
RXDAT	Receive Data	I		TTL. See also GPSI
RXCRS	Receive Carrier Sense	I/O	High	TTL input. Output in AUI
CLSN	Collision	I/O	High	TTL output. Input in GPSI
DXCVR	Disable Transceiver	O	High	TTL high
STDCLK	Serial Transmit Data Clock	I/O		Output. Input in GPSI
SRDCLK	Serial Receive Data Clock	I/O		Output. Input in GPSI
10BASE–T Interface				
TXD+/TXD–	Transmit Data	O		
TXP+/TXP–	Transmit Pre-distortion	O		
RXD+/RXD–	Receive Data	I		
$\overline{\text{LNKST}}$	Link Status	O	Low	Open Drain
$\overline{\text{RXPOL}}$	Receive Polarity	O	Low	Open Drain
TXEN	Transmit Enable	O	High	TTL. $\overline{\text{TXEN}}$ in DAI port
RXCRS	Receive Carrier Sense	I/O	High	TTL output. Input in DAI, GPSI port
CLSN	Collision	I/O	High	TTL output. Input in GPSI
DXCVR	Disable Transceiver	O	High	TTL high
STDCLK	Serial Transmit Data Clock	I/O		Output. Input in GPSI
SRDCLK	Serial Receive Data Clock	I/O		Output. Input in GPSI
General Purpose Serial Interface (GPSI)				
STDCLK	Serial Transmit Data Clock	I/O		Input
TXDAT+	Transmit Data +	O	High	TTL. See also DAI port
TXEN	Transmit Enable	O	High	TTL. $\overline{\text{TXEN}}$ in DAI port
SRDCLK	Serial Receive Data Clock	I/O		Input. See also EADI port
RXDAT	Receive Data	I		TTL. See also DAI port
RXCRS	Receive Carrier Sense	I/O	High	TTL input. Output in AUI
CLSN	Collision	I/O	High	TTL input
DXCVR	Disable Transceiver	O	Low	TTL low

PIN SUMMARY (continued)

Pin Name	Pin Function	Type	Active	Comment
External Address Detection Interface (EADI)				
SF/BD	Start Frame/Byte Delimiter	O	High	
SRD	Serial Receive Data	O	High	
$\overline{\text{EAM/R}}$	External Address Match/Reject	I	Low	
SRDCLK	Serial Receive Data Clock	I/O		Output except in GPSI
Host System Interface				
DBUS15–0	Data Bus	I/O	High	
ADD4–0	Address	I	High	
R/$\overline{\text{W}}$	Read/Write	I	High/Low	
$\overline{\text{RDTREQ}}$	Receive Data Transfer Request	O	Low	
$\overline{\text{TDTREQ}}$	Transmit Data Transfer Request	O	Low	
$\overline{\text{DTV}}$	Data Transfer Valid	O	Low	Tristate
$\overline{\text{EOF}}$	End Of Frame	I/O	Low	
$\overline{\text{BE0}}$	Byte Enable 0	I	Low	
$\overline{\text{BE1}}$	Byte Enable 1	I	Low	
$\overline{\text{CS}}$	Chip Select	I	Low	
$\overline{\text{FDS}}$	FIFO Data Strobe	I	Low	
$\overline{\text{INTR}}$	Interrupt	O	Low	Open Drain
EDSEL	Edge Select	I	High	
$\overline{\text{TC}}$	Timing Control	I	Low	Internal pull-up
SCLK	System Clock	I	High	
$\overline{\text{RESET}}$	Reset	I	Low	
IEEE 1149.1 Test Access Port (TAP) Interface				
TCK	Test Clock	I		Internal pull-up
TMS	Test Mode Select	I		Internal pull-up
TDI	Test Data Input	I		Internal pull-up
TDO	Test Data Out	O		
General Interface				
XTAL1	Crystal Input	I		CMOS
XTAL2	Crystal Output	O		CMOS
$\overline{\text{SLEEP}}$	Sleep Mode	I	Low	TTL
DV_{DD}	Digital Power (4 pins)	P		
DV_{SS}	Digital Ground (6 pins)	P		
AV_{DD}	Analog Power (4 pins)	P		
AV_{SS}	Analog Ground (2 pins)	P		

PIN DESCRIPTION

Network Interfaces

The MACE device has five potential network interfaces. Only one of the interfaces that provides physical network attachment can be used (active) at any time. Selection between the AUI, 10BASE-T, DAI or GPSI ports is provided by programming the PHY Configuration Control register. The EADI port is effectively active at all times. Some signals, primarily used for status reporting, are active for more than one single interface (the CLSN pin for instance). Under each of the descriptions for the network interfaces, the primary signals which are unique to that interface are described. Where signals are active for multiple interfaces, they are described once under the interface most appropriate.

Attachment Unit Interface (AUI)

CI+/CI

Control In ***(Input)***

A differential input pair, signalling the MACE device that a collision has been detected on the network media, indicated by the CI± inputs being exercised with 10 MHz pattern of sufficient amplitude and duration. Operates at pseudo-ECL levels.

DI+/DI

Data In ***(Input)***

A differential input pair to the MACE device for receiving Manchester encoded data from the network. Operates at pseudo-ECL levels.

DO+/DO

Data Out ***(Output)***

A differential output pair from the MACE device for transmitting Manchester encoded data to the network. Operates at pseudo-ECL levels.

Digital Attachment Interface (DAI)

TXDAT+/TXDAT–

Transmit Data ***(Output)***

When the DAI port is selected, TXDAT± are configured as a complementary pair for Manchester encoded data output from the MACE device, used to transmit data to a local external network transceiver. During valid transmission (indicated by $\overline{\text{TXEN}}$ low), a logical *1* is indicated by the TXDAT+ pin being in the high state and TXDAT– in the low state; and a logical *0* is indicated by the TXDAT+ pin being in the low state and TXDAT– in the high state. During idle ($\overline{\text{TXEN}}$ high), TXDAT+ will be in the high state, and TXDAT– in the low state. When the GPSI port is selected, TXDAT+ will provide NRZ data output from the MAC core, and TXDAT– will be held in the LOW state. Operates at TTL levels. The operations of TXDAT+ and TXDAT– are defined in the following tables:

TXDAT+ Configuration

$\overline{\text{SLEEP}}$	PORTSEL [1–0]	ENPLSIO	Interface Description	Pin Function
0	XX	X	Sleep Mode	High Impedance
1	00	1	AUI	High Impedance (Note 2)
1	01	1	10BASE-T	High Impedance (Note 2)
1	10	1	DAI Port	TXDAT+ Output
1	11	1	GPSI	TXDAT+ Output
1	XX	0	Status Disabled	High Impedance (Note 2)

TXDAT– Configuration

$\overline{\text{SLEEP}}$	PORTSEL [1–0]	ENPLSIO	Interface Description	Pin Function
0	XX	X	Sleep Mode	High Impedance
1	00	1	AUI	High Impedance
1	01	1	10BASE-T	High Impedance
1	10	1	DAI Port	TXDAT– Output
1	11	1	GPSI	LOW
1	XX	0	Status Disabled	High Impedance

Notes:

1. *PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14).*
2. *This pin should be externally terminated, if unused, to reduce power consumption.*

$\overline{\text{TXEN}}$/TXEN

Transmit Enable *(Output)*

When the AUI port is selected (PORTSEL [1–0] = 00), an output indicating that the AUI DO± differential output has valid Manchester encoded data is presented. When the 10BASE-T port is selected (PORTSEL [1–0] = 01), indicates that Manchester data is being output on the TXD±/TXP± complementary outputs. When the DAI port is selected (PORTSEL [1–0] = 10), indicates that Manchester data is being output on the DAI port TXDAT± complementary outputs. When the GPSI port is selected (PORTSEL [1–0] =11), indicates that NRZ data is being output from the MAC core of the MACE device, to an external Manchester encoder/decoder, on the TXDAT+ output. Active low when the DAI port is selected, active high when the AUI, 10 BASE-T or GPSI is selected. Operates at TTL levels.

RXDAT

Receive Data *(Input)*

When the DAI port is selected (PORTSEL [1–0] = 10), the Manchester encoded data input to the integrated clock recovery and Manchester decoder of the MACE device, from an external network transceiver. When the GPSI port is selected (PORTSEL [1–0] =11), the NRZ decoded data input to the MAC core of the MACE device, from an external Manchester encoder/decoder. Operates at TTL levels.

RXCRS

Receive Carrier Sense *(Input/Output)*

When the AUI port is selected (PORTSEL [1–0] = 00), an output indicating that the DI± input pair is receiving valid Manchester encoded data from the external transceiver which meets the signal amplitude and pulse width requirements. When the 10BASE-T port is selected (PORTSEL [1–0] = 01), an output indicating that the RXD± input pair is receiving valid Manchester encoded data from the twisted pair cable which meets the signal amplitude and pulse width requirements. RXCRS will be asserted high for the entire duration of the receive message. When the DAI port is selected (PORTSEL [1–0] = 10), an input signaling the MACE device that a receive carrier condition has been detected on the network, and valid Manchester encoded data is being presented to the MACE device on the RXDAT line. When the GPSI port is selected (PORTSEL [1–0] = 11), an input signalling the internal MAC core that valid NRZ data is being presented on the RXDAT input. Operates at TTL levels.

$\overline{\text{TXEN}}$/TXEN Configuration

$\overline{\text{SLEEP}}$	PORTSEL [1–0]	ENPLSIO	Interface Description	Pin Function
0	XX	X	Sleep Mode	High Impedance
1	00	1	AUI	TXEN Output
1	01	1	10BASE-T	TXEN Output
1	10	1	DAI Port	$\overline{\text{TXEN}}$ Output
1	11	1	GPSI	TXEN Output
1	XX	0	Status Disabled	High Impedance (Note 3)

Notes:

1. *PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14).*
2. *When the GPSI port is selected, TXEN should have an external pull-down attached (e.g. 3.3kΩ) to ensure the output is held inactivebefore ENPLSIO is set.*
3. *This pin should be externally terminated, if unused, to reduce power consumption.*

RXDAT Configuration

$\overline{\text{SLEEP}}$	PORTSEL [1–0]	ENPLSIO	Interface Description	Pin Function
0	XX	X	Sleep Mode	High Impedance
1	00	1	AUI	High Impedance (Note 2)
1	01	1	10BASE-T	High Impedance (Note 2)
1	10	1	DAI Port	RXDAT Input
1	11	1	GPSI	RXDAT Input
1	XX	0	Status Disabled	High Impedance (Note 2)

Notes:

1. *PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14).*
2. *This pin should be externally terminated, if unused, to reduce power consumption.*

RXCRS Configuration

$\overline{\text{SLEEP}}$	PORTSEL [1–0]	ENPLSIO	Interface Description	Pin Function
0	XX	X	Sleep Mode	High Impedance
1	00	1	AUI	RXCRS Output
1	01	1	10BASE-T	RXCRS Output
1	10	1	DAI Port	RXCRS Input
1	11	1	GPSI	RXCRS Input
1	XX	0	Status Disabled	High Impedance (Note 2)

Notes:

1. *PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14).*
2. *This pin should be externally terminated, if unused, to reduce power consumption.*

DXCVR

Disable Transceiver *(Output)*

An output from the MACE device to indicate the network port in use, as programmed by the ASEL bit or the PORTSEL [1–0] bits. The output is provided to allow power down of an external DC-to-DC converter, typically used to provide the voltage requirements for an external 10BASE2 transceiver.

When the Auto Select (ASEL) feature is enabled, the state of the PORTSEL [1–0] bits is overridden, and the network interface will be selected by the MACE device, dependent only on the status of the 10BASE-T link. If the link is active ($\overline{\text{LNKST}}$ pin driven LOW) the 10BASE-T port will be used as the active network interface. If the link is inactive ($\overline{\text{LNKST}}$ pin pulled HIGH) the AUI port will be used as the active network interface. Auto Select will continue to operate even when the $\overline{\text{SLEEP}}$ pin is asserted if the RWAKE bit has been set. The AWAKE bit does not allow the Auto Select function, and only the receive section of 10BASE-T port will be active (DXCVR = HIGH).

Active (HIGH) when either the 10BASE-T or DAI port is selected. Inactive (LOW) when the AUI or GPSI port is selected.

DXCVR Configuration—$\overline{\text{SLEEP}}$ Operation

$\overline{\text{SLEEP}}$ Pin	RWAKE Bit	AWAKE Bit	ASEL Bit	$\overline{\text{LNKST}}$ Pin	PORTSEL [1–0] Bits	Interface Description	Pin Function
0	0	0	X	High Impedance	XX	Sleep Mode	High Impedance
0	1	0	0	High Impedance	00	AUI with EADI port	LOW
0	1	0	0	High Impedance	01	10BASE-T with EADI port	HIGH
0	1	0	0	High Impedance	10	Invalid	HIGH
0	1	0	0	High Impedance	11	Invalid	LOW
0	1	0	1	High Impedance	0X	AUI with EADI port	LOW
0	1	0	1	High Impedance	0X	10BASE-T with EADI port	HIGH
0	1	1	1	HIGH	0X	AUI with EADI port	LOW
0	1	1	1	LOW	0X	10BASE-T with EADI port	HIGH
0	0	1	X	X	0X	10BASE-T	HIGH

Note: *RWAKE and ASEL are located in the PHY Configuration Control register (REG ADDR 15). PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14). All bits must be programmed prior to the assertion of the* $\overline{\text{SLEEP}}$ *pin.*

DXCVR Configuration—Normal Operation

$\overline{\text{SLEEP}}$ Pin	ASEL Bit	$\overline{\text{LNKST}}$ Pin	PORTSEL [1–0] Bits	ENPLSIO Bit	Interface Description	Pin Function
1	X	X	XX	X	SIA Test Mode	High Impedance
1	0	X	00	X	AUI	LOW
1	0	X	01	X	10BASE-T	HIGH
1	0	X	10	X	DAI Port	HIGH
1	0	X	11	X	GPSI	LOW
1	1	HIGH	0X	X	AUI	LOW
1	1	LOW	0X	X	10BASE-T	HIGH

Note: *RWAKE and ASEL are located in the PHY Configuration Control register (REG ADDR 15). PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14).*

10BASE-T Interface

TXD+, TXD–

Transmit Data ***(Output)***

10BASE-T port differential drivers.

TXP+, TXP–

Transmit Pre-Distortion ***(Output)***

Transmit wave form differential driver for pre-distortion.

RXD+, RXD–

Receive Data ***(Input)***

10BASE-T port differential receiver. These pins should be externally terminated to reduce power consumption if the 10BASE-T interface is not used.

$\overline{\text{LNKST}}$

Link Status ***(Output Open Drain)***

This pin is driven LOW if the link is identified as functional. If the link is determined to be nonfunctional, due to missing idle link pulses or data packets, then this pin is not driven (requires external pull-up). In the LOW output state, the pin is capable of sinking a maximum of 12 mA and can be used to drive an LED.

This feature can be disabled by setting the Disable Link Test (DLNKTST) bit in the PHY Configuration Control register. In this case the internal Link Test Receive function is disabled, the $\overline{\text{LNKST}}$ pin will be driven LOW, and the Transmit and Receive functions will remain active regardless of arriving idle link pulses and data. The internal 10BASE-T MAU will continue to generate idle link pulses irrespective of the status of the DLNKTST bit.

$\overline{\text{RXPOL}}$

Receive Polarity ***(Output, Open Drain)***

The twisted pair receiver is capable of detecting a receive signal with reversed polarity (wiring error). The $\overline{\text{RXPOL}}$ pin is normally in the LOW state, indicating correct polarity of the received signal. If the receiver detects a received packet with reversed polarity, then this pin is not driven (requires external pull–up) and the polarity of subsequent packets are inverted. In the LOW output state, this pin is capable of sinking a maximum of 12mA and can be used to drive an LED.

The polarity correction feature can be disabled by setting the Disable Auto Polarity Correction (DAPC) bit in the PHY Configuration Control register. In this case, the Receive Polarity correction circuit is disabled and the internal receive signal remains non-inverted, irrespective of the received signal. Note that $\overline{\text{RXPOL}}$ will continue to reflect the polarity detected by the receiver.

General Purpose Serial Interface (GPSI)

STDCLK

Serial Transmit Data Clock ***(Input/Output)***

When either the AUI, 10BASE-T or DAI port is selected, STDCLK is an output operating at one half the crystal or XTAL1 frequency. STDCLK is the encoding clock for Manchester data transferred to the output of either the AUI DO± pair, the 10BASE-T TXD±/TXP± pairs, or the DAI port TXDAT± pair. When using the GPSI port, STDCLK is an input at the network data rate, provided by the external Manchester encode/decoder, to strobe out the NRZ data presented on the TXDAT+ output.

STDCLK Configuration

$\overline{SLEEP}$	PORTSEL [1–0]	ENPLSIO	Interface Description	Pin Function
0	XX	X	Sleep Mode	High Impedance
1	00	1	AUI	STDCLK Output
1	01	1	10BASE-T	STDCLK Output
1	10	1	DAI Port	STDCLK Output
1	11	1	GPSI	STDCLK Input
1	XX	0	Status Disabled	High Impedance (Note 2)

Notes:

1. *PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14).*
2. *This pin should be externally terminated, if unused, to reduce power consumption.*

CLSN

Collision *(Input/Output)*

An external indication that a collision condition has been detected by the (internal or external) Medium Attachment Unit (MAU), and that signals from two or more nodes are present on the network. When the AUI port is selected (PORTSEL [1–0] = 00), CLSN will be activated when the CI± input pair is receiving a collision indication from the external transceiver. CLSN will be asserted high for the entire duration of the collision detection, but will not be asserted during the SQE Test message following a transmit message on the AUI. When the 10BASE-T port is selected (PORTSEL [1–0] = 01), CLSN will be asserted high when simultaneous transmit and receive activity is detected (logically detected when TXD±/TXP± and RXD± are both active). When the DAI port is selected (PORTSEL [1–0] = 10), CLSN will be asserted high when simultaneous transmit and receive activity is detected (logically detected when RXCRS and $\overline{TXEN}$ are both active). When the GPSI port is selected (PORTSEL [1–0] = 11), an input from the external Manchester encoder/decoder signaling the MACE device that a collision condition has been detected on the network, and any receive frame in progress should be aborted.

External Address Detection Interface (EADI)

SF/BD

Start Frame/Byte Delimiter *(Output)*

The external indication that a start of frame delimiter has been received. The serial bit stream will follow on the Serial Receive Data pin (SRD), commencing with the destination address field. SF/BD will go high for 4 bit times (400 ns) after detecting the second 1 in the SFD of a received frame. SF/BD will subsequently toggle every 400 ns (1.25 MHz frequency) with the rising edge indicating the start (first bit) in each subsequent byte of the received serial bit stream. SF/BD will be inactive during frame transmission.

SRD

Serial Receive Data *(Output)*

SRD is the decoded NRZ data from the network. It is available for external address detection. Note that when the 10BASE-T port is selected, transition on SRD will only occur during receive activity. When the AUI or DAI port is selected, transition on SRD will occur during both transmit and receive activity.

CLSN Configuration

$\overline{SLEEP}$	PORTSEL [1–0]	ENPLSIO	Interface Description	Pin Function
0	XX	X	Sleep Mode	High Impedance
1	00	1	AUI	CLSN Output
1	01	1	10BASE-T	CLSN Output
1	10	1	DAI Port	CLSN Output
1	11	1	GPSI	CLSN Input
1	XX	0	Status Disabled	High Impedance (Note 2)

Notes:

1. *PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14).*
2. *This pin should be externally terminated, if unused, to reduce power consumption.*

$\overline{\text{EAM}}/\overline{\text{R}}$

External Address Match/Reject *(Input)*

The incoming frame will be received dependent on the receive operational mode of the MACE device, and the polarity of the $\overline{\text{EAM/R}}$ pin. The $\overline{\text{EAM/R}}$ pin function is programmed by use of the M/$\overline{\text{R}}$ bit in the Receive Frame Control register. If the bit is set, the pin is configured as $\overline{\text{EAM}}$. If the bit is reset, the pin is configured as $\overline{\text{EAR}}$. $\overline{\text{EAM/R}}$ can be asserted during packet reception to accept or reject packets based on an external address comparison.

SRDCLK

Serial Receive Data Clock *(Input/Output)*

The Serial Receive Data (SRD) output is synchronous to SRDCLK running at the 10MHz receive data clock frequency. The pin is configured as an input, only when the GPSI port is selected. Note that when the 10BASE-T port is selected, transition on SRDCLK will only occur during receive activity. When the AUI or DAI port is selected, transition on SRDCLK will occur during both transmit and receive activity.

SRD Configuration

$\overline{\text{SLEEP}}$	PORTSEL [1–0]	ENPLSIO	Interface Description	Pin Function
0	XX	X	Sleep Mode	High Impedance
1	00	1	AUI	SRD Output
1	01	1	10BASE-T	SRD Output
1	10	1	DAI Port	SRD Output
1	11	1	GPSI	SRD Output
1	XX	0	Status Disabled	High Impedance

Note: *PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14).*

SRDCLK Configuration

$\overline{\text{SLEEP}}$	PORTSEL [1–0]	ENPLSIO	Interface Description	Pin Function
0	XX	X	Sleep Mode	High Impedance
1	00	1	AUI	SRDCLK Output
1	01	1	10BASE-T	SRDCLK Output
1	10	1	DAI Port	SRDCLK Output
1	11	1	GPSI	SRDCLK Input
1	XX	0	Status Disabled	High Impedance (Note 2)

Notes:

1. *PORTSEL [1–0] and ENPLSIO are located in the PLS Configuration Control register (REG ADDR 14).*
2. *This pin should be externally terminated, if unused, to reduce power consumption.*

HOST SYSTEM INTERFACE

DBUS15–0

Data Bus *(Input/Output/3-state)*

DBUS contains read and write data to and from internal registers and the Transmit and Receive FIFOs.

ADD4–0

Address Bus *(Input)*

ADD is used to access the internal registers and FIFOs to be read or written.

R/$\overline{W}$

Read/Write *(Input)*

Indicates the direction of data flow during the MACE device register, Transmit FIFO, or Receive FIFO accesses.

$\overline{RDTREQ}$

Receive Data Transfer Request *(Output)*

Receive Data Transfer Request indicates that there is data in the Receive FIFO to be read. When $\overline{RDTREQ}$ is asserted there will be a minimum of 16 bytes to be read except at the completion of the frame, in which case $\overline{EOF}$ will be asserted. $\overline{RDTREQ}$ can be programmed to request receive data transfer when 16, 32 or 64 bytes are available in the Receive FIFO, by programming the Receive FIFO Watermark (RCVFW bits) in the FIFO Configuration Control register. The first assertion of $\overline{RDTREQ}$ will not occur until at least 64 bytes have been received, and the frame has been verified as non runt. Runt packets will normally be deleted from the Receive FIFO with no external activity on $\overline{RDTREQ}$. When Runt Packet Accept is enabled (RPA bit) in the User Test Register, $\overline{RDTREQ}$ will be asserted when the runt packet completes, and the entire frame resides in the Receive FIFO. $\overline{RDTREQ}$ will be asserted only when Enable Receive (ENRCV) is set in the MAC Configuration Control register.

The RCVFW can be overridden by enabling the Low Latency Receive function (setting LLRCV bit) in the Receive Frame Control register, which allows $\overline{RDTREQ}$ to be asserted after only 12 bytes have been received. Note that use of this function exposes the system interface to premature termination of the receive frame, due to network events such as collisions or runt packets. It is the responsibility of the system designer to provide adequate recovery mechanisms for these conditions.

$\overline{TDTREQ}$

Transmit Data Transfer Request *(Output)*

Transmit Data Transfer Request indicates there is room in the Transmit FIFO for more data. $\overline{TDTREQ}$ is asserted when there are a minimum of 16 empty bytes in the Transmit FIFO. $\overline{TDTREQ}$ can be programmed to request transmit data transfer when 16, 32 or 64 bytes are available in the Transmit FIFO, by programming the Transmit FIFO Watermark (XMTFW bits) in the FIFO Configuration Control register. $\overline{TDTREQ}$ will be asserted only when Enable Transmit (ENXMT) is set in the MAC Configuration Control register.

$\overline{FDS}$

FIFO Data Select *(Input)*

FIFO Data Select allows direct access to the transmit or Receive FIFO without use of the ADD address bus. $\overline{FDS}$ must be activated in conjunction with R/$\overline{W}$. When the MACE device samples R/$\overline{W}$ as high and $\overline{FDS}$ low, a read cycle from the Receive FIFO will be initiated. When the MACE chip samples R/$\overline{W}$ and $\overline{FDS}$ low, a write cycle to the Transmit FIFO will be initiated. The $\overline{CS}$ line should be inactive (high) when FIFO access is requested using the $\overline{FDS}$ pin. If the MACE device samples both $\overline{CS}$ and $\overline{FDS}$ as active simultaneously, no cycle will be executed, and $\overline{DTV}$ will remain inactive.

$\overline{DTV}$

Data Transfer Valid *(Output/3-state)*

When asserted, indicates that the read or write operation has completed successfully. The absence of $\overline{DTV}$ at the termination of a host access cycle on the MACE device indicates that the data transfer was unsuccessful. $\overline{DTV}$ need not be used if the system interface can guarantee that the latency to $\overline{TDTREQ}$ and $\overline{RDTREQ}$ assertion and de-assertion will not cause the Transmit FIFO to be over-written or the Receive FIFO to be over-read. In this case, the latching or strobing of read or write data can be synchronized to the SCLK input rather than to the $\overline{DTV}$ output.

$\overline{EOF}$

End Of Frame *(Input/Output/3-state)*

End Of Frame will be asserted by the MACE device when the last byte/word of frame data is read from the Receive FIFO, indicating the completion of the frame data field for the receive message. End Of Frame must be asserted low to the MACE device when the last byte/word of the frame is written into the Transmit FIFO.

$\overline{BE1}$–$\overline{0}$

Byte Enable *(Input)*

Used to indicate the active portion of the data transfer to or from the internal FIFOs. For word (16-bit) transfers, both $\overline{BE0}$ and $\overline{BE1}$ should be activated by the external host/controller. Single byte transfers are performed by identifying the active data bus byte and activating only one of the two signals. The function of the $\overline{BE1}$–$\overline{0}$ pins is programmed using the BSWP bit (BIU Configuration Control register, bit 6). $\overline{BE1}$–$\overline{0}$ are not required for accesses to MACE device registers.

$\overline{CS}$

Chip Select *(Input)*

Used to access the MACE device FIFOs and internal registers locations using the ADD address bus. The FIFOs may alternatively be directly accessed without supplying the FIFO address, by using the $\overline{FDS}$ and R/$\overline{W}$ pins.

$\overline{\text{INTR}}$
Interrupt ***(Output, Open Drain)***

An attention signal indicating that one or more of the following status flags are set: XMTINT, RCVINT, MPCO, RPCO, RCVCCO, CERR, BABL or JAB. Each interrupt source can be individually masked. No interrupt condition can take place in the MACE device immediately after a hardware or software reset.

$\overline{\text{RESET}}$
Reset ***(Input)***

Reset clears the internal logic. Reset can be asynchronous to SCLK, but must be asserted for a minimum duration of 15 SCLK cycles.

SCLK
System Clock ***(Input)***

The system clock input controls the operational frequency of the slave interface to the MACE device and the internal processing of frames. SCLK is unrelated to the 20 MHz clock frequency required for the 802.3/Ethernet interface. The SCLK frequency range is 1 MHz–25 MHz.

EDSEL
System Clock Edge Select ***(Input)***

EDSEL is a static input that allows System Clock (SCLK) edge selection. If EDSEL is tied high, the bus interface unit will assume falling edge timing. If EDSEL is tied low, the bus interface unit will assume rising edge timing, which will effectively invert the SCLK as it enters the MACE device, i.e., the address, control lines ($\overline{\text{CS}}$, R/$\overline{\text{W}}$, $\overline{\text{FDS}}$, etc) and data are all latched on the rising edge of SCLK, and data out is driven off the rising edge of SCLK.

$\overline{\text{TC}}$
Timing Control ***(Input)***

The Timing Control input conditions the minimum number of System Clocks (SCLK) cycles taken to read or write the internal registers and FIFOs. $\overline{\text{TC}}$ can be used as a wait state generator, to allow additional time for data to be presented by the host during a write cycle, or allow additional time for the data to be latched during a read cycle. $\overline{\text{TC}}$ has an internal ($\overline{\text{SLEEP}}$ disabled) pull up.

Timing Control

$\overline{\text{TC}}$	Number of Clocks
1	2
0	3

IEEE 1149.1 TEST ACCESS PORT (TAP) INTERFACE

TCK
Test Clock ***(Input)***

The clock input for the boundary scan test mode operation. TCK can operate up to 10 MHz. TCK has an internal (not $\overline{\text{SLEEP}}$ disabled) pull up.

TMS
Test Mode Select ***(Input)***

A serial input bit stream used to define the specific boundary scan test to be executed. TMS has an internal (not $\overline{\text{SLEEP}}$ disabled) pull up.

TDI
Test Data Input ***(Input)***

The test data input path to the MACE device. TDI has an internal (not $\overline{\text{SLEEP}}$ disabled) pull up.

TDO
Test Data Out ***(Output)***

The test data output path from the MACE device.

GENERAL INTERFACE

XTAL1
Crystal Connection ***(Input)***

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. Internally, the 20 MHz crystal frequency is divided by two which determines the network data rate. Alternatively, an external 20 MHz CMOS-compatible clock signal can be used to drive this pin. The MACE device supports the use of 50 pF crystals to generate a 20 MHz frequency which is compatible with the IEEE 802.3 network frequency tolerance and jitter specifications.

XTAL2
Crystal Connection ***(Output)***

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. If an external clock generator is used on XTAL1, then XTAL2 should be left unconnected.

$\overline{\text{SLEEP}}$
Sleep Mode ***(Input)***

The optimal power savings made is extracted by asserting the $\overline{\text{SLEEP}}$ pin with both the Auto Wake (AWAKE bit) and Remote Wake (RWAKE bit) functions disabled. In this "deep sleep" mode, all outputs will be forced into their inactive or high impedance state, and all inputs will be ignored except for the $\overline{\text{SLEEP}}$, $\overline{\text{RESET}}$, SCLK, TCK,

TMS, and TDI pins. SCLK must run for 5 cycles after the assertion of $\overline{SLEEP}$. During the "Deep Sleep", the SCLK input can be optionally suspended for maximum power savings. Upon exiting "Deep Sleep", the hardware $\overline{RESET}$ pin must be asserted and the SCLK restored. The system must delay the setting of the bits in the MAC configuration Control Register of the internal analog circuits by 1 ns to allow for stabilization.

If the AWAKE bit is set prior to the activation of $\overline{SLEEP}$, the 10BASE-T receiver and the $\overline{LNKST}$ output pin remain operational.

If the RWAKE bit is set prior to $\overline{SLEEP}$ being asserted, the Manchester encoder/decoder, AUI and 10BASE-T cells remain operational, as do the SRD, SRDCLK and SF/BD outputs.

The input on XTAL1 must remain active for the AWAKE or RWAKE features to operate. After exit from the Auto Wake or Remote Wake modes, activation of hardware $\overline{RESET}$ is not required when $\overline{SLEEP}$ is reasserted.

On deassertion of $\overline{SLEEP}$, the MACE device will go through an internally generated hardware reset sequence, requiring re-initialization of MACE registers.

Power Supply

DV_{DD}
Digital Power

There are four Digital V_{DD} pins.

DV_{SS}
Digital Ground

There are six Digital V_{SS} pins.

AV_{DD}
Analog Power

There are four analog VDD pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on the supply to the PLL in the Manchester encoder/decoder (pins 66 and 83 in PLCC, pins 67 and 88 in PQFP). These supply lines should be kept separate from the DV_{DD} lines as far back to the power supply as is practically possible.

AV_{SS}
Analog Ground

There are two analog VSS pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on the PLL supply in Manchester encoder/decoder (pin 73 in PLCC, pin 74 in PQFP). These supply lines should be kept separate from the DV_{SS} lines as far back to the power supply as is practically possible.

PIN FUNCTIONS NOT AVAILABLE WITH THE 80-PIN TQFP PACKAGE

In the 84-pin PLCC configuration, *ALL* the pins are used while in the 100-pin PQFP version, 16 pins are specified as No Connects. Moving to the 80-pin TQFP configuration requires the removal of 4 pins. Since Ethernet controllers with integrated 10BASE-T have analog portions which are very sensitive to noise, power and ground pins are not deleted. The MACE device does have several sets of media interfaces which typically go unused in most designs, however. Pins from some of these interfaces are deleted instead. Removed are the following:

- TXDAT– (previously used for the DAI interface)
- SRD (previously used for the EADI interface)
- DTV (previously used for the host interface)
- RXPOL (previously used as a receive frame polarity LED driver)

Note that pins from four separate interfaces are removed rather than removing all the pins from a single interface. Each of these pins comes from one of the four sides of the device. This is done to maintain symmetry, thus avoiding bond out problems.

In general, the most critical of the four removed pins are TXDAT– and SRD. Depending on the application, either the DAI or the EADI interface may be important. In most designs, however, this will not be the case.

PINS REMOVED AND THEIR EFFECTS

TXDAT–

The removal of TXDAT– means that the DAI interface is no longer usable. The DAI interface was designed to be used with media types that do not require DC isolation between the MAU and the DTE. Media which do not require DC isolation can be implemented more simply using the DAI interface, rather than the AUI interface. In most designs this is not a problem because most media requires DC isolation (10BASE-T, 10BASE2, 10BASE5) and will use the AUI port. About the only media which does not require DC isolation is 10BASE-F.

SRD

The SRD pin is an output pin used by the MACE device to transfer a receive data stream to external address detection logic. It is part of the EADI interface. This pin is used to help interface the MACE device to an external CAM device. Use of an external CAM is typically required when an application will operate in promiscuous mode and will need perfect filtering (i.e., the internal hash filter will not suffice). Example applications for this

sort of operation are bridges and routers. Lack of perfect filtering in these applications forces the CPU to be more involved in filtering and thus either slows the forwarding rates achieved or forces the use of a more powerful CPU.

DTV

The DTV pin is part of the host interface to the MACE device. It is used to indicate that a read or write cycle to the MACE device was successful. If DTV is not asserted at the end of a cycle, the data transfer was not successful. Basically, this will happen on a write to a full transmit FIFO or a read from an empty receive FIFO. In general, there are ways to ensure that a transfer is always valid and so this pin is not required in many designs. For instance, the TDTREQ and RDTREQ pins can be used to monitor the state of the FIFOs to ensure that data transfer only occurs at the correct times.

RXPOL

RXPOL is typically used to drive an LED indicating the polarity of receive frames. This function is not necessary for correct operation of the Ethernet and serves strictly as a status indication to a user. The status of the receive polarity is still available through the PHYCC register.

FUNCTIONAL DESCRIPTION

The Media Access Controller for Ethernet (MACE) chip embodies the Media Access Control (MAC) and Physical Signaling (PLS) sub-layers of the 802.3 Standard. The MACE device provides the IEEE defined Attachment Unit Interface (AUI) for coupling to remote Media Attachment Units (MAUs) or on-board transceivers. The MACE device also provides a Digital Attachment Interface (DAI), by-passing the differential AUI interface.

The system interface provides a fundamental data conduit to and from an 802.3 network. The MACE device in conjunction with a user defined DMA engine, provides an 802.3 interface tailored to a specific application.

In addition, the MACE device can be combined with similarly architected peripheral devices and a multichannel DMA controller, thereby providing the system with access to multiple peripheral devices with a single master interface to memory.

Network Interfaces

The MACE device can be connected to an 802.3 network using any one of the AUI, 10 BASE-T, DAI and GPSI network interfaces. The Attachment Unit Interface (AUI) provides an IEEE compliant differential interface to a remote MAU or an on-board transceiver. An integrated 10BASE-T MAU provides a direct interface for twisted pair Ethernet networks. The DAI port can connect to local transceiver devices for 10BASE2, 10BASE-T or 10BASE-F connections. A General Purpose Serial Interface (GPSI) is supported, which effectively bypasses the integrated Manchester encoder/decoder, and allows direct access to/from the integral 802.3 Media Access Controller (MAC) to provide support for external encoding/decoding schemes. The interface in use is determined by the PORTSEL [1–0] bits in the PLS Configuration Control register.

The EADI port does not provide network connectivity, but allows an optional external circuit to assist in receive packet accept/reject.

System Interface

The MACE device is a slave register based peripheral. All transfers to and from the device, including data, are performed using simple memory or I/O read and write commands. Access to all registers, including the Transmit and Receive FIFOs, are performed with identical read or write timing. All information on the system interface is synchronous to the system clock (SCLK), which allows simple external logic to be designed to interrogate the device status and control the network data flow.

The Receive and Transmit FIFOs can be read or written by driving the appropriate address lines and asserting $\overline{CS}$ and R/$\overline{W}$. An alternative FIFO access mechanism allows the use of the $\overline{FDS}$ and the R/$\overline{W}$ lines, ignoring the address lines (ADD4–0). The state of the R/$\overline{W}$ line in conjunction with the $\overline{FDS}$ input determines whether the Receive FIFO is read (R/$\overline{W}$ high) or the Transmit FIFO written (R/$\overline{W}$ low). The MACE device system interface permits interleaved transmit and receive bus transfers, allowing the Transmit FIFO to be filled (*primed*) while a frame is being received from the network and/or read from the Receive FIFO.

In receive operation, the MACE device asserts Receive Data Transfer Request ($\overline{RDTREQ}$) when the FIFO contains adequate data. For the first indication of a new receive frame, 64 bytes must be received, assuming normal operation. Once the initial 64 byte threshold has been reached, $\overline{RDTREQ}$ assertion and de-assertion is dependent on the programming of the Receive FIFO Watermark (RCVFW bits in the BIU Configuration Control register). The $\overline{RDTREQ}$ can be programmed to activate when there are 16, 32 or 64 bytes of data available in the Receive FIFO. Enable Receive (ENRCV bit in MAC Configuration Control register) must be set to assert $\overline{RDTREQ}$. If the Runt Packet Accept feature is invoked (RPA bit in User Test Register), $\overline{RDTREQ}$ will be asserted for receive frames of less than 64 bytes on the basis of internal and/or external address match only. When RPA is set, $\overline{RDTREQ}$ will be asserted when the entire frame has been received or when the initial 64 byte threshold has been exceeded. See the FIFO Sub-Systems section for further details.

Note that the Receive FIFO may not contain 64 data bytes at the time $\overline{RDTREQ}$ is asserted, if the automatic pad stripping feature has been enabled (ASTRP RCV bit in the Receive Frame Control register) and a minimum length packet with pad is received. The MACE device will check for the minimum received length from the network, strip the pad characters, and pass only the data frame through the Receive FIFO.

If the Low Latency Receive feature is enabled (LLRCV bit set in Receive Frame Control Register), $\overline{RDTREQ}$ will be asserted once a low watermark threshold has been reached (12 bytes plus some additional synchronization time). Note that the system interface will therefore be exposed to potential disruption of the receive frame due to a network condition (see the FIFO Sub-System description for additional details).

In transmit operation, the MACE device asserts Transmit Data Transfer Request ($\overline{TDTREQ}$) dependent on the programming of the Transmit FIFO Watermark (XMTFW bits in the BIU Configuration Control register). $\overline{TDTREQ}$ will be permanently asserted when the Transmit FIFO is empty. The $\overline{TDTREQ}$ can be programmed to activate when there are 16, 32 or 64 bytes of space available in the Transmit FIFO. Enable Transmit (ENXMT bit in MAC Configuration Control register) must be set to assert $\overline{TDTREQ}$. Write cycles to the Transmit FIFO will not return $\overline{DTV}$ if ENXMT is disabled, and no data will be written. The MACE device will commence the preamble sequence once the Transmit Start Point (XMTSP bits in BIU Configuration Control register) threshold is reached in the Transmit FIFO.

The Transmit FIFO data will not be overwritten until at least 512 data bits have been transmitted onto the network. If a collision occurs within the slot time (512 bit time) window, the MACE device will generate a jam sequence (a 32-bit all zeroes pattern) before ceasing the transmission. The Transmit FIFO will be reset to point at the start of the transmit data field, and the message will be retried after the random back-off interval has expired.

DETAILED FUNCTIONS

Block Level Description

The following sections describe the major sub-blocks of and the external interfaces to the MACE device.

Bus Interface Unit (BIU)

The BIU performs the interface between the host or system bus and the Transmit and Receive FIFOs, as well as all chip control and status registers. The BIU can be configured to accept data presented in either little-endian or big endian format, minimizing the external logic required to access the MACE device internal FIFOs and registers. In addition, the BIU directly supports 8-bit transfers and incorporates features to simplify interfacing to 32-bit systems using external latches.

Externally, the FIFOs appear as two independent registers located at individual addresses. The remainder of the internal registers occupy 30 additional consecutive addresses, and appear as 8-bits wide.

BIU to FIFO Data Path

The BIU operates assuming that the 16-bit data path to/from the internal FIFOs is configured as two independent byte paths, activated by the Byte Enable signals $\overline{BE0}$ and $\overline{BE1}$.

$\overline{BE0}$ and $\overline{BE1}$ are only used during accesses to the 16-bit wide Transmit and Receive FIFOs. After hardware or software reset, the BSWP bit will be cleared. FIFO accesses to the MACE device will operate assuming an Intel 80x86 type memory convention (most significant byte of a word stored in the higher addressed byte). Word data transfers to/from the FIFOs over the DBUS15–0 lines will have the least significant byte located on DBUS7–0 (activated by $\overline{BE0}$) and the most significant byte located on DBUS15–8 (activated by $\overline{BE1}$).

FIFO data can be read or written using either byte and/or word operations.

If byte operation is required, read/write transfers can be performed on either the upper or lower data bus by asserting the appropriate byte enable. For instance with BSWP = 0, reading from or writing to DBUS15–8 is accomplished by asserting $\overline{BE1}$, and allows the data stream to be read from or written to the appropriate FIFO in byte order (byte 0, byte 1,....byte n). It is equally valid to read or write the data stream using DBUS7–0 and by asserting $\overline{BE0}$. For BSWP = 1, reading from or writing to DBUS15–8 is accomplished by asserting $\overline{BE0}$, and allows the byte stream to be transferred in byte order.

When word operations are required, BSWP ensures that the byte ordering of the target memory is compatible with the 802.3 requirement to send/receive the data stream in byte ascending order. With BSWP = 0, the data transferred to/from the FIFO assumes that byte n will be on DBUS7–0 (activated by $\overline{BE0}$) and byte n+1 will be on DBUS15–8 (activated by $\overline{BE1}$). With BSWP = 1, the data transferred to/from the FIFO assumes that byte n will be presented on DBUS15–8 (activated by $\overline{BE0}$), and byte n+1 will be on DBUS7–0 (activated by $\overline{BE1}$).

There are some additional special cases to the above generalized rules, which are as follows:

(a) When performing byte read operations, both halves of the data bus are driven with identical data, effectively allowing the user to arbitrarily read from either the upper or lower data bus, when only one of the byte enables is activated.

(b) When byte write operations are performed, the Transmit FIFO latency is affected. See the FIFO Sub-System section for additional details.

(c) If a word read is performed on the last data byte of a receive frame ($\overline{EOF}$ is asserted), and the message contained an odd number of bytes but the host requested a word operation by asserting both $\overline{BE0}$ and $\overline{BE1}$, then the MACE device will present one valid and one non-valid byte on the data bus. The placement of valid data for the data byte is dependent on the target memory architecture. Regardless of BSWP, the single valid byte will be read from the $\overline{BE0}$ memory bank. If BSWP = 0, $\overline{BE0}$ corresponds to DBUS7–0; if BSWP = 1, $\overline{BE0}$ corresponds to DBUS15–8.

(d) If a byte read is performed when the last data byte is read for a receive frame (when the MACE device activates the $\overline{EOF}$ signal), then the same byte will be presented on both the upper and lower byte of the data bus, regardless of which byte enable was activated (as is the case for all byte read operations).

(e) When writing the last byte in a transmit message to the Transmit FIFO, the portion of the data bus that the last byte is transferred over is irrelevant, providing the appropriate byte enable is used. For BSWP = 0, data can be presented on DBUS7–0 using $\overline{BE0}$ or DBUS15–8 using $\overline{BE1}$. For BSWP = 1, data can be presented on DBUS7–0 using $\overline{BE1}$ or DBUS15–8 using $\overline{BE0}$.

(f) When neither $\overline{BE0}$ nor $\overline{BE1}$ are asserted, no data transfer will take place. $\overline{DTV}$ will not be asserted.

Byte Alignment For FIFO Read Operations

$\overline{BE0}$	$\overline{BE1}$	BSWP	DBUS7–0	DBUS15–8
0	0	0	n	n+1
0	1	0	n	n
1	0	0	n	n
1	1	0	X	X
0	0	1	n+1	n
0	1	1	n	n
1	0	1	n	n
1	1	1	X	X

Byte Alignment For FIFO Write Operations

$\overline{BE0}$	$\overline{BE1}$	BSWP	DBUS7–0	DBUS15–8
0	0	0	n	n+1
0	1	0	n	X
1	0	0	X	n
1	1	0	X	X
0	0	1	n+1	n
0	1	1	X	n
1	0	1	n	X
1	1	1	X	X

BIU to Control and Status Register Data Path

All registers in the address range 2–31 are 8-bits wide. When a read cycle is executed on any of these registers, the MACE device will drive data on both bytes of the data bus, regardless of the programming of BSWP. When a write cycle is executed, the MACE device strobes in data based on the programming of BSWP as shown in the tables below. All accesses to addresses 2–31 are independent of the $\overline{BE0}$ and $\overline{BE1}$ pins.

Byte Alignment For Register Read Operations

$\overline{BE0}$	$\overline{BE1}$	BSWP	DBUS7–0	DBUS15–8
X	X	0	Read Data	Read Data
X	X	1	Read Data	Read Data

Byte Alignment For Register Write Operations

$\overline{BE0}$	$\overline{BE1}$	BSWP	DBUS7–0	DBUS15–8
X	X	0	Write Data	X
X	X	1	X	Write Data

FIFO Sub-System

The MACE device has two independent FIFOs, with 128-bytes for receive and 136-bytes for transmit operations. The FIFO sub-system contains both the FIFOs, and the control logic to handle normal and exception related conditions.

The Transmit and Receive FIFOs interface on the network side with the serializer/de-serializer in the MAC engine. The BIU provides access between the FIFOs and the host system to enable the movement of data to and from the network.

Internally, the FIFOs appear to the BIU as independent 16-bit wide registers. Bytes or words can be written to the Transmit FIFO (XMTFIFO), or read from the Receive FIFO (RCVFIFO). Byte and word transfers can be mixed in any order. The BIU will ensure correct byte ordering dependent on the target host system, as determined by the programming of the BSWP bit in the BIU Configuration Control register.

The XMTFIFO and RCVFIFO have three different modes of operation. These are Normal (Default), Burst and Low Latency Receive. Default operation will be used after the hardware $\overline{RESET}$ pin or software SWRST bit have been activated. The remainder of this general description applies to all modes except where specific differences are noted.

Transmit FIFO—General Operation:

When writing bytes to the XMTFIFO, certain restrictions apply. These restrictions have a direct influence on the latency provided by the FIFO to the host system. When a byte is written to the FIFO location, the entire word location is used. The unused byte is marked as a *hole* in the XMTFIFO. These *holes* are skipped during the serialization process performed by the MAC engine, when the bytes are unloaded from the XMTFIFO.

For instance, assume the Transmit FIFO Watermark (XMTFW) is set for 32 write cycles. If the host writes byte wide data to the XMTFIFO, after 36 write cycles there will be space left in the XMTFIFO for only 32 more write cycles. Therefore $\overline{TDTREQ}$ will de-assert even though only 36-bytes of data have been loaded into the XMTFIFO. Transmission will not commence until 64-bytes or the *End-of-Frame* are available in the XMFIFO, so transmission would not start, and $\overline{TDTREQ}$ would remain de-asserted. Hence for byte wide data transfers, the XMTFW should be programmed to the 8 or 16 write cycle limit, or the host should ensure that sufficient data will be written to the XMTFIFO after $\overline{TDTREQ}$ has been de-asserted (which is permitted), to guarantee that the transmission will commence. A third alternative is to program the Transmit Start Point (XMTSP) in the BIU Configuration Control register to below the 64-byte default; thereby imposing a lower latency to the host system requiring additional data to

ensure the XMTFIFO does not underflow during the transmit process, versus using the default XMTSP value. Note that if 64 single byte writes are executed on the XMTFIFO, and the XMTSP is set to 64-bytes, the transmission will commence, and all 64-bytes of information will be accepted by the XMTFIFO.

The number of write cycles that the host uses to write the packet into the Transmit FIFO will also directly influence the amount of space utilized by the transmit message. If the number of write cycles (n) required to transfer a packet to the Transmit FIFO is even, the number of bytes used in the Transmit FIFO will be 2*n. If the number of write cycles required to transfer a packet to the Transmit FIFO is odd, the number of bytes used in the Transmit FIFO will be 2*n + 2 because the *End Of Frame* indication in the XMTFIFO is always placed at the end of a 4-byte boundary. For example, a 32-byte message written as bytes (n = 32 cycles) will use 64-bytes of space in the Transmit FIFO (2*n = 64), whereas a 65-byte message written as 32 words and 1 byte (n = 33 cycles) would use 68-bytes (2*n + 2 = 68) .

The Transmit FIFO has been sized appropriately to minimize the system interface overhead. However, consideration must be given to overall system design if byte writes are supported. In order to guarantee that sufficient space is present in the XMTFIFO to accept the number of write cycles programmed by the XMTFW (including an *End Of Frame* delimiter), $\overline{\text{TDTREQ}}$ may go inactive before the XMTSP threshold is reached when using the non burst mode (XMTBRST = 0). For instance, assume that the XMTFW is programmed to allow 32 write cycles (default), and XMTSP is programmed to require 64 bytes (default) before starting transmission. Assuming that the host bursts the transmit data in a 32 cycle block, writing a single byte anywhere within this block will mean that XMTSP will not have been reached. This would be a typical scenario if the transmit data buffer was not aligned to a word boundary. The MACE device will continue to assert $\overline{\text{TDTREQ}}$ since an additional 36 write cycles can still be executed. If the host starts a second burst, the XMTSP will be reached, and $\overline{\text{TDTREQ}}$ will deassert when less that 32 write cycle can be performed although the data written by the host will continue to be accepted.

The host must be aware that additional space exists in the XMTFIFO although $\overline{\text{TDTREQ}}$ becomes inactive, and must continue to write data to ensure the XMTSP threshold is achieved. No transmit activity will commence until the XMTSP threshold is reached. Once 36 write cycles have been executed.

Note that write cycles can be performed to the XMTFIFO even if the $\overline{\text{TDTREQ}}$ is inactive. When $\overline{\text{TDTREQ}}$ is asserted, it guarantees that a minimum amount of space exists, when $\overline{\text{TDTREQ}}$ is deasserted, it does not necessarily indicate that there is no space in the XMTFIFO. The $\overline{\text{DTV}}$ pin will indicate the successful acceptance of data by the Transmit FIFO.

As another example, assume again that the XMTFW is programmed for 32 write cycles. If the host writes word wide data continuously to the XMTFIFO, the $\overline{\text{TDTREQ}}$ will deassert when 36 writes have executed on the XMTFIFO, at which point 72-bytes will have been written to the XMTFIFO, the 64-byte XMTSP will have been exceeded and the transmission of preamble will have commenced. $\overline{\text{TDTREQ}}$ will not re-assert until the transmission of the packet data has commenced and the possibility of losing data due to a collision within the *slot time* is removed (512 bits have been transmitted without a collision indication). Assuming that the host actually stopped writing data after the initial 72-bytes, there will be only 16-bytes of data remaining in the XMTFIFO (8-bytes of preamble/SFD plus 56-bytes of data have been transmitted), corresponding to 12.8 µs of latency before an XMTFIFO underrun occurs. This latency is considerably less than the maximum possible 57.6 µs the system may have assumed. If the host had continued with the block transfer until 64 write cycles had been performed, 128-bytes would have been written to the XMTFIFO, and 72-bytes of latency would remain (57.6 µs) when $\overline{\text{TDTREQ}}$ was re-asserted.

Transmit FIFO—Burst Operation:

The XMTFIFO burst mode, programmed by the XMTBRST bit in the FIFO Configuration Control register, modifies $\overline{\text{TDTREQ}}$ behavior. The assertion of $\overline{\text{TDTREQ}}$ is controlled by the programming of the XMTFW bits, such that when the specified number of write cycles can be guaranteed (8, 16 or 32), $\overline{\text{TDTREQ}}$ will be asserted. $\overline{\text{TDTREQ}}$ will be de-asserted when the XMTFIFO can only accept a single write cycle (one word write including an *End Of Frame* delimiter) allowing the external device to burst data into the XMTFIFO when $\overline{\text{TDTREQ}}$ is asserted, and stop when $\overline{\text{TDTREQ}}$ is deasserted.

Receive FIFO—General Operation:

The Receive FIFO contains additional logic to ensure that sufficient data is present in the RCVFIFO to allow the specified number of bytes to be read, regardless of the ordering of byte/word read accesses. This has an impact on the perceived latency that the Receive FIFO provides to the host system. The description and table below outline the point at which $\overline{\text{RDTREQ}}$ will be asserted when the first duration of the packet has been received and when any subsequent transfer of the packet to the host system is required.

No preamble/SFD bytes are loaded into the Receive FIFO. All references to bytes pass through the receive FIFO. These references are received after the preamble/SFD sequence.

The first assertion of $\overline{\text{RDTREQ}}$ for a packet will occur after the longer of the following two conditions is met:

- 64-bytes have been received (to assure runt packets and packets experiencing collision within the slot time will be rejected).
- The RCVFW threshold is reached plus an additional 12 bytes. The additional 12 bytes are necessary to ensure that any permutation of byte/word read access is guaranteed. They are required for all threshold values, but in the case of the 16 and 32-byte thresholds, the requirement that the slot time criteria is met dominates. Any subsequent assertion of $\overline{\text{RDTREQ}}$ necessary to complete the transfer of the packet will occur after the RCVFW threshold is reached plus an additional 12 bytes. The table below also outlines the latency provided by the MACE device when the $\overline{\text{RDTREQ}}$ is asserted.

Receive FIFO Watermarks, $\overline{\text{RDTREQ}}$ Assertion and Latency

RCVFW [1–0]	Bytes Required for First Assertion of $\overline{\text{RDTREQ}}$	Bytes of Latency After First Assertion of $\overline{\text{RDTREQ}}$	Bytes Required for Subsequent Assertion of $\overline{\text{RDTREQ}}$	Bytes of Latency After Subsequent Assertion of $\overline{\text{RDTREQ}}$
00	64	64	28	100
01	64	64	44	84
10	76	52	76	52
11	XX	XX	XX	XX

Receive FIFO—Burst Operation:

The RCVFIFO also provides a burst mode capability, programmed by the RCVBRST bit in the FIFO Configuration Control register, to modify the operation of $\overline{\text{RDTREQ}}$.The assertion of $\overline{\text{RDTREQ}}$ will occur according to the programming of the RCVFW bits. $\overline{\text{RDTREQ}}$ will be de-asserted when the RCVFIFO can only provide a single read cycle (one word read). This allows the external device to *burst* data from the RCVFIFO once $\overline{\text{RDTREQ}}$ is asserted, and stop when $\overline{\text{RDTREQ}}$ is deasserted.

Receive FIFO—Low Latency Receive Operation:

The LOW Latency Receive mode can be programmed using the Low Latency Receive bit (LLRCV in the Receive Frame Control register). This effectively causes the assertion of $\overline{\text{RDTREQ}}$ to be directly coupled to the low watermark of 12 bytes in the RCVFIFO. Once the 12-byte threshold is reached (plus some internal synchronization delay of less than 1 byte), $\overline{\text{RDTREQ}}$ will be asserted, and will remain active until the RCVFIFO can support only one read cycle (one word of data), as in the burst operation described earlier.

The intended use for the Low Latency Receive mode is to allow fast forwarding of a received packet in a bridge application. In this case, the receiving process is made aware of the receive packet after only 9.6 µs, instead of waiting up to 60.8 µs (76-bytes) necessary for the initial assertion of $\overline{\text{RDTREQ}}$. An Ethernet-to-Ethernet bridge employing the MACE device (on all the Ethernet connections) with the XMTSP of all MACE controller XMTFIFOs set to the minimum (4-bytes), forwarding of a receive packet can be achieved within a sub 20 µs delay including processing overhead.

Note however that this mode places significant burden on the host processor. The receiving MACE device will no longer delete runt packets. A runt packet will have the Receive Frame Status appended to the receive data which the host must read as normal. The MACE device will not attempt to delete runt packets from the RCVFIFO in the Low Latency Receive mode. Collision fragments will also be passed to the host if they are detected after the 12-byte threshold has been reached. If a collision occurs, the Receive Frame Status (RCVFS) will be appended to the data successfully received in the RCVFIFO up to the point the collision was detected. No additional receive data will be written to the RCVFIFO. Note that the RCVFS will not become available until after the receive activity ceases. The collision indication (CLSN) in the Receive Status (RCVSTS) will be set, and the Receive Message Byte Count (RCVCNT) will be the correct count of the total duration of activity, including the period that collision was detected. The detection of normal (slot time) collisions versus late collisions can only be made by counting the number of bytes that were successfully received prior to the termination of the packet data.

In all cases where the reception ends prematurely (runt or collision), the data that was successfully received prior to the termination of reception must be read from the RCVFIFO before the RCVFS bytes are available.

Media Access Control (MAC)

The Media Access Control engine is the heart of the MACE device, incorporating the essential protocol requirements for operation of a compliant Ethernet/802.3 node, and providing the interface between the FIFO

sub-system and the Manchester Encoder/Decoder (MENDEC).

The MAC engine is fully compliant to Section 4 of ISO/IEC 8802-3 (ANSI/IEEE Standard 1990 Second edition) and ANSI/IEEE 802.3 (1985).

The MAC engine provides enhanced features, programmed through the Transmit Frame Control and Receive Frame Control registers, designed to minimize host supervision and pre or post message processing. These features include the ability to disable retries after a collision, dynamic FCS generation on a packet-by-packet basis, and automatic pad field insertion and deletion to enforce minimum frame size attributes.

The two primary attributes of the MAC engine are:

- Transmit and receive message data encapsulation
  - Framing (frame boundary delimitation, frame synchronization)
  - Addressing (source and destination address handling)
  - Error detection (physical medium transmission errors)
- Media access management
  - Medium allocation (collision avoidance)
  - Contention resolution (collision handling)

Transmit and Receive Message Data Encapsulation

Data passed to the MACE device Transmit FIFO will be assumed to be correctly formatted for transmission over the network as a valid packet. The user is required to pass the data stream for transmission to the MACE chip in the correct order, according to the byte ordering convention programmed for the BIU.

The MACE device provides minimum frame size enforcement for transmit and receive packets. When APAD XMT = 1 (default), transmit messages will be padded with sufficient bytes (containing 00h) to ensure that the receiving station will observe an information field (destination address, source address, length/type, data and FCS) of 64-bytes. When ASTRP RCV = 1 (default), the receiver will automatically strip pad and FCS bytes from the received message if the value in the length field is below the minimum data size (46-bytes). Both features can be independently over-ridden to allow illegally short (less than 64-bytes of packet data) messages to be transmitted and/or received.

Framing (Frame Boundary Delimitation, Frame Synchronization)

The MACE device will autonomously handle the construction of the transmit frame. When the Transmit FIFO has been filled to the predetermined threshold (set by XMTSP), and providing access to the channel is currently permitted, the MACE device will commence the 7 byte preamble sequence (10101010b, where first bit transmitted is a 1). The MACE device will subsequently append the Start Frame Delimiter (SFD) byte (10101011) followed by the serialized data from the Transmit FIFO. Once the data has been completed, the MACE device will append the FCS (most significant bit first) computed on the entire data portion of the message.

Note that the user is responsible for the correct ordering and content in each of the fields in the frame, including the destination address, source address, length/type and packet data.

The receive section of the MACE device will detect an incoming preamble sequence and lock to the encoded clock. The internal MENDEC will decode the serial bit stream and present this to the MAC engine. The MAC will discard the first 8-bits of information before searching for the SFD sequence. Once the SFD is detected, all subsequent bits are treated as part of the frame. The MACE device will inspect the length field to ensure minimum frame size, strip unnecessary pad characters (if enabled), and pass the remaining bytes through the Receive FIFO to the host. If pad stripping is performed, the MACE device will also strip the received FCS bytes, although the normal FCS computation and checking will occur. Note that apart from pad stripping, the frame will be passed unmodified to the host. If the length field has a value of 46 or greater, the MACE device will not attempt to validate the length against the number of bytes contained in the message.

If the frame terminates or suffers a collision before 64-bytes of information (after SFD) have been received, the MACE device will automatically delete the frame from the Receive FIFO, without host intervention. Note however, that if the Low Latency Receive option has been enabled (LLRCV = 1 in the Receive Frame Control register), the MACE device will not delete receive frames which experience a collision once the 12-byte low watermark has been reached (see the FIFO Sub-System section for additional details).

Addressing (Source and Destination Address Handling)

The first 6-bytes of information after SFD will be interpreted as the destination address field. The MACE device provides facilities for physical, logical and broadcast address reception. In addition, multiple physical addresses can be constructed (perfect address filtering) using external logic in conjunction with the EADI interface.

Error Detection (Physical Medium Transmission Errors)

The MACE device provides several facilities which report and recover from errors on the medium. In addition, the network is protected from gross errors due to

inability of the host to keep pace with the MACE device activity.

On completion of transmission, the MACE device will report the Transmit Frame Status for the frame. The exact number of transmission retry attempts is reported (ONE, MORE used with XMTRC, or RTRY), and whether the MACE device had to Defer (DEFER) due to channel activity. In addition, Loss of Carrier is reported, indicting that there was an interruption in the ability of the MACE device to monitor its own transmission. Repeated LCAR errors indicate a potentially faulty transceiver or network connection. Excessive Defer (EXDEF) will be reported in the Transmit Retry Count register if the transmit frame had to wait for an abnormally long period before transmission.

Additional transmit error conditions are reported through the Interrupt Register.

The Late Collision (LCOL) error indicates that the transmission suffered a collision after the slot time. This is indicative of a badly configured network. Late collisions should not occur in normal operating network.

The Collision Error (CERR) indicates that the transceiver did not respond with an SQE Test message within the predetermined time after a transmission completed. This may be due to a failed transceiver, disconnected or faulty transceiver drop cable, or the fact the transceiver does not support this feature (or it is disabled).

In addition to the reporting of network errors, the MACE device will also attempt to prevent the creation of any network error caused by inability of the host to service the MACE device. During transmission, if the host fails to keep the Transmit FIFO filled sufficiently, causing an underflow, the MACE device will guarantee the message is either sent as a runt packet (which will be deleted by the receiving station) or has an invalid FCS (which will also allow the receiving station to reject the message).

The status of each receive message is passed via the Receive Frame Status bytes. FCS and Framing errors (FRAM) are reported, although the received frame is still passed to the host. The FRAM error will only be reported if an FCS error is detected and there are a non integral number of bytes in the message. The MACE device will ignore up to seven additional bits at the end of a message (dribbling bits), which can occur under normal network operating conditions. The reception of eight additional bits will cause the MACE device to de-serialize the entire byte, and will result in the received message and FCS being modified.

Received messages which suffer a collision after 64-byte times (after SFD) will be marked to indicate they have suffered a late collision (CLSN). Additional counters are provided to report the Receive Collision Count and Runt Packet Count to be used for network statistics and utilization calculations.

Note that if the MACE device detects a received packet which has a 00b pattern in the preamble (after the first 8-bits which are ignored), the entire packet will be ignored. The MACE device will wait for the network to go inactive before attempting to receive additional frames.

Media Access Management

The basic requirement for all stations on the network is to provide fairness of channel allocation. The 802.3/Ethernet protocols define a media access mechanism which permits all stations to access the channel with equality. Any node can attempt to contend for the channel by waiting for a predetermined time (Inter Packet Gap interval) after the last activity, before transmitting on the media. The channel is a bus or multidrop communications medium (with various topological configurations permitted) which allows a single station to transmit and all other stations to receive. If two nodes simultaneously contend for the channel, their signals will interact causing loss of data, defined as a collision. It is the responsibility of the MAC to attempt to avoid and recover from a collision, to guarantee data integrity for the end-to-end transmission to the receiving station.

Medium Allocation (Collision Avoidance)

The IEEE 802.3 Standard (ISO/IEC 8802-3 1990) requires that the CSMA/CD MAC monitors the medium for traffic by watching for carrier activity. When carrier is detected, the media is considered busy, and the MAC should defer to the existing message.

The IEEE 802.3 Standard also allows optional two part deferral after a receive message.

> *See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.1:*
>
> *"NOTE : It is possible for the PLS carrier sense indication to fail to be asserted during a collision on the media. If the deference process simply times the interFrame gap based on this indication it is possible for a short interFrame gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness the following optional measures, as specified in 4.2.8, are recommended when interFrameSpacingPart1 is other than zero:"*

(1) Upon completing a transmission, start timing the interpacket gap, as soon as transmitting and carrierSense are both false.

(2) When timing an interFrame gap following reception, reset the interFrame gap timing if carrierSense becomes true during the first 2/3 of the interFrame gap timing interval. During the final 1/3 of the interval the timer shall not be reset to ensure fair access to the me-

dium. An initial period shorter than 2/3 of the interval is permissible including zero."

The MAC engine implements the optional receive two part deferral algorithm, with a first part inter-frame-spacing time of 6.0 µs. The second part of the inter-frame-spacing interval is therefore 3.6 µs.

The MACE device will perform the two part deferral algorithm as specified in Section 4.2.8 (Process Deference). The Inter Packet Gap (IPG) timer will start timing the 9.6 µs InterFrameSpacing after the receive carrier is de-asserted. During the first part deferral (InterFrameSpacingPart1–IFS1) the MACE device will defer any pending transmit frame and respond to the receive message. The IPG counter will be reset to zero continuously until the carrier deasserts, at which point the IPG counter will resume the 9.6 µs count once again. Once the IFS1 period of 6.0µs has elapsed, the MACE device will begin timing the second part deferral (InterFrameSpacingPart2–IFS2) of 3.6 µs. Once IFS1 has completed, and IFS2 has commenced, the MACE chip will not defer to a receive packet if a transmit packet is pending. This means that the MACE device will not attempt to receive an incoming packet, and it will start to transmit at 9.6 µs regardless of network activity, forcing a collision if an existing transmission is in progress. The MACE device will guarantee to complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

In addition to the deferral after receive process, the MACE device also allows transmit two part deferral to be implemented as an option. The option can be disabled using the DXMT2PD bit in the MAC Configuration Control register. Two part deferral after transmission is useful for ensuring that severe IPG shrinkage cannot occur in specific circumstances, causing a transmit message to follow a receive message so closely, as to make them indistinguishable.

During the time period immediately after a transmission has been completed, the external transceiver (in the case of a standard AUI connected device), should generate the SQE Test message (a nominal 10 MHz burst of 5-15 BT duration) on the CI± pair (within 0.6–1.6 µs after the transmission ceases). During the time period in which the SQE Test message is expected the MACE device will not respond to receive carrier sense.

> *See ANSI/IEEE Std 802.3-1990 Edition, 7.2.4.6 (1)):*
>
> "At the conclusion of the output function, the DTE opens a time window during which it expects to see the *signal_quality_error* signal asserted on the Control In circuit. The time window begins when the CARRIER_STATUS becomes CARRIER_OFF. If execution of the output function does not cause CARRIER_ON to occur, no SQE test occurs in the DTE. The duration of the window shall be at least 4.0 µs but no more than 8.0 µs. During the time window the Carrier Sense Function is inhibited."

The MACE device implements a carrier sense *blinding* period within 0 µs–4.0 µs from deassertion of carrier sense after transmission. This effectively means that when transmit two part deferral is enabled (DXMT2PD in the MAC Configuration Control register is cleared) the IFS1 time is from 4 µs to 6 µs after a transmission. However, since IPG shrinkage below 4 µs will not be encountered on correctly configured networks, and since the fragment size will be larger than the 4 µs blinding window, then the IPG counter will be reset by a worst case IPG shrinkage/fragment scenario and the MACE device will defer its transmission. The MACE chip will not restart the carrier sense *blinding* period if carrier is detected within the 4.0–6.0 µs portion of IFS1, but will restart timing of the entire IFS1 period.

Contention Resolution (Collision Handling)

Collision detection is performed and reported to the MAC engine either by the integrated Manchester Encoder/Decoder (MENDEC), or by use of an external function (e.g. Serial Interface Adaptor, Am7992B) utilizing the GPSI.

If a collision is detected before the complete preamble/SFD sequence has been transmitted, the MACE device will complete the preamble/SFD before appending the jam sequence. If a collision is detected after the preamble/SFD has been completed, but prior to 512 bits being transmitted, the MACE device will abort the transmission, and append the jam sequence immediately. The jam sequence is a 32-bit all zeroes pattern.

The MACE device will attempt to transmit a frame a total of 16 times (initial attempt plus 15 retries) due to normal collisions (those within the slot time). Detection of collision will cause the transmission to be re-scheduled, dependent on the backoff time that the MACE device computes. Each collision which occurs during the transmission process will cause the value of XMTRC in the Transmit Retry Count register to be updated. If a single retry was required, the ONE bit will be set in the Transmit Frame Status. If more than one retry was required, the MORE bit will be set, and the exact number of attempts can be determined (XMTRC+1). If all 16 attempts experienced collisions, the RTRY bit will be set (ONE and MORE will be clear), and the transmit message will be flushed from the XMTFIFO, either by resetting the XMTFIFO (if no *End-of-Frame* tag exists) or by moving the XMTFIFO read pointer to the next free location (If an End-of-Frame tag is present). If retries have been disabled by setting the DRTRY bit, the MACE device will abandon transmission of the frame on detection of the first collision. In this case, only the RTRY bit

will be set and the transmit message will be flushed from the XMTFIFO. The RTRY condition will cause the de-assertion of $\overline{\text{TDTREQ}}$, and the assertion of the $\overline{\text{INTR}}$ pin, providing the XMTINTM bit is cleared.

If a collision is detected after 512 bit times have been transmitted, the collision is termed a late collision. The MACE device will abort the transmission, append the jam sequence and set the LCOL bit in the Transmit Frame Status. No retry attempt will be scheduled on detection of a late collision, and the XMTFIFO will be flushed. The late collision condition will cause the de-assertion of $\overline{\text{TDTREQ}}$, and the assertion of the $\overline{\text{INTR}}$ pin, providing the XMTINTM bit is cleared.

The IEEE 802.3 Standard requires use of a *truncated binary exponential backoff* algorithm which provides a controlled pseudo random mechanism to enforce the collision backoff interval, before re-transmission is attempted.

See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.5:

"At the end of enforcing a collision (jamming), the CSMA/CD sublayer delays before attempting to re-transmit the frame. The delay is an integer multiple of slotTime. The number of slot times to delay before the nth re-transmission attempt is chosen as a uniformly distributed random integer r in the range:

$0 \le r \le 2^k$, where k = min (n,10)."

The MACE device implements a random number generator, configured to ensure that nodes experiencing a collision, will not have their retry intervals track identically, causing retry errors.

The MACE device provides an alternative algorithm, which suspends the counting of the slot time/IPG during the time that receive carrier sense is detected. This aids in networks where large numbers of nodes are present, and numerous nodes can be in collision. It effectively accelerates the increase in the backoff time in busy networks, and allows nodes not involved in the collision to access the channel whilst the colliding nodes await a reduction in channel activity. Once channel activity is reduced, the nodes resolving the collision time-out their slot time counters as normal.

If a receive message suffers a collision, it will be either a runt, in which case it will be deleted in the Receive FIFO, or it will be marked as a receive late collision, using the CLSN bit in the Receive Frame Status register. All frames which suffer a collision within the slot time will be deleted in the Receive FIFO without requesting host intervention, providing that the LLRCV bit (Receive Frame Control) is not set. Runt packets which suffer a collision will be aborted regardless of the state of the RPA bit (User Test Register). If the collision commences after the slot time, the MACE device receiver will stop sending collided packet data to the Receive FIFO and the packet data read by the system will contain the amount of data received to the point of collision; the CLSN bit in the Receive Frame Status register will indicate the receive late collision. Note that the Receive Message Byte Count will report the total number of bytes during the receive activity, including the collision.

In all normal receive collision cases, the MACE device eliminates the transfer of packet data across the host bus. In a receive late collision condition, the MACE chip minimizes the amount transferred. These functions preserve bus bandwidth utilization.

Manchester Encoder/Decoder (MENDEC)

The integrated Manchester Encoder/Decoder provides the PLS (Physical Signaling) functions required for a fully compliant IEEE 802.3 station. The MENDEC block contains the AUI, DAI interfaces, and supports the 10BASE-T interface; all of which transfer data to appropriate transceiver devices in Manchester encoded format. The MENDEC provides the encoding function for data to be transmitted on the network using the high accuracy on-board oscillator, driven by either the crystal oscillator or an external CMOS level compatible clock generator. The MENDEC also provides the decoding function from data received from the network. The MENDEC contains a Power On Reset (POR) circuit, which ensures that all analog portions of the MACE device are forced into their correct state during power up, and prevents erroneous data transmission and/or reception during this time.

External Crystal Characteristics

When using a crystal to drive the oscillator, the following crystal specification should be used to ensure less than ±0.5 ns jitter at DO±:

Parameter	Min	Nom	Max	Units
1. Parallel Resonant Frequency		20		MHz
2. Resonant Frequency Error (CL = 20 pF)	–50		+50	PPM
3. Change in Resonant Frequency With Respect To Temperature (CL = 20 pF)*	–40		+40	PPM
4. Crystal Capacitance			20	pF
5. Motional Crystal Capacitance (C1)		0.022		pF
6. Series Resistance			35	ohm
7. Shunt Capacitance			7	pF

** Requires trimming crystal spec; no trim is 50 ppm total*

External Clock Drive Characteristics

When driving the oscillator from an external clock source, XTAL2 must be left floating (unconnected). An external clock having the following characteristics must be used to ensure less than ±0.5 ns jitter at DO±.

Clock Frequency:	20 MHz ±0.01%
Rise/Fall Time (tR/tF):	< 6 ns from 0.5 V to V_{DD}–0.5
XTAL1 HIGH/LOW Time (tHIGH/tLOW):	40 – 60% duty cycle
XTAL1 Falling Edge to Falling Edge Jitter:	< ±0.2 ns at 2.5 V input (V_{DD}/2)

MENDEC Transmit Path

The transmit section encodes separate clock and NRZ data input signals into a standard Manchester encoded serial bit stream. The transmit outputs (DO±) are designed to operate into terminated transmission lines. When operating into a 78 ohm terminated transmission line, signaling meets the required output levels and skew for Cheapernet, Ethernet and IEEE-802.3.

Transmitter Timing and Operation

A 20 MHz fundamental mode crystal oscillator provides the basic timing reference for the SIA portion of the MACE device. It is divided by two, to create the internal transmit clock reference. Both clocks are fed into the SIA's Manchester Encoder to generate the transitions in the encoded data stream. The internal transmit clock is used by the SIA to internally synchronize the Internal Transmit Data (ITXD) from the controller and Internal Transmit Enable (ITENA). The internal transmit clock is also used as a stable bit rate clock by the receive section of the SIA and controller.

The oscillator requires an external 0.005% crystal, or an external 0.01% CMOS-level input as a reference. The accuracy requirements if an external crystal is used are tighter because allowance for the on-chip oscillator must be made to deliver a final accuracy of 0.01%.

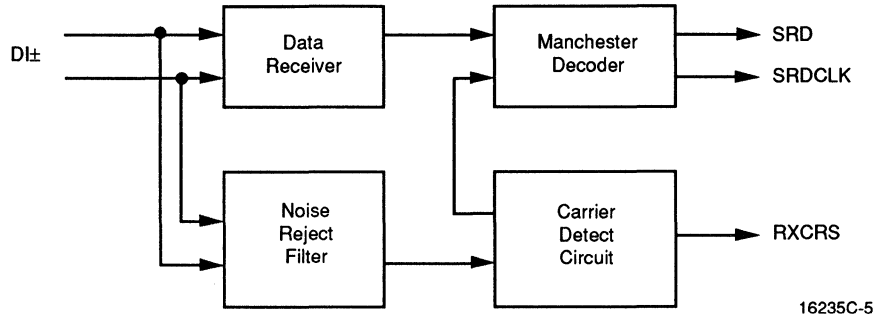
Transmission is enabled by the controller. As long as the ITENA request remains active, the serial output of the controller will be Manchester encoded and appear at DO±. When the internal request is dropped by the controller, the differential transmit outputs go to one of two idle states, dependent on TSEL in the Mode Register (CSR15, bit 9):

TSEL LOW:	The idle state of DO± yields "zero" differential to operate transformer-coupled loads.
TSEL HIGH:	In this idle state, DO+ is positive with respect to DO– (logical\HIGH).

Receive Path

The principal functions of the Receiver are to signal the MACE device that there is information on the receive pair, and separate the incoming Manchester encoded data stream into clock and NRZ data.

The Receiver section (see Receiver Block Diagram) consists of two parallel paths. The receive data path is a zero threshold, wide bandwidth line receiver. The carrier path is an offset threshold bandpass detecting line receiver. Both receivers share common bias networks to allow operation over a wide input common mode range.



Receiver Block Diagram

Input Signal Conditioning

Transient noise pulses at the input data stream are rejected by the Noise Rejection Filter. Pulse width rejection is proportional to transmit data rate. DC inputs more negative than minus 100 mV are also suppressed.

The Carrier Detection circuitry detects the presence of an incoming data packet by discerning and rejecting noise from expected Manchester data, and controls the stop and start of the phase-lock loop during clock acquisition. Clock acquisition requires a valid Manchester bit pattern of 1010 to lock onto the incoming message.

When input amplitude and pulse width conditions are met at DI±, the internal enable signal from the SIA to controller (RXCRS) is asserted and a clock acquisition cycle is initiated.

Clock Acquisition

When there is no activity at DI± (receiver is idle), the receive oscillator is phase locked to TCK. The first negative clock transition (bit cell center of first valid Manchester "0") after RXCRS is asserted interrupts the receive oscillator. The oscillator is then restarted at the second Manchester "0" (bit time 4) and is phase locked to it. As a result, the SIA acquires the clock from the incoming Manchester bit pattern in 4 bit times with a "1010" Manchester bit pattern.

SRDCLK and SRD are enabled 1/4 bit time after clock acquisition in bit cell 5 if the ENPLSIO bit is set in the PLS configuration control register. SRD is at a HIGH state when the receiver is idle (no SRDCLK). SRD however, is undefined when clock is acquired and may remain HIGH or change to LOW state whenever SRDCLK is enabled. At 1/4 bit time through bit cell 5, the controller portion of the MACE device sees the first SRDCLK transition. This also strobes in the incoming fifth bit to the SIA as Manchester "1". SRD may make a transition after the SRDCLK rising edge bit cell 5, but its state is still undefined. The Manchester "1" at bit 5 is clocked to SRD output at 1/4 bit time in bit cell 6.

PLL Tracking

After clock acquisition, the phase-locked clock is compared to the incoming transition at the bit cell center (BCC) and the resulting phase error is applied to a correction circuit. This circuit ensures that the phase-locked clock remains locked on the received signal. Individual bit cell phase corrections of the Voltage Controlled Oscillator (VCO) are limited to 10% of the phase difference between BCC and phase-locked clock.

Carrier Tracking and End of Message

The carrier detection circuit monitors the DI± inputs after RXCRS is asserted for an end of message. RXCRS deasserts 1 to 2 bit times after the last positive transition on the incoming message. This initiates the end of reception cycle. The time delay from the last rising edge of the message to RXCRS deassert allows the last bit to be strobed by SRDCLK and transferred to the controller section, but prevents any extra bit(s) at the end of message. When IRENA de-asserts (see Receive Timing-End of Reception (Last Bit = 0) and Receive Timing-End of Reception (Last Bit = 1) waveform diagrams) an RXCRS hold off timer inhibits RXCRS assertion for at least 2 bit times.

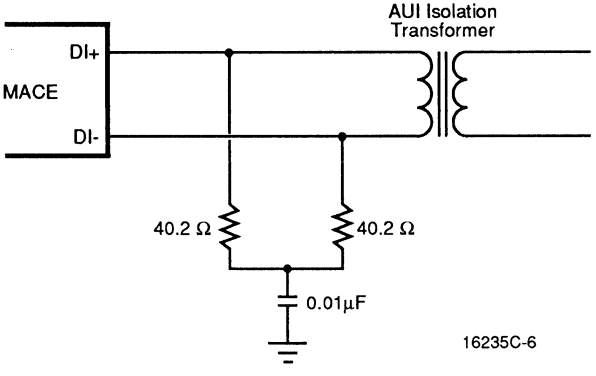
Data Decoding

The data receiver is a comparator with clocked output to minimize noise sensitivity to the DI± inputs. Input error is less than ± 35 mV to minimize sensitivity to input rise and fall time. SRDCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit, and clocks the data out on SRD on the following SRDCLK. The data receiver also generates the signal used for phase detector comparison to the internal SIA voltage controlled oscillator (VCO).

Differential Input Terminations

The differential input for the Manchester data (DI±) is externally terminated by two 40.2 ohm ±1% resistors and one optional common-mode bypass capacitor, as shown in the Differential Input Termination diagram

below. The differential input impedance, ZIDF, and the common-mode input impedance, ZICM, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. If SIP devices are used, 39 ohms is also a suitable value. The CI± differential inputs are terminated in exactly the same way as the DI± pair.



Differential Input Termination

Collision Detection

A transceiver detects the collision condition on the network and generates a differential signal at the CI± inputs. This collision signal passes through an input stage which detects signal levels and pulse duration. When the signal is detected by the MENDEC it sets the CLSN line HIGH. The condition continues for approximately 1.5 bit times after the last LOW-to-HIGH transition on CI±.

Jitter Tolerance Definition

The Receive Timing-Start of Reception Clock Acquisition waveform diagram shows the internal timing relationships implemented for decoding Manchester data in the SIA module. The SIA utilizes a clock capture circuit to align its internal data strobe with an incoming bit stream. The clock acquisition circuitry requires four valid bits with the values 1010. Clock is phase locked to the negative transition at the bit cell center of the second "0" in the pattern.

Since data is strobed at 1/4 bit time, Manchester transitions which shift from their nominal placement through 1/4 bit time will result in improperly decoded data. With this as the criteria for an error, a definition of "Jitter Handling" is:

The peak deviation approaching or crossing 1/4 bit cell position from nominal input transition, for which the SIA section will properly decode data.

Attachment Unit Interface (AUI)

The AUI is the PLS (Physical Signaling) to PMA (Physical Medium Attachment) interface which effectively connects the DTE to the MAU. The differential interface provided by the MACE device is fully compliant to Section 7 of ISO 8802-3 (ANSI/IEEE 802.3).

After the MACE device initiates a transmission it will expect to see data *looped-back* on the DI± pair (AUI port selected). This will internally generate a *carrier sense*, indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This *carrier sense* signal must be asserted during the transmission when using the AUI port (DO± transmitting). If *carrier sense* does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in the Transmit Frame Status (bit 7) after the packet has been transmitted.

Digital Attachment Interface (DAI)

The Digital Attachment Interface is a simplified electrical attachment specification which allows MAUs which do not require the DC isolation between the MAU and DTE (e.g. devices compatible with the 10BASE-T Standard and 10BASE-FL Draft document) to be implemented. All data transferred across the DAI port is Manchester Encoded. Decoding and encoding is performed by the MENDEC.

The DAI port will accept receive data on the basis that the RXCRS input is active, and will take the data presented on the RXDAT input as valid Manchester data. Transmit data is sent to the external transceiver by the MACE device asserting $\overline{\text{TXEN}}$ and presenting complimentary data on the TXDAT± pair. During idle, the MACE device will assert the TXDAT+ line high, and the TXDAT line low, while $\overline{\text{TXEN}}$ is maintained inactive (high). The MACE device implements logical collision detection and will use the simultaneous assertion of $\overline{\text{TXEN}}$ and RXCRS to internally detect a collision condition, take appropriate internal action (such as abort the current transmit or receive activity), and provide external indication using the CLSN pin. Any external

transceiver utilized for the DAI interface must not loop back the transmit data (presented by the MACE device) on the TXDAT± pins to the RXDAT pin. Neither should the transceiver assert the RXCRS pin when transmitting data to the network. Duplication of these functions by the external transceiver (unless the MACE device is in the external loop back test configuration) will cause false collision indications to be detected.

In order to provide an integrity test of the connectivity between the MACE device and the external transceiver similar to the SQE Test Message provided as a part of the AUI functionality, the MACE device can be programmed to operate the DAI port in an external loopback test. In this case, the external transceiver is assumed to loopback the TXDAT± data stream to the RXDAT pin, and assert RXCRS in response to the $\overline{\text{TXEN}}$ request. When in the external loopback mode of operation (programmed by LOOP [1–0] = 01), the MACE device will not internally detect a collision condition. The external transceiver is assumed to take action to ensure that this test will not disrupt the network. This type of test is intended to be operated for a very limited period (e.g. after power up), since the transceiver is assumed to be located physically close to the MACE device and with minimal risk of disconnection (e.g. connected via printed circuit board traces).

Note that when the DAI port is selected, LCAR errors will not occur, since the MACE device will internally loop back the transmit data path to the receiver. This loop back function must not be duplicated by a transceiver which is externally connected via the DAI port, since this will result in a condition where a collision is generated during any transmit activity.

The transmit function of the DAI port is protected by a jabber mechanism which will be invoked if the TXDAT± and $\overline{\text{TXEN}}$ circuit is active for an excessive period (20 – 150 ms). This prevents a single node from disrupting the network due to a *stuck-on* or faulty transmitter. If this maximum transmit time is exceeded, the DAI port transmitter circuitry is disabled, the CLSN pin is asserted, the Jabber bit (JAB in the Interrupt Register) is set and the $\overline{\text{INTR}}$ pin will be asserted providing the JABM bit (Interrupt Mask Register) is cleared. Once the internal transmit data stream from the MENDEC stops ($\overline{\text{TXEN}}$ deasserts), an *unjab* time of 250 ms–750 ms will elapse before the MACE device deasserts the CLSN indication and re-enables the transmit circuitry.

When jabber is detected, the MACE device will assert the CLSN pin, de-assert the $\overline{\text{TXEN}}$ pin (regardless of internal MENDEC activity) and set the TXDAT+ and TXDAT pins to their inactive state.

10BASE-T Interface

Twisted Pair Transmit Function

Data transmission over the 10BASE-T medium requires use of the integrated 10BASE-T MAU, and uses the differential driver circuitry in the TXD± and TXP± pins. The driver circuitry provides the necessary electrical driving capability and the pre-distortion control for transmitting signals over maximum length Twisted Pair cable, as specified by the 10BASE-T supplement to the IEEE 802.3 Standard. The transmit function for data output meets the propagation delays and jitter specified by the standard. During normal transmission, and providing that the 10BASE-T MAU is not in a Link Fail or jabber state, the $\overline{\text{TXEN}}$ pin will be driven LOW and can be used indirectly to drive a status LED.

Twisted Pair Receive Function

The receiver complies with the receiver specifications of the IEEE 802.3 10BASE-T Standard, including noise immunity and received signal rejection criteria (*Smart Squelch*). Signals meeting this criteria appearing at the RXD± differential input pair are routed to the internal MENDEC. The receiver function meets the propagation delays and jitter requirements specified by the 10BASE-T Standard. The receiver squelch level drops to half its threshold value after unsquelch to allow reception of minimum amplitude signals and to mitigate carrier fade in the event of worst case signal attenuation and crosstalk noise conditions. During receive, the RXCRS pin is driven HIGH and can be used indirectly to drive a status LED.

Note that the 10BASE-T Standard defines the receive input amplitude at the external Media Dependent Interface (MDI). Filter and transformer loss are not specified. The 10BASE-T MAU receiver squelch levels are defined to account for a 1dB insertion loss at 10 MHz, which is typical for the type of receive filters/transformers recommended (see the Appendix for additional details).

Normal 10BASE-T compatible receive thresholds are employed when the LRT bit is inactive (PHY Configuration Control register). When the LRT bit is set, the Low Receive Threshold option is invoked, and the sensitivity of the 10BASE-T MAU receiver is increased. This allows longer line lengths to be employed, exceeding the 100m target distance of normal 10BASE-T (assuming typical 24 AWG cable). The additional cable distance attributes directly to increased signal attenuation and reduced signal amplitude at the 10BASE-T MAU receiver. However, from a system perspective, making the receiver more sensitive means that it is also more susceptible to

extraneous noise, primarily caused by coupling from co-resident services (crosstalk). For this reason, it is recommended that when using the Low Receive Threshold option that the service should be installed on 4-pair cable only. Multi-pair cables within the same outer sheath have lower crosstalk attenuation, and may allow noise emitted from adjacent pairs to couple into the receive pair, and be of sufficient amplitude to falsely unsquelch the 10BASE-T MAU receiver.

Link Test Function

The link test function is implemented as specified by 10BASE-T standard. During periods of transmit pair inactivity, Link Test pulses will be periodically sent over the twisted pair medium to constantly monitor medium integrity.

When the link test function is enabled, the absence of Link Test pulses and receive data on the RXD± pair will cause the 10BASE-T MAU to go into a Link Fail state. In the Link Fail state, data transmission, data reception, data loopback and the collision detection functions are disabled, and remain disabled until valid data or >5 consecutive link pulses appear on the RXD± pair. During Link Fail, the $\overline{\text{LNKST}}$ pin is inactive (externally pulled HIGH), and the Link Fail bit (LNKFL in the PHY Configuration Control register) will be set. When the link is identified as functional, the $\overline{\text{LNKST}}$ pin is driven LOW (capable of directly driving a Link OK LED using an integrated 12 mA driver) and the LNKFL bit will be cleared. In order to inter-operate with systems which do not implement link test, this function can be disabled by setting the the Disable Link Test bit (DLNKTST in the PHY Configuration Control register). With link test disabled, the data driver, receiver and loopback functions as well as collision detection remain enabled irrespective of the presence or absence of data or link pulses on the RXD± pair.

The MACE devices integrated 10BASE-T transceiver will mimic the performance of an externally connected device (such as a 10BASE-T MAU connected using an AUI). When the 10BASE-T transceiver is in link fail, the receive data path of the transceiver must be disabled. The MACE device will report a Loss of Carrier error (LCAR bit in the Transmit Frame Status register) due to the absence of the normal loopback path, for every packet transmitted during the link fail condition. In addition, a Collision Error (CERR bit in the Transmit Frame Status register) will also be reported (see the section on Signal Quality Error Test Function for additional details).

If the AWAKE bit is set in the PHY Configuration Control register prior to the assertion of the hardware $\overline{\text{SLEEP}}$ pin, the 10BASE-T receiver remains operable, and is able to detect and indicate (using the $\overline{\text{LNKST}}$ output) the presence of legitimate Link Test pulses or receive activity. The transmission of Link Test pulses is suspended to reduce power consumption.

If the RWAKE bit is set in the PHY Configuration Control register prior to the assertion of the hardware $\overline{\text{SLEEP}}$ pin, the 10BASE-T receiver and transmitter functions remain active, the $\overline{\text{LNKST}}$ output is disabled, and the EADI output pins are enabled. In addition the AUI port (transmit and receive) remains active. Note that since the MAC core will be in a sleep mode, no transmit activity is possible, and the transmission of Link Test pulses is also suspended to reduce power consumption.

Polarity Detection and Reversal

The Twisted Pair receive function includes the ability to invert the polarity of the signals appearing at the RXD± pair if the polarity of the received signal is reversed (such as in the case of a wiring error). This feature allows data packets received from a reverse wired RXD± input pair to be corrected in the 10BASE-T MAU prior to transfer to the MENDEC. The polarity detection function is activated following reset or Link Fail, and will reverse the receive polarity based on both the polarity of any previous Link Test pulses and the polarity of subsequent packets with a valid End Transmit Delimiter (ETD).

When in the Link Fail state, the internal 10BASE-T receiver will recognize Link Test pulses of either positive or negative polarity. Exit from the Link Fail state is made due to the reception of five to six consecutive Link Test pulses of identical polarity. On entry to the Link Pass state, the polarity of the last five Link Test pulses is used to determine the initial receive polarity configuration and the receiver is reconfigured to subsequently recognize only Link Test pulses of the previously recognized polarity. This link pulse algorithm is employed only until ETD polarity determination is made as described later in this section.

Positive Link Test pulses are defined as received signal with a positive amplitude greater than 520 mV (LRT = LOW) with a pulse width of 60 ns–200 ns. This positive excursion may be followed by a negative excursion. This definition is consistent with the expected received signal at a correctly wired receiver, when a Link Test pulse which fits the template of Figure 14-12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

Negative Link Test pulses are defined as received signals with a negative amplitude greater than 520 mV (LRT = LOW) with a pulse width of 60 ns–200 ns. This negative excursion may be followed by a positive excursion. This definition is consistent with the expected received signal at a reverse wired receiver, when a Link Test pulse which fits the template of Figure 14-12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

The polarity detection/correction algorithm will remain *armed* until two consecutive packets with valid ETD of identical polarity are detected. When armed, the

receiver is capable of changing the initial or previous polarity configuration based on the most recent ETD polarity.

On receipt of the first packet with valid ETD following reset or Link Fail, the MACE device will utilize the inferred polarity information to configure its RXD± input, regardless of its previous state. On receipt of a second packet with a valid ETD with correct polarity, the detection/correction algorithm will *lock-in* the received polarity. If the second (or subsequent) packet is not detected as confirming the previous polarity decision, the most recently detected ETD polarity will be used as the default. Note that packets with invalid ETD have no effect on updating the previous polarity decision. Once two consecutive packets with valid ETD have been received, the MACE device will disable the detection/correction algorithm until either a Link Fail condition occurs or a hardware or software reset occurs.

During polarity reversal, the $\overline{\text{RXPOL}}$ pin should be externally pulled HIGH and the Reversed Polarity bit (REVPOL in the PHY Configuration Control register) will be set. During normal polarity conditions, the $\overline{\text{RXPOL}}$ pin is driven LOW (capable of directly driving a Polarity OK LED using an integrated 12 mA driver) and the REVPOL bit will be cleared.

If desired, the polarity correction function can be disabled by setting the Disable Auto Polarity Correction bit (DAPC bit in the PHY Configuration Control register). However, the polarity detection portion of the algorithm continues to operate independently, and the $\overline{\text{RXPOL}}$ pin and the REVPOL bits will reflect the polarity state of the receiver.

Twisted Pair Interface Status

Three outputs (TXEN, RXCRS and CLSN) indicate whether the MACE device is transmitting (MENDEC to Twisted Pair), receiving (Twisted Pair to MENDEC), or in a collision state with both functions active simultaneously.

The MACE device will power up in the Link Fail state. The normal algorithm will apply to allow it to enter the Link Pass state. On power up, the TXEN, RXCRS and CLSN) pins will be in a high impedance state until they are enabled by setting the Enable PLS I/O bit (ENPLSIO in the PLS Configuration Control register) and the 10BASE-T port enters the Link Pass state.

In the Link Pass state, transmit or receive activity which passes the pulse width/amplitude requirements of the DO± or RXD± inputs, will be indicated by the TXEN or RXCRS pin respectively going active. TXEN, RXCRS and CLSN are all asserted during a collision.

In the Link Fail state, TXEN, RXCRS and CLSN are inactive.

In jabber detect mode, the MACE device will activate the CLSN pin, disable TXEN (regardless of Manchester data output from the MENDEC), and allow the RXCRS pin to indicate the current state of the RXD± pair. If there is no receive activity on RXD±, only CLSN will be active during jabber detect. If there is RXD± activity, both CLSN and RXCRS will be active.

If the $\overline{\text{SLEEP}}$ pin is asserted (regardless of the programming of the AWAKE or RWAKE bits in the PHY Configuration Control register), the TXEN, RXCRS and CLSN outputs will be placed in a high impedance state.

Collision Detect Function

Simultaneous activity (presence of valid data signals) from both the internal MENDEC transmit function (indicated externally by TXEN active) and the twisted pair RXD± pins constitutes a collision, thereby causing an external indication on the CLSN pin, and an internal indication which is returned to the MAC core. The TXEN, RXCRS and CLSN pins are driven high during collision.

Signal Quality Error (SQE) Test (Heartbeat) Function

The SQE Test message (a 10 MHz burst normally returned on the AUI CI± pair at the end of every transmission) is intended to be a self-test indication to the DTE that the MAU collision circuitry is functional and the AUI cable/connection is intact. This has minimal relevance when the 10BASE-T MAU is embedded in the LAN controller. A Collision Error (CERR bit in the Interrupt Register) will be reported only when the 10BASE-T port is in the link fail state, since the collision circuit of the MAU will be disabled, causing the absence of the SQE Test message. In GPSI mode the external encoder/decoder is responsible for asserting the CLSN pin after each transmission. In DAI mode SEQ Test has no relevance.

Jabber Function

The Jabber function inhibits the twisted pair transmit function of the MACE device if the TXD±/TXP± circuits are active for an excessive period (20–150 ms). This prevents any one node from disrupting the network due to a stuck-on or faulty transmitter. If this maximum transmit time is exceeded, the data path through the 10BASE-T transmitter circuitry is disabled (although Link Test pulses will continue to be sent), the CLSN pin is asserted, the Jabber bit (JAB in the Interrupt Register) is set and the $\overline{\text{INTR}}$ pin will be asserted providing the JABM bit (Interrupt Mask Register) is cl eared. Once the internal transmit data stream from the MENDEC stops (TXEN deasserts), an *unjab* time of 250–750 ms will elapse before the MACE device deasserts the CLSN indication and re-enables the transmit circuitry.

When jabber is detected, the MACE device will assert the CLSN pin, de-assert the TXEN pin (regardless of

internal MENDEC activity), and allow the RXCRS pin to indicate the current state of the RXD± pair. If there is no receive activity on RXD±, only CLSN will be active during jabber detect. If there is RXD± activity, both CLSN and RXCRS will be active.

External Address Detection Interface (EADI)

This interface is provided to allow external *perfect address filtering.* This feature is typically utilized for terminal server, bridge and/or router type products. The use of external logic is required, to capture the serial bit stream from the MACE device, and compare this with a table of stored addresses or identifiers. See the EADI port diagram in the Systems Applications section, Network Interfaces sub-section, for details.

The EADI interface operates directly from the NRZ decoded data and clock recovered by the Manchester decoder. This allows the external address detection to be performed in parallel with frame reception and address comparison in the MAC Station Address Detection (SAD) block.

SRDCLK is provided to allow clocking of the receive bit stream from the MACE device, into the external address detection logic. Once a received packet commences and data and clock are available from the decoder, the EADI interface logic will monitor the alternating (*1,0*) preamble pattern until the two ones of the Start Frame Delimiter (*1,0,1,0,1,0,1,1*) are detected, at which point the SF/BD output will be driven high.

After SF/BD is asserted the serial data from SRD should be de-serialized and sent to a Content Addressable Memory (CAM) or other address detection device.

To allow simple serial to parallel conversion, SF/BD is provided as a strobe and/or marker to indicate the delineation of bytes, subsequent to the SFD. This feature provides a mechanism to allow not only capture and/or decoding of the physical or logical (group) address, but also facilitates the capture of header information to determine protocol and or inter-networking information. The $\overline{\text{EAM/R}}$ pin is driven by the external address comparison logic, to either reject or accept the packet. Two alternative modes are permitted, allowing the external logic to either accept the packet based on address match, or reject the packet if there is no match. The two alternate methods are programmed using the Match/Reject (M/$\overline{\text{R}}$) bit in the Receive Frame Control register.

If the M/$\overline{\text{R}}$ bit is set, the pin is configured as $\overline{\text{EAM}}$ (External Address Match). The MACE device can be configured with Physical, Logical or Broadcast Address comparison operational. If an internal address match is detected, the packet will be accepted regardless of the condition of $\overline{\text{EAM}}$. Additional addresses can be located in the external address detection logic. If a match is detected, $\overline{\text{EAM}}$ must go active within 600 ns of the last bit in the destination address field (end of byte 6) being presented on the SRD output, to guarantee frame reception. In addition, $\overline{\text{EAM}}$ must go inactive after a match has been detected on a previous packet, before the next match can take place on any subsequent packet. $\overline{\text{EAM}}$ must be asserted for a minimum pulse width of 200 ns.

If the M/$\overline{\text{R}}$ bit is clear (default state after either the $\overline{\text{RESET}}$ pin or SWRST bit have been activated), the pin is configured as $\overline{\text{EAR}}$ (External Address Reject). The MACE device can be configured with Physical, Logical or Broadcast Address comparison operational. If an internal address match is detected, the packet will be accepted regardless of the condition of $\overline{\text{EAR}}$. Incoming packets which do not pass the internal address comparison will continue to be received by the MACE device. $\overline{\text{EAR}}$ must be externally presented to the MACE chip prior to the first assertion of $\overline{\text{RDTREQ}}$, to guarantee rejection of unwanted packets. This allows approximately 58 byte times after the last destination address bit is available to generate the $\overline{\text{EAR}}$ signal, assuming the MACE device is not configured to accept runt packets. $\overline{\text{EAR}}$ will be ignored by the MACE device from 64 byte times after the SFD, and the packet will be accepted if $\overline{\text{EAR}}$ has not been asserted before this time. If the MACE device is configured to accept runt packets, the $\overline{\text{EAR}}$ signal must be generated prior to the receive message completion, which could be as short as 12 byte times (assuming six bytes for source address, two bytes for length, no data, four bytes for FCS) after the last bit of the destination address is available. $\overline{\text{EAR}}$ must have a pulse width of at least 200 ns.

Note that setting the PROM bit (MAC Configuration Control) will cause all receive packets to be received, regardless of the programming of M/$\overline{\text{R}}$ or the state of the $\overline{\text{EAM/R}}$ input. The following table summarizes the operation of the EADI features.

Internal/External Address Recognition Capabilities

PROM	M/R	EAM/R	Required Timing	Received Messages
1	X	X	No timing requirements	All Received Frames
0	0	H	No timing requirements	All Received Frames
0	0	↓	Low for 200 ns within 512-bits after SFD	Physical/Logical/Broadcast Matches
0	1	H	No timing requirements	Physical/Logical/Broadcast Matches
0	1	↓	Low for 200 ns within 8-bits after DA field	All Received Frames

General Purpose Serial Interface (GPSI)

The GPSI port provides the signals necessary to present an interface consistent with the non encoded data functions observed to/from a LAN controller such as the Am7990 Local Area Network Controller for Ethernet (LANCE). The actual GPSI pins are functionally identical to some of the pins from the DAI and the EADI ports, the GPSI replicates this type of interface.

The GPSI allows use of an external Manchester encoder/decoder, such as the Am7992B Serial Interface Adapter (SIA). In addition, it allows the MACE device to be used as a MAC sublayer engine in a repeater based on the Am79C980 Integrated Multiport Repeater (IMR). Simple connection to the IMR Expansion Bus allows the MAC to view all packet data passing through a number of interconnected IMRs, allowing statistics and network management information to be collected.

The GPSI functional pins are duplicated as follows:

Pin Configuration for GPSI Function

Function	Type	LANCE Pin	MACE Pin
Receive Data	I	RX	RXDAT
Receive Clock	I	RCLK	SRDCLK
Receive Carrier Sense	I	RENA	RXCRS
Collision	I	CLSN	CLSN
Transmit Data	O	TX	TXDAT+
Transmit Clock	I	TCK	STDCLK
Transmit Enable	O	TENA	TXEN

IEEE 1149.1 Test Access Port Interface

An IEEE 1149.1 compatible boundary scan Test Access Port is provided for board level continuity test and diagnostics. All digital input, output and input/output and input/output pins are tested. Analog pins, including the AUI differential driver (DO±) and receivers DI±, CI±), and the crystal input (XTAL1/XTAL2) pins, are not tested.

The following is a brief summary of the IEEE 1149.1 compatible test functions implemented in the MACE device. For additional details, consult the IEEE Standard Test Access Port and Boundary-Scan Architecture document (IEEE Std 1149.1–1990).

The boundary scan test circuit requires four pins (TCK, TMS, TDI and TDO), defined as the Test Access Port (TAP). It includes a finite state machine (FSM), an instruction register, a data register array and a power on reset circuit. Internal pull-up resistors are provided for the TCK, TDI and TMS pins.

The TAP engine is a 16 state FSM, driven by the Test Clock (TCK) and the Test Mode Select (TMS) pins. An independent power on reset circuit is provided to ensure the FSM is in the TEST_LOGIC_RESET state at power up.

In addition to the minimum IEEE 1149.1 instruction requirements (EXTEST, SAMPLE and BYPASS), three additional instructions (IDCODE, TRI_ST and SET_I/O) are provided to further ease board level testing. All unused instruction codes are reserved.

IEEE 1149.1 Supported Instruction Summary

Inst Name	Description	Selected Data Reg	Reg Mode	Inst Code
EXTEST	External Test	BSR	Test	0000
ID Code	ID Code Inspection	ID Reg	Normal	0001
Sample	Sample Boundary	BSR	Normal	0010
TRI_ST	Force Tristate	Bypass	Normal	0011
SET_I/0	Control Boundary To I/0	Bypass	Test	0100
Bypass	Bypass Scan	Bypass	Normal	1111

After hardware or software reset, the IDCODE instruction is always invoked. The decoding logic provides signals to control the data flow in the DATA registers according to the current instruction.

Each Boundary Scan Register (BSR) cell also has two stages. A flip-flop and a latch are used in the SERIAL SHIFT STAGE and the PARALLEL OUTPUT STAGE respectively.

There are four possible operational modes in the BSR cell:

(1) CAPTURE

(2) SHIFT

(3) UPDATE

(4) SYSTEM FUNCTION

Other Data Registers

- BYPASS REG (1 bit)
- Device Identification Register (32 bits)
  Bits 31–28: Version (4 bits)
  Bits 27–12: Part number (16 bits) is 9400H
  Bits 11–1: Manufacturer ID (11 bits). The manufacturer ID code for AMD is 00000000001 in accordance with JEDEC Publication 106-A.
  Bit 0: Always a logic 1

SLAVE ACCESS OPERATION

Internal register accesses are based on a 2 or 3 SCLK cycle duration, dependent on the state of the $\overline{TC}$ input pin. $\overline{TC}$ must be externally pulled low to force the MACE device to perform a 3-cycle access. $\overline{TC}$ is internally pulled high if left unconnected, to configure the 2-cycle access by default.

All register accesses are byte wide with the exception of the data path to and from the internal FIFOs.

Data exchanges to/from register locations will take place over the appropriate half of the data bus to suit the host memory organization (as programmed by the BSWP bit in the BIU Configuration Control register).

The $\overline{BE0}$, $\overline{BE1}$ and $\overline{EOF}$ signals are provided to allow control of the data flow to and from the FIFOs. Byte read operations from the Receive FIFO cause data to be duplicated on both the upper and lower bytes of the data bus. Byte write operations to the Transmit FIFO must use the $\overline{BE0}$ and $\overline{BE1}$ inputs to define the active data byte to the MACE device.

Read Access

Details of the read access timing are located in the AC Waveforms section, Host System Interface, figures: Two-Cycle Receive FIFO/Register Read Timing and Three-Cycle Receive FIFO/Register Read Timing.

$\overline{TC}$ can be dynamically changed on a cycle by cycle basis to program the slave cycle execution for two ($\overline{TC}$ = HIGH) or three ($\overline{TC}$ = LOW) SCLK cycles. $\overline{TC}$ must be stable by the falling edge of SCLK (EDSEL = High) in S0 at the start of a cycle, and should only be changed in S0 in a multiple cycle burst.

A read cycle is initiated when either $\overline{CS}$ or $\overline{FDS}$ is sampled low on the falling edge of SCLK at S0. $\overline{FDS}$ and $\overline{CS}$ must be asserted exclusively. If they are active simultaneously when sampled, the MACE device will not execute any read or write cycle.

If $\overline{CS}$ is low, a Register Address read will take place. The state of the ADD4–0 will be used to commence decoding of the appropriate internal register/FIFO.

If $\overline{FDS}$ is low, a FIFO Direct read will take place from the RCVFIFO. The state of the ADD4–0 bus is irrelevant for the FIFO Direct mode.

With either the $\overline{CS}$ or $\overline{FDS}$ input active, the state of the ADD0-4 (for Register Address reads), R/$\overline{W}$ (high to indicate a read cycle), $\overline{BE0}$ and $\overline{BE1}$ will also be latched on the falling (EDSEL = HIGH) edge of SCLK at S0.

From the falling edge of SCLK in S1 (EDSEL = HIGH), the MACE device will drive data on DBUS15–0 and activate the $\overline{DTV}$ output (providing the read cycle completed successfully). If the cycle read the last byte/word of data for a specific frame from the RCVFIFO, the MACE device will also assert the $\overline{EOF}$ signal. DBUS15–0, $\overline{DTV}$ and $\overline{EOF}$ will be guaranteed valid and can be sampled on the falling (EDSEL = HIGH) edge of SCLK at S2.

If the Register Address mode is being used to access the RCVFIFO, once $\overline{EOF}$ is asserted during the last byte/word read for the frame, the Receive Frame Status can be read in one of two ways. The Register Address mode can be continued, by placing the appropriate address (00110b) on the address bus and executing four read cycles ($\overline{CS}$ active) on the Receive Frame Status location. In this case, additional Register Address read requests from the RCVFIFO will be ignored, and no $\overline{DTV}$ returned, until all four bytes of the Receive Frame Status register have been read. Alternatively, a FIFO Direct read can be performed, which will effectively route the Receive Frame Status through the RCVFIFO location. This mechanism is explained in more detail below.

If the FIFO Direct mode is used, the Receive Frame Status can be read directly from the RCVFIFO by continuing to execute read cycles (by asserting $\overline{FDS}$ low and R/$\overline{W}$ high) after $\overline{EOF}$ is asserted indicating the last byte/word read for the frame. Each of the four bytes of Receive Frame Status will appear on both halves of the data bus, as if the actual Receive Frame Status register were being accessed. Alternatively, the status can be read as normal using the Register Address mode by placing the appropriate address (00110b) on the address bus and executing four read cycles ($\overline{CS}$ active).

Either the FIFO Direct or Register Address modes can be interleaved at any time to read the Receive Frame Status, although this is considered unlikely due to the additional overhead it requires. In either case, no additional data will be read from the RCVFIFO until the Receive Frame Status has been read, as four bytes appended to the end of the packet when using the FIFO Direct mode, or as four bytes from the Receive Frame Status location when using the Register Address mode.

$\overline{EOF}$ will only be driven by the MACE device when reading received packet data from the RCVFIFO. At all other times, including reading the Receive Frame Status

using the FIFO Direct mode, the MACE device will place $\overline{EOF}$ in a high impedance state.

$\overline{RDTREQ}$ should be sampled on the falling edge of SCLK. The assertion of RDTREQ is programmed by RCVFW, and the de-assertion is modified dependent on the state of the RCVBRST bit (both in the FIFO Configuration Control register). See the section Receive FIFO Read for additional details.

Write Access

Details of the write access timing are located in the AC Waveforms section, Host System Interface, figures: Two-Cycle Transmit FIFO/Register Write Timing and Three-Cycle Transmit FIFO/Register Write Timing.

Write cycles are executed in a similar manner as the read cycle previously described, but with the R/$\overline{W}$ input low, and the host responsible to provide the data with sufficient set up to the falling edge of SCLK after S2.

After a FIFO write, $\overline{TDTREQ}$ should be sampled on or after the falling (EDSEL = HIGH) edge of SCLK after S3 of the FIFO write. The state of $\overline{TDTREQ}$ at this time will reflect the state of the XMTFIFO.

After going active (low), $\overline{TDTREQ}$ will remain low for two or more XMTFIFO writes.

The minimum high (inactive) time of $\overline{TDTREQ}$ is one SCLK cycle. When $\overline{EOF}$ is written to the Transmit FIFO, $\overline{TDTREQ}$ will go inactive after one SCLK cycle, for a minimum of one SCLK cycle.

Initialization

After power-up, RESET should be asserted for a minimum of 15 SCLK cycles to set the MACE device into a defined state. This will set all MACE registers to their default values. The receive and transmit functions will be turned off. A typical sequence to initialize the MACE device could look like this:

- Write the BIU Configuration Control (BIUCC) register to change the Byte Swap mode to big endian or to change the Transmit Start Point.
- Write the FIFO Configuration Control (FIFOCC) register to change the FIFO watermarks or to enable the FIFO Burst Mode.
- Write the Interrupt Mask Register (IMR) to disable unwanted interrupt sources.
- Write the PLS Configuration Control (PLSCC) register to enable the active network port. If the GPSI interface is used, the register must be written twice. The first write access should only set PORTSEL[1–0] = 11. The second access must write again PORTSEL[1–0] = 11 and additionally set ENPLSIO = 1. This sequence is required to avoid contention on the clock, data and/or carrier indication signals.
- Write the PHY Configuration Control (PHYCC) register to configure any non-default mode if the 10BASE-T interface is used.
- Program the Logical Address Filter (LADRF) register or the Physical Address Register (PADR). The Internal Address Configuration (IAC) register must be accessed first. Set the Address Change (ADDRCHG) bit to request access to the internal address RAM. Poll the bit until it is cleared by the MACE device indicating that access to the internal address RAM is permitted. In the case of an address RAM access after hardware or software reset (ENRCV has not been set), the MACE device will return ADDRCHG = 0 right away. Set the LOGADDR bit in the IAC register to select writing to the Logical Address Filter register. Set the PHYADDR bit in the IAC register to select writing to the Physical Address Register. Either bit can be set together with writing the ADDRCHG bit. Initializing the Logical Address Filter register requires 8 write cycles. Initializing the Physical Address Register requires 6 write cycles.
- Write the User Test Register (UTR) to set the MACE device into any of the user diagnostic modes such as loopback.
- Write the MAC Configuration Control (MACCC) register as the last step in the initialization sequence to enable the receiver and transmitter. Note that the system must guarantee a delay of 1 ms after power-up before enabling the receiver and transmitter to allow the MACE phase lock loop to stabilize.
- The Transmit Frame Control (XMTFC) and the Receive Frame Control (RCVFC) registers can be programmed on a per packet basis.

Reinitialization

The SWRST bit in the BIU Configuration Control (BIUCC) register can be set to reset the MACE device into a defined state for reinitialization. The same sequence described in the initialization section can be used. The 1 ms delay for the MACE phase lock loop stabilization need not to be observed as it only applies to a power–up situation.

TRANSMIT OPERATION

The transmit operation and features of the MACE device are controlled by programmable options. These options are programmed through the BIU, FIFO and MAC Configuration Control registers.

Parameters controlled by the MAC Configuration Control register are generally programmed only once, during initialization, and are therefore static during the normal operation of the MACE device (see the Media Access Control section for a detailed description). The features controlled by the FIFO Configuration Control

register and the Transmit Frame Control register can be re-programmed if the MACE device is not transmitting.

Transmit FIFO Write

The Transmit FIFO is accessed by performing a host generated write sequence on the MACE device. See the Slave Access Operation-Write Access section and the AC Waveforms section, Host System Interface, figures: Two-Cycle Transmit FIFO/Register Write Timing and Three-Cycle Transmit FIFO/Register Write Timing for details of the write access timing.

There are two fundamentally different access methods to write data into the FIFO. Using the Register Address mode, the FIFO can be addressed using the ADD0-4 lines, (address 00001b), initiating the cycle with the $\overline{CS}$ and R/$\overline{W}$ (low) signals. The FIFO Direct mode allows write access to the Transmit FIFO without use of the address lines, and using only the $\overline{FDS}$ and R/$\overline{W}$ lines. If the MACE device detects both signals active, it will not execute a write cycle. The write cycle timing for the Register Address or Direct FIFO modes are identical. $\overline{FDS}$ and $\overline{CS}$ should be mutually exclusive.

The data stream to the Transmit FIFO is written using multiple byte and/or word writes. $\overline{CS}$ or $\overline{FDS}$ does not have to be returned inactive to commence execution of the next write cycle. If $\overline{CS}$/$\overline{FDS}$ is detected low at the falling edge of S0, a write cycle will commence. Note that $\overline{EOF}$ must be asserted by the host/controller during the last byte/word transfer.

Transmit Function Programming

The Transmit Frame Control register allows programming of dynamic transmit attributes. Automatic transmit features such as retry on collision, FCS generation/transmission and pad field insertion can all be programmed, to provide flexibility in the (re-)transmission of messages.

The disable retry on collision (DRTRY bit) and automatic pad field insertion (APAD XMT bit) features should not be changed while data remains in the Transmit FIFO. Writing to either the DRTRY or APAD XMT bits in this case may have unpredictable results. These bits are not internally latched or protected. When writing to the Transmit Frame Control register the DRTRY and APAD XMT bits should be programmed consistently. Once the Transmit FIFO is empty, DRTRY and APAD XMT can be reprogrammed.

This can be achieved with no risk of transmit data loss or corruption by clearing ENXMT after the packet data for the current frame has been completely loaded. The transmission will complete normally and the activation of the $\overline{INTR}$ pin can be used to determine if the transmit frame has completed (XMTINT will be set in the Interrupt Register). Once the Transmit Frame Status has been read, APAD XMT and/or DRTRY can be changed and ENXMT set to restart the transmit process with the new parameters.

APAD XMT is sampled if there are less than 60 bytes in the transmit packet when the last bit of the last byte is transmitted. If APAD XMT is set, a pad field of pattern *00h* is added until the minimum frame size of 64 bytes (excluding preamble and SFD) is achieved. If APAD XMT is clear, no pad field insertion will take place and runt packet transmission is possible. When APAD XMT is enabled, the DXMTFCS feature is over-ridden and the four byte FCS will be added to the transmitted packet unconditionally.

The disable FCS generation/transmission feature can be programmed dynamically on a packet by packet basis. The current state of the DXMTFCS bit is internally latched on the last write to the Transmit FIFO, when the $\overline{EOF}$ indication is asserted by the host/controller.

The programming of static transmit attributes are distributed between the BIU, FIFO and MAC Configuration Control registers.

The point at which transmission begins in relation to the number of bytes of a frame in the FIFO is controlled by the XMTSP bits in the BIU Configuration Control register. Depending on the bus latency of the system, XMTSP can be set to ensure that the Transmit FIFO does not underflow before more data is written to the FIFO. When the entire frame is in the FIFO, or the FIFO becomes full before the threshold is reached, transmission of preamble will commence regardless of the value in XMTSP. The default value of XMTSP is 64 bytes after reset.

The point at which $\overline{TDTREQ}$ is asserted in relation to the number of empty bytes present in the Transmit FIFO is controlled by the XMTFW bits in the FIFO Configuration Control register. $\overline{TDTREQ}$ will be asserted when one of the following conditions is true:

- The number of bytes free in the Transmit FIFO relative to the current Saved Read Pointer value is greater than or equal to the threshold set by the XMTFW (16, 32 or 64 bytes). The *Saved Read Pointer* is the first byte of the current transmit frame, either in progress or awaiting channel availability.
- The number of bytes free in the Transmit FIFO relative to the current *Read Pointer* value is greater than or equal to the threshold set by the XMTFW (16, 32 or 64 bytes). The *Read Pointer* becomes available only after a minimum of 64 byte frame length has been transmitted on the network (eight bytes of preamble plus 56 bytes of data), and points to the current byte of the frame being transmitted.

Depending on the bus latency of the system, XMTFW can be set to ensure that the Transmit FIFO does not underflow before more data is written into the FIFO. When the entire frame is in the FIFO, $\overline{\text{TDTREQ}}$ will remain asserted if sufficient bytes remain empty. The default value of XMTFW is 64 bytes after hardware or software reset. Note that if the XMTFW is set below the 64 byte limit, the transmit latency for the host to service the MACE device is effectively increased, since $\overline{\text{TDTREQ}}$ will occur earlier in the transmit sequence and more bytes will be present in the Transmit FIFO when the $\overline{\text{TDTREQ}}$ is de-asserted.

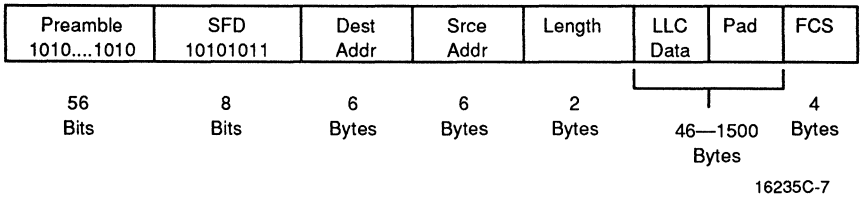
The transmit operation of the MACE device can be halted at any time by clearing the ENXMT bit (bit 1) in the MAC Configuration Control register. Note that any complete transmit frame that is in the Transmit FIFO and is currently in progress will complete, prior to the transmit function halting. Transmit frames in the FIFO which have not commenced will not be started. Transmit frames which have commenced but which have not been fully transferred into the Transmit FIFO will be aborted, in one of two ways. If less than 544 bits (68 bytes) have been transmitted onto the network, the transmission will be terminated immediately, generating a runt packet which can be deleted at the receiving station. If greater than 544 bits have been transmitted, the messages will have the current CRC inverted and appended at the next byte boundary, to guarantee an error is detected at the receiving station. This feature ensures that packets will not be generated with potential undetected data corruption. An explanation of the 544 bit derivation appears in the "*Automatic Pad Generation*" section.

Automatic Pad Generation

Transmit frames can be automatically padded to extend them to 64 data bytes (excluding preamble) permitting the minimum frame size of 64 bytes (512 bits) for 802.3/Ethernet to be guaranteed, with no software intervention from the host system.

APAD XMT = 1 enables the automatic padding feature. The pad is placed between the LLC Data field and FCS field in the 802.3 frame. The FCS is always added if APAD XMT = 1, regardless of the state of DXMTFCS. The transmit frame will be padded by bytes with the value of 00h. The default value of APAD XMT will enable auto pad generation after hardware or software reset.

It is the responsibility of upper layer software to correctly define the actual length field contained in the message to correspond to the total number of LLC Data bytes encapsulated in the packet (length field as defined in the IEEE 802.3 standard). The length value contained in the message is not used by the MACE device to compute the actual number of pad bytes to be inserted. The MACE chip will append pad bytes dependent on the actual number of bits transmitted onto the network. Once the last data byte of the frame has completed, prior to appending the FCS, the MACE device will check to ensure that 544 bits have been transmitted. If not, pad bytes are added to extend the frame size to this value, and the FCS is then added.



IEEE 802.3 Format Data Frame

The 544 bit count is derived from the following:

Minimum frame size (excluding preamble, including FCS)	64 bytes	512 bits
Preamble/SFD size	8 bytes	64 bits
FCS size	4 bytes	32bits

To be classed as a minimum size frame at the receiver, the transmitted frame must contain:

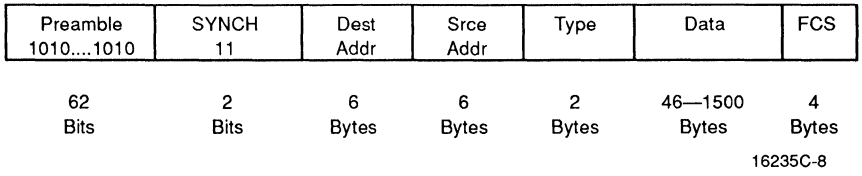
Preamble + (Min Frame Size + FCS) bits

At the point that FCS is to be appended, the transmitted frame should contain:

Preamble + (Min Frame Size - FCS) bits
64 + (512 - 32) bits

A minimum length transmit frame from the MACE device will therefore be 576 bits, after the FCS is appended.

The Ethernet specification makes no use of the LLC pad field, and assumes that minimum length messages will be at least 64 bytes in length.



Ethernet Format Data Frame

Transmit FCS Generation

Automatic generation and transmission of FCS for a transmit frame depends on the value of DXMTFCS (Disable Transmit FCS) when the $\overline{EOF}$ is asserted indicating the last byte/word of data for the transmit frame is being written to the FIFO. The action of writing the last data byte/word of the transmit frame, latches the current contents of the Transmit Frame Control register, and therefore determines the programming of DXMTFCS for the transmit frame. When DXMTFCS = 0 the transmitter will generate and append the FCS to the transmitted frame. If the automatic padding feature is invoked (APAD XMT in Transmit Frame Control), the FCS will be appended regardless of the state of DXMTFCS. Note that the calculated FCS is transmitted most significant bit first. The default value of DXMTFCS is 0 after hardware or software reset.

Transmit Status Information

Although multiple transmit frames can be queued in the Transmit FIFO, the MACE device will not permit loss of Transmit Frame Status information. The Transmit Frame Status and Transmit Retry Count can only be buffered internally for a maximum of two frames. The MACE device will therefore not commence a third transmit frame, until the status from the first frame is read. Once the Transmit Retry Count and Transmit Frame Status for the first transmit packet is read, the MACE device will autonomously begin the next transmit frame, provided that a transmit frame is pending, the XMTSP threshold has been exceeded (or the XMTFIFO is full), the network medium is free, and the IPG time has elapsed.

Indication of valid Transmit Frame Status can be obtained by servicing the hardware interrupt and testing the XMTINT bit in the Interrupt Register, or by polling the XMTSV bit in the Poll register if a continuous polling mechanism is required. If the Transmit Retry Count data is required (for loading, diagnostic, or management information), XMTRC must be read prior to XMTFS. Reading the XMTFS register when the XMTSV bit is set will clear both the XMTRC and XMTFS values.

Transmit Exception Conditions

Exception conditions for frame transmission fall into two distinct categories; those which are the result of normal network operation and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the MACE device are:

(a) Collisions within the slot time with automatic retry

(b) Deletion of packets due to excessive transmission attempts.

(a) The MACE device will ensure that collisions which occur within 512 bit times from the start of transmission (including preamble) will be automatically retried with no host intervention. The Transmit FIFO ensures this by guaranteeing that data contained within the Transmit FIFO will not be overwritten until at least 64 bytes (512 bits) of data have been successfully transmitted onto the network. This criteria will be met, regardless of whether the transmit frame was the first (or only) frame in the Transmit FIFO, or if the transmit frame was queued pending completion of the preceding frame.

(b) If 16 total attempts (initial attempt plus 15 retries) have been made to transmit the frame, the MACE device will abandon the transmit process for the particular frame, de-assert the $\overline{TDTREQ}$ pin, report a Retry Error (RTRY) in the Transmit Frame Status, and set the XMTINT bit in the Interrupt Register, causing activation of the external $\overline{INTR}$ pin providing the interrupt is unmasked.

Once the XMTINT condition has been externally recognized, the Transmit Frame Counter (XMTFC) can be read to determine whether the tail end of the frame that suffers the RTRY error is still in the host memory (i.e., when XMTFC = 0). This XMTFC read should be requested before the Transmit Frame Status read since reading the XMTFS would cause the XMTFC to decrement. If the tail end of the frame is indeed still in the host memory, the host is responsible for ensuring that the tail end of the frame does not get written into the FIFO and does not get transmitted as a whole frame. It is recommended that the host clear the tail end of the frame from the host memory before requesting the XMTFS read so that after the XMTFS read, when MACE device re-asserts $\overline{TDTREQ}$, the tail end of the frame does not get written into the FIFO. The Transmit Frame Status read will indicate that the RTRY error occurred. The read operation on the Transmit Frame Status will update the FIFO read and write pointers. If no *End-of-Frame* write ($\overline{EOF}$ pin assertion) had occurred during the FIFO write sequence, the entire transmit path will be reset (which will update the Transmit FIFO watermark with the

current XMTFW value in the FIFO Configuration Control register). If a whole frame does reside in the FIFO, the read pointer will be moved to the start of the next frame or free location in the FIFO, and the write pointer will be unaffected. $\overline{\text{TDTREQ}}$ will not be re-asserted until the Transmit Frame Status has been read.

After a RTRY error, all further packet transmission will be suspended until the Transmit Frame Status is read, regardless of whether additional packet data exists in the FIFO to be transmitted. Receive FIFO read operations are not impaired.

Packets experiencing 16 unsuccessful attempt to transmit will not be re-tried. Recovery from this condition must be performed by upper layer software.

Abnormal network conditions include:

(a) Loss of carrier.
(b) Late collision.
(c) SQE Test Error.

These should not occur on a correctly configured 802.3 network, but will be reported if the network has been incorrectly configured or a fault condition exists.

(a) A loss of carrier condition will be reported if the MACE device cannot observe receive activity while it is transmitting. After the MACE device initiates a transmission it will expect to see data *looped-back* on the receive input path. This will internally generate a carrier sense, indicating that the integrity of the data path to and from the external MAU is intact, and that the MAU is operating correctly.

When the AUI port is selected, if carrier sense does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in the Transmit Frame Status (bit 7) after the packet has been transmitted. The packet will not be re-tried on the basis of an LCAR error.

When the 10BASE-T port is selected, LCAR will be reported for every packet transmitted during the Link fail condition.

When the GPSI port is selected, LCAR will be reported if the RXCRS input pin fails to become active during a transmission, or once active, goes inactive before the end of transmission.

When the DAI port is selected, LCAR errors will not occur, since the MACE device will internally loop back the transmit data path to the receiver. The loop back feature must not be performed by the external transceiver when the DAI port is used.

During internal loopback, LCAR will not be set, since the MACE device has direct control of the transmit and receive path integrity. When in external loopback, LCAR will operate normally according to the specific port which has been selected.

(b) A late collision will be reported if a collision condition exists or commences 64 byte times (512 bit times) after the transmit process was initiated (first bit of preamble commenced). The MACE device will abandon the transmit process for the particular frame, complete transmission of the jam sequence (32-bit all zeroes pattern), de-assert the $\overline{\text{TDTREQ}}$ pin, report the Late Collision (LCOL) and Transmit Status Valid (XMTSV) in the Transmit Frame Status, and set the XMTINT bit in the Interrupt Register, causing activation of the external $\overline{\text{INTR}}$ pin providing the interrupt is unmasked.

Once the XMTINT condition has been externally recognized, the Transmit Frame Counter (XMTFC) can be read to determine whether the tail end of the frame that suffers the LCOL error is still in the host memory (i.e., when XMTFC = 0). This XMTFC read should be requested before the Transmit Frame Status read since reading the XMTFS would cause the XMTFC to decrement. If the tail end of the frame is indeed still in the host memory, the host is responsible for ensuring that the tail end of the frame does not get written into the FIFO and does not get transmitted as a whole frame. It is recommended that the host clear the tail end of the frame from the host memory before requesting the XMTFS read so that after the XMTFS read,when the MACE device re-asserts $\overline{\text{TDTREQ}}$, the tail end of the frame does not get written into the FIFO. The Transmit Frame Status read will indicate that the LCOL error occurred. The read operation on the Transmit Frame Status will update the FIFO read and write pointers. If no *End-of-Frame* write ($\overline{\text{EOF}}$ pin assertion) had occurred during the FIFO write sequence, the entire transmit path will be reset (which will update the Transmit FIFO watermark with the current XMTFW value in the FIFO Configuration Control register). If a whole frame resides in the FIFO, the read pointer will be moved to the start of the next frame or free location in the FIFO, and the write pointer will be unaffected. $\overline{\text{TDTREQ}}$ will not be re-asserted until the Transmit Frame Status has been read.

After an LCOL error, all further packet transmission will be suspended until the Transmit Frame Status is read, regardless of whether additional packet data exists in the FIFO to be transmitted. Receive FIFO operations are unaffected.

Packets experiencing a late collision will not be re-tried. Recovery from this condition must be performed by upper layer software.

(c) During the inter packet gap time following the completion of a transmitted message, the AUI CI± pair is asserted by some transceivers as a self-test. When the AUI port has been selected, the integral Manchester Encoder/Decoder will expect the SQE Test Message

(nominal 10 MHz sequence) to be returned via the CI± pair, within a 40 network bit time period after DI± goes inactive. If the CI± input is not asserted within the 40 network bit time period following the completion of transmission, then the MACE device will set the CERR bit (bit 5) in the Interrupt Register. The $\overline{INTR}$ pin will be activated if the corresponding mask bit CERRM = 0.

When the GPSI port is selected, the MACE device will expect the CLSN input pin to be asserted 40 bit times after the transmission has completed (after TXEN output pin has gone inactive). When the DAI port has been selected, the CERR bit will not be reported. A transceiver connected via the DAI port is not expected to support the SQE Test Message feature.

Host related transmit exception conditions include:

(a) Overflow caused by excessive writes to the Transmit FIFO ($\overline{DTV}$ will not be issued if the Transmit FIFO is full).

(b) Underflow caused by lack of host writes to the Transmit FIFO.

(c) Not reading current Transmit Frame Status.

(a) The host may continue to write to the Transmit FIFO after the $\overline{TDTREQ}$ has been de-asserted, and can safely do so on the basis of knowledge of the number of free bytes remaining (set by XMTFW in the FIFO Configuration Control register). If however the host system continues to write data to the point that no additional FIFO space exists, the MACE device will not return the $\overline{DTV}$ signal and hence will effectively not acknowledge acceptance of the data. It is the host's responsibility to ensure that the data is re-presented at a future time when space exists in the Transmit FIFO, and to track the actual data written into the FIFO.

(b) If the host fails to respond to the $\overline{TDTREQ}$ from the MACE device before the Transmit FIFO is emptied, a FIFO underrun will occur. The MACE device will in this case terminate the network transmission in an orderly sequence. If less than 512 bits have been transmitted onto the network the transmission will be terminated immediately, generating a runt packet. If greater than 512 bits have been transmitted, the message will have the current CRC inverted and appended at the next byte boundary, to guarantee an FCS error is detected at the receiving station. The MACE device will report this condition to the host by de-asserting the $\overline{TDTREQ}$ pin, setting the UFLO and XMTSV bits (in the Transmit Frame Status) and the XMTINT bit (in the Interrupt Register), and asserting the $\overline{INTR}$ pin providing the corresponding XMTINTM bit (in the Interrupt Mask Register) is cleared.

Once the XMTINT condition has been externally recognized, the Transmit Frame Counter (XMTFC) can be read to determine whether the tail end of the frame that suffers the UFLO error is still in the host memory (i.e., when XMTFC = 0). In the case of FIFO underrun, this will definitely be the case and the host is responsible for ensuring that the tail end of the frame does not get written into the FIFO and does not get transmitted as a whole frame. It is recommended that the host clear the tail end of the frame from the host memory before requesting the XMTFS read so that after the XMTFS read, when the MACE device re-asserts $\overline{TDTREQ}$, the tail end of the frame does not get written into the FIFO. The Transmit Frame Status read will indicate that the UFLO error occurred. The read operation on the Transmit Frame Status will update the FIFO read and write pointers and the entire transmit path will be reset (which will update the Transmit FIFO watermark with the current XMTFW value in the FIFO Configuration Control register). $\overline{TDTREQ}$ will not be re-asserted until the Transmit Frame Status has been read.

(c) The MACE device will internally store the Transmit Frame Status for up to two packets. If the host fails to read the Transmit Frame Status and both internal entries become occupied, the MACE device will not commence any subsequent transmit frames to prevent overwriting of the internally stored values. This will occur regardless of the number of bytes written to the Transmit FIFO.

RECEIVE OPERATION

The receive operation and features of the MACE device are controlled by programmable options. These options are programmed through the BIU, FIFO and MAC Configuration Control registers.

Parameters controlled by the MAC Configuration Control register are generally programmed only once, during initialization, and are therefore static during the normal operation of the MACE device (see the Media Access Control section for a detailed description). The features controlled by the FIFO Configuration Control register and the Receive Frame Control register can be programmed without performing a reset on the part. The host is responsible for ensuring that no data is present in the Receive FIFO when re-programming the receive attributes.

Receive FIFO Read

The Receive FIFO is accessed by performing a host generated read sequence on the MACE device. See the Slave Access Operation-Read Access section and the AC Waveforms section, Host System Interface, figures: "2 Cycle Receive FIFO/Register Read Timing" and "3 Cycle Receive FIFO/Register Read Timing" for details of the read access timing.

Note that $\overline{EOF}$ will be asserted by the MACE device during the last data byte/word transfer.

Receive Function Programming

The Receive Frame Control register allows programming of the automatic pad field stripping feature and the configuration of the Match/Reject (M/$\overline{R}$) pin. ASTRP RCV and M/$\overline{R}$ must be static when the receive function is enabled (ENRCV = 1). The receiver should be disabled before (re-) programming these options.

The EADI port can be used to permit reception of frames to commence whilst external address decoding takes place. The M/$\overline{R}$ bit defines the function of the $\overline{EAM/R}$ pin, and hence whether frames will be accepted or rejected by the external address comparison logic.

The programming of additional receive attributes are distributed between the FIFO and MAC Configuration Control registers, and the User Test Register.

All receive frames can be accepted by setting the PROM bit (bit 7) in the MAC Configuration Control register. When PROM is set, the MACE device will attempt to receive all messages, subject to minimum frame enforcement. Setting PROM will override the use of the EADI port to force the rejection of unwanted messages. See the sections *External Address Detection Interface* for more details.

The point at which $\overline{RDTREQ}$ is asserted in relation to the number of bytes of a frame that are present in the Receive FIFO (RCVFIFO) is controlled by the RCVFW bits in the FIFO Configuration Control register, or the LLRCV bit in the Receive Frame Control register. $\overline{RDTREQ}$ will be asserted when one of the following conditions is true:

(i) There are at least 64 bytes in the RCVFIFO.

(ii) The received packet has passed the 64 byte minimum criteria, and the number of bytes in the RCVFIFO is greater than or equal to the threshold set by the RCVFW (16 or 32 bytes).

(iii) A receive packet has completed, and part or all of it is present in the RCVFIFO.

(iv) The LLRCV bit has been set and greater than 12-bytes of at least 8 bytes have been received.

Note that if the RCVFW is set below the 64-byte limit, the MACE device will still require 64-bytes of data to be received before the initial assertion of $\overline{RDTREQ}$. Subsequently, $\overline{RDTREQ}$ will be asserted at any time the RCVFW threshold is exceeded. The only times that the $\overline{RDTREQ}$ will be asserted when there are not at least an initial 64-bytes of data in the RCVFIFO are:

(i) When the ASTRP RCV bit has been set in the Receive Frame Control register, and the pad is automatically stripped from a minimum length packet.

(ii) When the RPA bit has been set in the User Test Register, and a runt packet of at least 8 bytes has been received.

(iii) When the LLRCV bit has been set in the Receive Frame Control register, and at least 12-bytes (after SFD) has been received.

No preamble/SFD bytes are loaded into the Receive FIFO. All references to bytes past through the receive FIFO are received after the preamble/SFD sequence.

Depending on the bus latency of the system, RCVFW can be set to ensure that the RCVFIFO does not overflow before more data is read. When the entire frame is in the RCVFIFO, $\overline{RDTREQ}$ will be asserted regardless of the value in RCVFW. The default value of RCVFW is 64-bytes after hardware or software reset.

The receive operation of the MACE device can be halted at any time by clearing the ENRCV bit in the MAC Configuration Control register. Note that any receive frame currently in progress will be accepted normally, and the MACE device will disable the receive process once the message has completed. The Missed Packet Count (MPC) will be incremented for subsequent packets that would have normally been passed to the host, and are now ignored due to the disabled state of the receiver.

Note that clearing the ENRCV bit disables the assertion of $\overline{RDTREQ}$. If ENRCV is cleared during receive activity and remains cleared for a long time and if the tail end of the receive frame currently in progress is longer than the amount of space available in the Receive FIFO, Receive FIFO overflow will occur. However, even with $\overline{RDTREQ}$ deasserted, if there is valid data in the Receive FIFO to be read, successful slave reads to the Receive FIFO can be executed (indicated by valid $\overline{DTV}$). It is the host's responsibility to avoid the overflow situation.

Automatic Pad Stripping

During reception of a frame the pad field can be stripped automatically. ASTRP RCV = 1 enables the automatic pad stripping feature. The pad field will be stripped before the frame is passed to the FIFO, thus preserving FIFO space for additional frames. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has the pad characters stripped.

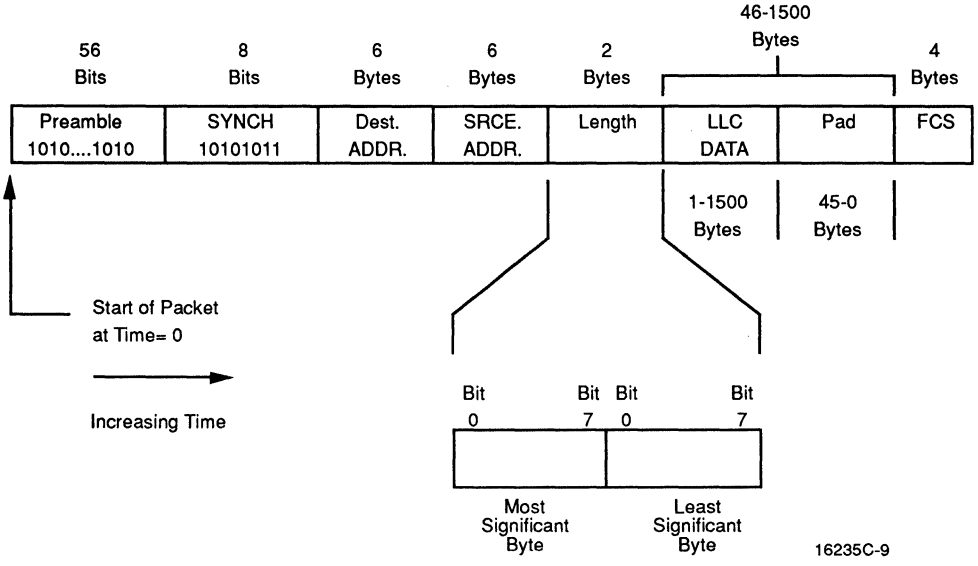
The number of bytes to be stripped is calculated from the embedded length field (as defined in the IEEE 802.3 definition) contained in the packet. The length indicates the actual number of LLC data bytes contained in the message. Any received frame which contains a length field less than 46 bytes will have the pad field stripped.

Receive frames which have a length field of 46 bytes or greater will be passed to the host unmodified.

Since any valid Ethernet Type field value will always be greater than a normal 802.3 Length field, the MACE device will not attempt to strip valid Ethernet frames.

Note that for some network protocols, the value passed in the Ethernet Type and/or 802.3 Length field is not compliant with either standard and may cause problems.

The diagram below shows the byte/bit ordering of the received length field for an 802.3 compatible frame format.



802.3 Packet and Length Field Transmission Order

Receive FCS Checking

Reception and checking of the received FCS is performed automatically by the MACE device. Note that if the Automatic Pad Stripping feature is enabled, the received FCS will be verified against the value computed for the incoming bit stream including pad characters, but it will not be passed through the Receive FIFO to the host. If an FCS error is detected, this will be reported by the FCS bit (bit 4) in the Receive Frame Status.

Receive Status Information

The $\overline{EOF}$ indication signals that the last byte/word of data has been passed from the FIFO for the specific frame. This will be accompanied by a RCVINT indication in the the Interrupt Register signaling that the Receive Frame Status has been updated, and must be read. The Receive Frame Status is a single location which must be read four times to allow the four bytes of status information associated with each frame to be read. Further data read operations from the Receive FIFO using the Register Address mode, will be ignored by the MACE device (indicated by the MACE chip not returning $\overline{DTV}$) until all four bytes of the Receive Frame Status have been read. Alternatively, the FIFO Direct access mode may be used to read the Receive Frame Status through the Receive FIFO. In either case, the 4-byte total must be read before additional receive data can be read from the Receive FIFO. However, the $\overline{RDTREQ}$ indication will continue to reflect the state of the Receive FIFO as normal, regardless of whether the Receive Frame Status has been read. $\overline{DTV}$ will not be returned when a read operation is performed on the Receive Frame Status location and no valid status is present or ready.

Note that the Receive Frame Status can be read using either the Register Address or FIFO Direct modes. For additional details, see the section *Receive FIFO Read.*

Receive Exception Conditions

Exception conditions for frame reception fall into two distinct categories; those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the MACE device are basically collisions within the slot time and automatic runt packet deletion. The MACE device will ensure that any receive packet which experiences a collision within 512 bit times

from the start of reception (excluding preamble) will be automatically deleted from the Receive FIFO with no host intervention (the state of the RPA bit in the User Test Register; or the RCVFW bits in the FIFO Configuration Control register have no effect on this). This criteria will be met, regardless of whether the receive frame was the first (or only) frame in the Receive FIFO, or if the receive frame was queued behind a previously received message.

Abnormal network conditions include:

- FCS errors
- Framing errors
- Dribbling bits
- Late collision

These should not occur on a correctly configured 802.3 network, but may be reported if the network has been incorrectly configured or a fault condition exists.

Host related receive exception conditions include:

(a) Underflow caused by excessive reads from the Receive FIFO ($\overline{DTV}$ will not be issued if the Receive FIFO is empty)

(b) Overflow caused by lack of host reads from the Receive FIFO

(c) Missed packets due to lack of host reads from the Receive FIFO and/or the Receive Frame Status

(a) Successive read operations from the Receive FIFO after the final byte of data/status has been read, will cause the $\overline{DTV}$ pin to remain de-asserted during the read operation, indicating that no valid data is present. There will be no adverse effect on the Receive FIFO.

(b) Data present in the Receive FIFO from packets which completed before the overflow condition occurred, can be read out by accessing the Receive FIFO normally. Once this data (and the associated Receive Frame Status) has been read, the $\overline{EOF}$ indication will be asserted by the MACE device during the first read operation takes place from the Receive FIFO, for the packet which suffered the overflow. If there were no other packets in the FIFO when the overflow occurred, the $\overline{EOF}$ will be asserted on the first read from the FIFO. In either case, the $\overline{EOF}$ indication will be accompanied by assertion of the $\overline{INTR}$ pin, providing that the RCVINTM bit in the Interrupt Mask Register is not set. If the Register Address mode is being used, the host is required to access the Receive Frame Status location using four separate read cycles. Further access to the Receive FIFO will be ignored by the MACE device until all four bytes of the Receive Frame Status have been read. $\overline{DTV}$ will not be returned if a Receive FIFO read is attempted. If the FIFO Direct mode is being used, the host can read the Receive Frame Status through the Receive FIFO, but the host must be aware that the subsequent four cycles will yield the receive status bytes, and not data from the same or a new packet. Only the OFLO bit will be valid in the Receive Frame Status, other error/status and the RCVCNT fields are invalid.

While the Receive FIFO is in the overflow condition, it is *deaf* to additional receive data on the network. However, the MACE device internal address detect logic continues to operate and counts the number of packets that would have been passed to the host under normal (non overflow) conditions. The Missed Packet Count (MPC) is an 8–bit count (in register 24) that maintains the number of packets which pass the address match criteria, and complete without collision. The MPC counter will wrap around when the maximum count of 255 is reached, setting the MPCO (Missed Packet Count Overflow) bit in the Interrupt Register, and asserting the $\overline{INTR}$ pin providing that MPCOM (Missed Packet Count Overflow Mask) in the Interrupt Mask Register is clear. MPCO will be cleared (the interrupt will be unmasked) after hardware or software reset. However, until the first time that the receiver is enabled, MPC will not increment, hence no interrupt will occur due to missed packets after a reset.

(c) Failure to read packet data from the Receive FIFO will eventually cause an overflow condition. The FIFO will maintain any previously completed packet(s), which can be read by the host at its convenience. However, packet data on the network will no longer be received, regardless of destination address, until the overflow is cleared by reading the remaining Receive FIFO data and Receive Status. The MACE device will increment the Missed Packet Count (MPC) register to indicate that a packet which would have been normally passed to the host, was dropped due to the error condition.

LOOPBACK OPERATION

During loopback, the FCS logic can be allocated to the receiver by setting RCVFCSE = 1 in User Test Register. This permits both the transmit and receive FCS operations to be verified during the loopback process. The state of RCVFCSE is only valid during loopback operation.

If RCVFCSE = 0, the MACE device will calculate and append the FCS to the transmitted message. The receive message passed to the host will therefore contain an additional four bytes of FCS. The Receive Frame Status will indicate the result of the loopback operation and the RCVCNT.

If RCVFCSE = 1, the last four bytes of the transmit message must contain the FCS computed for the transmit data preceding it. The MACE device will transmit the

data without addition of an FCS field, and the FCS will be calculated and verified at the receiver.

The loopback facilities of the MACE device allow full operation to be verified without disturbance to the network. Loopback operation is also affected by the state of the Loopback Control bits (LOOP [0–1]) in the User Test Register. This affects whether the internal MENDEC is considered part of the internal or external loopback path.

When in the loopback mode(s), the multicast address detection feature of the MACE device, programmed by the contents of the Logical Address Filter (LADR [63–0]) can only be tested when RCVFCSE = 1, allocating the CRC generator to the receiver. All other features operate identically in loopback as in normal operation, such as automatic transmit padding and receive pad stripping.

USER ACCESSIBLE REGISTERS

The following registers are provided for operation of the MACE device. All registers are 8-bits wide unless otherwise stated. Note that all reserved register bits should be written as zero.

Receive FIFO (RCVFIFO) (REG ADDR 0)

RCVFIFO [15–0]

This register provides a 16-bit data path from the Receive FIFO. Reading this register will read one word/byte from the Receive FIFO. The RCVFIFO should only be read when Receive Data Transfer Request ($\overline{RDTREQ}$) is asserted. If the RCVFIFO location is read before 64-bytes are available in the RCVFIFO (or 12-bytes in the case that LLRCV is set in the Receive Frame Control register), $\overline{DTV}$ will not be returned. Once the 64-byte threshold has been achieved and $\overline{RDTREQ}$ is asserted, the de-assertion of $\overline{RDTREQ}$ does not prevent additional data from being read from the RCVFIFO, but indicates the number of additional bytes which are present, before the RCVFIFO is emptied, and subsequent reads will not return $\overline{DTV}$ (see the FIFO Sub-System section for additional details). Write operations to this register will be ignored and $\overline{DTV}$ will not be returned.

Byte transfers from the RCVFIFO are supported, and will be fully aligned to the target memory architecture, defined by the BSWP bit in the BIU Configuration Control register. The Byte Enable inputs ($\overline{BE}1–0$) will define which half of the data bus should be used for the transfer. The external host/controller will be informed that the last byte/word of data in a receive frame is being read from the RCVFIFO, when the MACE device asserts the $\overline{EOF}$ signal.

Transmit FIFO (XMTFIFO) (REG ADDR 1)

XMTFIFO [15–0]

This register provides a 16-bit data path to the Transmit FIFO. Byte/word data written to this register will be placed in the Transmit FIFO. The XMTFIFO can be written at any time the Transmit Data Transfer Request ($\overline{TDTREQ}$) is asserted. The de-assertion of $\overline{TDTREQ}$ does not prevent data being written to the XMTFIFO, but indicates the number of additional write cycles which can take place, before the XMTFIFO is filled, and subsequent writes will not return $\overline{DTV}$ (see the FIFO Sub-System section for additional details). Read operations to this register will be ignored and $\overline{DTV}$ will not be returned.

Byte transfers to the XMTFIFO are supported, and accept data from the source memory architecture to ensure the correct byte ordering for transmission, defined by the BSWP bit in the MAC Configuration Control register. The Byte Enable inputs ($\overline{BE}1–0$) will define which half of the data bus should be used for the transfer. The use of byte transfers have implications on the latency time provided by the XMTFIFO (see the *FIFO Sub-System* section for additional details). The external host/controller must indicate the last byte/word of data in a transmit frame is being written to the XMTFIFO, by asserting the $\overline{EOF}$ signal.

Transmit Frame Control (XMTFC) (REG ADDR 2)

The Transmit Frame Control register is latched internally on the last write to the Transmit FIFO for each individual packet, when $\overline{EOF}$ is asserted. This permits automatic transmit padding and FCS generation on a packet-by-packet basis.

DRTRY	RES	RES	RES	DXMTFCS	RES	RES	APAD XMT

Bit	Name	Description
Bit 7	DRTRY	Disable Retry. When DRTRY is set, the MACE device will provide a single transmission attempt for the packet, all further retries will be suspended. In the case of a collision during the attempt, a Retry Error (RTRY) will be reported in the Transmit Status. With DRTRY cleared, the MACE device will attempt up to 15 retries (16 attempts total) before indicating a Retry Error. DRTRY is cleared by activation of the $\overline{RESET}$ pin or SWRST bit. DRTRY is sampled during the transmit process when a collision occurs. DRTRY should not be changed whilst data remains in the Transmit FIFO since this may cause an unpredictable retry response to a collision. Once the Transmit FIFO is empty, DRTRY can be reprogrammed.
Bit 6–4	RES	Reserved. Read as zeroes. Always write as zeroes.
Bit 3	DXMTFCS	Disable Transmit FCS. When DXMTFCS = 0 the transmitter will generate and append an FCS to the transmitted frame. When DXMTFCS = 1, no FCS will be appended to the transmitted frame, providing that APAD XMT is also clear. If APAD XMT is set, the calculated FCS will be appended to the transmitted message regardless of the state of DXMTFCS. The value of DXMTFCS for each frame is programmed when $\overline{EOF}$ is asserted to transfer the last byte/word for the transmit packet to the FIFO. DXMTFCS is cleared by

activation of the RESET pin or SWRST bit. DXMTFCS is sampled only when EOF is asserted during a Transmit FIFO write.

Bit	Name	Description
Bit 2–1	RES	Reserved. Read as zeroes. Always write as zeroes.
Bit 0	APAD XMT	Auto Pad Transmit. APAD XMT enables the automatic padding feature. Transmit frames will be padded to extend them to 64 bytes including FCS. The FCS is calculated for the entire frame including pad, and appended after the pad field. APAD XMT will override the programming of the DXMTFCS bit. APAD XMT is set by activation of the RESET pin or SWRST bit. APAD XMT is sampled only when EOF is asserted during a Transmit FIFO write.

Transmit Frame Status (XMTFS) (REG ADDR 3)

The Transmit Frame Status is valid when the XMTSV bit is set. The register is read only, and is cleared when XMTSV is set and a read operation is performed. The XMTINT bit in the Interrupt Register will be set when any bit is set in this register.

Note that if XMTSV is not set, the values in this register can change at any time, including during a read operation. This register should be read after the Transmit Retry Count (XMTRC). See the description of the Transmit Retry Count (XMTRC) for additional details.

XMTSV	UFLO	LCOL	MORE	ONE	DEFER	LCAR	RTRY

Bit	Name	Description
Bit 7	XMTSV	Transmit Status Valid. Transmit Status Valid indicates that this status is valid for the last frame transmitted. The value of XMTSV will not change during a read operation.
Bit 6	UFLO	Underflow. Indicates that the Transmit FIFO emptied before the end of frame was reached. The transmitted frame is truncated at that point. If UFLO is set, TDTREQ will be de-asserted, and will not be re-asserted until the XMTFS has been read.
Bit 5	LCOL	Late Collision. Indicates that a collision occurred after the slot time of the channel elapsed. If LCOL is set, TDTREQ will be de-asserted, and will not be re-asserted until the XMTFS has been read. The MACE device does not retry after a late collision.
Bit 4	MORE	More. Indicates that more than one retry was needed to transmit the frame. ONE, MORE and RTRY are mutually exclusive.
Bit 3	ONE	One. Indicates that exactly one retry was needed to transmit the frame. ONE, MORE and RTRY are mutually exclusive.
Bit 2	DEFER	Defer. Indicates that MACE device had to defer transmission of the frame. This condition results if the channel is busy when the MACE device is ready to transmit.
Bit 1	LCAR	Loss of Carrier. Indicates that the carrier became false during a transmission. The MACE device does not retry upon Loss of Carrier. LCAR will not be set when the DAI port is selected, when the 10BASE-T port is selected and in the link pass state, or during any internal loopback mode. When the 10BASE-T port is selected and in the link fail state, LCAR will will be reported for any transmission attempt.
Bit 0	RTRY	Retry Error. Indicates that all attempts to transmit the frame were unsuccessful, and that further attempts have been aborted. If Disable Retry (DRTRY in the Transmit Frame Control register) is cleared, RTRY will be set when a total of 16 unsuccessful attempts were made to transmit the frame. If DRTRY is set, RTRY indicates that the first and only attempt to transmit the frame was unsuccessful. ONE, MORE and RTRY are mutually exclusive. If RTRY is set, TDTREQ will be de-asserted, and will not be re-asserted until the XMTFS has been read.

Transmit Retry Count (XMTRC) (REG ADDR 4)

The Transmit Retry Count should be read only in response to a hardware interrupt request (INTR asserted) when XMTINT is set in the Interrupt Register, or after XMTSV is set in the Poll Register. The register should be read before the Transmit Frame Status register. Reading the Transmit Frame Status with XMTSV set will cause the XMTRC value to be reset. This register is read only.

EXDEF	RES	RES	RES	XMTRC[3–0]

Bit	Name	Description
Bit 3-0	EXDEF	Excessive Defer. The EXDEF bit will be set if a transmit frame waited for an excessive period for transmission. An excessive defer time is defined in accordance with the following (from page 34, section 5.2.4.1 of IEEE Std 802.3h–1990 Layer Management): maxDeferTime = {2 x (max frame size x 8)} bits where maxFrameSize = 1518 bytes (from page 68, section 4.4.2.1 of ANSI/IEEE Std 802.3–1990). So, the maxDeferTime = 24288 bits = $2^{14}+2^{12}+2^{11}+2^{10}+2^{9}+2^{7}+2^{6}+2^{5}$
Bit 6–4	RES	Reserved. Read as zeroes. Always write as zeroes.
Bit 3–0	XMTRC [3–0]	Transmit Retry Count. Contains the count of the number of retry attempts made by the MACE device to transmit the current transmit packet. The value of the counter will be zero if the first transmission attempt was successful, and a maximum of 15 if all retry attempts were utilized. RTRY will be set in Transmit Frame Status if all 16 attempts were unsuccessful.

Receive Frame Control (RCVFC) (REG ADDR 5)

RES	RES	RES	RES	LLRCV	M/R	RES	ASTRPRCV

Bit	Name	Description
Bit 7–4	RES	Reserved. Read as zeroes. Always write as zeroes.
Bit 3	LLRCV	Low Latency Receive. A programmable option to allow access to the Receive FIFO before the 64-byte threshold has been reached. When set, data can be read from the RCVFIFO once a low threshold (12-bytes after SFD plus synchronization) has been exceeded, causing RDTREQ to be asserted. RDTREQ will remain asserted as long as one read cycle can be performed on the RCVFIFO (identical to the burst mode). Indication of a valid read cycle from the RCVFIFO will return DTV asserted. Reading the RCVFIFO before data is available, or while waiting for additional data once a packet is in progress will not cause the RCVFIFO to underflow, and will be indicated by DTV being invalid. The MACE device will no longer be able to reject runts in this mode, this responsibility is transferred to the host system. In the case of a collided packet (normal slot time collision or late collision), the MACE device will abort the reception, and return the RCVFS. Note that all collisions in this mode will appear as late collisions and be reported by the CLSN bit in the Receive Status (RCVSTS) byte. If the host does not keep up with the incoming receive data, normal RCVFIFO overflow recovery is provided.
Bit 2	M/R	Match/Reject. The Match/Reject option sets the criteria for the External Address Detection Interface. If set, the EAM/R pin is configured as External Address Match, and is used to signal the acceptance of a receive frame to the MACE device. If cleared, the pin functions as External Address Reject and is used to flush unwanted packets from the Receive FIFO prior to the first assertion of RDTREQ. M/R is cleared by activation of the RESET pin or SWRST bit. When the EADI feature is disabled, the EAM/R pin must be tied active (low) and all normal receive address recognition configurations are supported (physical, logical and promiscuous). See the section "External Address Detection Interface" for additional details.
Bit 1	RES	Reserved. Read as zero. Always write as zero.

Bit 0 ASTRP RCV — Auto Strip Receive. ASTRP RCV enables the automatic pad stripping feature. The pad and FCS fields will be stripped from receive frames and not placed in the FIFO. ASTRP RCV is set by activation of the RESET pin or the SWRST bit.

Receive Frame Status (RCVFS) (REG ADDR 6)

RCVFS [31–00]

The Receive Frame Status is a single byte location which must be read by four read cycles to obtain the four bytes (32-bits) of status associated with each receive frame. Receive Frame Status can be read using either the Register Direct or FIFO Direct access modes.

In Register Direct mode, access to the Receive FIFO will be denied until all four status bytes for the completed frame have been read from the Receive Frame Status location. In FIFO Direct mode, the Receive Frame Status is read through the Receive FIFO location, by continuing to execute four read cycles after the completion of packet data (and assertion of EOF). The Receive Frame Status can be read using either mode, or a combination of both modes, however each status byte will be presented only once regardless of access method. Other register reads and/or writes can be interleaved at any time, during the Receive Frame Status sequence.

The Receive Frame Status consists of the following four bytes of information:

RFS0 Receive Message Byte Count (RCVCNT) [7–0]
RFS1 Receive Status, (RCVSTS) [11–8]
RFS2 Runt Packet Count (RNTPC) [7–0]
RFS3 Receive Collision Count (RCVCC) [7–0]

RFS0—Receive Message Byte Count (RCVCNT)

RCVCNT [7:0]

Bit	Name	Description
Bit 7-0	RCVCNT [7:0]	The Receive Message Byte Count indicates the number of whole bytes in the received message. If pad bytes were stripped from the received frame, RCVCNT indicates the number of bytes received less the number of pad bytes and less the number of FCS bytes. RCVCNT is 12 bits long. If a late collision is detected (CLSN set in RCVSTS), the count is an indication of the length (in byte times) of the duration of the receive activity including the collision. RCVCNT [10:8] correspond to bits 3–0 in RFS1 of the Receive Frame Status. RCVCNT [11–0] will be invalid when OFLO is set.

RFS1—Receive Status (RCVSTS)

OFLO	CLSN	FRAM	FCS	RCVCNT [10:8]

Bit	Name	Description
Bit 7	OFLO	Overflow flag. Indicates that the Receive FIFO over flowed due to the inability of the host/controller to read data fast enough to keep pace with the receive serial bit stream and the latency provided by the Receive FIFO itself. OFLO is indicated on the receive frame that caused the overflow condition; complete frames in the Receive FIFO are not affected. While the Receive FIFO is in the overflow condition, it ignores additional receive data on the network. The internal address detect logic will continue to operate and the Missed Packet Count (MPC in register 24) will be incremented for each packet which passes the address match criteria, and complete without collision.
Bit 6	CLSN	Collision Flag. Indicates that the receive operation suffered a collision during reception of the frame. If CLSN is set, it indicates that the receive frame suffered a late collision, since a frame experiencing collision within the slot time will be automatically deleted from the RCVFIFO (providing LLRCV in the Receive Frame Control register is cleared). Note that if the LLRCV bit is enabled, the late collision threshold is effectively moved from the normal 64-byte (512-bit) level to the 12-byte (96-bit) level. Runt packets suffering a collision will be flushed from the RCVFIFO regardless of the state of the RPA bit (User Test Register). CLSN will not be set if OFLO is set.
Bit 5	FRAM	Framing Error flag. Indicates that the received frame contained a non-integer multiple of bytes and an FCS error. If there was no FCS error then FRAM will not be set. FRAM is not valid during internal loopback. FRAM will not be set if OFLO is set.
Bit 4	FCS	FCS Error flag. Indicates that there is an FCS error in the frame. The receive FCS is computed and checked normally when ASTRP RCV = 1, but is not

passed to the host. FCS will not be set if OFLO is set.

Bit	Name	Description
Bit 3–0	RCVCNT [11:8]	The Receive Message Byte Count indicates the number of whole bytes in the received message from the network. RCVCNT is 12 bits long, and valid (accurate) only when there are no errors reported in the Receive Status (RCVSTS). If a late collision is detected (CLSN set in RCVSTS), the count is an indication of the length (in byte times) of the duration of the receive activity including the collision. RCVCNT [7:0] correspond to bits 7–0 in RFS0 of the Receive Frame Status. RCVCNT [11–0} will be invalid when OFLO is set.

RFS2—Runt Packet Count (RNTPC)

RNTPC [7–0]

Bit	Name	Description
Bit 7–0	RNTPC [7–0]	The Runt Packet Count indicates the number of runt packets received, addressed to this node, since the last successfully received packet. The value does not roll over after 255 runt packets have been detected, and will remain frozen at the maximum count.

RFS3—Receive Collision Count (RCVCC)

RCVCC [7–0]

Bit	Name	Description
Bit 7–0	RCVCC [7–0]	The Receive Collision Count indicates the number of collisions detected on the network since the last successfully received packet. The value does not roll over after 255 collisions have been detected, and will remain frozen at the maximum count.

FIFO Frame Count (FIFOFC) (REG ADDR 7)

RCVFC[3–0]	XMTFC[3–0]

Bit	Name	Description
Bit 7–4	RCVFC [3–0]	Receive Frame Count. The (read only) count of the frames in the Receive FIFO. A frame is counted when the last byte is put in the FIFO. The counter is decremented when the last byte of the frame is read. If the RCVFC reaches its maximum value of 15, additional receive frames will be ignored, and the Missed Packet Count (MPC) register will be incremented for frames which match the internal address(es) of the MACE device.
Bit 3–0	XMTFC [3–0]	Transmit Frame Count. The (read only) count of the frames in the Transmit FIFO. A frame is counted when the last byte is put in the FIFO. The counter is decremented when XMTSV (in the Transmit Frame Status and Poll Register) is set and the Transmit Frame Status read access is performed.

Interrupt Register (IR) (REG ADDR 8)

All status bits are set upon occurrence of an event and cleared when read. The resister is read only. In addition all status bits are cleared by hardware or software reset. Bit assignments for the register are as follows:

JAB	BABL	CERR	RCVCCO	RNTPCO	MPCO	RCVINT	XMTINT

Bit	Name	Description
Bit 7	JAB	Jabber Error. JAB indicates that the MACE device attempted to transmit for an excessive time period (20–150 ms), when using either the DAI port or the 10BASE-T port. If the internal jabber timer expires during transmission, the transmit bit stream will be interrupted, until the internal transmission ceases and the *unjab* timer (0.5 s ±0.25 s) expires. The jabber function will be disabled, and JAB will not be set, regardless of transmission length, when either the AUI or GPSI ports have been selected. JAB is READ/CLEAR only, and is set by the MACE device and reset when read. Writing has no effect. It is also cleared by activation of the $\overline{\text{RESET}}$ pin or SWRST bit.
Bit 6	BABL	Babble Error. BABL is the transmitter time-out error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum packet. It will be set after 1519 bytes (or greater) have been transmitted. The MACE device will continue to transmit until the current packet transmission is over. The $\overline{\text{INTR}}$ pin will be acti-

vated if the corresponding mask bit BABLM = 0.

BABL is READ/CLEAR only, and is set by the MACE device and reset when read. Writing has no effect. It is also cleared by activation of the RESET pin or SWRST bit.

Bit 5 CERR — Collision Error. CERR indicates the absence of the Signal Quality Error Test (SQE Test) message after a packet transmission. The SQE Test message is a transceiver test feature. Detection depends on the MACE network interface selected. In all cases, CERR will be set if the MACE device failed to observe the SQE Test message within 20 network bit times after the packet transmission ended. When CERR is set, the INTR pin will be activated if the corresponding mask bit CERRM = 0.

When the AUI port is selected, the SQE Test message is returned over the CI± pair as a brief (5–15 bit times) burst of 10 MHz activity. When the 10BASE-T port is selected, CERR will be reported after a transmission only when the internal transceiver is in the link fail state (LNKST pin = HIGH). When the GPSI port is selected, the CLSN pin must be asserted by the external encoder/decoder to provide the SQE Test function. When the DAI port is selected, CERR will not be reported at any time.

CERR is READ/CLEAR only. It is set by the MACE and reset when read. Writing has no effect. It is also cleared by activation of the RESET pin or SWRST bit.

Bit 4 RCVCCO — Receive Collision Count Overflow. Indicates that the Receive Collision Count register rolled over at a value of 255 receive collisions. Receive collisions are defined as received frames which suffered a collision. The INTR pin will be activated if the corresponding mask bit RCVCCOM = 0. Note that the RCVCC value returned in the Receive Frame Status (RFS3) will freeze at a value of 255, whereas this register based version of RCVCC (REG ADDR 27) is free running.

RCVCCO is READ/CLEAR only. It is set by the MACE device and reset when read. Writing has no effect. It is also cleared by asserting the RESET pin or SWRST bit.

Bit 3 RNTPCO — Runt Packet Count Overflow. Indicates that the Runt Packet Count register rolled over at a value of 255 runt packets. Runt packets are defined as received frames which passed the internal address match criteria but did not contain a minimum of 64-bytes of data after SFD. The INTR pin will be activated if the corresponding mask bit RNTPCOM = 0. Note that the RNTPC value returned in the Receive Frame Status (RFS2) will freeze at a value of 255, whereas this register based version of RNTPC (REG ADDR 26) is free running.

RNTPCO is READ/CLEAR only. It is set by the MACE device and reset when read. Writing has no effect. It is also cleared by asserting the RESET pin or SWRST bit.

Bit 2 MPCO — Missed Packet Count Overflow. Indicates that the Missed Packet Count register rolled over at a value of 255 missed frames. Missed frames are defined as received frames which passed the internal address match criteria but were missed due to a Receive FIFO overflow, the receiver being disabled (ENRCV = 0) or an excessive receive frame count (RCVFC > 15). The INTR pin will be activated if the corresponding mask bit MPCOM = 0.

MPCO is READ/CLEAR only. It is set by the MACE device and reset when read. Writing has no effect. It is also cleared by asserting the RESET pin or SWRST bit.

Bit 1 RCVINT — Receive Interrupt. Indicates that the host read the last byte/word of a packet from the Receive FIFO. The Receive Frame Status is available immediately on the next host read operation. The INTR pin will be activated if the corresponding mask bit RCVINTM = 0.

RCVINT is READ/CLEAR only. It is set by the MACE device and reset when read. Writing has no effect. It is also cleared by activation of the RESET pin or SWRST bit.

Bit 0 XMTINT — Transmit Interrupt. Indicates that the MACE device has completed

the transmission of a packet and updated the Transmit Frame Status. The INTR pin will be activated if the corresponding mask bit XMTINTM = 0.

XMTINT is READ/CLEAR only. It is set by the MACE device and reset when read. Writing has no effect. It is also cleared by activation of the RESET pin or SWRST bit.

Interrupt Mask Register (IMR) (REG ADDR 9)

This register contains the mask bits for the interrupts. Read/write operations are permitted. Writing a one into a bit will mask the corresponding interrupt. Writing a zero to any previously set bit will unmask the corresponding interrupt. Bit assignments for the register are as follows:

RES	BABLM	CERRM	RCVCCOM	RNTPCOM	MPCOM	RCVINTM	XMTINTM

Bit	Name	Description
Bit 7	JABM	Jabber Error Mask. JABM is the mask for JAB. The INTR pin will not be asserted by the MACE device regardless of the state of the JAB bit, if JABM is set. It is cleared by activation of the RESET pin or SWRST bit.
Bit 6	BABLM	Babble Error Mask. BABLM is the mask for BABL. The INTR pin will not be asserted by the MACE device regardless of the state of the BABL bit, if BABLM is set. It is cleared by activation of the RESET pin or SWRST bit.
Bit 5	CERRM	Collision Error Mask. CERRM is the mask for CERR. The INTR pin will not be asserted by the MACE device regardless of the state of the CERR bit, if CERRM is set. It is cleared by activation of the RESET pin or SWRST bit.
Bit 4	RCVCCOM	Receive Collision Count Overflow Mask. RCVCCOM is the mask for RCVCCO(Receive Collision Count Overflow). The INTR pin will not be asserted by the MACE device regardless of the state of the RCVCCO bit, if RCVCCOM is set. It is cleared by activation of the RESET pin or SWRST bit.
Bit 3	RNTPCOM	Runt Packet Count Overflow Mask. RNTPCOM is the mask for RNTPCO (Runt Packet Count Overflow). The INTR pin will not be asserted by the MACE device regardless of the state of the RNTPCO bit, if RNTPCOM is set. It is cleared by activation of the RESET pin or SWRST bit.
Bit 2	MPCOM	Missed Packet Count Overflow Mask. MPCOM is the mask for MPCO (Missed Packet Count Overflow). The INTR pin will not be asserted by the MACE device regardless of the state of the MPCO bit, if MPCOM is set. It is cleared by activation of the RESET pin or SWRST bit.
Bit 1	RCVINTM	Receive Interrupt Mask. RCVINTM is the mask for RCVINT. The INTR pin will not be asserted by the MACE device regardless of the state of the RCVINT bit, if RCVINTM is set. It is cleared by activation of the RESET pin or SWRST bit.
Bit 0	XMTINTM	Transmit Interrupt Mask. XMTINTM is the mask for XMTINT. The INTR pin will not be asserted by the MACE device regardless of the state of the XMTINT bit, if XMTINT is set. It is cleared by activation of the RESET pin or SWRST bit.

Poll Register (PR) (REG ADDR 10)

This register contains copies of internal status bits to simplify a host implementation which is non-interrupt driven. The register is read only, and its status is unaffected by read operations. All register bits are cleared by hardware or software reset. Bit assignments are as follows:

XMTSV	TDTREQ	RDTREQ	RES	RES	RES	RES	RES

Bit	Name	Description
Bit 7	XMTSV	Transmit Status Valid. Transmit Status Valid indicates that the Transmit Frame Status is valid.
Bit 6	TDTREQ	Transmit Data Transfer Request. An internal indication of the current request status of the Transmit FIFO. TDTREQ is set when the external TDTREQ signal is asserted.

Bit 5	RDTREQ	Receive Data Transfer Request. An internal indication of the current request status of the Receive FIFO. RDTREQ is set when the external RDTREQ signal is asserted.
Bit 4–0	RES	Reserved. Read as zeroes. Always write as zeroes.

BIU Configuration Control (BIUCC) (REG ADDR 11)

All bits within the BIU Configuration Control register will be set to their default state upon a hardware or software reset. Bit assignments are as follows:

RES	BSWP	XMTSP [1–0]	RES	RES	RES	SWRST

Bit	Name	Description
Bit 7	RES	Reserved. Read as zero. Always write as zero.
Bit 6	BSWP	Byte Swap. The BSWP function allows data to and from the FIFOs to be orientated according to little endian or big endian byte ordering conventions. BSWP is cleared by by activation of the RESET pin or SWRST bit, defaulting to Intel byte ordering.
Bit 5-4	XMTSP [1–0]	Transmit Start Point. XMTSP controls the point preamble transmission commences in relation to the number of bytes written to the XMTFIFO. When the entire frame is in the XMTFIFO (or the XMTFIFO becomes full before the threshold is achieved), transmission of preamble will start regardless of the value in XMTSP (once the IPG time has expired). XMTSP is given a value of 10 (64 bytes) after hardware or software reset. Regardless of XMTSP, the FIFO will not internally over write its data until at least 64 bytes, or the entire frame, has been transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be re-written to the XMTFIFO, and re-tries will be handled autonomously by the MACE device.

Transmit Start Point

XMTSP [1–0]	Bytes
00	4
01	16
10	64
11	112

Bit 3-1	RES	Reserved. Read as zeroes. Always write as zeroes.
Bit 0	SWRST	Software Reset. When set, provides an equivalent of the hardware RESET pin function. All register bits will be set to their default values. The MACE device will require re-initialization after SWRST has been activated. The MACE device will clear SWRST during its internal reset sequence.

FIFO Configuration Control (REG ADDR 12) (FIFOCC)

All bits within the FIFO Configuration Control register will be set to their default state upon a hardware or software reset. Bit assignments are as follows:

XMTFW[1–0]	RCVFW [1–0]	XMTFWU	RCVFWU	XMTBRST	RCVBRST

Bit	Name	Description
Bit 7-6	XMTFW [1–0]	Transmit FIFO Watermark. XMTFW controls the point TDTREQ is asserted in relation to the number of write cycles to the Transmit FIFO. TDTREQ will be asserted at any time that the number of write cycles specified by XMTFW can be executed. XMTFW is set to a value of 00 (8 cycles) after hardware or software reset.

Transmit FIFO Watermarks

XMTFW [1–0]	Write Cycles
00	8
01	16
10	32
11	XX

The XMTFW value will only be updated when the XMTFWU bit is set.

To ensure that sufficient space is present in the XMTFIFO to accept the specified number of write cycles (including an End-Of-Frame delimiter), TDTREQ may go inactive before the XMTSP threshold is reached when using the non burst mode (XMTBRST = 0). The host must be aware that despite TDTREQ going inactive, additional space exists in the XMTFIFO, and the data write must continue to ensure the XMTSP threshold is achieved. No transmit activity will commence until the XMTSP

threshold is reached. When using the burst mode, TDTREQ will not be de-asserted until only a single write cycle can be performed. See the FIFO Sub-system section for additional details.

Bit 5-4 RCVFW [1–0] Receive FIFO Watermark. RCVFW controls the point RDTREQ is asserted in relation to the number of bytes available in the RCVFIFO. RCVFW specifies the number of bytes which must be present (once the packet has been verified as a non-runt), before the RDTREQ is asserted. Note however that in order for RDTREQ to be activated for a new frame, at least 64-bytes must have been received. This effectively avoids reacting to receive frames which are runts or suffer a collision during the slot time (512 bit times). If the Runt Packet Accept feature (RPA in Receive Frame Control) is enabled, the RDTREQ pin will be activated as soon as either 64-bytes are received, or a complete valid receive frame is detected (regardless of length). RCVFW is set to a value of 10 (64 bytes) after hardware or software reset.

Receive FIFO Watermarks

RCVFW [1–0]	Bytes
00	16
01	32
10	64
11	XX

The RCVFW value will only be updated when the RCVFWU bit is set.

Bit 3 XMTFWU Transmit FIFO Watermark Update. Allows update of the Transmit FIFO Watermark bits. The XMTFW can be written at any point, and will be read back as written. However, the new value in the XMTFW bits will be ignored until XMTFWU is set (or the transmit path is reset due to a retry failure). The recommended procedure to change the XMTFW is to write the new value with XMTFWU set, in a single write cycle. The XMTFIFO should be empty and all transmit activity complete before attempting a watermark update, since the XMTFIFO will be reset to allow the new pointer values to be loaded. It is recommended that the transmitter be disabled by clearing the ENXMT bit. XMTFWU will be cleared by the MACE device after the new XMTFW value has been loaded, or by activation of the RESET pin or SWRST bit.

Bit 2 RCVFWU Receive FIFO Watermark Update. Allows update of the Receive FIFO Watermark bits. The RCVFW bits can be written at any point, and will read back as written. However, the new value in the RCVFW bits will be ignored until RCVFWU is set. The recommended procedure to change the RCVFW is to write the new value with RCVFWU set, in a single write cycle. The RCVFIFO should be empty before attempting a watermark update, since the RCVFIFO will be reset to allow the new pointer values to be loaded. It is recommended that the receiver be disabled by clearing the ENRCV bit. RCVFWU will be cleared by the MACE device after the new RCVFW value has been loaded, or by activation of the RESET pin or SWRST bit.

Bit 1 XMTBRST Transmit Burst. When set, the transmit burst mode is selected. The behavior of the Transmit FIFO high watermark, and hence the de-assertion of TDTREQ, will be modified. TDTREQ will be deasserted if there are only two bytes of space available in the XMTFIFO (so that a full word write can still occur) or if four bytes of space exist and the EOF pin is asserted by the host.

		TDTREQ will be asserted identically in both normal and burst modes, when there is sufficient space in the XMTFIFO to allow the specified number of write cycles to occur (programmed by the XMTFW bits). Cleared by activation of the RESET pin or SWRST bit.
Bit 0	RCVBRST	Receive Burst. When set, the receive burst mode is selected. The behavior of the Receive FIFO low watermark, and hence the de-assertion of RDTREQ, will be modified. RDTREQ will de-assert when there are only 2-bytes of data available in the RCVFIFO (so that a full word read can still occur). RDTREQ will be asserted identically in both normal and burst modes, when a minimum of 64-bytes have been received for a new frame (or a runt packet has been received and RPA is set). Once the 64-byte limit has been exceeded, RDTREQ will be asserted providing there is sufficient data in the RCVFIFO to exceed the threshold, as programmed by the RCVFW bits. Cleared by activation of the RESET pin or SWRST bit.

MAC Configuration Control (MACCC) (REG ADDR 13)

This register programs the transmit and receive operation and behavior of the internal MAC engine. All bits within the MAC Configuration Control register are cleared upon hardware or software reset. Bit assignments are as follows:

PROM	DXMT2PD	EMBA	RES	DRCVPA	DRCVBC	ENXMT	ENRCV

Bit	Name	Description
Bit 7	PROM	Promiscuous. When PROM is set all incoming frames are received regardless of the destination address. PROM is cleared by activation of the RESET pin or SWRST bit.
Bit 6	DXMT2PD	Disable Transmit Two Part Deferral. When set, disables the transmit two part deferral option. DXMT2PD is cleared by activation of the RESET pin or SWRST bit.
Bit 5	EMBA	Enable Modified Back-off Algorithm. When set, enables the modified backoff algorithm. EMBA is cleared by activation of the RESET pin or SWRST bit.
Bit 4	RES	Reserved. Read as zeroes. Always write as zeroes.
Bit 3	DRCVPA	Disable Receive Physical Address. When set, the physical address detection (Station or node ID) of the MACE device will be disabled. Packets addressed to the nodes individual physical address will not be recognized (although the packet may be accepted by the EADI mechanism). DRCVPA is cleared by activation of the RESET pin or SWRST bit.
Bit 2	DRCVBC	Disable Receive Broadcast. When set, disables the MACE device from responding to broadcast messages. Used for protocols that do not support broadcast addressing, except as a function of multicast. DRCVBC is cleared by activation of the RESET pin or SWRST bit (broadcast messages will be received).
Bit 1	ENXMT	Enable Transmit. Setting ENXMT = 1 enables transmission. With ENXMT = 0, no transmission will occur. If ENXMT is written as 0 during frame transmission, a packet transmission which is incomplete will have a guaranteed CRC violation appended before the internal Transmit FIFO is cleared. No subsequent attempts to load the FIFO should be made until ENXMT is set and TDTREQ is asserted. ENXMT is cleared by activation of the RESET pin or SWRST bit.
Bit 0	ENRCV	Enable Receive. Setting ENRCV = 1 enables reception of frames. With ENRCV = 0, no frames will be received from the network into the internal FIFO. When ENRCV is written as 0, any receive frame currently in progress will be completed (and valid data contained in the RCVFIFO can be read by the host) and the MACE device will enter the monitoring state for missed packets. Note that clearing the ENRCV bit disables the

assertion of RDTREQ. If ENRCV is cleared during receive activity and remains cleared for a long time and if the tail end of the receive frame currently in progress is longer than the amount of space available in the Receive FIFO, Receive FIFO overflow will occur. However, even with RDTREQ deasserted, if there is valid data in the Receive FIFO to be read, successful slave reads to the Receive FIFO can be executed (indicated by valid DTV). It is the host's responsibility to avoid the overflow situation. ENRCV is cleared by activation of the RESET pin or SWRST bit.

PLS Configuration Control (PLSCC) (REG ADDR 14)

All bits within the PLS Configuration Control register are cleared upon a hardware or software reset. Bit assignments are as follows:

RES	RES	RES	RES	XMTSEL	PORTSEL [1–0]	ENPLSIO

Bit	Name	Description
Bit 7–4	RES	Reserved. Read as zeroes. Always write as zeroes.
Bit 3	XMTSEL	Transmit Mode Select. XMTSEL provides control over the AUI DO+ and DO– operation while the MACE device is not transmitting. With XMTSEL = 0, DO+ and DO will be equal during transmit idle state, providing zero differential to operate transformer coupled loads. The turn off and return to zero delays are controlled internally. With XMTSEL = 1, DO+ is positive with respect to DO during the transmit idle state .
Bit 2–1	PORTSEL [1–0]	Port Select. PORTSEL is used to select between the AUI, 10BASE-T, DAI or GPSI ports of the MACE device. PORTSEL is cleared by hardware or software reset. PORTSEL will determine which of the interfaces is used during normal operation, or tested when utilizing the loopback options (LOOP [1–0]) in the User Test Register. Note that the PORTSEL [1–0] programming will be overridden if the ASEL bit in the PHY Configuration Control register is set.

PORTSEL Interface Definition

PORTSEL [1–0]	Active Interface	DXCVR Pin
00	AUI	LOW
01	10BASE-T	HIGH
10	DAI Port	HIGH
11	GPSI	LOW

Bit 0	ENPLSIO	Enable PLS I/O. ENPLSIO is used to enable the optional I/O functions from the PLS function. The following pins are affected by the ENPLSIO bit: RXCRS, RXDAT, TXEN, TXDAT+, TXDAT–, CLSN, STDCLK, SRDCLK and SRD. Note that if an external SIA is being utilized via the GPSI, PORTSEL [1–0] = 11 must be programmed before ENPLSIO is set, to avoid contention of clock, data and/or carrier indicator signals.

PHY Configuration Control (PHYCC) (REG ADDR 15)

All bits within the PHY Configuration Control register with the exception of LNKFL, are cleared by hardware or software reset. Bit assignments are as follows:

LNKFL	DLNKTST	REVPOL	DAPC	LRT	ASEL	RWAKE	AWAKE

Bit	Name	Description
Bit 7	LNKFL	Link Fail. Reports the link integrity of the 10BASE-T receiver. When the link test function is enabled (DLNKTST = 0), the absence of link beat pulses on the RXD± pair will cause the integrated 10BASE-T transceiver to go into the link fail state. In the link fail state, data transmission, data reception, data loopback and the collision detection functions are disabled, and remain disabled until valid data or >5 consecutive link pulses appear on the RXD± pair. During link fail, the LNKFL bit will be set and the LNKST pin should be externally pulled HIGH. When the link is identified as functional, the LNKFL bit will be cleared and the LNKST pin is driven LOW, which is capable of directly driving a Link OK LED. In order to interoperate with systems which do

not implement Link Test, this function can be disabled by setting the DLNKTST bit. With Link Test disabled (DLNKTST = 1), the data driver, receiver and loopback functions as well as collision detection remain enabled irrespective of the presence or absence of data or link pulses on the RXD± pair. The transmitter will continue to generate link beat pulses during periods of transmit data inactivity. Set by hardware or software reset.

Bit 6 DLNKTST Disable Link Test. When set, the integrated 10BASE-T transceiver will be forced into the link pass state, regardless of receive link test pulses or receive packet activity.

Bit 5 REVPOL Reversed Polarity. Indicates the receive polarity of the RD± pair. When normal polarity is detected, the REVPOL bit will be cleared, and the RXPOL pin (capable of driving a *Polarity OK* LED) will be driven LOW. When reverse polarity is detected, the REVPOL bit will be set, and the RXPOL pin should be externally pulled HIGH.

Bit 4 DAPC Disable Auto Polarity Correction. When set, the automatic polarity correction will be disabled. Polarity detection and indication will still be possible via the RXPOL pin.

Bit 3 LRT Low Receive Threshold. When set, the threshold of the twisted pair receiver will be reduced by 4.5 dB, to allow extended distance operation.

Bit 2 ASEL Auto Select. When set, the PORTSEL [1–0] bits are overridden, and the MACE device will automatically select the operating media interface port. When the 10BASE-T transceiver is in the link pass state (due to receiving valid packet data and/or Link Test pulses or the DLNKTST bit is set), the 10BASE-T port will be used. When the 10BASE-T port is in the link fail state, the AUI port will be used. Switching between the ports will not occur during transmission in order to avoid any type of fragment generation.

Bit 1 RWAKE Remote Wake. When set prior to the SLEEP pin being activated, the AUI and 10BASE-T receiver sections and the EADI port will continue to operate even during SLEEP. Incoming packet activity will be passed to the EADI port pins permitting detection of specific frame contents used to initiate a wake-up sequence. RWAKE must be programmed prior to SLEEP being asserted for this function to operate. RWAKE is not cleared by SLEEP, only by activation of the SWRST bit or RESET pin.

Bit 0 AWAKE Auto Wake. When set prior to the SLEEP pin being activated, the 10BASE-T receiver section will continue to operate even during SLEEP, and will activate the LNKST pin if Link Pass is detected. AWAKE must be programmed prior to SLEEP being asserted for this function to operate. AWAKE is not cleared by SLEEP, only by activation of the SWRST bit or RESET pin.

Chip Identification Register (CHIPID [15–00]) (REG ADDR 16 &17)

This 16-bit value corresponds to the specific version of the MACE device being used. The value will be programmed to X940h, where *X* is a value dependent on version.

CHIPID [07–00]
CHIPID [15–08]

Internal Address Configuration (IAC) (REG ADDR 18)

This register allows access to and from the multi-byte Physical Address and Logical Address Filter locations, using only a single byte location.

The MACE device will reset the IAC register PHYADDR and LOGADDR bits after the appropriate number of read or write cycles have been executed on the Physical Address Register or the Logical Address Filter. Once the LOGADDR bit is set, the MACE device will reset the bit after 8 read or write operations have been performed. Once the PHYADDR bit is set, the MACE device will reset the bit after 6 read or write operations have been performed. The MACE device makes no distinction between read or write operations, advancing the internal address RAM pointer with each access. If both PHYADDR and LOGADDR bits are set, the MACE device will accept only the LOGADDR bit. If the PHYADDR bit is set and the Logical Address Filter location is accessed, a DTV will not be returned. Similarly, if the LOGADDR bit is set and the Physical Address Register location is accessed, DTV will not be returned. PHYADDR or LOGADDR can be set in the same cycle as ADDRCHG.

ADDRCHG	RES	RES	RES	RES	PHYADDR	LOGADDR	RES

Bit	Name	Description
Bit 7	ADDRCHG	Address Change. When set, allows the physical and/or logical address to be read or programmed. When ADDRCHG is set, ENRCV will be cleared, the MPC will be stopped, and the last or current in progress receive frame will be received as normal. After the frame completes, access to the internal address RAM will be permitted, indicated by the MACE device clearing the ADDRCHG bit. Please refer to the register description of the ENRCV bit in the MAC Configuration Control register (REG ADDR 13) for the effect of clearing the ENRCV bit. Normal reception can be resumed once the physical/logical address has been changed, by setting ENRCV.
Bit 6–3	RES	Reserved. Read as zeroes. Always write as zeroes.
Bit 2	PHYADDR	Physical Address Reset. When set, successive reads or writes to the Physical Address Register will occur in the order PADR [07–00], PADR [15–08],...., PADR [47–40]. Each read or write operation on the PADR location will auto-increment the internal pointer to access the next most significant byte.
Bit 1	LOGADDR	Logical Address Reset. When set, successive reads or writes to the Logical Address Filter will occur in the order LADRF [07–00], LADRF [15–08],....,LADRF [63–56]. Each read or write operation on the LADRF location will auto-increment the internal pointer to access the next most significant byte.
Bit 0	RES	Reserved. Read as zero. Always write as zero.

Logical Address Filter (LADRF [63–00]) **(REG ADDR 20)**

LADRF [63–00]

This 64-bit mask is used to accept incoming Logical Addresses. The Logical Address Filter is expected to be programmed at initialization (after hardware or software reset). After a hardware or software reset and before the ENRCV bit in the MAC Configuration Control register has been set, the Logical Address can be accessed by setting the LOG ADDR bit in the Internal Address Configuration register (REG ADDR 18) and then by performing 8 reads or writes to the Logical Address Filter. Once ENRCV has been set, the ADDR CHG bit in the Internal Address Configuration register must be set and be polled until it is cleared by the MACE device before setting the LOGADDR bit and before accessing of the Logical Address Filter is allowed.

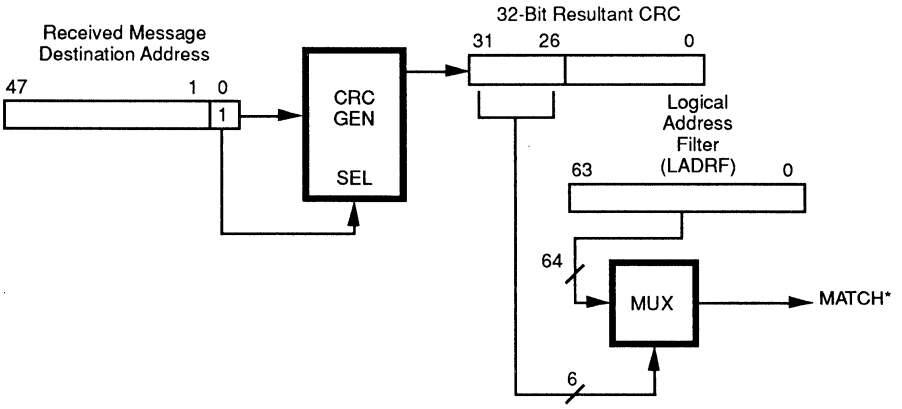
If the least significant address bit of a received message is set (Destination Address bit 00 = 1), then the address is deemed logical, and passed through the FCS generator. After processing the 48-bit destination address, a 32-bit resultant FCS is produced and strobed into an internal register. The high order 6-bits of this resultant FCS are used to select one of the 64-bit positions in the Logical Address Filter (see diagram). If the selected filter bit is a *1*, the address is accepted and the packet will be placed in memory.

The first bit of the incoming address must be a *1* for a logical address. If the first bit is a *0*, it is a physical address and is compared against the value stored in the Physical Address Register at initialization.

The Logical Address Filter is used in multicast addressing schemes. The acceptance of the incoming frame based on the filter value indicates that the message may be intended for the node. It is the user's responsibility to determine if the message is actually intended for the node by comparing the destination address of the stored message with a list of acceptable logical addresses.

The Broadcast address, which is all ones, does not go through the Logical Address Filter and is always enabled providing that the Disable Receive Broadcast bit (DRCVBC in the MAC Configuration Control register) is cleared. If the Logical Address Filter is loaded with all zeroes (and PROM = 0), all incoming logical addresses except broadcast will be rejected.

Multicast addressing can only be performed when using external loopback (LOOP [1–0] = 0) by programming RCVFCSE = 1 in the User Test Register. The FCS logic is internally allocated to the receiver section, allowing the FCS to be computed on the incoming logical address.



Logical Address Match Logic

Physical Address (PADR [47–00]) **(REG ADDR 21)**

PADR [47–00]

This 48-bit value represents the unique node value assigned by the IEEE and used for internal address comparison. After a hardware or software reset and before the ENRCV bit in the MAC Configuration Control register has been set, the Physical Address can be accessed by setting the PHYADDR bit in the Internal Address Configuration register (REG ADDR 18) and then by performing 6 reads or writes to the Physical Address. Once ENRCV has been set, the ADDRCHG bit in the Internal Address Configuration register must be set and be polled until it is cleared by the MACE device before setting the PHYADDR bit and before accessing of the Physical Address is allowed. The first bit of the incoming address must be a 0 for a physical address. The incoming address is compared against the value stored in the Physical Address register at initialization provided that the DRCVPA bit in the MAC Configuration Control register is cleared.

Missed Packet Count (MPC) **(REG ADDR 24)**

MPC [7–0]

The Missed Packet Count (MPC) is a read only 8-bit counter. The MPC is incremented when the receiver is unable to respond to a packet which would have normally been passed to the host. The MPC will be reset to zero when read. The MACE device will be *deaf* to receive traffic due to any of the following conditions :

- The host disabled the receive function by clearing the ENRCV bit in the MAC Configuration Control register.
- A Receive FIFO overflow condition exists, and must be cleared by reading the Receive FIFO and the Receive Frame Status.
- The Receive Frame Count (RCVFC) in the FIFO Frame Count register exceeds its maximum value, indicating that greater than 15 frames are in the Receive FIFO.

If the number of received frames that have been missed exceeds 255, the MPC will roll over and continue counting from zero, the MPCO (Missed Packet Count Overflow) bit in the Interrupt Register will be set (at the value 255), and the $\overline{\text{INTR}}$ pin will be asserted providing that MPCOM (Missed Packet Count Overflow Mask) in the Interrupt Mask Register is clear. MPCOM will be cleared (the interrupt will be unmasked) after a hardware or software reset.

Note that the following conditions apply to the MPC:

- After hardware or software reset, the MPC will not increment until the first time the receiver is enabled (ENRCV = 1). Once the receiver has been enabled, the MPC will count all missed packet events, regardless of the programming of ENRCV.
- The packet must pass the internal address match to be counted. Any of the following address match conditions will increment MPC while the receiver is *deaf*:
  Physical Address match;
  Logical Address match;
  Broadcast reception;
  Any receive in promiscuous mode (PROM = 1 in the MAC Configuration Control register);
  EADI feature match mode and $\overline{\text{EAM}}$ is asserted;
  EADI feature reject mode and $\overline{\text{EAR}}$ is not asserted.
- Any packet which suffers a collision within the slot time will not be counted.
- Runt packets will not be counted unless RPA in the User Test Register is enabled.
- Packets which pass the address match criteria but experience FCS or Framing errors will be counted, since they are normally passed to the host.

Runt Packet Count (RNTPC) **(REG ADDR 26)**

RNTPC [7–0]

The Runt Packet Count (RNTPC) is a read only 8-bit counter, incremented when the receiver detects a runt packet that is addressed to this node. Runt packets are defined as received frames which passed the internal address match criteria but did not contain a minimum of 64-bytes of data after SFD. Note that the RNTPC value returned in the Receive Frame Status (RFS2) will freeze at a value of 255, whereas this register based version of RNTPC is free running. The value will roll over after 255 runt packets have been detected, setting the RNTPCO bit (in the Interrupt Register and asserting the $\overline{\text{INTR}}$ pin if the corresponding mask bit (RNTPCOM in the Interrupt Mask Register) is cleared. RNTPC will be reset to zero when read.

Receive Collision Count (RCVCC) (REG ADDR 27)

RCVCC [7–0]

The Receive Collision Count (RCVCC) is a read only 8-bit counter, incremented when the receiver detects a collision on the network. Note that the RCVCC value returned in the Receive Frame Status (RFS3) will freeze at a value of 255, whereas this register based version of RCVCC is free running. The value will roll over after 255 receive collisions have been detected, setting the RCVCCO bit (in the Interrupt Register and asserting the $\overline{\text{INTR}}$ pin if the corresponding mask bit (RCVCCOM in the Interrupt Mask Register) is cleared. RCVCC will be reset to zero when read.

User Test Register (UTR) (REG ADDR 29)

The User Test Register is used to put the chip into test configurations. All bits within the Test Register are cleared upon a hardware or software reset. Bit assignments are as follows:

RTRE	RTRD	RPA	FCOLL	RCVFCSE	LOOP [1–0]	RES

Bit	Name	Description
Bit 7	RTRE	Reserved Test Register Enable. Access to the Reserved Test Registers should not be attempted by the user. **Note that access to the Reserved Test Register may cause damage to the MACE device if configured in a system board application.** Access to the Reserved Test Register is prevented, regardless of the state of RTRE, once RTRD has been set. RTRE is cleared by activation of the $\overline{\text{RESET}}$ pin or SWRST bit.
Bit 6	RTRD	Reserved Test Register Disable. When set, access to the Reserved Test Registers is inhibited, and further writes to the RTRD bit are ignored. Access to the Reserved Test Register is prevented, regardless of the state of RTRE, once RTRD has been set. RTRD can only be cleared by hardware or software reset.
Bit 5	RPA	Runt Packet Accept. Allows receive packets which are less than the legal minimum as specified by IEEE 802.3/Ethernet, to be passed to the host interface via the Receive FIFO. The receive packets must be at least 8 bytes (after SFD) in length to be accepted. RPA is cleared by activation of the $\overline{\text{RESET}}$ pin or SWRST bit.
Bit 4	FCOLL	Force Collision. Allows the collision logic to be tested. The MACE device should be in an internal loopback test for the FCOLL test. When FCOLL = 1, a collision will be forced during the next transmission attempt. This will result in 16 total transmission attempts (if DRTRY = 0) with the Retry Error reported in the Transmit Frame Status register. FCOLL is cleared by the activation of the $\overline{\text{RESET}}$ pin or SWRST bit.
Bit 3	RCVFCSE	Receive FCS Enable. Allows the hardware associated with the FCS generation to be allocated to the transmitter or receiver during loopback diagnostics. When clear, the FCS will be generated and appended to the transmit message (providing that DXMTFCS in the Transmit Frame Control is clear), and received after the loopback process through the Receive FIFO. When set, the hardware associated with the FCS generation is allocated to the receiver. A transmit packet will be assumed to contain the FCS in the last four bytes of the frame passed through the Transmit FIFO. The received frame will have the FCS calculated on the data field and compared with the last four bytes contained in the received message. An FCS error will be flagged in the Received Status (RFS1) if the received and calculated values do not match. RCVFCSE is only valid when in any one of the loopback modes as defined by LOOP [0–1]. Note that if the receive frame is expected to be recognized on the basis of a multicast address match, the FCS logic must be allocated to the receiver (RCVFCSE = 1). RCVFCSE is cleared by activation of the $\overline{\text{RESET}}$ pin or SWRST bit.
Bit 2–1	LOOP [1–0]	Loopback Control. The loopback functions allow the MACE device to receive its own transmitted frames. Three levels of loopback are provided as shown in the following table. During loopback operation a multicast address can only be recognized if RCVFCSE = 1. LOOP [0–1] are cleared by activation of the $\overline{\text{RESET}}$ pin or SWRST bit.

Loopback Functions

Loop [1–0]	Function
00	No Loopback
01	External Loopback
10	Internal Loopback, excludes MENDEC
11	Internal Loopback, includes MENDEC

External loopback allow the MACE device to transmit to the physical medium, using either the AUI, 10BASE-T, DAI or GPSI port, dependent on the PORTSEL [1–0] bits in the PLS Configuration Control register. Using the internal loopback test will ensure that transmission does not disturb the physical medium and will prohibit frame reception from the network. One Internal loopback function includes the MENDEC in the loop.

Bit 0 RES Reserved. Read as zero. Always write as zero.

Reserved Test Register 1 (RTR1) (REG ADDR 30)

Reserved for AMD internal use only.

Reserved Test Register 2 (RTR2) (REG ADDR 31)

Reserved for AMD internal use only.

Register Table Summary

Address	Mnemonic	Contents	Comments
0	RCVFIFO	Receive FIFO [15–00]	Read only
1	XMTFIFO	Transmit FIFO [15–00]	Write only
2	XMTFC	Transmit Frame Control	Read/Write
3	XMTFS	Transmit Frame Status	Read only
4	XMTRC	Transmit Retry Count	Read only
5	RCVFC	Receive Frame Control	Read/Write
6	RCVFS	Receive Frame Status (4-bytes)	Read only
7	FIFOFC	FIFO Frame Count	Read only
8	IR	Interrupt Register	Read only
9	IMR	Interrupt Mask Register	Read/Write
10	PR	Poll Register	Read only
11	BIUCC	BIU Configuration Control	Read/Write
12	FIFOCC	FIFO Configuration Control	Read/Write
13	MACCC	MAC Configuration Control	Read/Write
14	PLSCC	PLS Configuration Control	Read/Write
15	PHYCC	PHY Configuration Control	Read/Write
16	CHIPID	Chip Identification Register [07–00]	Read only
17	CHIPID	Chip Identification Register [15–08]	Read only
18	IAC	Internal Address Configuration	Read/Write
19		Reserved	Read/Write as 0
20	LADRF	Logical Address Filter (8-bytes)	Read/Write
21	PADR	Physical Address (6-bytes)	Read/Write
22		Reserved	Read/Write as 0
23		Reserved	Read/Write as 0
24	MPC	Missed Packet Count	Read only
25		Reserved	Read/Write as 0
26	RNTPC	Runt Packet Count	Read only
27	RCVCC	Receive Collision Count	Read only
28		Reserved	Read/Write as 0
29	UTR	User Test Register	Read/Write
30	RTR1	Reserved Test Register 1	Read/Write as 0
31	RTR2	Reserved Test Register 2	Read/Write as 0

Register Bit Summary

16-Bit Registers

0	RCVFIFO [15–0]
1	XMTFIFO [15–0]

8-Bit Registers

Address	Mnemonic							
2	DRTRY	RES	RES	RES	DXMTFCS	RES	RES	APADXMT
3	XMTSV	UFLO	LCOL	MORE	ONE	DEFER	LCAR	RTRY
4	EXDEF	RES	RES	RES	XMTRC [3–0]			
5	RES	RES	RES	RES	LLRCV	$\overline{M/R}$	RES	ASTRPRCV
6	RCVFS [31–00]							
7	RCVFC [3–0]				XMTFC [3–0]			
8	JAB	BABL	CERR	RCVCCO	RNTPCO	MPCO	RCVINT	XMTINT
9	JABM	BABLM	CERRM	RCVCCOM	RNTPCOM	MPCOM	RCVINTM	XMTINTM
10	XMTSV	TDTREQ	RDTREQ	RES	RES	RES	RES	RES
11	RES	BSWP	XMTSP [1–0]		RES	RES	RES	SWRST
12	XMTFW [1–0]		RCVFW [1–0]		XMTFWU	RCVFWU	XMTBRST	RCVBRST
13	PROM	DXMT2PD	EMBA	RES	DRCVPA	DRCVBC	ENXMT	ENRCV
14	RES	RES	RES	RES	XMTSEL	PORTSEL [1–0]		ENPLSIO
15	LNKFL	DLNKTST	REVPOL	DAPC	LRT	ASEL	RWAKE	AWAKE
16	CHIPID [07–00]							
17	CHIPID [15–08]							
18	ADDRCHG	RES	RES	RES	RES	PHYADDR	LOGADDR	RES
19	RESERVED							
20	LADRF [63–00]							
21	PADR [47–00]							
22	RESERVED							
23	RESERVED							
24	MPC [7–0]							
25	RESERVED							
26	RNTPC [7–0]							
27	RCVCC [7–0]							
28	RESERVED							
29	RTRE	RTRD	RPA	FCOLL	RCVFCSE	LOOP [1–0]		RES
30	RESERVED							
31	RESERVED							

Receive Frame Status

Address	Mnemonic				
RFS0	RCVCNT [7:0]				
RFS1	OFLO	CLSN	FRAM	FCS	RCVCNT [10:8]
RFS2	RNTPC [7–0]				
RFS3	RCVCC [7–0]				

Programmer's Register Model

Addr	Mnemonic	Contents			R/W
0	RCVFIFO	Receive FIFO—16 bits			RO
1	XMTFIFO	Transmit FIFO—16 bits			WO
2	XMTFC	Transmit Frame Control			R/W
		80	DRTRY	Disable Retry	
		08	DXMTFCS	Disable Transmit FCS	
		01	APADXMT	Auto Pad Transmit	
3	XMTFS	Transmit Frame Status			RO
		80	XMTSV	Transmit Status Valid	
		40	UFLO	Underflow	
		20	LCOL	Late Collision	
		10	MORE	MORE than one retry was needed	
		08	ONE	Exactly ONE retry occurred	
		04	DEFER	Transmission was deferred	
		02	LCAR	Loss of Carrier	
		01	RTRY	Transmit aborted after 16 attempts	
4	XMTRC	80	EXDEF	Excessive Defer	RO
		40	—		
		20	—		
		10	—		
		0F	XMTRC [3:0]	4-bit Transmit Retry Count	
5	RCVFC	Receive Frame Control			R/W
		08	LLRCV	Low Latency Receive	
		04	M/$\overline{R}$	Match/$\overline{\text{Reject}}$ for external address detection	
		01	ASTRPRCV	Auto Strip Receive—Strips pad and FCS from received frames	
6	RCVFS	Receive Frame Status—4 bytes—read in 4 read cycles			RO
		RFS0	RCVCNT [7:0]	Receive Message Byte Count	
		RFS1	RCVSTS, RCVCNT [11:8]—Receive Status & Receive Msg Byte Count MSBs		
			80	OFLO	Receive FIFO Overflow
			40	CLSN	Collision during reception
			20	FRAM	Framing Error
			10	FCS	FCS (CRC) error
			0F	RCVCNT [11:8]	4 MSBs of Receive Msg. Byte Count
		RFS2	RNTPC [7:0]	Runt Packet Count (since last successful reception)	
		RFS3	RCVCC [7:0]	Receive Collision Count (since last successful reception)	
7	FIFOFC	FIFO Frame Count			RO
		F0	RCVFC	Receive Frame Count—# of RCV frames in FIFO	
		0F	XMTFC	Transmit Frame Count—# of XMT frames in FIFO	RO
8	IR	Interrupt Register			RO
		80	JAB	Jabber Error—Excessive transmit duration (20–150ms)	
		40	BABL	Babble Error→1518 bytes transmitted	
		20	CERR	Collision Error—No SQE Test Message	
		10	RCVCCO	Receive Collision Count Overflow—Reg Addr 27 overflow	
		08	RNTPCO	Runt Packet Count Overflow—Reg Addr 26 overflow	
		04	MPCO	Missed Packet Count Overflow—Reg Addr 24 overflow	
		02	RCVINT	Receive Interrupt—Host has read last byte of packet	
		01	XMTINT	Transmit Interrupt—Transmission is complete	

Programmer's Register Model (continued)

Addr	Mnemonic	Contents			R/W
9	IMR	Interrupt Mask Register			
		80	JABM	Jabber Error Mask	
		40	BABLM	Babble Error Mask	
		20	CERRM	Collision Error Mask	
		10	RCVCCOM	Receive Collision Count Overflow Mask	R/W
		08	RNTPCOM	Runt Packet Count Overflow Mask	
		04	MPCOM	Missed Packet Count Overflow Mask	
		02	RCVINTM	Receive Interrupt Mask	
		01	XMTINTM	Transmit Interrupt Mask	
10	PR	Poll Register			
		80	XMTSV	Transmit Status Valid	RO
		40	TDTREQ	Transmit Data Transfer Request	
		20	RDTREQ	Receive Data Transfer Request	
11	BIUCC	Bus Interface Unit Configuration Control			
		80	—		
		40	BSWP	Byte Swap	
		30	XMTSP—Transmit Start Point (2 bits)		
			00	Transmit after 4 bytes have been loaded	R/W
			01	Transmit after 16 bytes have been loaded	
			10	Transmit after 64 bytes have been loaded	
			11	Transmit after 112 bytes have been loaded	
		01	SWRST	Software Reset	
12	FIFOCC	FIFO Configuration Control			
		C0	XMTFW	Transmit FIFO Watermark (2 bits)	
			00	Assert TDTREQ after 8 write cycles can be made	
			01	Assert TDTREQ after 16 write cycles can be made	
			10	Assert TDTREQ after 32 write cycles can be made	
			11	XX	
		30	RCVFW	Receive FIFO Watermark (2 bits)	
			00	Assert RDTREQ after 16 bytes are present	R/W
			01	Assert RDTREQ after 32 bytes are present	
			10	Assert RDTREQ after 64 bytes are present	
			11	XX	
		08	XMTFWU	Transmit FIFO Watermark Update—loads XMTFW bits	
		04	RCVFWU	Receive FIFO Watermark Update—loads RCVFW bits	
		02	XMTBRST	Select Transmit Burst mode	
		01	RCVBRST	Select Receive Burst mode	
13	MACCC	Media Access Control (MAC) Configuration Control			
		80	PROM	Promiscuous mode	
		40	DXMT2PD	Disable Transmit Two Part Deferral	
		20	EMBA	Enable Modified Back-off Algorithm	
		10	—		R/W
		08	DRCVPA	Disable Receive Physical Address	
		04	DRCVBC	Disable Receive Broadcast	
		02	ENXMT	Enable Transmit	
		01	ENRCV	Enable Receive	

Programmer's Register Model (continued)

Addr	Mnemonic	Contents	R/W
14	PLSCC	Physical Layer Signalling (PLS) Configuration Control 08 XMTSEL Transmit Mode Select: 1→DO± = 1 during IDLE 06 PORTSEL [1:0]—Port Select (2 bits) 00 AUI selected 01 10BASE-T selected 10 DAI port selected 11 GPSI selected 01 ENPLSIO Enable Status	R/W
15	PHYCC	Physical Layer (PHY) Configuration Control 80 LNKFL Link Fail—Reports 10BASE-T receive inactivity 40 DLNKTST Disable Link Test—Force 10BASE-T port into Link Pass 20 REVPOL Reversed Polarity—Reports 10BASE-T receiver wiring error 10 DAPC Disable Auto Polarity Correction—Detection remains active 08 LRT Low Receive Threshold—Extended distance capability 04 ASEL Auto Select—Select 10BASE-T port when active, otherwise AUI 02 RWAKE Remote Wake—10BASE-T, AUI and EADI features active during sleep 01 AWAKE Auto Wake—10BASE-T receive and $\overline{\text{LNKST}}$ active during sleep	R/W
16	CHIPID	Chip Identification Register LSB—CHIPID [7:0]	RO
17	CHIPID	Chip Identification Register MSB—CHIPID [15:8]	RO
18	IAC	Internal Address Configuration 80 ADDRCHG Address Change—Write to PHYADDR or LOGADDR after ENRCV 40 — 20 — 10 — 08 — 04 — 04 PHYADDR Reset Physical Address pointer 02 LOGADDR Reset Logical Address pointer 01 —	R/W
19	—	Reserved	R/W as 0
20	LADRF	Logical Address Filter—8 bytes—8 reads or writes—LS Byte first	R/W
21	PADR	Physical 6 bytes—6 reads or writes—LS Byte first	R/W
22	—	Reserved	R/W as 0
23	—	Reserved	R/W as 0
24	MPC	Missed Packet Counter—Number of receive packets missed	RO
25	—	Reserved	R/W as 0
26	RNTPC	Runt Packet Count—Number of runt packets addressed to this node	RO
27	RCVCC	Receive Collision Count—Number of receive collision frames on network	RO
28	—	Reserved	R/W as 0
29	UTR	User Test Register 80 RTRE Reserved Test Register Enable—*must be 0* 40 RTRD Reserved Test Register Disable 20 RPA Runt Packet Accept 10 FCOLL Force Collision 08 RCVFCSE Receive FCS Enable 06 LOOP Loopback control (2 bits) 00 No loopback 01 External loopback 10 Internal loopback, excludes MENDEC 11 Internal loopback, includes MENDEC 01 —	R/W

Programmer's Register Model (continued)

Addr	Mnemonic	Contents	R/W
30	—	Reserved	R/W as 0
31	—	Reserved	R/W as 0

SYSTEM APPLICATIONS

Host System Examples

Motherboard DMA Controller

The block diagram shows the MACE device interfacing to a 8237 type DMA controller. Two external latches are used to provide a 24 bit address capability. The first latch stores the address bits A[15:8], which the 8237 will output on the data line DB[7:0], while the signal ADSTB is active. The second latch is used as a page register. It extends the addressing capability of the 8237 from 16-bit to 24-bit. This latch must be programmed by the system using an I/0 command to generate the signal LATCHHIGHADR.

The MACE device uses two of the four DMA channels. One is dedicated to fill the Transmit FIFO and the other to empty the Receive FIFO. Both DMA channels should be programmed in the following mode:

Command Register:
- Memory to memory disabled
- DREQ sense active high
- DACK sense active low
- Normal timing
- Late Write

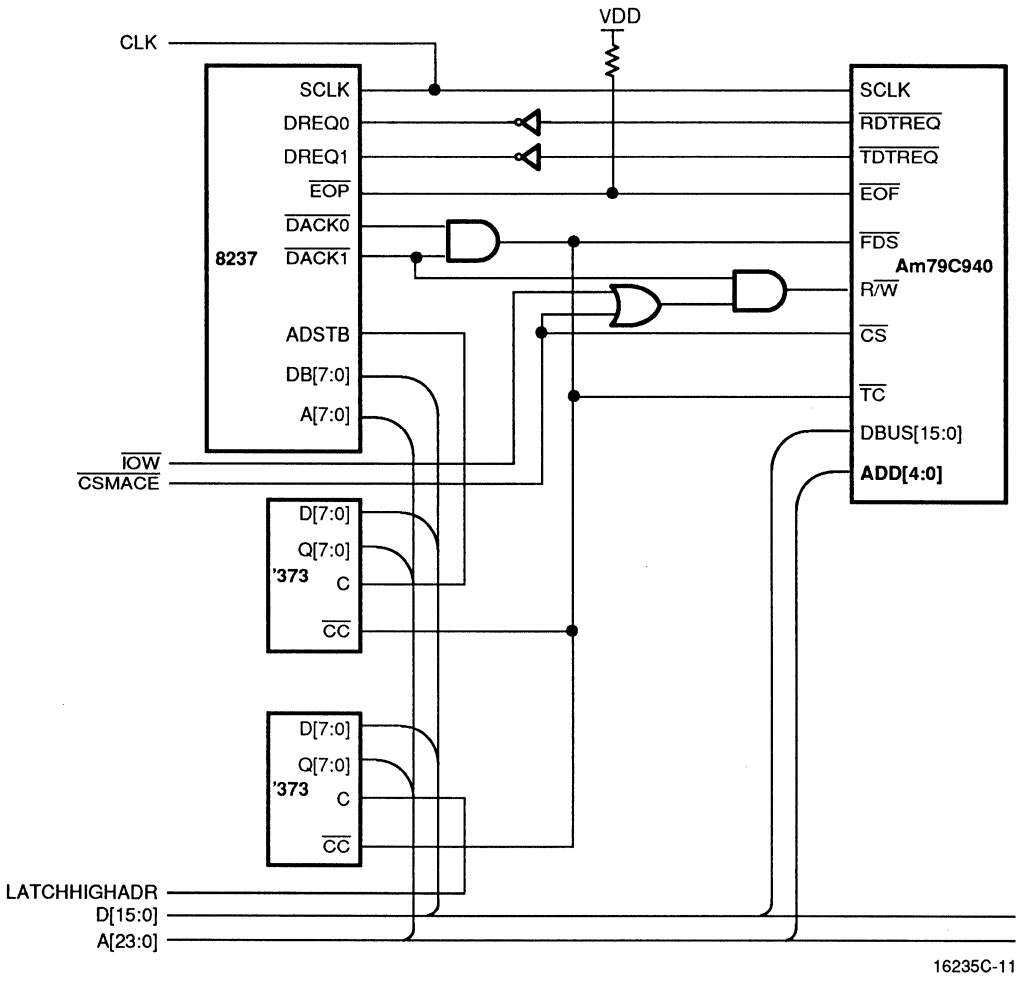
Note:

This is the same configuration as used in the IBM PC.

The 8237 and the MACE device run synchronous to the same SCLK. The 8237 is programmed to execute a transfer in three clock cycles This requires an extra wait state in the MACE device during FIFO accesses. A system not using the same configuration as in the IBM PC can minimize the bus bandwidth required by the MACE device by programming the DMA controller in the compressed timing mode.

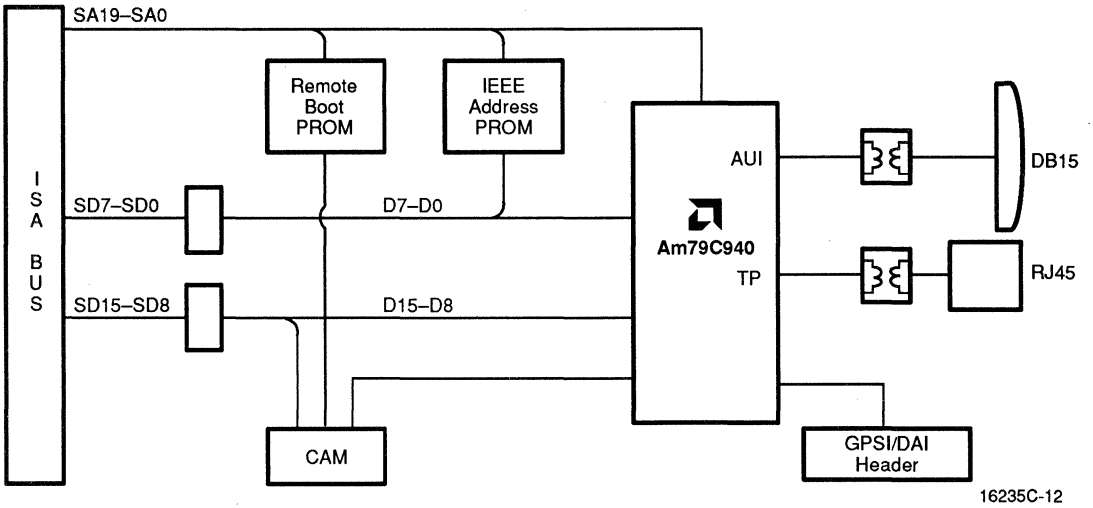
Care must be taken with respect to the number of transfers within a burst. The 8237 will drive the signal $\overline{EOP}$ low every time the internal counter reaches the zero. The MACE device however only expects $\overline{EOF}$ asserted on the last byte/word of a packet. This means, that the word counter of the 8237 should be initially loaded with the number of bytes/words in the whole packet. If the application requires that the packet will be constructed from several buffers at transmit time, some extra logic is required to suppress the assertion of $\overline{EOF}$ at the end of all but the last buffer transferred by the DMA controller. Also note that the DMA controller can only handle either bytes or words at any time. It requires special handling if a packet is transferred to the MACE device Transmit FIFO in word quantities and it ends in an odd byte.

The 8237 requires an extra clock cycle to update the external address latch every 256 transfer cycles. This example assumes that an update of the external address latch occurs only at the beginning of the block transfer.



System Interface – Motherboard DMA Example

PC/AT Ethernet Adapter Card

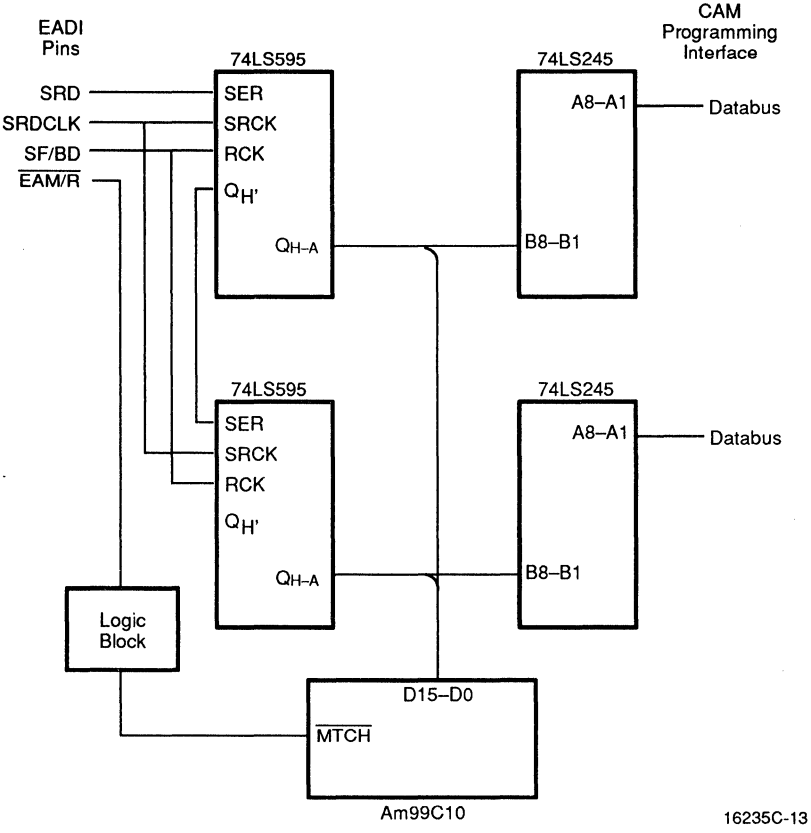


System Interface – Simple PC/AT Ethernet Adapter Card Example

NETWORK INTERFACES

External Address Detection Interface (EADI)

The External Address Detection Interface can be used to implement alternative address recognition schemes outside the MACE device, to complement the physical, logical and promiscuous detection supported internally. The address matching, and the support logic necessary to capture and present the relevant data to the external table of address is application specific. Note that since the entire 802.3 packet after SFD is made available, recognition is not limited to the destination address and/or type fields (Ethernet only). Inter-networking protocol recognition can be performed on specific header or LLC information fields.

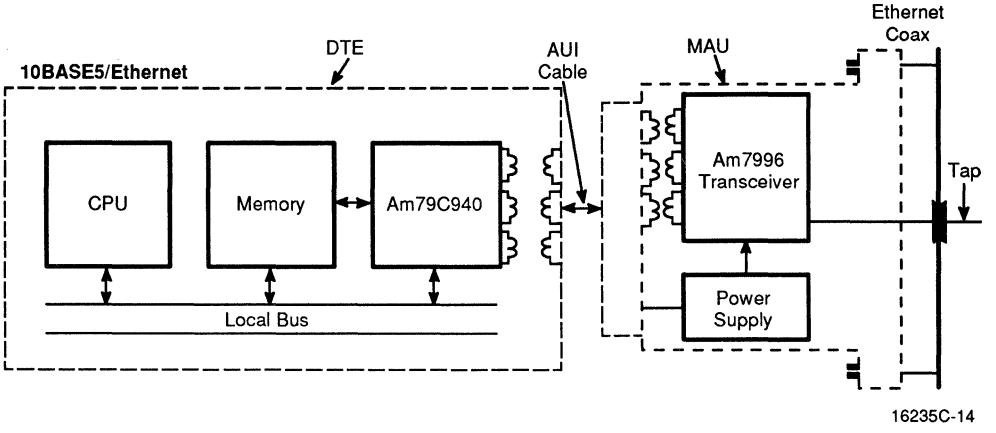


EADI Feature – Simple External CAM Interface

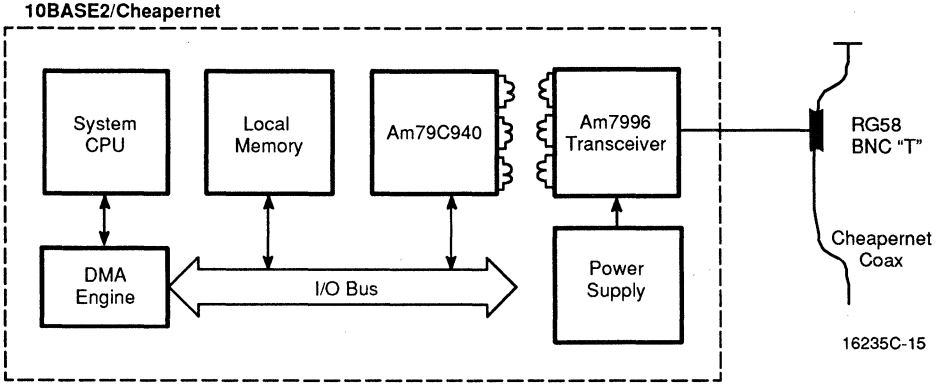
Attachment Unit Interface (AUI)

The AUI can drive up to 50 m of standard drop cable to allow the transceiver to be remotely located, as is typically the case in IEEE 803.3 10BASE5 or thick Ethernet® installations. For a locally mounted transceiver, such as 802.3 10BASE2 or Cheapernet interface, the isolation transformer requirements between the transceiver and the MACE device can be reduced.

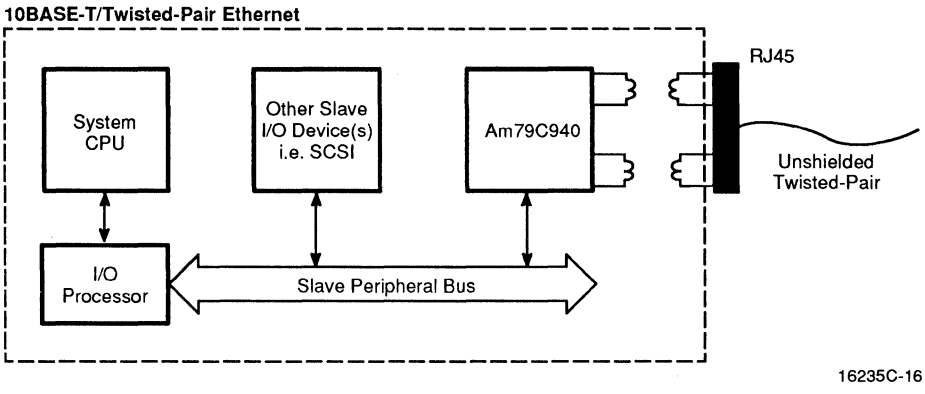
When used with the Am79C98 TPEX™ (Twisted Pair Ethernet Transceiver), the isolation requirements of the AUI are completely removed providing that the transceiver is mounted locally. For remote location of the TPEX via an AUI drop cable, the isolation requirement is necessary to meet IEEE 802.3 specifications for fault tolerance and recovery.



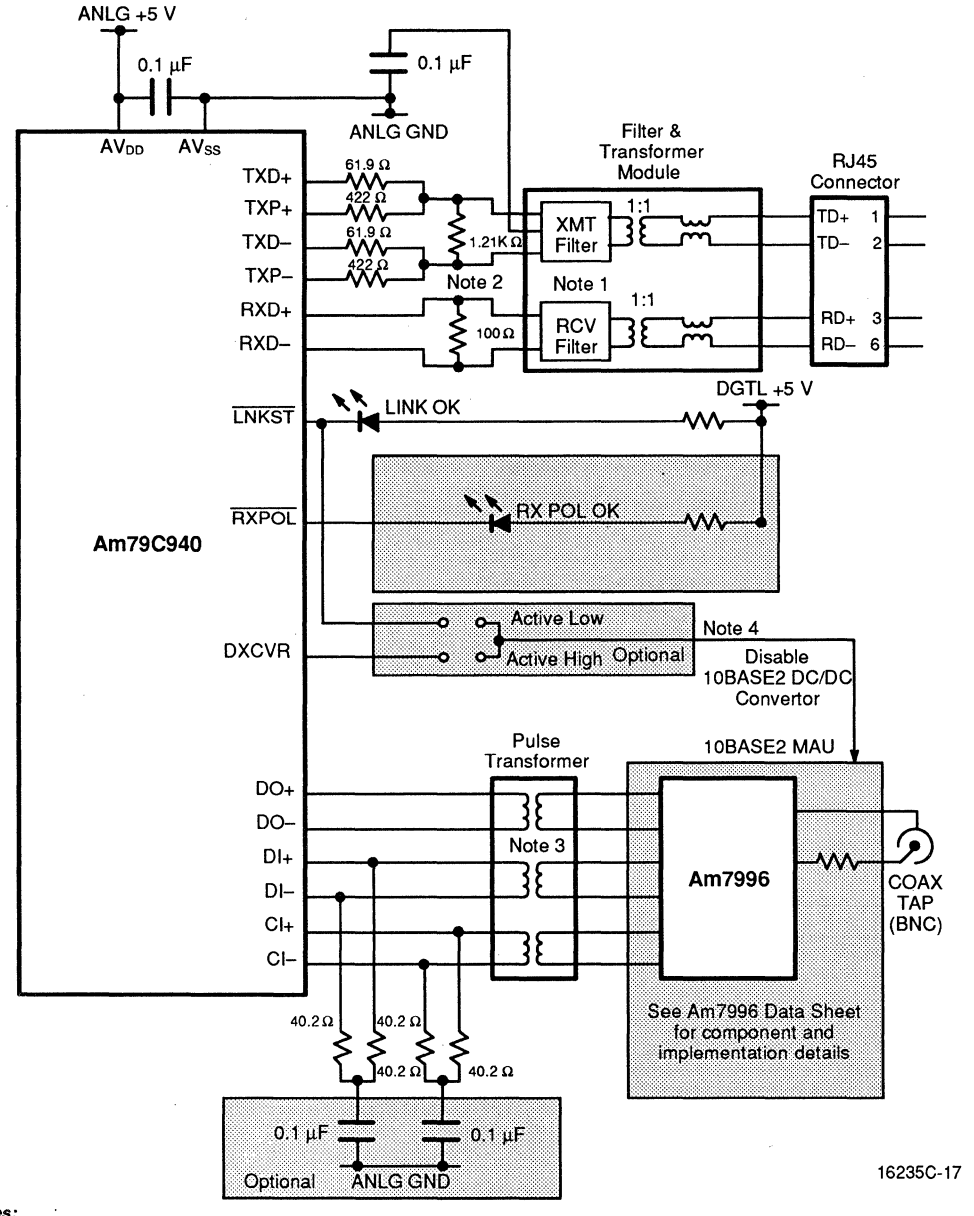
AUI-10BASE5/Ethernet Example



AUI-10BASE2/Cheapernet Example



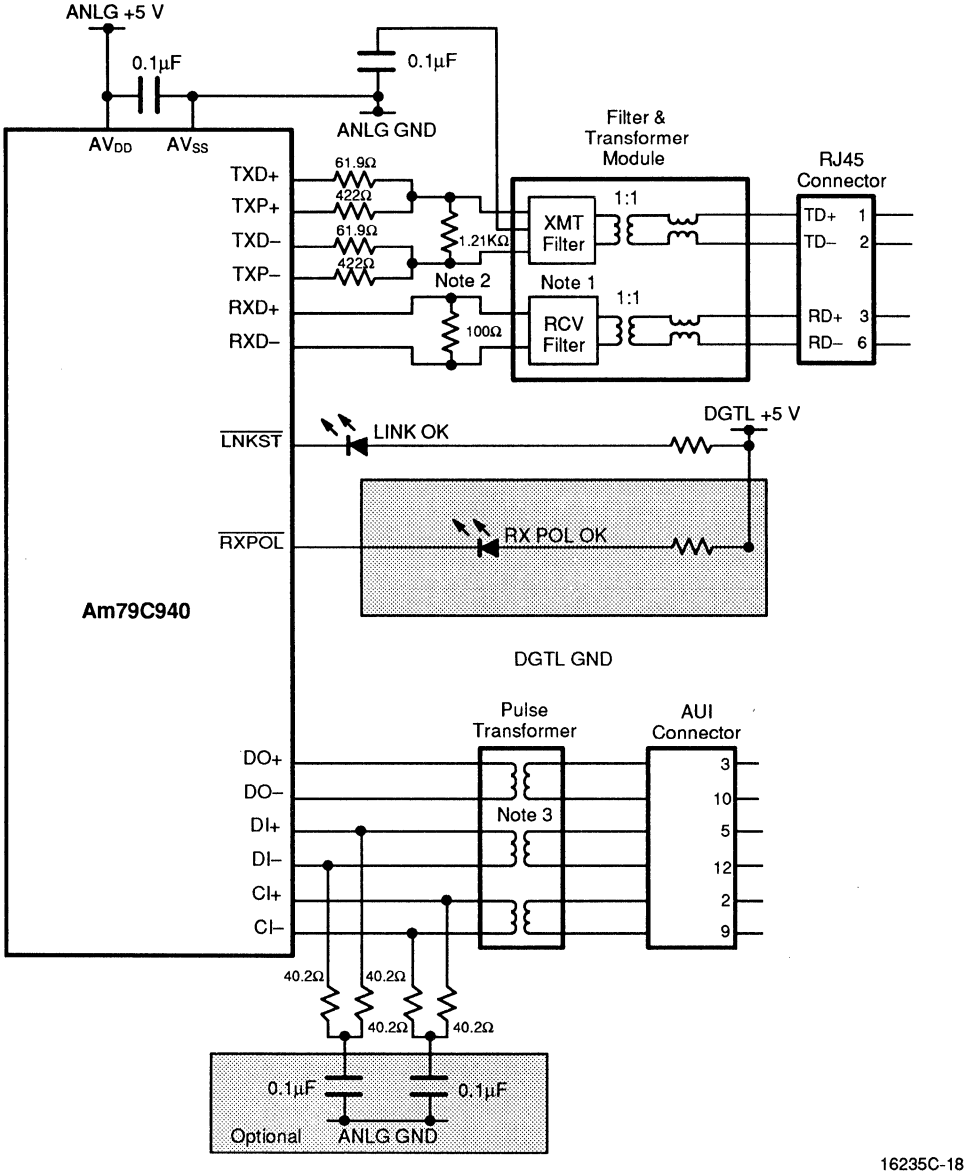
AUI-10BASE-T/Unshielded Twisted-Pair Interface



Notes:

1. *Compatible filter modules, with a brief description of package type and features are included in the following section.*
2. *The resistor values are recommended for general purpose use and should allow compliance to the 10BASE-T specification for template fit and jitter performance. However, the overall performance of the transmitter is also affected by the transmit filter configuration. All resistors are ± 1%.*
3. *Compatible AUI transformer modules, with a brief description of package type and features are included in the following section.*
4. *Active High indicates the external convertor should be turned off. The Disable Transceiver (DXCVR) output is used to indicate the active network port. A high level indicates the 10BASE-T port is selected and the AUI port is disabled. A low level indicates the AUI port is selected and the Twisted Pair interface is disabled.*
   Active Low: indicates the external converter should be turned off. The LNKST output can be used to indicate the active network port. A high level indicates the 10BASE-T port is in the Link Fail state, and the external convertor should be on. A low level indicates the 10BASE-T port is in the Link Pass state, and the external convertor should be off.

10BASE-T and 10BASE2 Configuration of Am79C940



16235C-18

Notes:

1. *Compatible filter modules, with a brief description of package type and features are included in the following section.*
2. *The resistor values are recommended for general purpose use and should allow compliance to the 10BASE-T specification for template fit and jitter performance. However, the overall performance of the transmitter is also affected by the transmit filter configuration. All resistors are ± 1%.*
3. *Compatible AUI transformer modules, with a brief description of package type and features are included in the following section.*

10BASE-T and AUI Implementation of Am79C940

MACE Compatible 10BASE-T Filters and Transformers

The table below provides a sample list of MACE compatible 10BASE-T filter and transformer modules available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part #	Package	Filters and Transformers	Filters Transformers and Choke	Filters Transformers Dual Chokes	Filters Transformers Resistors Dual Chokes
Bel Fuse	A556-2006-DE	16-pin 0.3 DIL	√			
Bel Fuse	0556-2006-00	14-pin SIP	√			
Bel Fuse	0556-2006-01	14-pin SIP			√	
Bel Fuse	0556-6392-00	16-pin 0.5 DIL			√	
Halo Electronics	FD02-101G	16-pin 0.3 DIL	√			
Halo Electronics	FD12-101G	16-pin 0.3 DIL		√		
Halo Electronics	FD22-101G	16-pin 0.3 DIL			√	
PCA Electronics	EPA1990A	16-pin 0.3 DIL	√			
PCA Electronics	EPA2013D	16-pin 0.3 DIL		√		
PCA Electronics	EPA2162	16-pin 0.3 SIP			√	
Pulse Engineering	PE-65421	16-pin 0.3 DIL	√			
Pulse Engineering	PE-65434	16-pin 0.3 SIL			√	
Pulse Engineering	PE-65445	16-pin 0.3 DIL			√	
Pulse Engineering	PE-65467	12-pin 0.5 SMT				√
Valor Electronics	PT3877	16-pin 0.3 DIL	√			
Valor Electronics	FL1043	16-pin 0.3 DIL			√	

MACE Compatible AUI Isolation Transformers

The table below provides a sample list of MACE compatible AUI isolation transformers available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part #	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3 DIL	50 µH
Bel Fuse	S553-0756-AE	16-pin 0.3 SMD	75 µH
Halo Electronics	TD01-0756K	16-pin 0.3 DIL	75 µH
Halo Electronics	TG01-0756W	16-pin 0.3 SMD	75 µH
PCA Electronics	EP9531-4	16-pin 0.3 DIL	50 µH
Pulse Engineering	PE64106	16-pin 0.3 DIL	50 µH
Pulse Engineering	PE65723	16-pin 0.3 SMT	75 µH
Valor Electronics	LT6032	16-pin 0.3 DIL	75 µH
Valor Electronics	ST7032	16-pin 0.3 SMD	75 µH

MACE Compatible DC/DC Converters

The table below provides a sample list of MACE compatible DC/DC converters available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part #	Package	Voltage	Remote On/Off
Halo Electronics	DCU0-0509D	24-pin DIP	5/-9	No
Halo Electronics	DCU0-0509E	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1007P	24-pin DIP	5/-9	No
PCA Electronics	EPC1054P	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1078	24-pin DIP	5/-9	Yes
Valor Electronics	PM7202	24-pin DIP	5/-9	No
Valor Electronics	PM7222	24-pin DIP	5/-9	Yes

MANUFACTURER CONTACT INFORMATION

Contact the following companies for further information on their products.

Company	U.S. and Domestic		Asia	Europe
Bel Fuse	Phone:	(201) 432-0463	852-328-5515	33-1-69410402
	FAX:	(201) 432-9542	852-352-3706	33-1-69413320
Halo Electronics	Phone:	(415) 969-7313	65-285-1566	
	FAX:	(415) 367-7158	65-284-9466	
PCA Electronics	Phone:	(818) 892-0761	852-553-0165	33-1-44894800
(HPC in Hong Kong)	FAX:	(818) 894-5791	852-873-1550	33-1-42051579
Pulse Engineering	Phone:	(619) 674-8100	852-425-1651	353-093-24107
	FAX:	(619) 675-8262	852-480-5974	353-093-24459
Valor Electronics	Phone:	(619) 537-2500	852-513-8210	49-89-6923122
	FAX:	(619) 537-2525	852-513-8214	49-89-6926542

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... –65°C to +150°C

Ambient Temperature Under Bias ..... 0°C to +70°C

Supply Voltage to AV_{SS} or DV_{SS} (AV_{DD}, DV_{DD}) ..... –0.3 V to +6.0 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A) ..... 0°C to +70°C

Supply Voltages (AV_{DD}, DV_{DD}) ..... 5 V ±5%

All inputs within the range: .. AV_{DD} + 0.5 V ≤ Vin ≤ ..... AV_{SS} – 0.5 V, or ..... DV_{DD} = 0.5 V ≤ Vin ≤ ..... DV_{SS} – 0.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
V_{IL}	Input LOW Voltage			0.8	V
V_{IH}	Input HIGH Voltage		2.0		V
V_{ILX}	XTAL1 Input LOW Voltage (External Clock Signal)	V_{SS} = 0.0 V	–0.5	0.8	V
V_{IHX}	XTAL1 Input HIGH Voltage (External Clock Signal)	V_{SS} = 0.0 V	V_{DD} – 0.8	V_{DD} + 0.5	V
V_{OL}	Output LOW Voltage	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output HIGH Voltage	I_{OH} = –0.4 mA (Note 1)	2.4		V
I_{IL1}	Input Leakage Current	V_{DD} = 5 V, V_{IN} = 0 V (Note 2)	–10	10	µA
I_{IL2}	Input Leakage Current	V_{DD} = 5 V, V_{IN} = 0 V (Note 2)	–200	200	µA
I_{IH}	Input Leakage Current	V_{DD} = 5 V, V_{IN} = 2.7 V (Note 3)		–100	µA
I_{IAXD}	Input Current at DI+ and DI–	–1 V < V_{IN} < AV_{DD} + 0.5 V	–500	+500	µA
I_{IAXC}	Input current at CI+ and CI–	–1 V < V_{IN} < AV_{DD} + 0.5 V	–500	+500	µA
I_{ILXN}	XTAL1 Input LOW Current during normal operation	V_{IN} = 0 V $\overline{SLEEP}$ = HIGH		–92	µA
I_{IHXN}	XTAL1 Input HIGH Current during normal operation	V_{IN} = 5.5 V $\overline{SLEEP}$ = HIGH		92	µA
I_{ILXS}	XTAL1 Input LOW Current during Sleep	V_{IN} = 0 V $\overline{SLEEP}$ = LOW		<10	µA
I_{IHXS}	XTAL1 Input HIGH Current during Sleep	V_{IN} = 5.5 V $\overline{SLEEP}$ = LOW		410	µA
I_{OZ}	Output Leakage Current	0.4 V < V_{OUT} < V_{DD} (Note 4)	–10	10	µA
V_{AOD}	Differential Output Voltage |(DO+)–(DO–)|	R_L = 78 Ω	630	1200	mV
V_{AODOFF}	Transmit Differential Output Idle Voltage	R_L = 78 Ω (Note 5)	–40	+40	mV
I_{AODOFF}	Transmit Differential Output Idle Current	R_L = 78 Ω	–1	+1	mA

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
V_{AOCM}	DO± Common Mode Output Voltage	R_L = 78 Ω	2.5	AVDD	V
V_{ODI}	DO± Differential Output Voltage Imbalance	R_L = 78 Ω (Note 6)	–25	25	mV
V_{ATH}	Receive Data Differential Input Threshold	RL = 78 Ω (Note 6)	–35	35	mV
V_{ASQ}	DI± and CI± Differential Input Threshold Squelch	RL = 78 Ω (Note 6)	–160	–275	mV
V_{IRDVD}	DI± and CI± Differential Mode Input Voltage Range			1.5	V
V_{ICM}	DI± and CI± Input Bias Voltage	I_{IN} = 0 mA	AV_{DD}–3.0	AV_{DD}–0.8	V
V_{OPD}	DO± Undershoot Voltage at Zero Differential on Transmit Return to Zero (ETD)	(Note 5)		–100	mV
I_{DD}	Power Supply Current	SCLK = 25 MHz XTAL1 = 20 MHz		75	mA
$I_{DDSLEEP}$	Power Supply Current	$\overline{SLEEP}$ Asserted, AWAKE = 0 RWAKE = 0 (Note 7)		100	µA
$I_{DDSLEEP}$	Power Supply Current	$\overline{SLEEP}$ Asserted, AWAKE = 1 RWAKE = 0 (Note 7)		10	mA
$I_{DDSLEEP}$	Power Supply Current	$\overline{SLEEP}$ Asserted, AWAKE = 0 RWAKE = 1 (Note 7)		20	mA
Twisted Pair Interface					
I_{IRXD}	Input Current at RXD±	$AV_{SS} < V_{IN} < AV_{DD}$	–500	500	µA
R_{RXD}	RXD± Differential Input Resistance	(Note 8)	10		KΩ
V_{TIVB}	RXD+, RXD– Open Circuit Input Voltage (Bias)	I_{IN} = 0 mA	AV_{DD} – 3.0	AV_{DD} – 1.5	V
V_{TIDV}	Differential Mode Input Voltage Range (RXD±)	AV_{DD} = +5 V	–3.1	+3.1	V
V_{TSQ+}	RXD Positive Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	300	520	mV
V_{TSQ-}	RXD Negative Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	–520	–300	mV
V_{THS+}	RXD Post-Squelch Positive Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	150	293	mV
V_{THS-}	RXD Post-Squelch Negative Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	–293	–150	mV
V_{LTSQ+}	RXD Positive Squelch Threshold (Peak)	$\overline{LRT}$ = LOW	180	312	mV
V_{LTSQ-}	RXD Negative Squelch Threshold (Peak)	$\overline{LRT}$ = LOW	–312	–180	mV
V_{LTHS+}	RXD Post-Squelch Positive Threshold (Peak)	$\overline{LRT}$ = LOW	90	156	mV
V_{LTHS-}	RXD Post-Squelch Negative Threshold (Peak)	$\overline{LRT}$ = LOW	–156	–90	mV

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
V_{RXDTH}	RXD Switching Threshold	(Note 4)	–35	35	mV
V_{TXH}	TXD± and TXP± Output HIGH Voltage	DV_{SS} = 0 V	DV_{DD} – 0.6	DV_{DD}	V
V_{TXL}	TXD± and TXP± Output LOW Voltage	DV_{DD} = +5 V	DV_{SS}	DV_{SS} + 0.6	V
V_{TXI}	TXD± and TXP± Differential Output Voltage Imbalance		–40	+40	mV
V_{TXOFF}	TXD± and TXP± Idle Output Voltage	DV_{DD} = +5 V		40	mV
R_{TX}	TXD± Differential Driver Output Impedance	(Note 8)		40	Ω
	TXP± Differential Driver Output Impedance	(Note 8)		80	Ω

Notes:

1. *V_{OH} does not apply to open-drain output pins.*
2. *I_{IL1} and I_{IL2} applies to all input only pins except DI±, CI±, and XTAL1.*
   I_{IL1} = ADD4–0, $\overline{BE1}$–0, $\overline{CS}$, $\overline{EAM/R}$, $\overline{FDS}$, $\overline{RESET}$, RXDAT, R/$\overline{W}$, SCLK.
   I_{IL2} = $\overline{TC}$, TDI, TCK, TMS.
3. *Specified for input only pins with internal pull-ups: $\overline{TC}$, TDI, TCK, TMS.*
4. *I_{OZ} applies to all three-state output pins and bi-directional pins.*
5. *Test not implemented to data sheet specification.*
6. *Tested, but to values in excess of limits. Test accuracy not sufficient to allow screening guard bands.*
7. *During the activation of $\overline{SLEEP}$:*
   - *The following pins are placed in a high impedance state: SRD, SF/BD, TXDAT, DXCVR, $\overline{DTV}$, $\overline{TDTREQ}$, $\overline{RDTREQ}$, $\overline{NTR}$ and TDO.*
   - *The following I/O pins are placed in a high impedance mode and have their internal TTL level translators disabled: DBUS15–0, $\overline{EOF}$, SRDCLK, RXCRS, RXDAT, CLSN, $\overline{TXEN}$, STDCLK and TXDAT+.*
   - *The following input pin has its internal pull-up and TTL level translator disabled: $\overline{TC}$.*
   - *The following input pins have their internal TTL level translators disabled and do not have internal pull-ups: $\overline{CS}$, $\overline{FDS}$, R/$\overline{W}$, ADD4–0, SCLK, $\overline{BE0}$, $\overline{BE1}$ and $\overline{EAM/R}$.*
   - *The following pins are pulled low: XTAL1 (XTAL2 feedback is cut off from XTAL1), TXD+, TXD–, TXP+, TXP–, DO+ and DO.*
   - *The following pins have their input voltage bias disabled: DI+, DI, CI+ and CI.*
   - *AWAKE and RWAKE are reset to zero. $I_{DDSLEEP}$, with either AWAKE set or RWAKE set, will be much higher and its value remains to be determined.*
8. *Parameter not tested.*

AC CHARACTERISTICS

No.	Parameter Symbol	Parameter Description	Test Conditions	Min (ns)	Max (ns)
Clock and Reset Timing					
1	t_{SCLK}	SCLK period		40	1000
2	t_{SCLKL}	SCLK LOW pulse width		0.4*t_{SCLK}	0.6*t_{SCLK}
3	t_{SCLKH}	SCLK HIGH pulse width		0.4*t_{SCLK}	0.6*t_{SCLK}
4	t_{SCLKR}	SCLK rise time			5
5	t_{SCLKF}	SCLK fall time			5
6	t_{RST}	$\overline{\text{RESET}}$ pulse width		15*t_{SCLK}	
7	t_{BT}	Network Bit Time (BT) =2*tX1 or tSTDC)		99	101
Internal MENDEC Clock Timing					
9	t_{X1}	XTAL1 period		49.995	50.005
11	t_{X1H}	XTAL1 HIGH pulse width		20	
12	t_{X1L}	XTAL1 LOW pulse width		20	
13	t_{X1R}	XTAL1 rise time			5
14	t_{X1F}	XTAL1 fall time			5
BIU Timing (Note 1)					
31	t_{ADDS}	Address valid setup to SCLK↓		9	
32	t_{ADDH}	Address valid hold after SCLK↓		2	
33	t_{SLVS}	$\overline{\text{CS}}$ or $\overline{\text{FDS}}$ and $\overline{\text{TC}}$, $\overline{\text{BE}}$1–0, R/$\overline{\text{W}}$ setup to SCLK↓		9	
34	t_{SLVH}	$\overline{\text{CS}}$ or $\overline{\text{FDS}}$ and $\overline{\text{TC}}$, $\overline{\text{BE}}$1–0, R/$\overline{\text{W}}$ hold after SCLK↓		2	
35	t_{DATD}	Data out valid delay from SCLK↓	C_L = 100 pF (Note 2)		32
36	t_{DATH}	Data out valid hold after SCLK↓		6	
37	t_{DTVD}	$\overline{\text{DTV}}$ valid delay from SCLK↓	C_L = 100 pF (Note 2)		32
38	t_{DTVH}	$\overline{\text{DTV}}$ valid hold after SCLK↓		6	
39	t_{EOFD}	$\overline{\text{EOF}}$ valid delay from SCLK↓	C_L = 100 pF (Note 2)		32
40	t_{EOFH}	$\overline{\text{EOF}}$ output valid hold after SCLK↓		6	
41	t_{CSIS}	$\overline{\text{CS}}$ inactive prior to SCLK↓		9	
42	t_{EOFS}	$\overline{\text{EOF}}$ input valid setup to SCLK↓		9	
43	t_{EOFH}	$\overline{\text{EOF}}$ input valid hold after SCLK↓		2	
44	t_{RDTD}	$\overline{\text{RDTREQ}}$ valid delay from SCLK↓	C_L = 100 pF (Note 2)		32
45	t_{RDTH}	$\overline{\text{RDTREQ}}$ valid hold after SCLK↓		6	
46	t_{TDTD}	$\overline{\text{TDTREQ}}$ valid delay from SCLK↓	C_L = 100 pF (Note 2)		32
47	t_{TDTH}	$\overline{\text{TDTREQ}}$ valid hold after SCLK↓		6	
48	t_{DATS}	Data in valid setup to SCLK↓		9	
49	t_{DATIH}	Data in valid setup after SCLK↓		2	
50	t_{DATE}	Data output enable delay from SCLK↓ (Note 3)		0	
51	t_{DATD}	Data output disable delay from SCLK↓ (Notes 3, 4)			25

Notes:

1. *The following BIU timing assumes that EDSEL = 1. Therefore, these parameters are specified with respect to the falling edge of SCLK (SCLK↓). If EDSEL = 0, the same parameters apply but should be referenced to the rising edge of SCLK (SCLK↑).*
2. *Tested with C_L set at 100 pF and derated to support the Indicated distributed capacitive Load. See the BIU output valid delay vs. Load Chart.*
3. *Guaranteed by design—not tested.*
4. *t_{DATD} is defined as the time required for outputs to turn high impedence and is not referred to as output voltage lead.*

AC CHARACTERISTICS (continued)

No.	Parameter Symbol	Parameter Description	Test Conditions	Min (ns)	Max (ns)
AUI Timing					
53	t_{DOTD}	XTAL1 (externally driven) to DO± output			100
54	t_{DOTR}	DO± rise time (10% to 90%)		2.5	5.0
55	t_{DOTF}	DO± fall time (10% to 90%)		2.5	5.0
56	t_{DOETM}	DO± rise and fall mismatch			1
57	t_{DOETD}	DO± End of Transmit Delimiter		200	375
58	t_{PWRDI}	DI± pulse width to reject	|input| > |V_{ASQ}|		15
59	t_{PWODI}	DI± pulse width to turn on internal DI carrier sense	|input| > |V_{ASQ}|	45	
60	t_{PWMDI}	DI± pulse width to maintain internal DI carrier sense on	|input| > |V_{ASQ}|	45	136
61	t_{PWKDI}	DI± pulse width to turn internal DI carier sense off	|input| > |V_{ASQ}|	200	
62	t_{PWRCI}	CI± pulse width to reject	|input| > |V_{ASQ}|		10
63	t_{PWOCI}	CI± pulse width to turn on internal SQE sense	|input| > |V_{ASQ}|	26	
64	t_{PWMCI}	CI± pulse width to maintain internal SQE sense on	|input| > |V_{ASQ}|	26	90
65	t_{PWKCI}	CI± pulse width to turn internal SQE sense off	|input| > |V_{ASQ}|	160	
66	t_{SQED}	CI± SQE Test delay from O± inactive	|input| > |V_{ASQ}|		
67	t_{SQEL}	CI± SQE Test length	|input| > |V_{ASQ}|		
79	t_{CLSHI}	CLSN high time		t_{STDC}+30	
80	t_{TXH}	$\overline{TXEN}$ or DO± hold time from CLSN↑	|input| > |V_{ASQ}|	32*t_{STDC}	96*t_{STDC}
DAI Port Timing					
70	t_{TXEND}	STDCLK↑ delay to $\overline{TXEN}$↓	C_L = 50 pF		70
72	t_{TXDD}	STDCLK↑ delay to TXDAT± change	C_L = 50 pF		70
80	t_{XH}	$\overline{TXEN}$ or TXDAT± hold time from CLSN↑		32*t_{STDC}	96*t_{STDC}
95	t_{DOTF}	Mismatch in STDCLK≠ to $\overline{TXEN}$↓ and TXDAT± change			15
96	t_{TXDTR}	TXDAT± rise time	See Note 1		5
97	t_{TXDTF}	TXDAT± fall time	See Note 1		5
98	t_{TXDTM}	TXDAT± rise and fall mismatch	See Note 1		1
99	$t_{TXENETD}$	$\overline{TXEN}$ End of Transmit Delimiter		250	350
100	t_{FRXDD}	First RXDAT↓ delay to RXCRS↑			100
101	t_{LRXDD}	Last RXDAT≠ delay to RXCRS↓			120
102	$t_{CRSCLSD}$	RXCRS↑ delay to CLSN↑ ($\overline{TXEN}$ = 0)			100

AC CHARACTERISTICS (continued)

No.	Parameter Symbol	Parameter Description	Test Conditions	Min (ns)	Max (ns)
GPSI Clock Timing					
17	t_{STDC}	STDCLK period		99	101
18	t_{STDCL}	STDCLK low pulse width	See Note 1	45	
19	t_{STDCH}	STDCLK high pulse width		45	
20	t_{STDCR}	STDCLK rise time	See Note 1		5
21	t_{STDCF}	STDCLK fall time	See Note 1		5
22	t_{SRDC}	SRDCLK period		85	115
23	t_{SRDCH}	SRDCLK HIGH pulse width		38	
24	t_{SRDCL}	SRDCLK LOW pulse width		38	
25	t_{SRDCR}	SRDCLK rise time	See Note 1		5
26	t_{SRDCF}	SRDCLK fall time	See Note 1		5
GPSI Timing					
70	t_{TXEND}	STDCLK↑ delay to $\overline{TXEN}$↑	(C_L=50 pF)		70
71	t_{TXENH}	$\overline{TXEN}$ hold time from STDCLK↑	(C_L=50 pF)	5	
72	t_{TXDD}	STDCLK↑ delay to TXDAT+ change	(C_L=50 pF)		70
73	t_{TXDH}	TXDAT+ hold time from STDCLK↑	(C_L=50 pF)	5	
74	t_{RXDR}	RXDAT rise time	See Note 1		8
75	t_{RXDF}	RXDAT fall time	See Note 1		8
76	t_{RXDH}	RXDAT hold time (SRDCLK↑ to RXDAT change)		25	
77	t_{RXDS}	RXDAT setup time (RXDAT stable to SRDCLK↑)		0	
78	t_{CRSL}	RXCRS low time		t_{STDC}+20	
79	t_{CLSHI}	CLSN high time		t_{STDC}+30	
80	t_{TXH}	$\overline{TXEN}$ or TXDAT± hold time from CLSN↑		32*t_{STDC}	96*t_{STDC}
81	t_{CRSH}	RXCRS hold time from SRDCLK↑		0	
EADI Feature Timing					
85	t_{DSFBDR}	SRDCLK↓ delay to SF/BD↑			20
86	t_{DSFBDF}	SRDCLK↓ delay to SF/BD↓			20
87	t_{EAMRIS}	$\overline{EAM}$/R invalid setup prior to SRDCLK↓ after SFD		–150	
88	t_{EAMS}	$\overline{EAM}$ setup to SRDCLK↓ at bit 6 of Source Address byte 1 (match packet)		0	
89	t_{EAMRL}	$\overline{EAM}$/R low time		200	
90	$t_{SFBDHIH}$	SF/BD high hold from last SRDCLK↓		100	
91	t_{EARS}	$\overline{EAR}$ setup to SRDCLK↓ at bit 6 of message byte 64 (reject normal packet)		0	

Note:

1. Not tested but data available upon request.

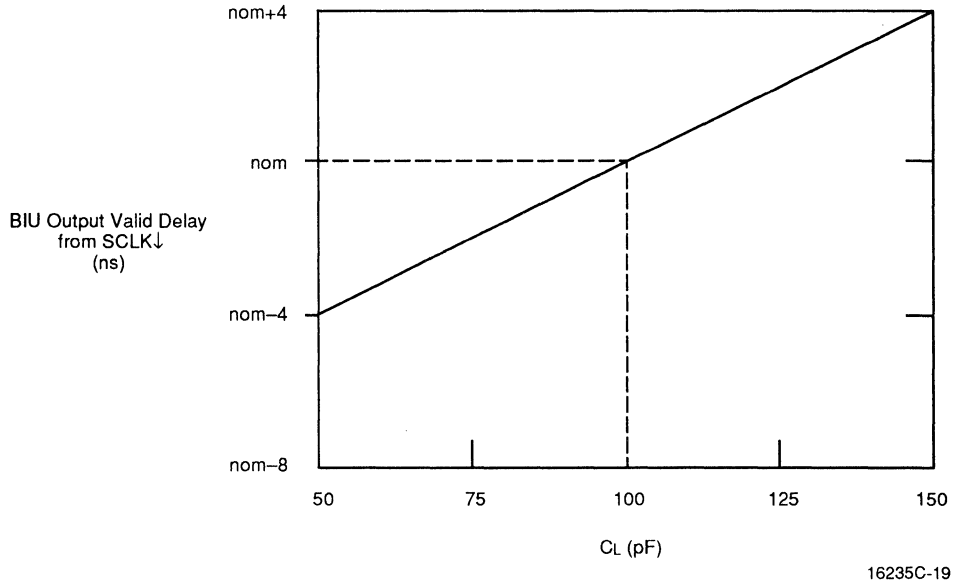
AC CHARACTERISTICS (continued)

No.	Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
IEEE 1149.1 Timing						
109	t_{TCLK}	TCK Period, 50% duty cycle (+5%)		100		
110	t_{su1}	TMS setup to TCK↑		8		
111	t_{su2}	TDI setup to TCK↑		5		
112	t_{hd1}	TMS hold time from TCK↑		5		
113	t_{hd2}	TDI hold time from TCK↑		10		
114	t_{d1}	TCK↓ delay to TDO			30	
115	t_{d2}	TCK↓ delay to SYSTEM OUTPUT			35	
10BASE-T Transmit Timing				**Min**	**Max**	**Unit**
125	t_{TETD}	Transmit Start of Idle		250	350	ns
126	t_{TR}	Transmitter Rise Time	(10% to 90%)		5.5	ns
127	t_{TF}	Transmitter Fall Time	(90% to 10%)		5.5	ns
128	t_{TM}	Transmitter Rise and Fall Time Mismatch			1	ns
129	t_{XMTON}	XMT# Asserted Delay			100	ns
130	t_{XMTOFF}	XMT# De-asserted Delay		TBD	TBD	ms
131	t_{PERLP}	Idle Signal Period		8	24	ms
132	t_{PWLP}	Idle Link Pulse Width	(Note 1)	75	120	ns
133	t_{PWPLP}	Predistortion Idle Link Pulse Width	(Note 1)	45	55	ns
134	t_{JA}	Transmit Jabber Activation Time		20	150	ms
135	t_{JR}	Transmit Jabber Reset Time		250	750	ms
136	t_{JREC}	Transmit Jabber Recovery Time (Minimum Time Gap Between Transmitted Packets to Prevent Jabber Activation)		1.0		µs
10BASE-T Receive Timing						
140	t_{PWNRD}	RXD Pulse Width Not to Turn Off Internal Carrier Sense	VIN > VTHS (min)	136	–	ns
141	t_{PWROFF}	RXD Pulse Width to Turn Off VIN > VTHS (min)		200		ns
142	t_{RETD}	Receive Start of Idle		200		ns
143	t_{RCVON}	RCV# Asserted Delay		t_{RON}–50	t_{RON}+100	ns
144	t_{RCVOFF}	RCV# De-asserted Delay		TBD	TBD	ms

Note:

1. Not tested but data available upon request.

BIU Output Valid Delay vs. Load Chart



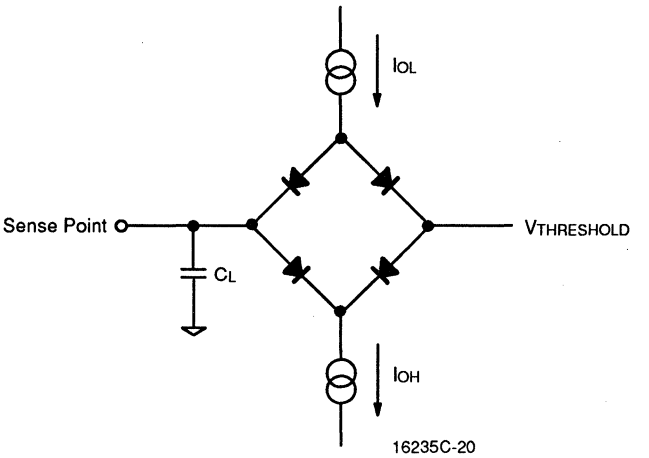
16235C-19

KEY TO SWITCHING WAVEFORMS

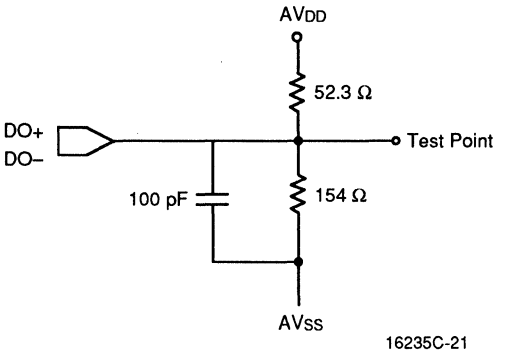
WAVEFORM	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off" State

KS000010

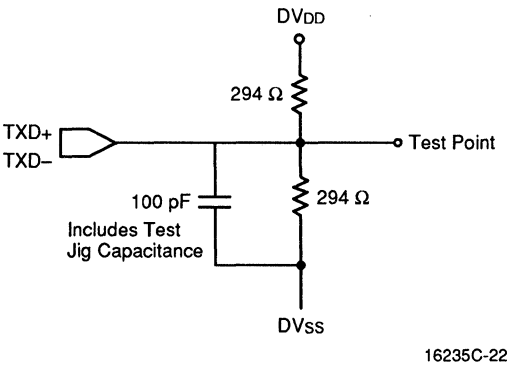
SWITCHING TEST CIRCUITS



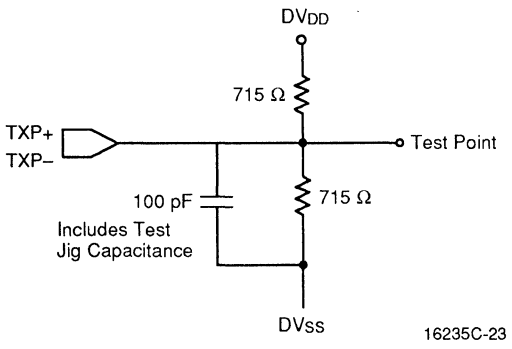
Normal and Three-State Outputs



AUI DO Switching Test Circuit

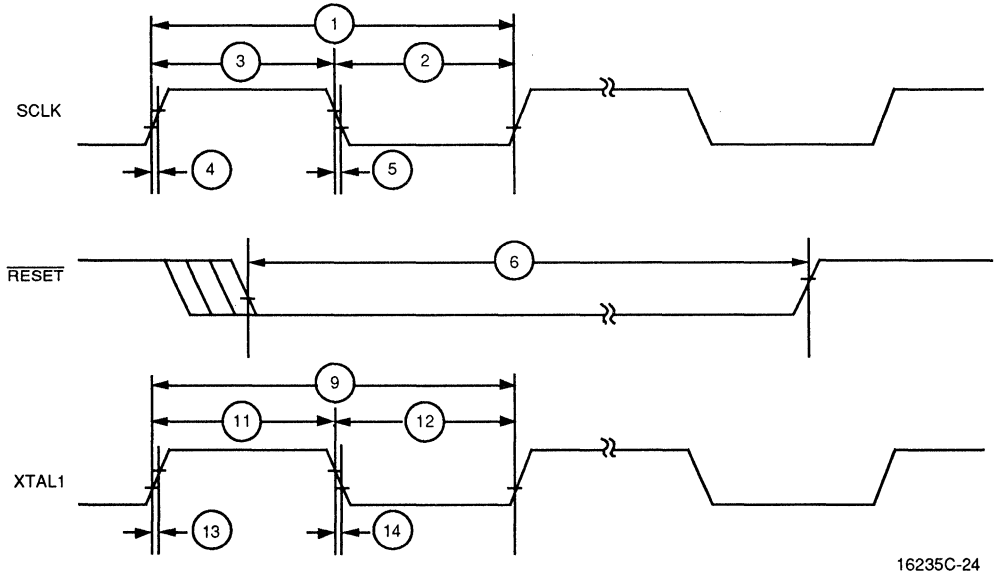


TXD Switching Test Circuit



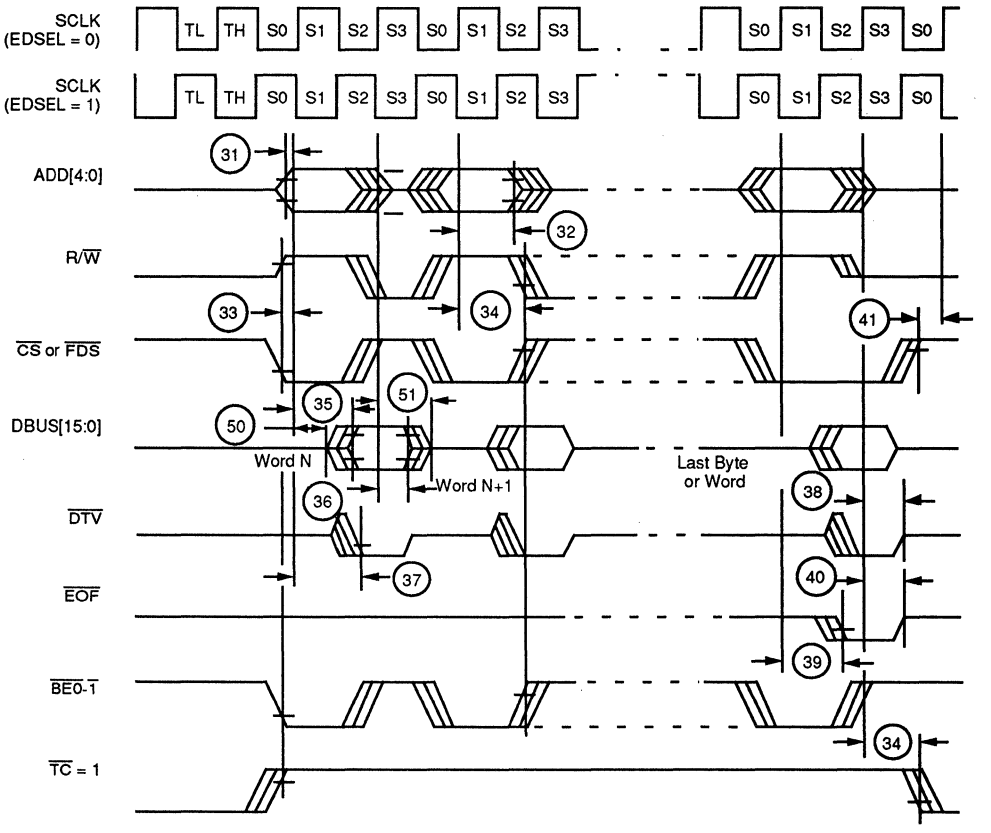
TXP Outputs Test Circuit

AC WAVEFORMS



Clock and Reset Timing

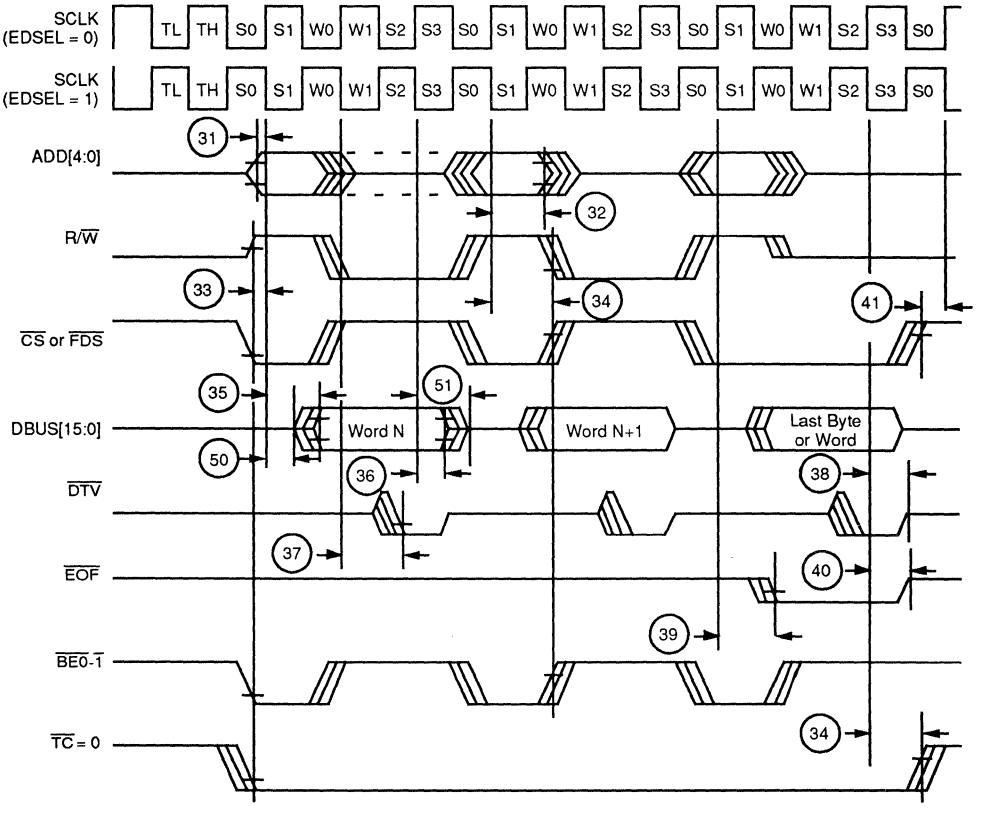
AC WAVEFORMS



16235C-25

Host System Interface—2-Cycle Receive FIFO/Register Read Timing

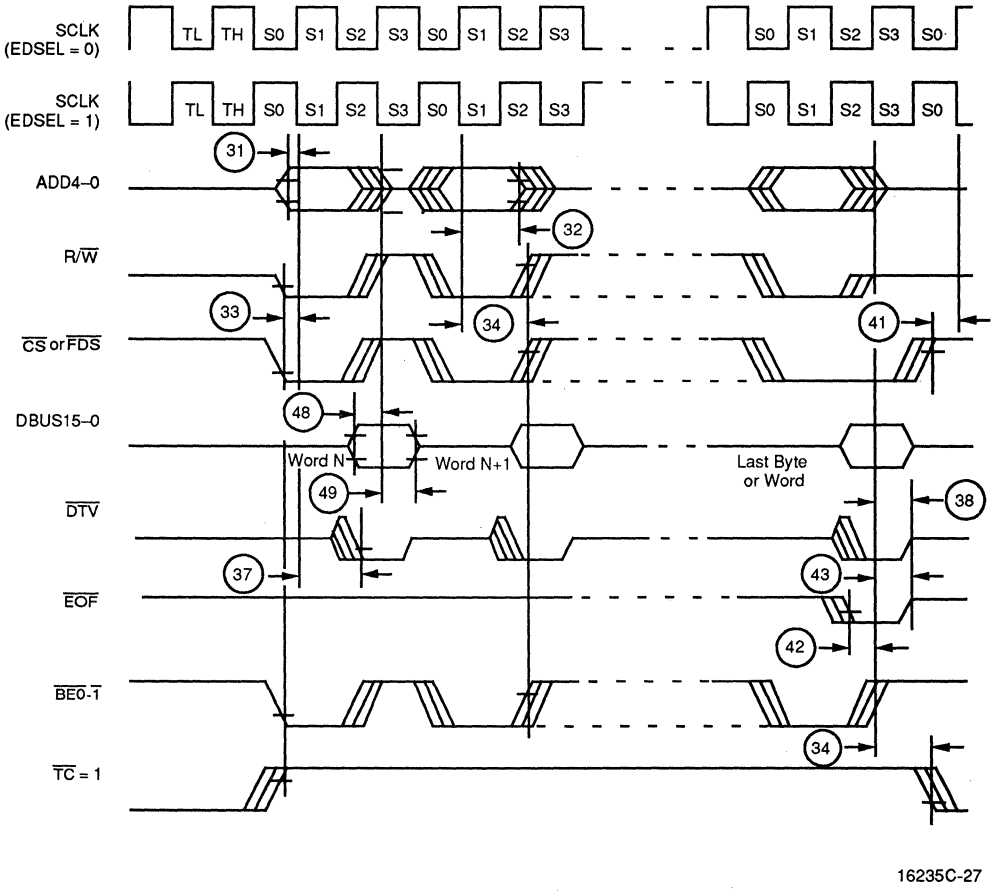
AC WAVEFORMS



16235C-26

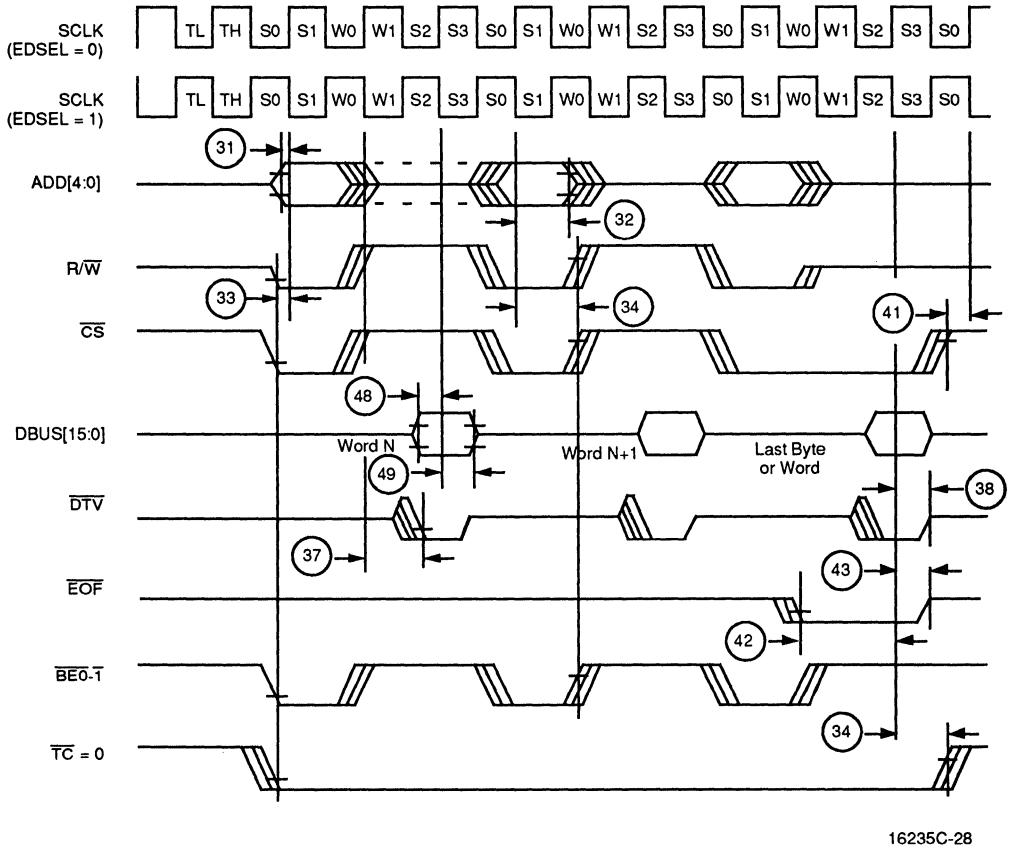
Host System Interface—3-Cycle Receive FIFO/Register Read Timing

AC WAVEFORMS

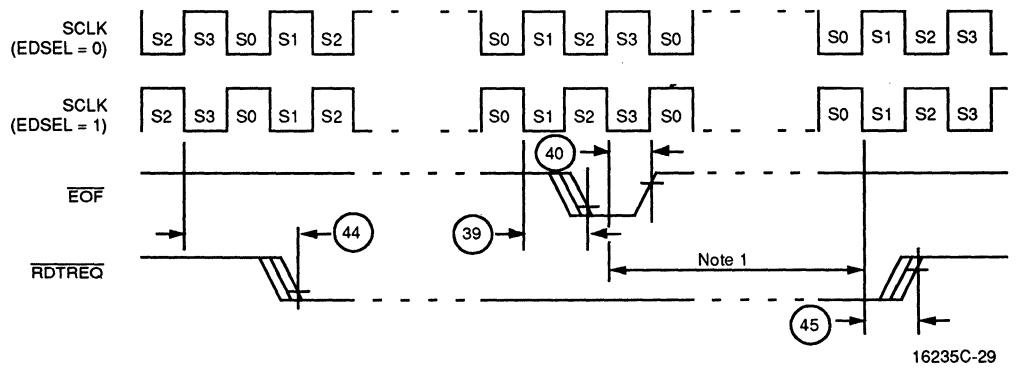


Host System Interface—2-Cycle Transmit FIFO/Register Write Timing

AC WAVEFORMS



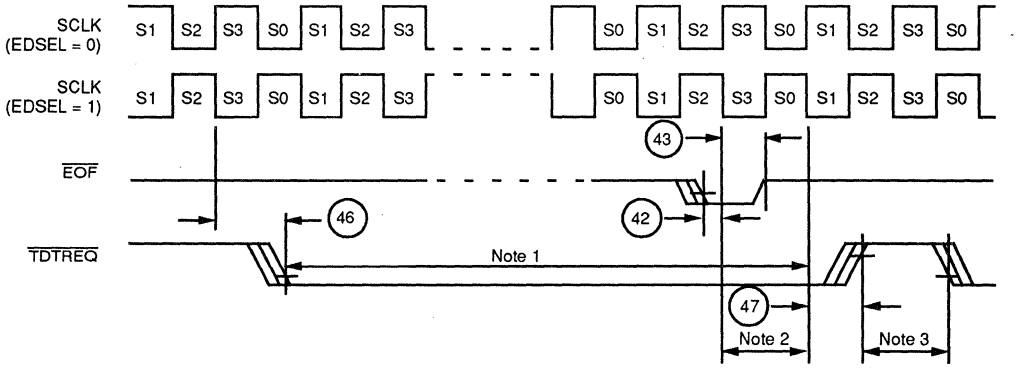
Host System Interface—3-Cycle Transmit FIFO/Register Write Timing



Note: *Once the host detects the EOF output active from the MACE device (S2/S3 edge), if no other receive packet exists in the RCVFIFO which meets the assert conditions for RDTREQ, the MACE device will deassert RDTREQ within 4 SCLK cycles (S0/S1 edge). This is consistent for both 2 or 3 cycle read operations.*

Host System Interface—RDTREQ Read Timing

AC WAVEFORMS

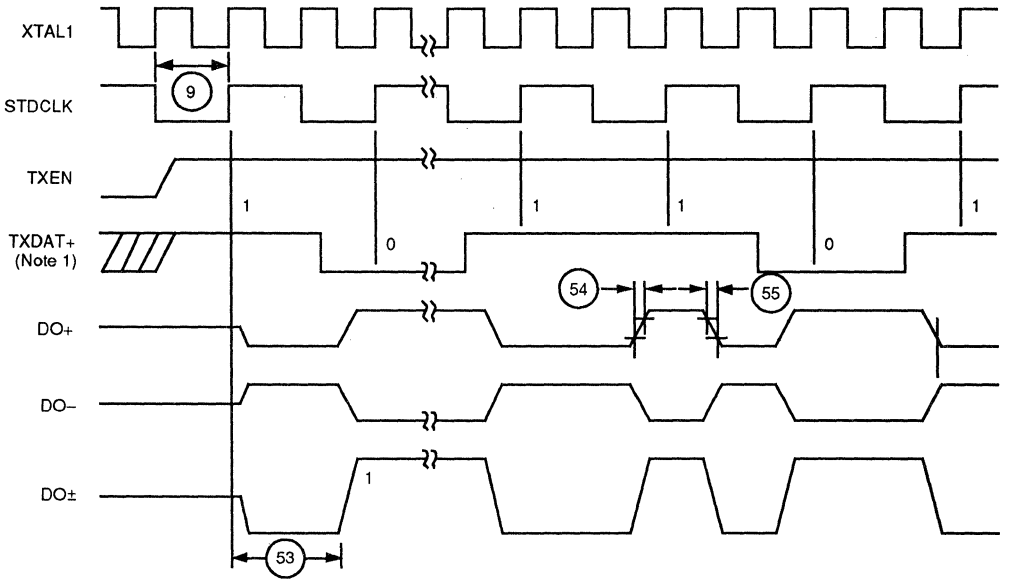


16235C-30

Notes:

1. *$\overline{TDTREQ}$ will be asserted for two write cycles (4 SCLK cycles) minimum.*
2. *$\overline{TDTREQ}$ will deassert 1 SCLK cycle after $\overline{EOF}$ is detected (S2/S3 edge).*
3. *When $\overline{EOF}$ is written, $\overline{TDTREQ}$ will go inactive for 1 SCLK cycle minimum.*

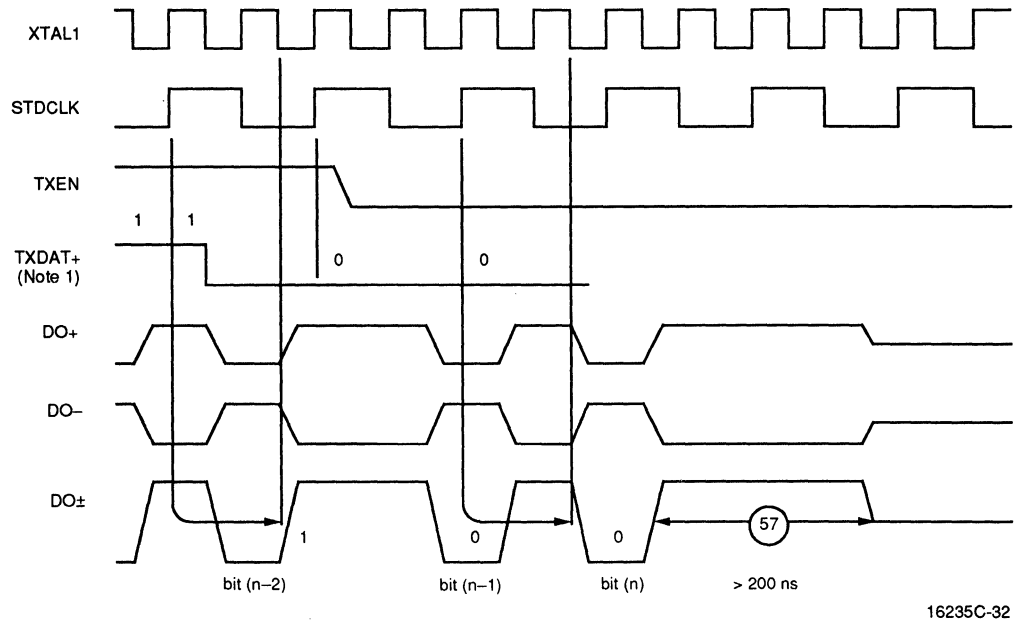
Host System Interface—$\overline{TDTREQ}$ Write Timing



16235C-31

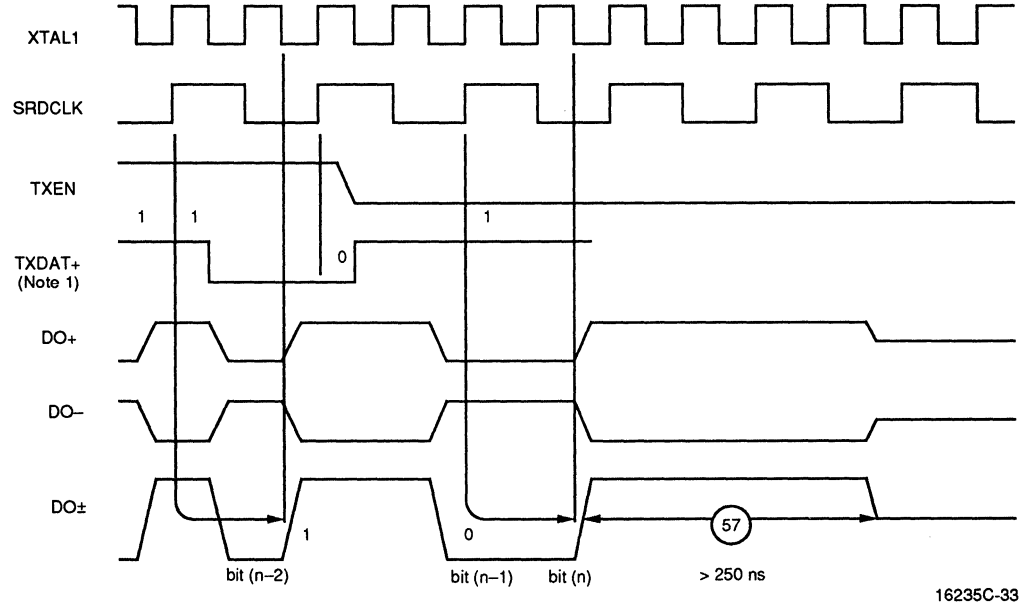
Note: *TXDAT+ is the internal version of the signal, and is shown for clarification only.*

AUI Transmit Timing—Start of Packet



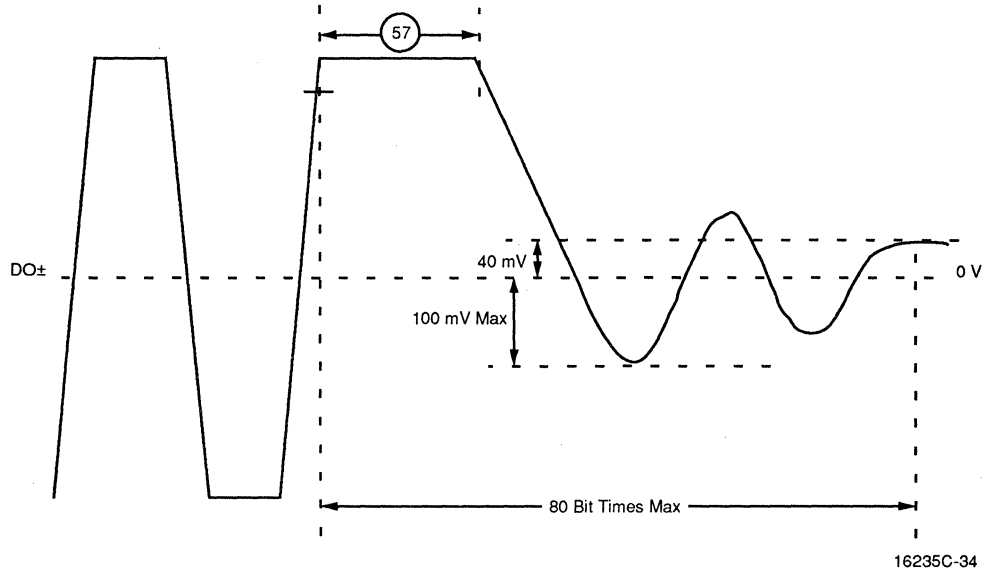
Note: *TXDAT+ is the internal version of the signal, and is shown for clarification only.*

AUI Transmit Timing—End of Packet (Last Bit = 0)

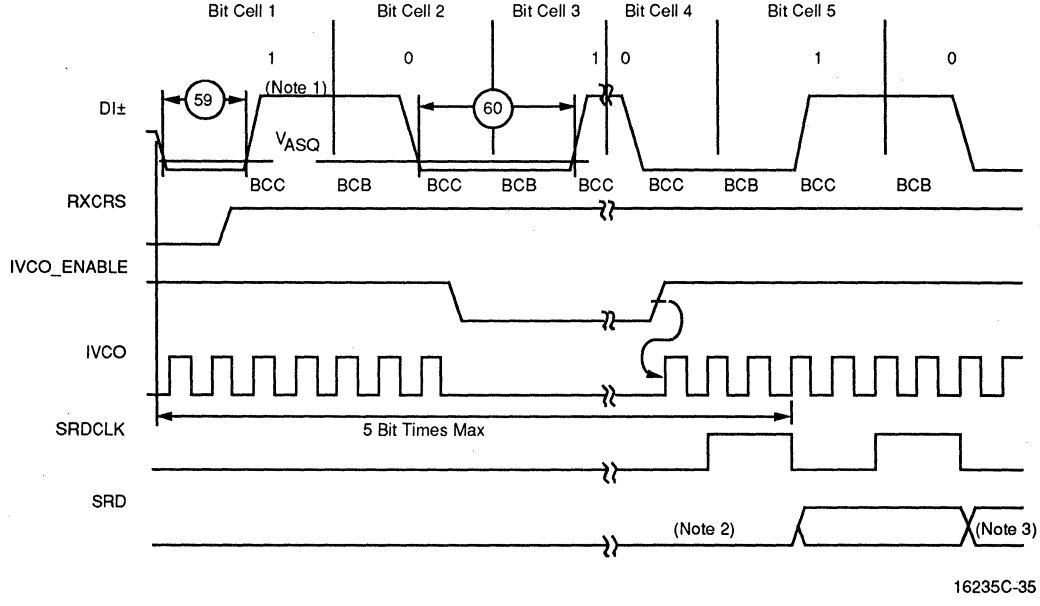


Note: *TXDAT+ is the internal version of the signal, and is shown for clarification only.*

AUI Transmit Timing—End of Packet (Last Bit = 1)



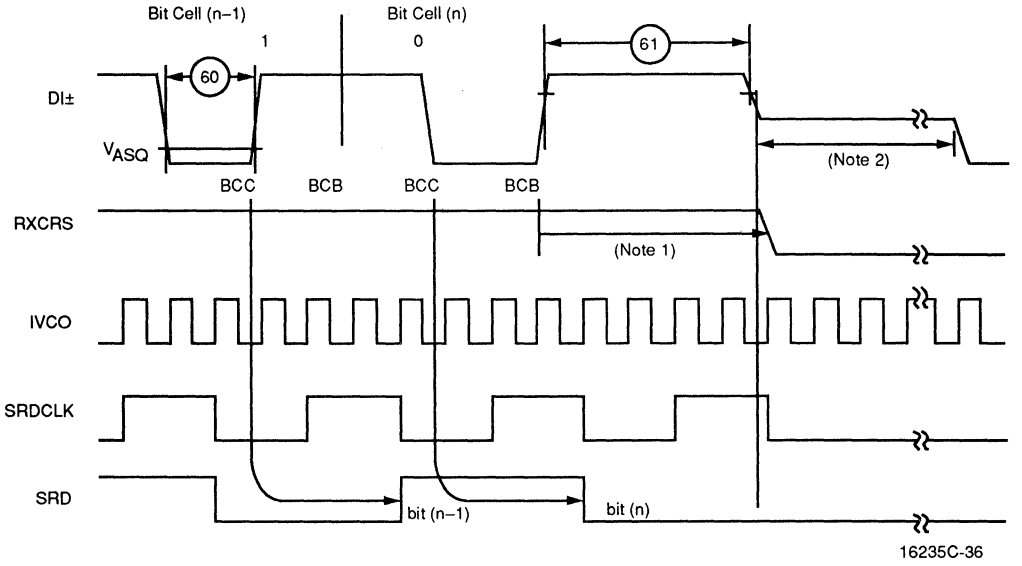
AUI Transmit Timing—End Transmit Delimiter (ETD)



Notes:

1. *Minimum pulse width >45 ns with amplitude > −160 mV.*
2. *SRD first decoded bit might not be defined until bit time 5.*
3. *First valid data bit.*
4. *IVCO and VCO ENABLE are internal signals shown for clarification only.*

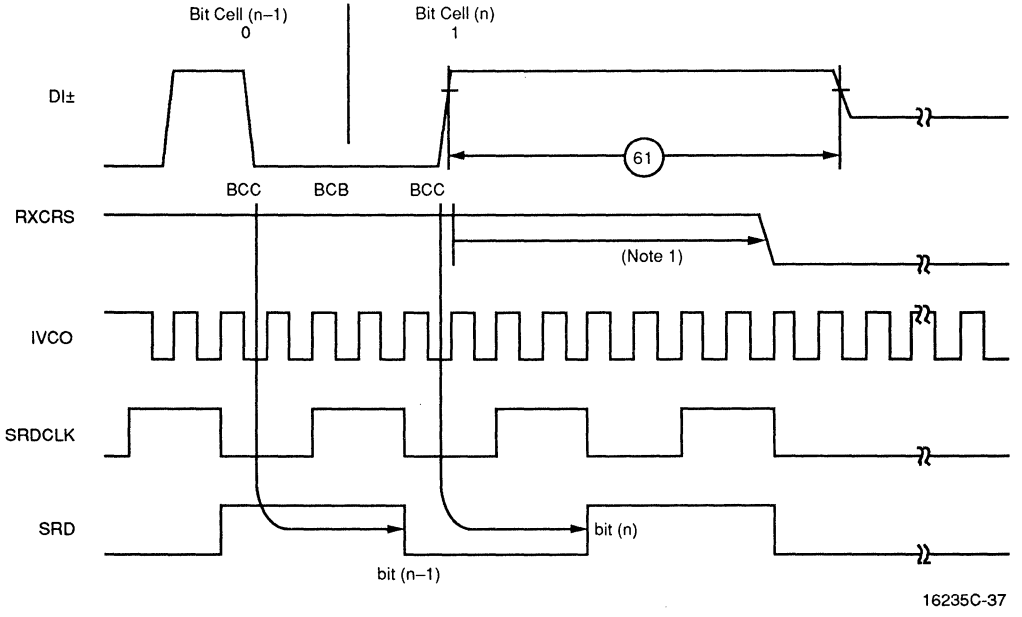
AUI Receive Timing—Start of Packet



Notes:

1. *RXCRS deasserts in less than 3 bit times after last DI± rising edge.*
2. *Start of next packet reception (2 bit times).*
3. *IVCO is an internal signal shown for clarification only.*

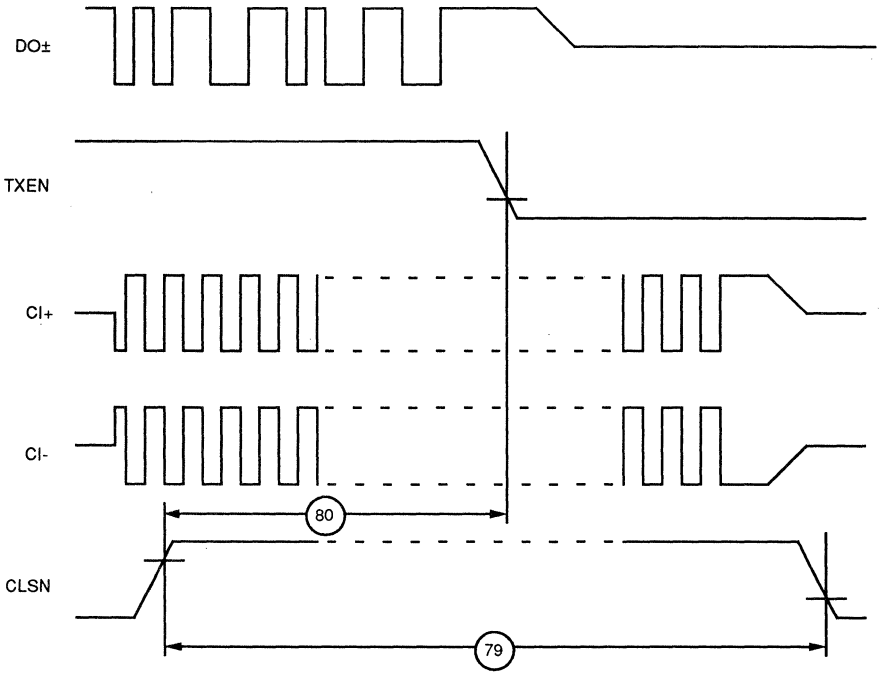
AUI Receive Timing—End of Packet (Last Bit = 0)



Notes:

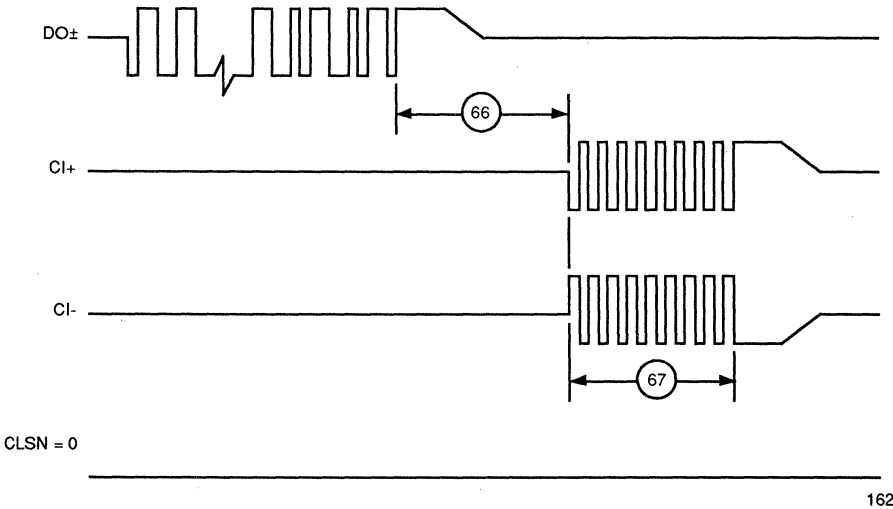
1. *RXCRS deasserts in less than 3 bit times after last DI± rising edge.*
2. *IVCO is an internal signal shown for clarification only.*

AUI Receive Timing—End of Packet (Last Bit = 1)



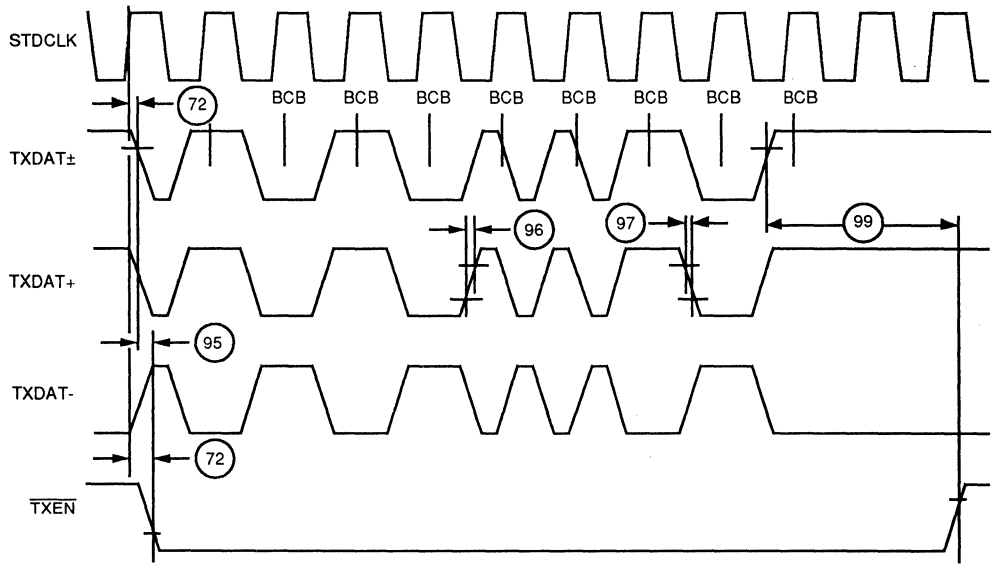
16235C-38

AUI Collision Timing



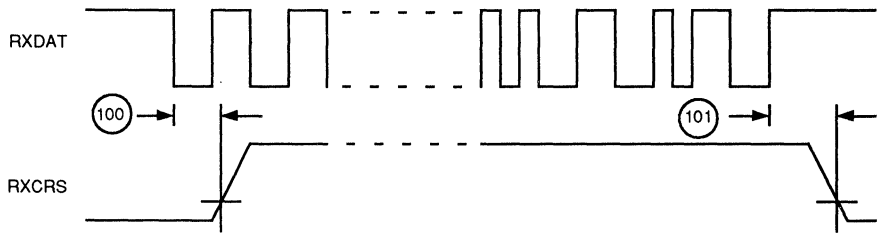
16235C-39

AUI SQE Test Timing



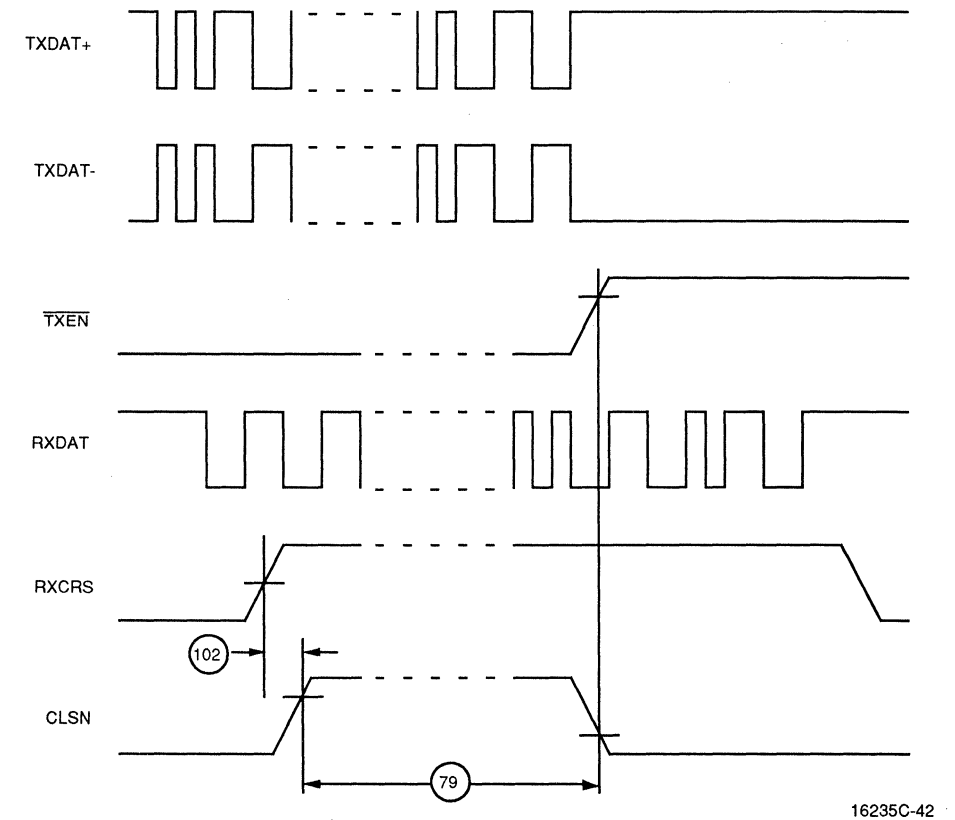
16235C-40

DAI Port Transmit Timing

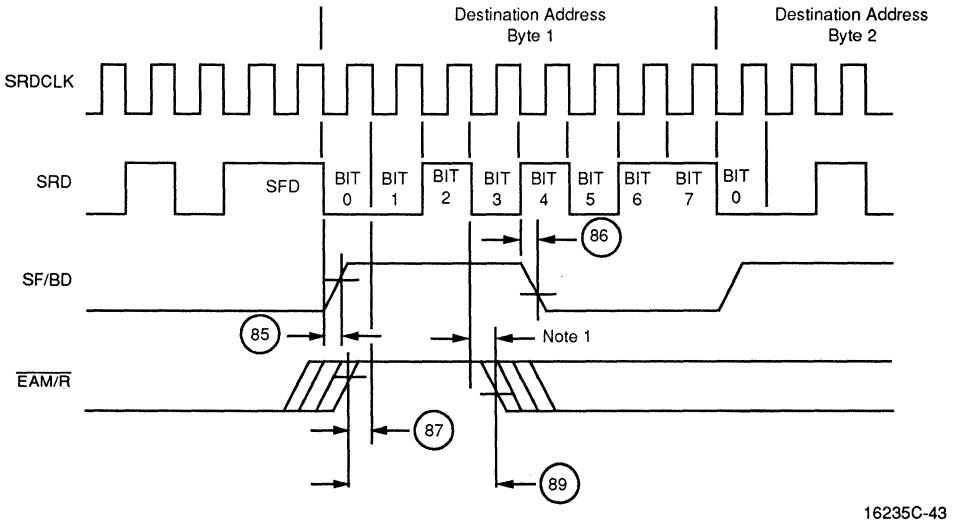


16235C-41

DAI Port Receive Timing

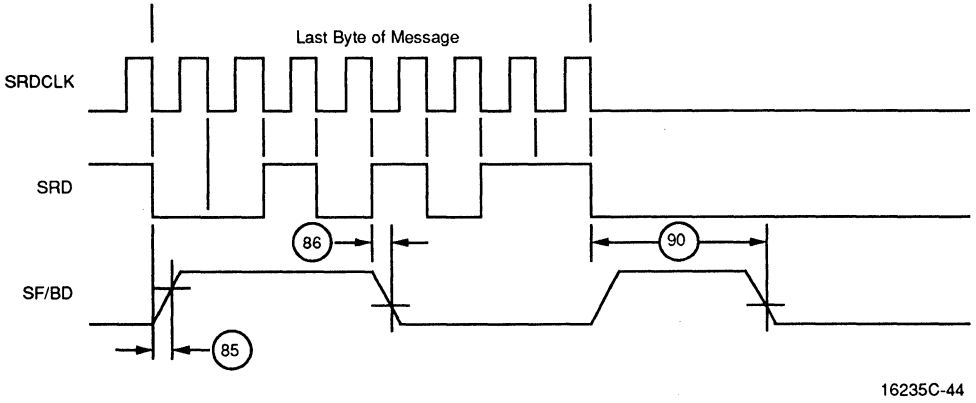


DAI Port Collision Timing



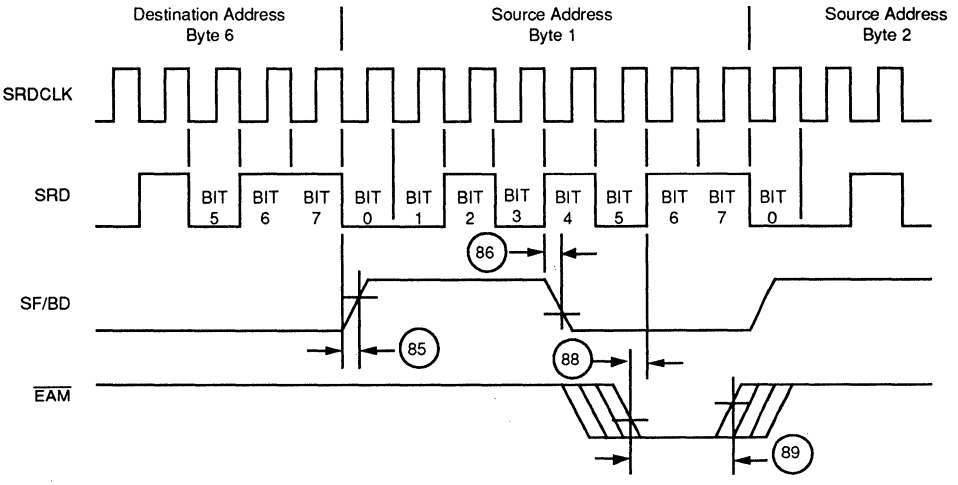
Note: *First assertion of EAM/R must occur after bit 2/3 boundary of preamble.*

EADI Feature Timing—Start of Address



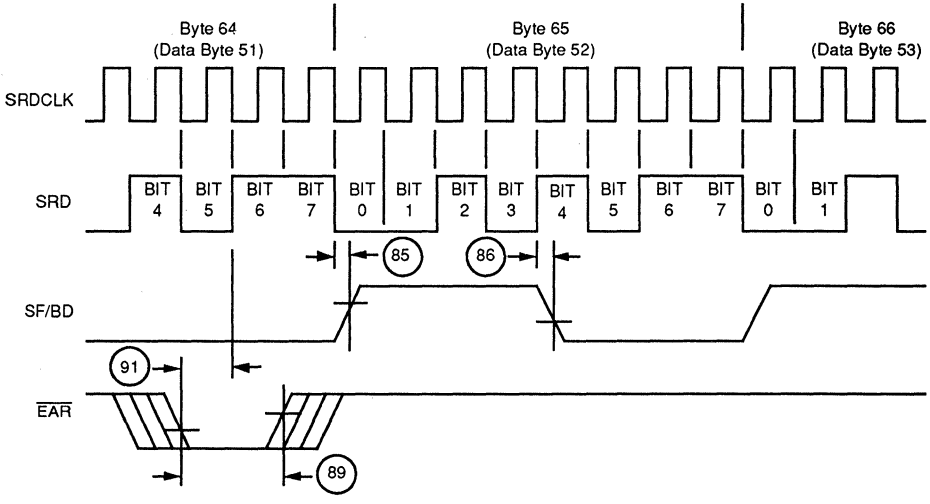
16235C-44

EADI Feature—End of Packet Timing



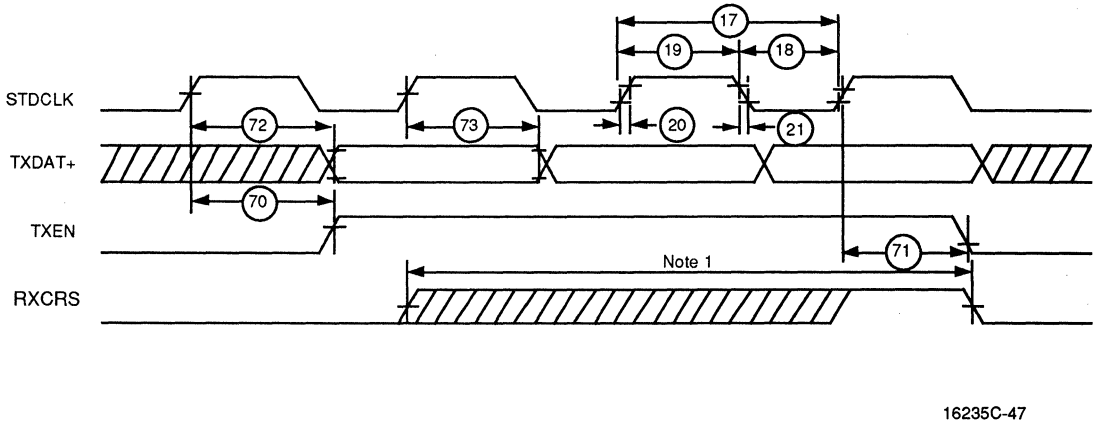
16235C-45

EADI Feature-Match Timing



16235C-46

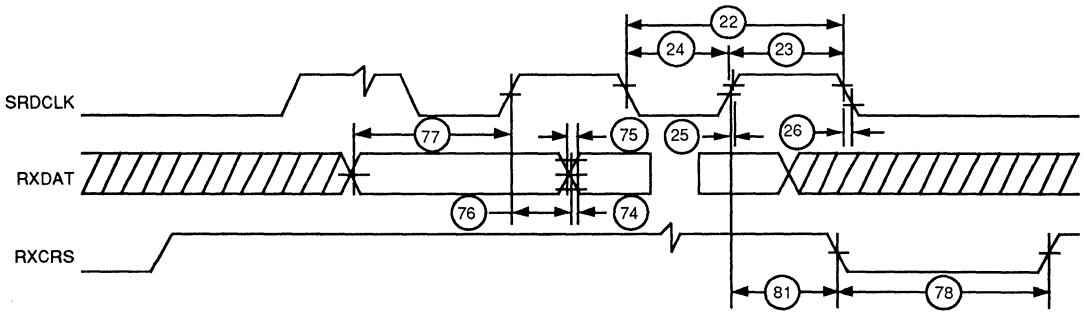
EADI Feature Reject Timing



16235C-47

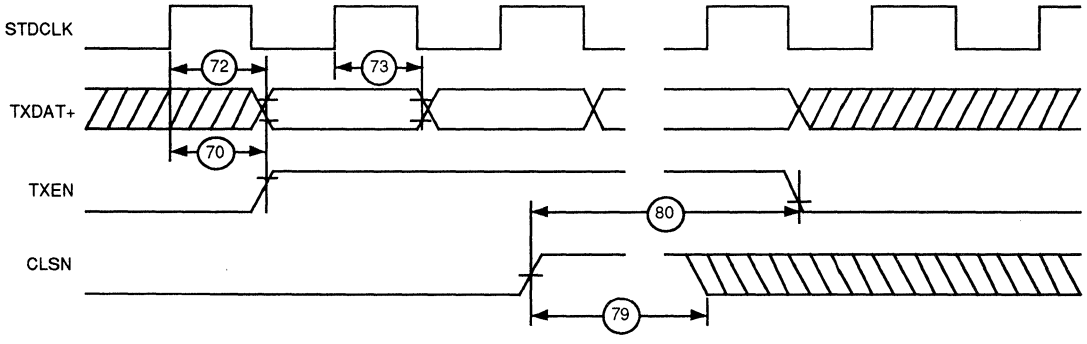
Note: *During transmit, the RXCRS input must be asserted (high) and remain active-high after TXEN goes active (high). If RXCRS is deasserted before TXEN is deasserted, LCAR will be reported (Transmit Frame Status) after the transmission is completed by the MACE device.*

GPSI Transmit Timing



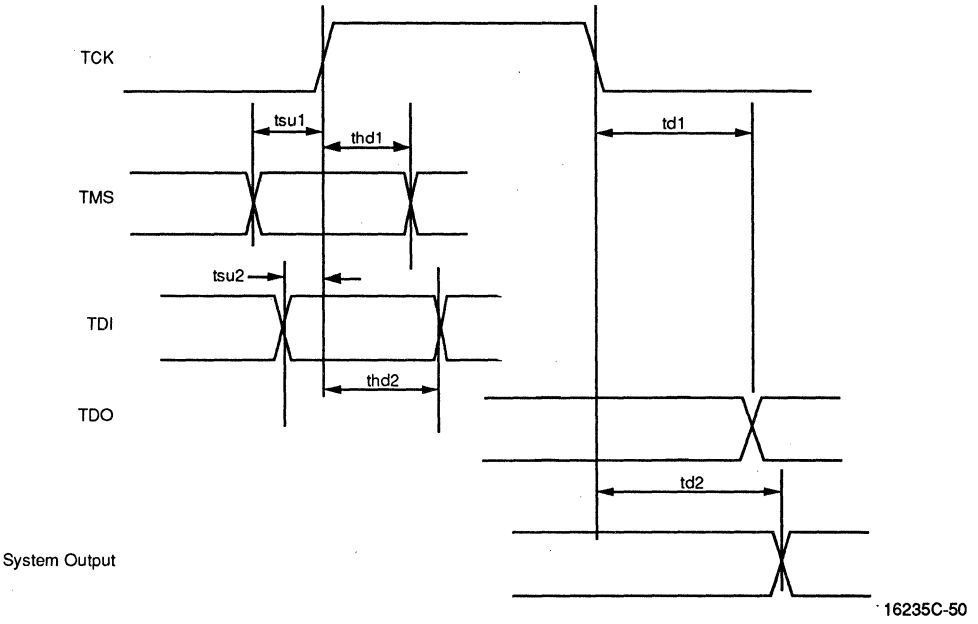
16235C-48

GPSI Receive Timing

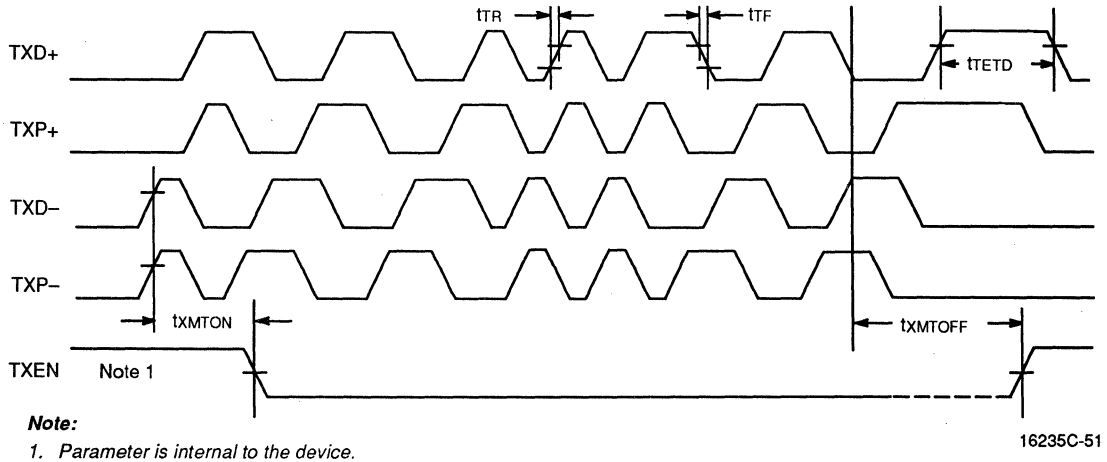


16235C-49

GPSI Collision Timing



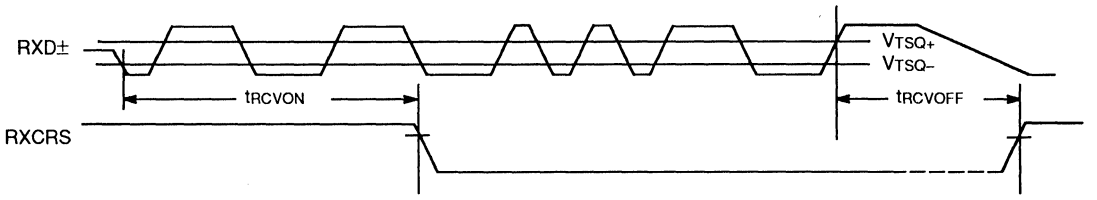
IEEE 1149.1 TAP Timing



Note:

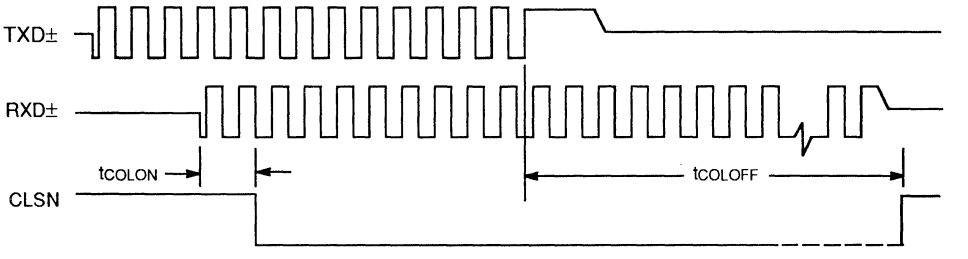
1. Parameter is internal to the device.

10BASE-T Transmit Timing



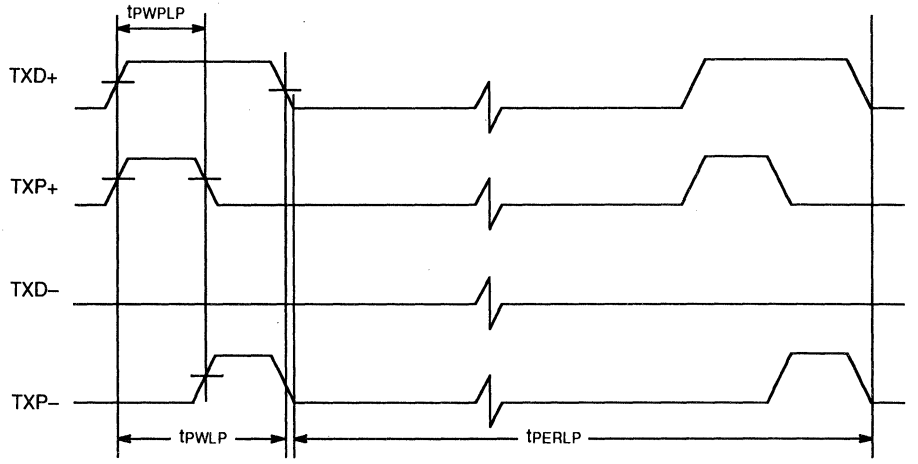
16235C-52

10BASE-T Receive Timing



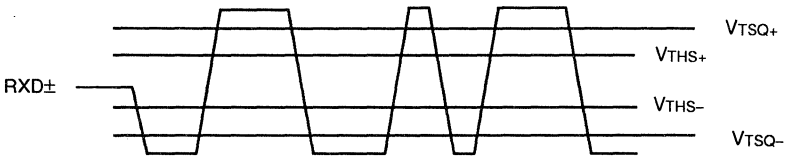
16235C-53

10BASE-T Collision Timing



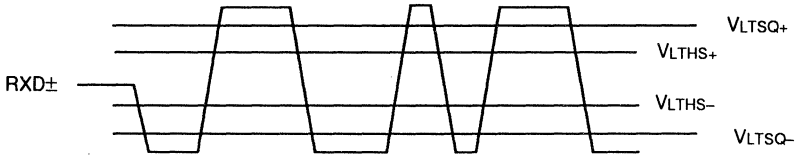
16235C-54

10BASE-T Idle Link Test Pulse



16235C-55

10BASE-T Receive Thresholds (LRT = 0)



16235C-56

10BASE-T Receive Thresholds (LRT = 1)

APPENDIX A

Logical Address Filtering For Ethernet

The purpose of logical (or group or multicast) addresses is to allow a group of nodes in a network to receive the same message. Each node can maintain a list of multicast addresses that it will respond to. The logical address filter mechanism in AMD Ethernet controllers is a hardware aide that reduces the average amount of host computer time required to determine whether or not an incoming packet with a multicast destination address should be accepted.

The logical address filter hardware is an implementation of a hash code searching technique commonly used by software programmers. If the multicast bit in the destination address of an incoming packet is set, the hardware maps this address into one of 64 categories then accepts or rejects the packet depending on whether or not the bit in the logical address filter register corresponding the selected category is set. For example, if the address maps into category 24, and bit 24 of the logical address filter register is set, the packet is accepted.

Since there are more than 10^{14} possible multicast addresses and only 64 categories, this scheme is far from unambiguous. This means that the software will still have to compare the address of a received packet with its list of acceptable multicast addresses to make the final decision whether to accept or discard the packet. However, the hardware prevents the software from having to deal with the vast majority of the unacceptable packets.

The efficiency of this scheme depends on the number of multicast groups that are used on a particular network and the number of groups to which a node belongs. At one extreme if a node happens to belong to 64 groups that map into 64 different categories, the hardware will accept all multicast addresses, and all filtering must be done by software. At the other extreme (which is closer to a practical network), if multicast addresses are assigned by the local administrator, and fewer than 65 groups are set up, the addresses can be assigned so that each address maps into a different category, and no software filtering will be needed at all.

In the latter case described above, a node can be made a member of several groups by setting the appropriate bits in the logical address filter register. The administrator can use the table *Mapping of Logical Address to Filter Mask* to find a multicast address that maps into a particular address filter bit. For example address 0000 0000 00BB maps into bit 15. Therefore, any node that has bit 15 set in its logical address filter register will receive all packets addressed to 0000 0000 00BB. (Addresses in this table are not shown in the standard Ethernet format. In the table the rightmost byte is the first byte to appear on the network with the least significant bit appearing first).

Driver software that manages a list of multicast addresses can work as follows. First the multicast address list and the logical address filter must be initialized. Some sort of management function such as the driver initialization routine passes to the driver a list of addresses. For each address in the list the driver uses a subroutine similar to the one listed in the Am7990 LANCE data sheet to set the appropriate bit in a software copy of the logical address filter register. When the complete list of addresses has been processed, the register is loaded.

Later, when a packet is received, the driver first looks at the Individual/Group bit of the destination address of the packet to find out whether or not this is a multicast address. If it is, the driver must search the multicast address list to see if this address is in the list. If it is not in the list, the packet is discarded.

The broadcast address, which consists of all ones is a special multicast address. Packets addressed to the broadcast address must be received by all nodes. Since broadcast packets are usually more common than other multicast packets, the broadcast address should be the first address in the multicast address list.

MAPPING OF LOGICAL ADDRESS TO FILTER MASK

Byte #	Bit #	LADRF Bit	Destination Address Accepted	Byte #	Bit #	LADRF Bit	Destination Address Accepted
0	0	0	85 00 00 00 00 00	4	0	32	21 00 00 00 00 00
0	1	1	A5 00 00 00 00 00	4	1	33	01 00 00 00 00 00
0	2	2	E5 00 00 00 00 00	4	2	34	41 00 00 00 00 00
0	3	3	C5 00 00 00 00 00	4	3	35	71 00 00 00 00 00
0	4	4	45 00 00 00 00 00	4	4	36	E1 00 00 00 00 00
0	5	5	65 00 00 00 00 00	4	5	37	C1 00 00 00 00 00
0	6	6	25 00 00 00 00 00	4	6	38	81 00 00 00 00 00
0	7	7	05 00 00 00 00 00	4	7	39	A1 00 00 00 00 00
1	0	8	2B 00 00 00 00 00	5	0	40	8F 00 00 00 00 00
1	1	9	0B 00 00 00 00 00	5	1	41	BF 00 00 00 00 00
1	2	10	4B 00 00 00 00 00	5	2	42	EF 00 00 00 00 00
1	3	11	6B 00 00 00 00 00	5	3	43	CF 00 00 00 00 00
1	4	12	EB 00 00 00 00 00	5	4	44	4F 00 00 00 00 00
1	5	13	CB 00 00 00 00 00	5	5	45	6F 00 00 00 00 00
1	6	14	8B 00 00 00 00 00	5	6	46	2F 00 00 00 00 00
1	7	15	BB 00 00 00 00 00	5	7	47	0F 00 00 00 00 00
2	0	16	C7 00 00 00 00 00	6	0	48	63 00 00 00 00 00
2	1	17	E7 00 00 00 00 00	6	1	49	43 00 00 00 00 00
2	2	18	A7 00 00 00 00 00	6	2	50	03 00 00 00 00 00
2	3	19	87 00 00 00 00 00	6	3	51	23 00 00 00 00 00
2	4	20	07 00 00 00 00 00	6	4	52	A3 00 00 00 00 00
2	5	21	27 00 00 00 00 00	6	5	53	83 00 00 00 00 00
2	6	22	67 00 00 00 00 00	6	6	54	C3 00 00 00 00 00
2	7	23	47 00 00 00 00 00	6	7	55	E3 00 00 00 00 00
3	0	24	69 00 00 00 00 00	7	0	56	CD 00 00 00 00 00
3	1	25	49 00 00 00 00 00	7	1	57	ED 00 00 00 00 00
3	2	26	09 00 00 00 00 00	7	2	58	AD 00 00 00 00 00
3	3	27	29 00 00 00 00 00	7	3	59	8D 00 00 00 00 00
3	4	28	A9 00 00 00 00 00	7	4	60	0D 00 00 00 00 00
3	5	29	89 00 00 00 00 00	7	5	61	2D 00 00 00 00 00
3	6	30	C9 00 00 00 00 00	7	6	62	6D 00 00 00 00 00
3	7	31	E9 00 00 00 00 00	7	7	63	4D 00 00 00 00 00

APPENDIX B

BSDL Description of Am79C940 MACE JTAG Structure

```
entity Am79C940 is
generic (PHYSICAL_PIN_MAP : string := "undefined");
port (
 DO0,DO1,DTV_L,INTR_L,LNKST_L,DXRCV_L,RDTREQ_L,RXPOL_L,SF_BD,SRD,
 TDO,TDTREQ_L,TXD0,TXD1,TXDAT0,TXP0,TXP1,XTAL2 : out bit;
 BE0_L,BE1_L,CI0,CI1,CS_L,DI0,DI1,EAM_R_L,EDSEL,FDS_L,RESET_L,RXD0,
 RXD1,R_W_L,SCLK,SLEEP_L,TCLK,TC_L,TDI,TMS,XTAL1 : in bit;
 ADD : in bit_vector (4 downto 0);
 CLSN,EOF_L,RXCRS,RXDAT,SRDCLK,STDCLK,TXDAT1,TXEN_L : inout bit;
 DBUS : inout bit_vector (15 downto 0);
 AVDD1,AVDD2,AVDD3,AVDD4,AVSS1,AVSS2,
 DVDD1,DVDD2,DVDDN,DVDDP,DVSS1,DVSS2,DVSSN1,DVSSN2,DVSSN3,DVSSP : linkage bit
);

use STD_1149_1_1990.all;  - get std 1149.1 1990 attributes and definitions

attribute PIN_MAP of am79c940 : entity is PHYSICAL_PIN_MAP;

constant PQFP_PACKAGE : PIN_MAP_STRING :=
  "SRDCLK:5, EAM_R_L:6, SRD:7, SF_BD:8, RESET_L:9, SLEEP_L:10," &
  "DVDDP:11," &
  "INTR_L:12, TC_L:13," &
  "DBUS:(36, 35, 33, 32, 31, 29, 25, 24, 23, 22, 21, 19, 18, 17, 16, 14)," &
  "DVSSN1:15, DVSSN2:20, DVDDN:34, DVSSN3:37," &
  "EOF_L:38, DTV_L:39, FDS_L:40, BE0_L:41, BE1_L:42, SCLK:43," &
  "TDTREQ_L:44, RDTREQ_L:45, ADD: (50, 49, 48, 47, 46)," &
  "R_W_L:55, CS_L:56, RXPOL_L:57, LNKST_L:58," &
  "TDO:59, TMS:60, TCK:61," &
  "DVSS1:62," &
  "TDI:63," &
  "DVDD1:64," &
  "RXD0:65, RXD1:66," &
  "AVDD1:67," &
  "TXP0:68, TXD0:69, TXP1:70, TXD1:71," &
  "AVDD2:72," &
  "XTAL1:73," &
  "AVSS1:74," &
  "XTAL2:75," &
  "AVSS2:79," &
  "DO0:81, DO1:82," &
  "AVDD3:83," &
  "DI0:84, DI1:85, CI0:86, CI1:87," &
  "AVDD4:88," &
  "DVDD2:89," &
  "DXRCV_L:90, EDSEL:91," &
```

```
  "DVSS2:92," &
  "TXDAT1:93, TXDAT0:94," &
  "DVSSP:95," &
  "STDCLK:96, TXEN_L:97, CLSN:98, RXDAT:99, RXCRS:100);
constant PLCC_PACKAGE : PIN_MAP_STRING :=
  "SRDCLK:12, EAM_R_L:13, SRD:14, SF_BD:15, RESET_L:16, SLEEP_L:17," &
  "DVDDP:18," &
  "INTR_L:19, TC_L:20," &
  "DBUS:(39, 38, 36, 35, 34, 33, 32, 31, 30, 29, 28, 26, 25, 24, 23, 21)," &
  "DVSSN1:22, DVSSN2:27, DVDDN:37, DVSSN3:39," &
  "EOF_L:41, DTV_L:42, FDS_L:43, BE0_L:44, BE1_L:45, SCLK:46," &
  "TDTREQ_L:47, RDTREQ_L:48, ADD: (53, 52, 51, 50, 49)," &
  "R_W_L:54, CS_L:55, RXPOL_L:56, LNKST_L:57," &
  "TDO:58, TMS:59, TCK:60," &
  "DVSS1:61," &
  "TDI:62," &
  "DVDD1:63," &
  "RXD0:64, RXD1:65," &
  "AVDD1:66," &
  "TXP0:67, TXD0:68, TXP1:69, TXD1:70," &
  "AVDD2:71," &
  "XTAL1:72," &
  "AVSS1:73," &
  "XTAL2:74," &
  "AVSS2:75," &
  "DO0:76, DO1:77," &
  "AVDD3:78," &
  "DI0:79, DI1:80, CI0:81, CI1:82," &
  "AVDD4:83," &
  "DVDD2:84," &
  "DXRCV_L:1, EDSEL:2," &
  "DVSS2:3," &
  "TXDAT1:4, TXDAT0:5," &
  "DVSSP:6," &
  "STDCLK:7, TXEN_L:8, CLSN:9, RXDAT:10, RXCRS:11);

attribute TAP_SCAN_IN of TDI : signal is true;
attribute TAP_SCAN_MODE of TMS : signal is true;
attribute TAP_SCAN_OUT of TDO : signal is true;
attribute TAP_SCAN_CLOCK of TCK : signal is (10.0e6, BOTH);

attribute INSTRUCTION_LENGTH of am79c940 : entity is 4;

attribute INSTRUCTION_OPCODE of am79c940 : entity is
  "Extest   (0000)," &
  "Idcode   (0001)," &
  "Sample   (0010)," &
  "Tribyp   (0011)," &
  "Setbyp   (0100)," &
  "Selftst  (0101)," &
  "Bypass   (0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111)";
attribute INSTRUCTION_CAPTURE of am79c940 : entity is "0001";
attribute INSTRUCTION_DISABLE of am79c940 : entity is "Tribyp";
```

```
attribute INSTRUCTION_PRIVATE of am79c940 : entity is "Selftst";
attribute IDCODE_REGISTER of am79c940 : entity is
  "0000" &         - 4 bit version
  "1001010000000000" &  - 16 bit part number
  "00000000001" & - 11 bit manufacturer
  "1";                  - mandatory LSB

attribute REGISTER_ACCESS of am79c940 : entity is
  "Boundary (Extest, Sample, Selftst)," &
  "Bypass (Bypass, Tribyp, Setbyp)," &
  "Idcode (Idcode)";

attribute BOUNDARY_CELL of am79c940 : entity is "BC_1,BC_4";
attribute BOUNDARY_LENGTH of am79c940 : entity is 99

- num cell  port        function    safe  (ccell      disval      rslt)
  "98 (BC_1,      *,          internal,   0)," &      - COL_SQL
  "97 (BC_1,      *,          internal,   0)," &      - AUI_NSQ
  "96 (BC_1,      *,          internal,   0)," &      - XMTD
  "95 (BC_1,      *,          internal,   0)," &      - AUIEN
  "94 (BC_1,      *,          internal,   0)," &      - TXD0L
  "93 (BC_1,      *,          internal,   0)," &      - TXP0L
  "92 (BC_1,      *,          internal,   0)," &      - TXEN
  "91 (BC_1,      *,          internal,   0)," &      - PSQ_O xor FIXPOL
  "90 (BC_1,      *,          internal,   0)," &      - CLK20
  "89 (BC_1,      DXRCV_L,    output3,    X,    88,   0,    Z)," &
  "88 (BC_1,      *,          control,    0)," &      - TRI PWDNBAR
  "87 (BC_1,      EDSEL,            input,            1)," &
  "86 (BC_1,      TXDAT1,           input,            1)," &
  "85 (BC_1,      TXDAT1,           output3,    X,    83,   0,    Z)," &
  "84 (BC_1,      TXDAT0,           output3,    X,    83,   0,    Z)," &
  "83 (BC_1,      *,          control,    0)," &      - TRI TXDAT+/TXDAT-
  "82 (BC_1,      STDCLK,           input,            0)," &
  "81 (BC_1,      STDCLK,           output3,    X,    80,   0,    Z)," &
  "80 (BC_1,      *,          control,    0)," &      - TRI STDCLK
  "79 (BC_1,      TXEN_L,           input,            0)," &
  "78 (BC_1,      TXEN_L,           output3,    X,    77,   0,    Z)," &
  "77 (BC_1,      *,          control,    0)," &      - TRI TXEN_L
  "76 (BC_1,      CLSN,       input,            0)," &
  "75 (BC_1,      CLSN,       output3,    X,    74,   0,    Z)," &
  "74 (BC_1,      *,          control,    0)," &      - TRI CLSN
  "73 (BC_1,      RXDAT,            input,            0)," &
  "72 (BC_1,      RXDAT,            output3,    X,    71,   0,    Z)," &
  "71 (BC_1,      *,          control,    0)," &      - TRI RXDAT
  "70 (BC_1,      RXCRS,            input,            0)," &
  "69 (BC_1,      RXCRS,            output3,    X,    68,   0,    Z)," &
  "68 (BC_1,      *,          control,    0)," &      - TRI RXCRS
  "67 (BC_1,      SRDCLK,           input,            0)," &
  "66 (BC_1,      SRDCLK,           output3,    X,    65,   0,    Z)," &
  "65 (BC_1,      *,          control,    0)," &      - TRI SRDCLK
  "64 (BC_1,      EAM_R,            input,            0)," &
  "63 (BC_1,      SRD,        output3,    X,    61,   0,    Z)," &
  "62 (BC_1,      SF_BD,            output3,    X,    61,   0,    Z)," &
```

```
"61 (BC_1,      *,          control,    0)," &      - TRI SF_BD/SRD
"60 (BC_1,      RESET_L,    input,          1)," &
"59 (BC_1,      SLEEP_L,    input,          1)," &
"58 (BC_1,      INTR_L,         output3,    1,    57,   0,  Weak1)," &
"57 (BC_1,      *,          control,    0)," &      - TRI INTR_L
"56 (BC_1,      TC_L,       input,          1)," &
"55 (BC_1,      DBUS(0),    input,          0)," &
"54 (BC_1,      DBUS(0),    output3,    X,    35,   0,    Z); &
"53 (BC_1,      DBUS(1),    input,          0)," &
"52 (BC_1,      DBUS(1),    output3,    X,    35,   0,    Z); &
"51 (BC_1,      DBUS(2),    input,          0)," &
"50 (BC_1,      DBUS(2),    output3,    X,    35,   0,    Z); &
"49 (BC_1,      DBUS(3),    input,          0)," &
"48 (BC_1,      DBUS(3),    output3,    X,    35,   0,    Z); &
"47 (BC_1,      DBUS(4),    input,          0)," &
"46 (BC_1,      DBUS(4),    output3,    X,    35,   0,    Z); &
"45 (BC_1,      DBUS(5),    input,          0)," &
"44 (BC_1,      DBUS(5),    output3,    X,    35,   0,    Z); &
"43 (BC_1,      DBUS(6),    input,          0)," &
"42 (BC_1,      DBUS(6),    output3,    X,    35,   0,    Z); &
"41 (BC_1,      DBUS(7),    input,          0)," &
"40 (BC_1,      DBUS(7),    output3,    X,    35,   0,    Z); &
"39 (BC_1,      DBUS(8),    input,          0)," &
"38 (BC_1,      DBUS(8),    output3,    X,    35,   0,    Z); &
"37 (BC_1,      DBUS(9),    input,          0)," &
"36 (BC_1,      DBUS(9),    output3,    X,    35,   0,    Z); &
"35 (BC_1,  *,          control,    0)," &      - TRI DBUS(9:0)
"34 (BC_1,      DBUS(10),   input,          0)," &
"33 (BC_1,      DBUS(10),   output3,    X,    22,   0,    Z); &
"32 (BC_1,      DBUS(11),   input,          0)," &
"31 (BC_1,      DBUS(11),   output3,    X,    22,   0,    Z); &
"30 (BC_1,      DBUS(12),   input,          0)," &
"29 (BC_1,      DBUS(12),   output3,    X,    22,   0,    Z); &
"28 (BC_1,      DBUS(13),   input,          0)," &
"27 (BC_1,      DBUS(13),   output3,    X,    22,   0,    Z); &
"26 (BC_1,      DBUS(14),   input,          0)," &
"25 (BC_1,      DBUS(14),   output3,    X,    22,   0,    Z); &
"24 (BC_1,      DBUS(15),   input,          0)," &
"23 (BC_1,      DBUS(15),   output3,    X,    22,   0,    Z); &
"22 (BC_1,      *,          control,    0)," &      - TRI DBUS(15:10)
"21 (BC_1,      EOF_L,          input,          1)," &
"20 (BC_1,      EOF_L,          output3,    X,    19,   0,    Z); &
"19 (BC_1,      *,          control,    0)," &      - TRI EOF_L
"18 (BC_1,      DTV_L,          output3,    X,    17,   0,    Z); &
"17 (BC_1,      *,          control,    0)," &      - TRI DTV_L
"16 (BC_1,      FDS_L,          input,          1)," &
"15 (BC_1,      BE0_L,          input,          1)," &
"14 (BC_1,      BE1_L,          input,          1)," &
"13 (BC_4,      SCLK,       clock,          1)," &
"12 (BC_1,      TDTREQ_L,   output3,    X,    10,   0,    Z); &
"11 (BC_1,      RDTREQ_L,   output3,    X,    10,   0,    Z); &
"10 (BC_1,      *,          control,    0)," &      - TRI TDTREQ_L/RDTREQ_L
"9  (BC_1,      ADD(0),         input,          0)," &
```

```
  "8   (BC_1,       ADD(1),          input,              0)," &
  "7   (BC_1,       ADD(2),          input,              0)," &
  "6   (BC_1,       ADD(3),          input,              0)," &
  "5   (BC_1,       ADD(4),          input,              0)," &
  "4   (BC_1,       R_W_L,           input,              1)," &
  "3   (BC_1,       CS_L,        input,              1)," &
  "2   (BC_1,       RXPOL_L,     output3,    X,      0,     0,  Weak1)," &
  "1   (BC_1,       LNKST_L,     output3,    X,      0,     0,  Weak1)," &
  "0   (BC_1,       *,           control,    0)";  -  TRI RXPOL_L/LNKST_L

end am79c940
```

FINAL

Am7992B

Serial Interface Adapter (SIA)

Advanced Micro Devices

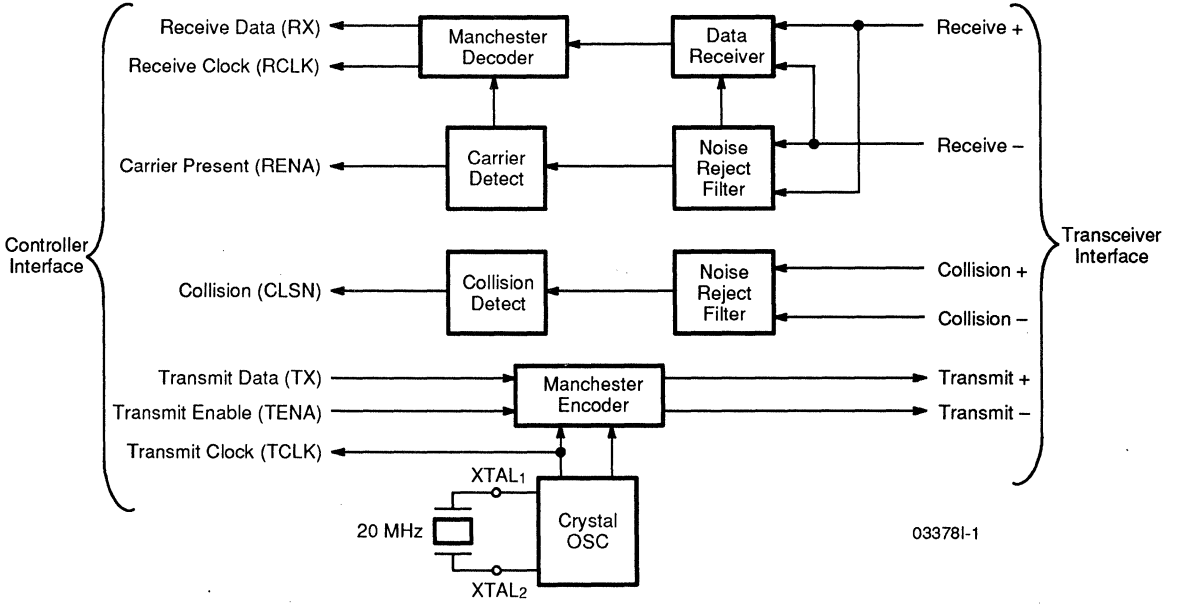
DISTINCTIVE CHARACTERISTICS

- **Compatible with IEEE 802.3/Ethernet/ Cheapernet specifications**
- **Crystal/TTL oscillator controlled Manchester Encoder**
- **Manchester Decoder acquires clock and data within four bit times with an accuracy of ±3 ns**
- **Guaranteed carrier and collision detection squelch threshold limits**
  - — Carrier/collision detected for inputs greater than –275 mV
  - — No carrier/collision for inputs less than –175 mV
- **Input signal conditioning rejects transient noise**
  - — Transients <10 ns for collision detector inputs
  - — Transients <20 ns for carrier detector inputs
- **Receiver decodes Manchester data with worst case ±19 ns of clock jitter (at 10 MHz)**
- **TTL compatible host interface**
- **Transmit accuracy +0.01% (without adjustments)**

GENERAL DESCRIPTION

The Am7992B Serial Interface Adapter (SIA) is a Manchester Encoder/Decoder compatible with IEEE 802.3, Cheapernet and Ethernet specifications. In an IEEE 802.3/Ethernet application, the Am7992B interfaces the Am7990 Local Area Network Controller for Ethernet (LANCE) to the Ethernet transceiver cable, acquires clock and data within four bit times, and decodes Manchester data with worst case ±19 ns phase jitter at 10 MHz. SIA provides both guaranteed signal threshold limits and transient noise suppression circuitry in both data and collision paths to minimize false start conditions.

BLOCK DIAGRAM



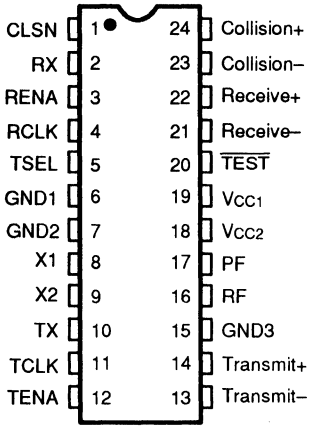
Publication# 03378 Rev. I Amendment/0
Issue Date: May 1993

RELATED PRODUCTS

Part No.	Description
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am7996	IEEE 802.3/Ethernet/Cheapernet/Transceiver
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)

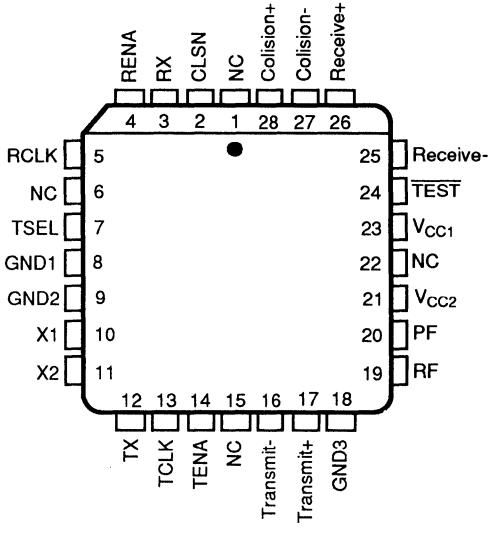
CONNECTION DIAGRAMS

DIP



03378I-2

PLCC



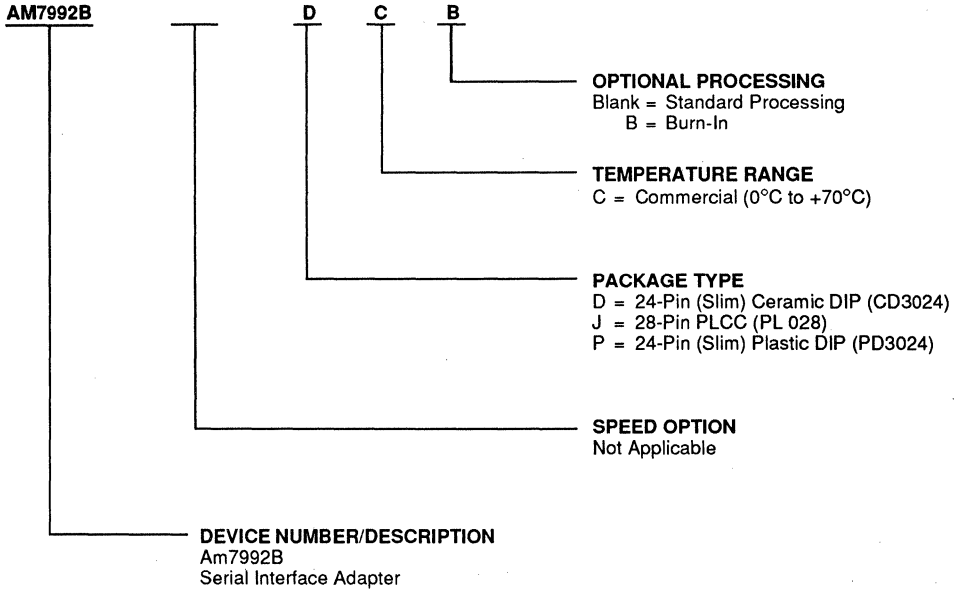
03378I-3

Note:
Pin 1 is marked for orientation.

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM7992B	DC, DCB, JC, JCTR, PC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

PIN DESCRIPTION

CLSN
Collision (Output, TTL Active HIGH)

Signals at the Collision± terminals meeting threshold and pulse width requirements will produce a logic HIGH at CLSN output. When no signal is present at Collision±, CLSN output will be LOW.

RX
Receive Data (Output)

A MOS/TTL output, recovered data. When there is no signal at Receive± and $\overline{TEST}$ is HIGH, RX is HIGH. RX is actuated with RCLK and remains active until RENA is deasserted at the end of message. During reception, RX is synchronous with RCLK and changes after the rising edge of RCLK. When $\overline{TEST}$ is LOW, RX is enabled.

RENA
Receive Enable (Output, TTL Active HIGH)

When there is no signal at Receive+ RENA is LOW. Signals meeting threshold and pulse width "on" requirements will produce a logic HIGH at RENA. When RENA is HIGH, Receive+ signals meeting threshold and pulse width "off" requirements will produce a LOW at RENA.

RCLK
Receive Clock (Output)

A MOS/TTL output, recovered clock. When there is no signal at Receive± and $\overline{TEST}$ is HIGH, RCLK is LOW. RCLK is activated 1/4 bit time after the second negative Manchester preamble clock transition at Receive±, and remains active until after an end of message. When $\overline{TEST}$ is LOW, RCLK is enabled and meets minimum pulse width specifications.

TX
Transmit (Input)

TTL-compatible input. When TENA is HIGH, signals at TX meeting setup and hold time to TCLK will be encoded as normal Manchester at Transmit+ and Transmit–.

TX HIGH: Transmit+ is negative with respect to Transmit– for first half of data bit cell.

TX LOW: Transmit+ is positive with respect to Transmit– for first half of data bit cell.

TENA
Transmit Enable (Input)

TTL-compatible input. Active HIGH data encoder enable. Signals meeting setup and hold time to TCLK will allow encoding of Manchester data from TX to Transmit+ and Transmit–.

TCLK
Transmit Clock (Output)

MOS/TTL output. TCLK provides symmetrical HIGH and LOW clock signals at data rate for reference timing of data to be encoded. It also provides clock signals for the controller chip (Am7990 – LANCE) and an internal timing reference for receive path voltage controlled oscillators.

Transmit+, Transmit–
Transmit (Outputs)

A differential line output. This line pair is intended to operate into terminated transmission lines. For signals meeting setup and hold time to TCLK at TENA and TX, Manchester clock and data are outputted at Transmit+/Transmit–. When operating into a 78 Ω terminated transmission line, signaling meets the required output levels and skew for both Ethernet and IEEE 802.3 drop cables.

Receive+, Receive–
Receiver (Inputs)

A differential input. A pair of internally biased line receivers consisting of a carrier detect receiver with offset threshold and noise filtering to detect the line activity, and a data recovery receiver with no offset for Manchester data decoding.

Collision+, Collision–
Collision (Inputs)

A differential input. An internally biased line receiver input with offset threshold and noise filtering. Signals at Collision± have no effect on data-path functions.

TSEL
Transmit Mode Select (Output, Open Collector; Input, Sense Amplifier)

TSEL LOW: Idle transmit state Transmit+ is positive with respect to Transmit–.

TSEL HIGH: Idle transmit state Transmit+ and Transmit– are equal, providing "zero" differential to operate transformer coupled loads.

When connected with an RC network, TSEL is held LOW during transmission. At the end of transmission the open collector output is disabled, allowing TSEL to rise and provide a smooth transmission from logic HIGH to "zero" differential idle. Delay and output return to zero are externally controlled by the RC network at TSEL and Transmit± load inductance.

X1, X2
Biased Crystal Oscillator (Input)

X1 is the input and X2 is the bypass port. When connected for crystal operation, the system clock which appears at TCLK is half the frequency of the crystal oscillator. X1 may be driven from an external source of two times the data rate.

RF
Frequency Setting Voltage Controlled Oscillator (V_{CO}) Loop Filter (Output)

This loop filter output is a reference voltage for the receive path phase detector. It also is a reference for timing noise immunity circuits in the collision and receive enable path. Nominal reference V_{CO} gain is 1.25 TCLK frequency MHz/V.

PF
Receive Path V_{CO} Phase-Lock Loop Filter (Input)

This loop filter input is the control for receive path loop damping. Frequency of the receive V_{CO} is internally limited to transmit frequency ±12%. Nominal receive V_{CO} gain is 0.25 reference V_{CO} gain MHz/V.

$\overline{TEST}$
Test Control (Input)

A static input that is connected to V_{CC} for Am7992B/Am7990 operation and to Ground for testing of Receive± path threshold and RCLK output high parameters. When $\overline{TEST}$ is grounded, RX is enabled and RCLK is enabled except during Clock acquisition when RCLK is HIGH.

GND1
High Current Ground

GND2
Logic Ground

GND3
Voltage Controlled Oscillator Ground

V_{CC1}
High Current and Logic Supply

V_{CC2}
Voltage Controlled Oscillator Supply

FUNCTIONAL DESCRIPTION

The Am7992B Serial Interface Adapter (SIA) has three basic functions. It is a Manchester Encoder/line driver in the transmit path, a Manchester Decoder with noise filtering and quick lock-on characteristics in the receive path, and a signal detect/converter (10 MHz differential to TTL) in the collision path. In addition, the SIA provides the interface between the TTL logic environment of the Local Area Network Controller for Ethernet (LANCE) and the differential signaling environment in the transceiver cable.

Transmit Path

The transmit section encodes separate clock and NRZ data input signals meeting the set-up and hold time to TCLK at TENA and TX, into a standard Manchester II serial bit stream. The transmit outputs (Transmit+ /Transmit–) are designed to operate into terminated transmission lines. When operating into a 78 Ω terminated transmission line, signaling meets the required output levels and skew for IEEE 802.3/Ethernet/ Cheapernet.

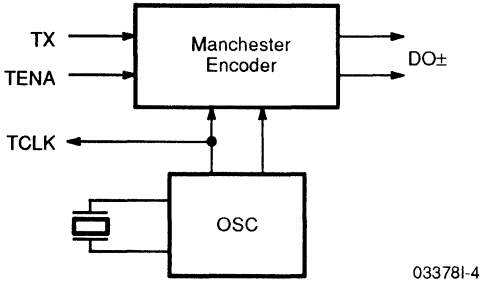


Figure 1. Transmit Section

Transmitter Timing and Operation

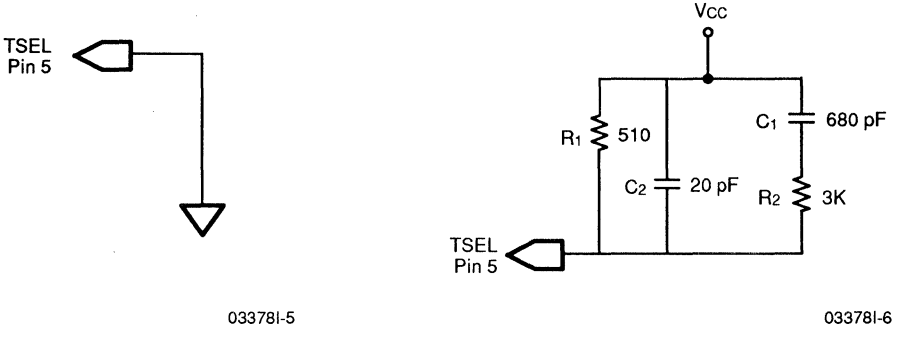
A 20 MHz fundamental mode crystal oscillator provides the basic timing reference in the SIA. It is divided by two to create the Transmit Clock reference (TCLK). Both 20 MHz and 10 MHz clocks are fed into the Manchester Encoder to generate the transitions in the encoded data stream. The 10 MHz clock, TCLK, is used by the SIA to internally synchronize Transmit (TX) data and Transmit Enable (TENA). TCLK is also used as a stable bit rate clock by the receive section of the SIA and by other devices in the system (the Am7990 LANCE uses TCLK to drive its internal state machine). The oscillator may use an external .005% crystal or an external TTL-level input as a reference which will achieve a transmit accuracy of .01% (no external adjustments are required).

Transmission is enabled when TENA is activated. As long as TENA remains HIGH, signals at TX will be encoded as Manchester and will appear at Transmit+ and Transmit–. When TENA goes LOW, the differential transmit outputs go to one of two idle states determined by the circuit configuration of TSEL:

TSEL HIGH: The idle state of Transmit± yields "zero" differential to operate transformer-coupled loads (see Figure 2, Transmitter Timing – End of Transmission waveform diagram and Typical Performance Curve diagram).

TSEL LOW: In this idle state, Transmit+ is positive to Transmit– (logical HIGH) (see Figures and diagrams as referenced above).

The End of Transmission – Return to Zero is determined by the external RX network at TSEL and by the load at Transmit±.



A. TSEL LOW **B. TSEL HIGH**

Figure 2. Transmit Mode Select (TSEL) Connection

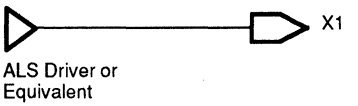


Figure 3. TTL Clock Driver Circuit for X1

SIA Oscillator

Specification for External Crystal

When using a crystal to drive the Am7992B oscillator, the following crystal specification should be used to ensure a transmit accuracy of 0.01%:

	Limit			
	Min	Nominal	Max	Unit
Resonant Frequency Error with C_L = 50 pF	–50	0	+50	PPM
Change in Resonant Frequency Temperature with C_L = 50 pF	–40		+40	PPM
Parallel Resonant Frequency with C_L= 50 pF		20		MHz
Motional Crystal Capacitance, C_1		0.022		pF

Some crystal manufacturers have generated crystals to this specification. One such manufacturer is Reeves-Hoffman. Their ordering part number tor this crystal is RH#04-20423-312. Another manufacturer is Epson – Part #MA 506-200M-50 pF which is a surface-mounted crystal.

Specification for External TTL Level

When driving the oscillator from an external clock source, X2 must be left floating (unconnected). An external clock having the following characteristics must be used to ensure less than +0.5 ns jitter at Transmit+ (see the X1 Driven from External Source waveform diagram and the TTL Clock Driver Circuit for X1, Figure 3):

Clock Frequency: 20 MHz ±0.01%

Rise/Fall Time (t_R/t_F): <4 ns, monotonic

X1 HIGH/LOW Time (t_{HIGH}/t_{LOW}): > 20 ns

X1 Falling Edge to Falling Edge Jitter:
< ±0.2 ns at 1.5 V input

Receiver Path

The principle functions of the Receiver are to signal the LANCE that there is information on the receive pair, and separate the incoming Manchester-encoded data stream into clock and NRZ data.

The Receiver section (see Figures 4 and 5) consists of two parallel paths. The receive data path is a zero threshold, wide bandwidth line receiver. The carrier path is an offset threshold bandpass detecting line receiver. Both receivers share common bias networks to allow operation over an input common mode range of 0 V to 5.5 V.

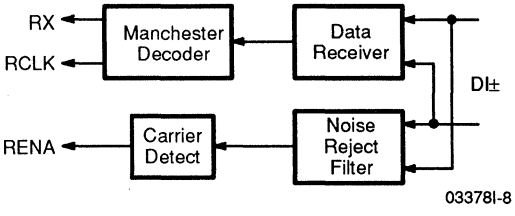


Figure 4. Receiver

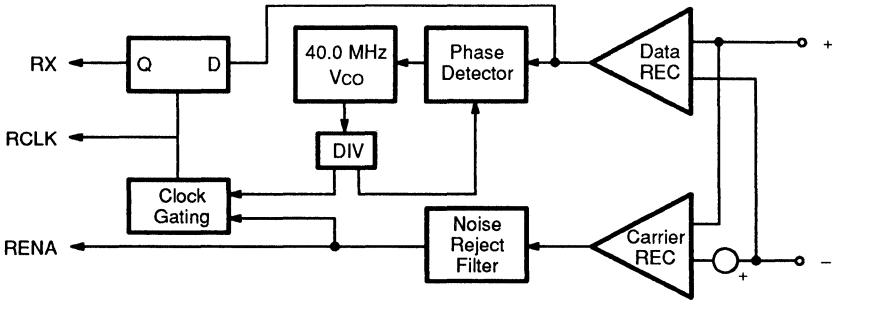


Figure 5. Receiver Section Detail

Input Signal Conditioning

The Carrier Receiver detects the presence of an incoming data packet by discerning and rejecting noise from expected Manchester data. It also controls the stop and start of the phase-lock loop during clock acquisition. In the Am7992B, clock acquisition requires a valid Manchester bit pattern of 1010 to lock on the incoming message (see Receive Timing – Start of Reception Clock Acquisition waveform diagram).

Transient noise pulses less than 20 ns wide are rejected by the Carrier Receiver as noise and DC inputs more positive than –175 mV are also suppressed. Carrier is detected for input signal wider than 45 ns with amplitude more negative than –275 mV. When input amplitude and pulse width conditions are met at Receive±, RENA is asserted and a clock acquisition cycle is initiated.

Clock Acquisition

When there is no activity at Receive± (receiver is idle), the receive oscillator is phase locked to TCLK. The first negative clock transition (first valid Manchester "0") after RENA is asserted interrupts the receive oscillator and presets the INTRCLK (internal clock) to the HIGH state. The oscillator is then restarted at the second Manchester "0" (bit time 4) and is phase locked to it. As a result, the SIA acquires the clock from the incoming Manchester bit stream in four bit times with "1010" Manchester bit pattern. The 10 MHz INTRCLK and INTPLLCLK are derived from the internal oscillator which runs at 4 times the data rate (40.0 MHz). The three clocks generated internally are utilized in the following manner:

INTRCLK: After clock acquisition, INTRCLK strobes the incoming data at 1/4 bit time. Receive data path sets the input to the data decode register (Figure 5).

INTPLLCLK: At clock acquisition, INTPLLCLK is phase locked to the incoming Manchester clock transition at Bit Cell Center (BCC). The transition at BCC is compared to INTPLLCLK and phase correction is applied to maintain INTRCLK at 1/4 bit time in the Manchester cell.

INTCARR: From start to end of a message, INTCARR is active and establishes RENA Turn-off synchronously with RCLK rising edge. Internal carrier goes active when there is a negative transition that is more negative than –275 mV and has a pulse width greater or equal to 45 ns. Internal carrier goes inactive typically 155 ns after the last positive transition at Receive±.

When $\overline{\text{TEST}}$ is strapped LOW, RCLK and RX are enabled 1/4 bit time after clock acquisition in bit cell 5. RX is at HIGH state when the receiver is idle and $\overline{\text{TEST}}$ is strapped HIGH (no RLCK). RX, however, is undefined when clock is acquired and may remain HIGH or change to LOW state whenever RCLK is enabled. At the 1/4 bit time of clock transition in bit cell 5, RCLK makes its first external transition. It also strobes the incoming fifth bit Manchester "1." RX may make a transition after the RCLK rising edge in bit cell 5, but its state is still undefined. The Manchester "1" at bit 5 is clocked to RX output at 1/4 bit time in bit cell 6.

PLL Tracking

After clock acquisition, the INTPLLCLK is compared to the incoming transitions at BCC and the resulting phase error is applied to a correction circuit. This circuit ensures that INTPLLCLK remains locked on the received signal. Individual bit cell phase corrections of the VCO are limited to 10% of the phase difference between BCC and INTPLLCLK. Hence, input data jitter is reduced in RCLK by 10 to 1.

Carrier Tracking and End of Message

The carrier receiver monitors Receive± input after RENA is asserted for an end of message. INTCARR deasserts typically 155 ns to 165 ns after the incoming message transitions positive. This initiates the end of reception cycle. INTCARR is strobed at 3/4 bit time by the

falling edge of INTRCLK. The time delay from the last rising edge of the message to INTCARR deassert allows the last bit to be strobed by RCLK and transferred by the LANCE without an extra bit at the end of message. When RENA deasserts (see Receive Timing—End of Reception waveform diagrams), a RENA hold off timer inhibits RENA assertion for at least 120 ns.

Data Decoding

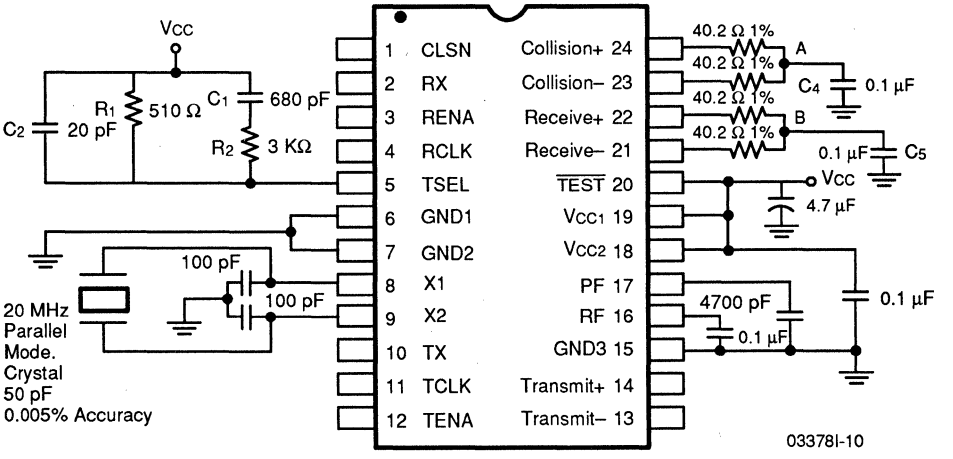
The data receiver is a comparator with clocked output to minimize noise sensitivity to the Receive± inputs. Input error (VIRD) is less than ±35 mV to minimize sensitivity to input rise and fall time. RCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit and clocks the data out at RX on the following RCLK. The data receiver also generates the signal used for phase detector comparison to the internal Am7992B V_{CO}.

Differential I/O Terminations

The differential input for the Manchester data (Receive±) is externally terminated by two 40.2 ohm ±1% resistors and one optional common-mode bypass capacitor. The differential input impedance, Z_{IDF} and the common-mode input, Z_{ICM}. are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. The Collision± differential inputs are terminated in exactly the same way as the receive inputs (see Figure 6).

Collision Detection

A transceiver detects collisions on the network and generates a 10 MHz signal at the Collision± inputs. This collision signal passes through an input stage which detects signal levels and pulse duration. When the signal is detected by the Am7992B it sets the CLSN line HIGH. This condition continues for approximately 160 ns after the last LOW-to-HIGH transition on Collision±.



Notes:

1. *Connect R1, R2, C1, C2 for 0 differential nontransmit. Connect to ground for logic 1 differential nontransmit.*
2. *Pin 20 shown for normal device operation.*
3. *The inclusion of C4 and C5 is necessary to reduce the common-mode loading on certain transceivers which are direct coupled.*
4. *C2 reduces the amount of noise from the power supply and crosstalk from RCLK that can be coupled from TSEL through to the transmit± outputs.*

Figure 6. External Component Diagram

Jitter Tolerance Definition and Test

The Receive Timing-Start of Reception Clock Acquisition waveform diagram shows the internal timing relationships implemented for decoding Manchester data in the Am7992B. The Am7992B utilizes a clock capture circuit to align its internal data strobe with an incoming bit stream. The clock acquisition circuitry requires four valid bits with the values 1010. Clock is phase locked to the negative transition at BCC of the second "0" in the pattern.

Since data is strobed at 1/4 bit time, Manchester transitions which shift from their nominal placement through 1/4 bit time will result in improperly decoded data. For IEEE 802.3/Ethernet, this results in the loss of a message. With this as the criteria for an error, a definition of "Jitter Handling" is:

> That peak deviation from nominal input transition approaching or crossing 1/4 bit cell position for which the Am7992B will properly decode data.

Four events of signal are needed to adequately test the ability of the Am7992B to properly decode data trom the Manchester bit stream. For each of the four events two time points within a received message are tested; (See Input Jitter Timing Waveforms):

1. Jitter tolerance at clock acquisition, the measure of clock capture, (case 1–4).
2. Jitter tolerance within a message after the analogue PLL has reduced clock acquisition error to a minimum, (case 5–8).

The four events to test are shown the Input Jitter Timing Waveform diagram. They are:

1. BCC jitter for a 01 bit pattern
2. BCC jitter for a 10 bit pattern
3. BCB jitter for an 11 bit pattern
4. BCB jitter for an X0 bit pattern

The test signals utilized to jitter the input data are artificial in that they may not be realizable on networks (examples are cases 2, 3 and 4 at clock acquisition). However, each pattern relates to setup and hold time measurements for the data decode register (Figure 5). Receive+ and Receive– are driven with the inputs shown to produce the zero crossing distortion at the differential inputs for the applicable test. Case 4 and 8 require only a single zero to implement when tested at the end of message.

Levels used to test jitter are within the common-mode and differential-mode range of the receive inputs and also are available from automatic test equipment. It is assumed that the incoming message is asynchronous with the local TCLK frequency for the Am7992B. This ensures that proper clock acquisition has been established with random phase and frequency error in incoming message. An additional condition placed on the jitter tolerance test is that it must meet all test requirements within 10 ms after power is applied. This forces the Am7992B crystal oscillator to start and lock the analogue PLL to within acceptable limits for receiving from a cold start.

Case 1 of the test corresponds to the expected Manchester data at clock acquisition and average values for clock leading jitter tolerance are 21.5 ns. For cases 5 through 8, average values are 24.4 ns. Cases 5 through 8 are jittered at bit times 55 or 56 as applicable. The Am7992B, then, has on average 0.6 ns static phase error for the noise-free case.

APPLICATION

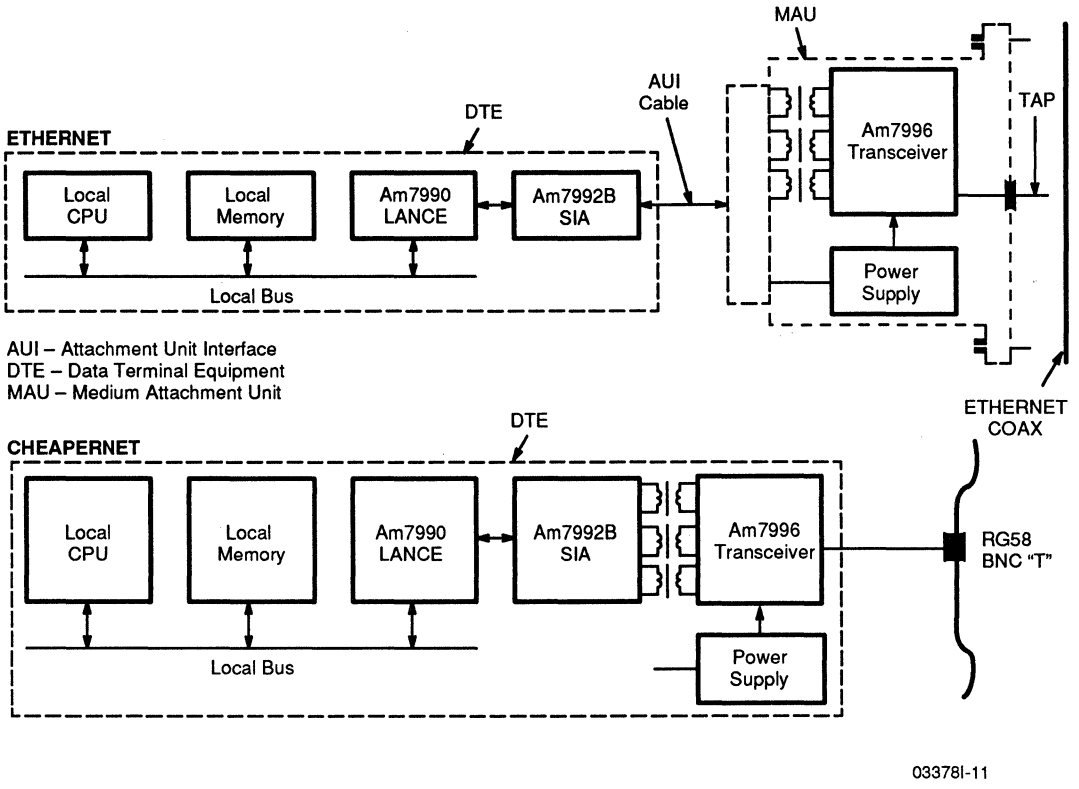


Figure 7. Typical ETHERNET Node

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C

Ambient Temperature with Power Applied ..... 0°C to +70°C

Supply Voltage Continuous ..... +7.0 V

DC Voltage Applied to Outputs ... −0.5 V to Vcc Max

DC Input Voltage (Logic Inputs) ..... +5.5 V

DC Input Voltage (Receive±/Collision±) ..... −6 V to +16 V

Transmit± Output Current ..... −50 mA to +25 mA

DC Output Current, Into Outputs ..... 100 mA

DC Input Current (Logic Inputs) ..... ±30 mA

Transmit± Applied Voltage ..... 0 V to +16 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ.

OPERATING RANGES

Commercial (C) Devices

Temperature (Tc) ..... 0°C to +70°C

Supply Voltage (Vcc) ..... +5.0 V ±10%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions		Com'l Min	Com'l Max	Unit
V_{OH}	Output HIGH Voltage RX, RENA, CLSN, TCLK, RCLK	I_{OH} = −1.0 mA, V_{CC} = Min		2.4		V
V_{OL}	Output LOW Voltage	I_{OL} = 16 mA, V_{CC} = Min			0.5	V
	RCLK, TSEL, TCLK, RENA, RX, CLSN	I_{OL} = 1 mA, V_{CC} = Min			0.4	V
V_{OD}	Differential Output Voltage TX+ > TX− for V_o	R_L = 78 Ω		550	770	mV
	(Transmit+) − (Transmit−) TX+ < TX− for $\overline{V_o}$			−550	−770	mV
$V_{OD\ OFF}$	Transmit Differential Output Idle Voltage	V_{CC} = Min, R_L = 78 Ω	(Note 1)	−20	20	mV
$I_{OD\ OFF}$	Transmit Differential Output Idle Current	TSEL = HIGH	(Note 2)	−0.5	0.5	mA
V_{CMT}	Transmit Output Common-Mode Voltage	R_L = 78 Ω, V_{CC} = Min		0	5	V
V_{ODI}	Transmit Differential Output Voltage Imbalance $\|\|V_o\| - \|\overline{V_o}\|\|$		(Note 1)		20	mV
V_{IH}	Input HIGH Voltage TX, TENA			2.0		V
I_{IH}	Input HIGH Current TX, TENA, $\overline{TEST}$	V_{CC} = Max, V_{IN} = 2.7 V			+50	µA
V_{IL}	Input LOW Current TX, TENA				0.8	V
I_{IL}	Input LOW Current TX, TENA, $\overline{TEST}$	V_{CC} = Max, V_{IN} = 0.4 V			−400	µA
V_{IRD}	Differential Input Threshold (Receive Data)	V_{CM} = 0 V, (Note 4)	Ceramic Package	−35	+35	mV
			Plastic Package	−65	+65	mV
V_{IRVD}	Differential Mode Input Voltage Range (Receive ±/Collision ±)	(Note 3)		−1.5	+1.5	V
V_{IRVC}	Receive ± and Collision ± Common Mode Voltage	(Note 2)		0	5.5	V
V_{IDC}	Differential Input Threshold to Detect Carrier	V_{CM} = 0 V (Note 4)		−175	−275	mV
I_{CC}	Power Supply Current	V_{CC} = Max (Note 5)			180	mA
V_{IB}	Input Breakdown Voltage (TX, TENA, $\overline{TEST}$)	I_I = 1 mA, V_{CC} = Max		5.5		V
V_{IC}	Input Clamp Voltage	I_{IN} = −18 mA, V_{CC} = Min			−1.2	V
V_{ODP}	Undershoot Voltage on Transmit Return to Zero (End of Message)	(Note 3)			−100	mV
I_{SC}	Short Circuit Current RCLK, RX, TCLK, CLSN, RENA	V_{CC} = Max (Note 6)		−40	−150	mA
R_{IDF}	Differential Input Resistance	V_{CC} = 0 to Max (Note 3)		6		kΩ
R_{ICM}	Common Mode Input Resistance	V_{CC} = 0 to Max (Note 3)		1.5		kΩ
V_{ICM}	Receive and Collision Input Bias Voltage	I_{IN} = 0, V_{CC} = Max		1.5	4.2	V
I_{ILD}	Receive and Collision Input LOW Current	V_{IN} = −1 V, V_{CC} = Max			−1.64	mA
I_{IHD}	Receive and Collision Input HIGH Current	V_{IN} = 6 V, V_{CC} = Min			+1.10	mA
I_{IHZ}	Receive and Collision Input HIGH Current Power Off	V_{CC} = 0, V_{IN} = +6 V			1.86	mA
I_{IHX}	Oscillator (X1) Input HIGH Current	V_{IN} = 2.4 V, V_{CC} = Max			+800	µA
I_{ILX}	Oscillator (X1) Input LOW Current	V_{IN} = 0.4 V, V_{CC} = Max			−1.2	mA
V_{IHX}	Oscillator (X1) Input HIGH Voltage	(Note 3)		2.0		V
V_{ILX}	Oscillator (X1) Input LOW Voltage	(Note 3)			0.8	V

Note:

See notes following Switching Characteristics table.

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified

No.	Parameters	Description	Test Conditions	Min	Max	Unit
Receiver Specification						
1	t_{RCT}	RCLK Cycle Time	(Note 8)	85	118	ns
2	t_{RCH}	RCLK HIGH Time		38		ns
3	t_{RCL}	RCLK LOW Time		38		ns
4	t_{RCR}	RCLK Rise Time			8	ns
5	t_{RCF}	RCLK Fall Time			8	ns
6	t_{RDR}	RX Rise Time			8	ns
7	t_{RDF}	RX Fall Time			8	ns
8	t_{RDH}	RX Hold Time (RCLK ↑ to RX Change)		5		ns
9	t_{RDS}	RX Prop Delay (RCLK ↑ to RX Stable)			25	ns
10	t_{DPH}	RENA Turn-On Delay (V_{IDC} Max on Receive ± to $RENA_H$)			80	ns
11	t_{DPO}	RENA Turn-On Delay (V_{IDC} Min on Receive ± to $RENA_L$)	(Note 9)		300	ns
12	t_{DPL}	RENA LOW Time	(Note 10)	120		ns
13	t_{RPWR}	Receive ± Input Pulse Width to Reject ($\|Input\| > \|V_{IDC}$ Max$\|$)	(Note 4)		20	ns
14	t_{RPWO}	Receive ± Input Pulse Width to Turn-On ($\|Input\| > \|V_{IDC}$ Max$\|$)		45		ns
15	t_{RLT}	Decoder Acquisition Time			450	ns
16	t_{REDH}	RENA Hold Time (RCLK ↑ to $RENA_L$)		40	80	ns
17	t_{RPWN}	Receive ± Input Pulse Width to Not Turn-Off INTCARR			165	ns
Collision Specification						
18	t_{CPWR}	Collision ± Input Pulse Width to Not Turn-On CLSN ($\|Input\| > \|V_{IDC}$ Min$\|$)	(Note 4)		10	ns
19	t_{CPWO}	Collision ± Input Pulse Width to Turn-On CLSN ($\|Input\| > \|V_{IDC}$ Max$\|$)		26		ns
20	t_{CPWE}	Collision ± Input Pulse Width to Turn-Off CLSN ($\|Input\| > \|V_{IDC}$ Max$\|$)		160		ns
21	t_{CPWN}	Collision ± Input Pulse Width to Not Turn-Off CLSN ($\|Input\| < \|V_{IDC}$ Max$\|$)			80	ns
22	t_{CPH}	CLSN Turn-On Delay (V_{IDC} Max on Collision ± to $CLSN_H$)			50	ns
23	t_{CPO}	CLSN Turn-Off Delay (V_{IDC} Max on Collision ± to $CLSN_L$)			160	ns

SWITCHING CHARACTERISTICS (continued)

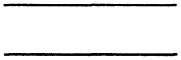
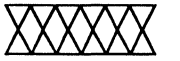
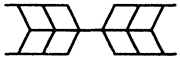
No.	Parameters	Description	Test Conditions	Min	Max	Unit
Transmitter Specification						
24	t_{TCL}	TCLK LOW Time	(Note 11)	45		ns
25	t_{TCH}	TCLK HIGH Time		45		ns
26	t_{TCR}	TCLK Rise Time			8	ns
27	t_{TCF}	TCLK Rise Time			8	ns
28	t_{TDS}, t_{TES}	TX and TENA Setup Time to TCLK	(Note 1)	5		ns
29	t_{TDH}, t_{TEH}	TX and TENA Hold Time to TCLK		5		ns
30	t_{TOCE}	Transmit ± Output, (Bit Cell Center to Edge)		49.5	50.5	ns
31	t_{OD}	TCLK HIGH to Transmit ± Output			100	ns
32	t_{TOR}	Transmit ± Output Rise Time	20% – 80%		4	ns
33	t_{TOF}	Transmit ± Output Fall Time			4	ns
34	t_{XTCH}	X1 to TCLK Propagation Delay for HIGH	(Notes 7 & 12)	5	18	ns
35	t_{XTCL}	X1 to TCLK Propagation Delay for LOW		5	18	ns
36	t_{EJ1}	Clock Acquisition Jitter Tolerance	V_{CC} = 5.0 V (Note 1)	16	21.5	ns
37	t_{EJ51}	Jitter Tolerance After 50 Bit Times	V_{CC} = 5.0 V (Note 1)	19	24.4	ns

**Min = 4.5 V, Max = 5.5 V, T_{OSC} = 50 ns; in production test, all differential input test conditions are done single-ended, non-V_{IRD} levels are forces on DUT for waveform swing (levels chosen are due to tester limitations) and a distortion-free preamble is applied to Receive± inputs.*

Notes:

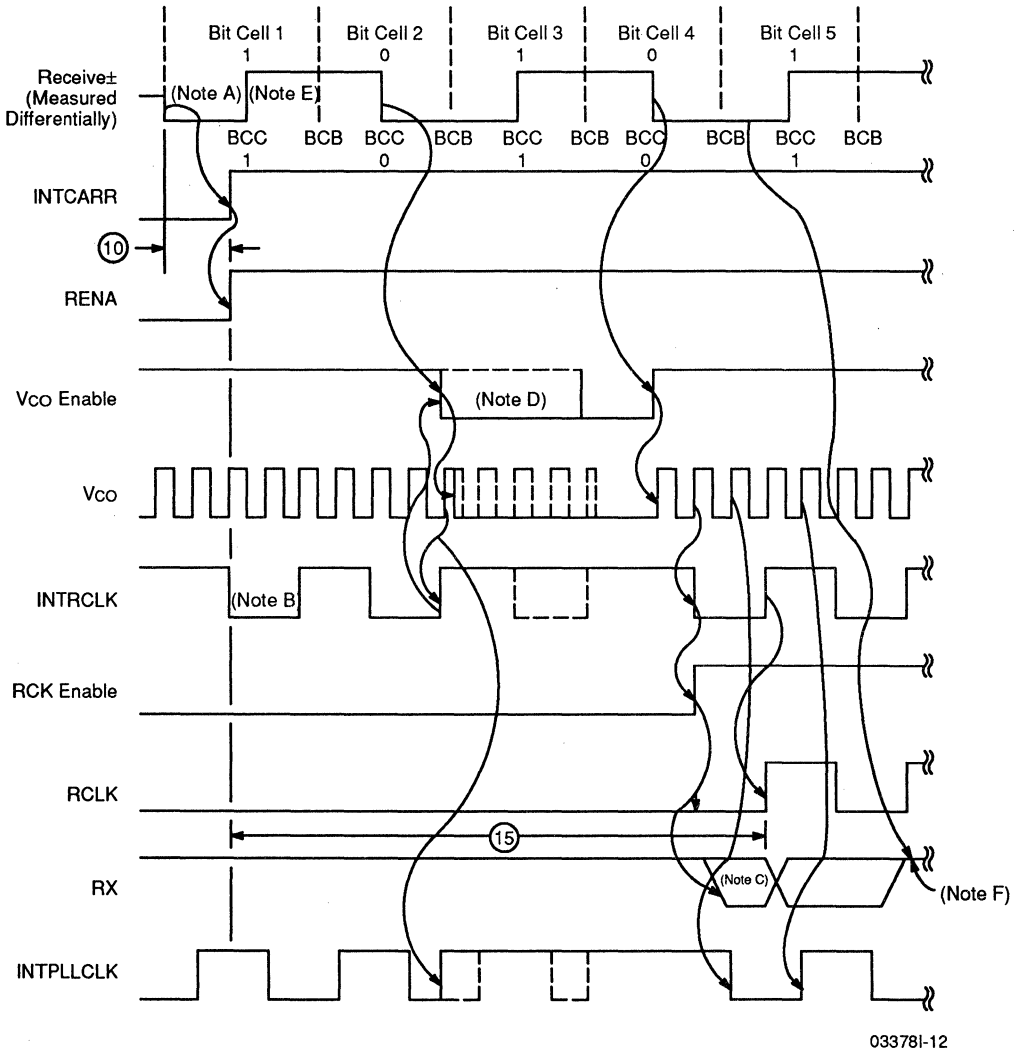
1. *Tested but to values in excess of limits. Test accuracy not sufficient to allow screening guardbands.*
2. *Correlated to other tested parameter: I_{OD} OFF = V_{OD} OFF/R_L.*
3. *Not tested.*
4. *Test done by monitoring output functionally.*
5. *Receive, Collision and Transmit functions are inactive: X1 driven by 20 MHz.*
6. *Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.*
7. *TCLK changes state on X1 rising edge, but initial state of TCLK is not defined. When TENA is High, TX data is Manchester encoded on the falling edge of X1 after the rising edge of TCLK.*
8. *Assumes 50 pF capacitance loading on RCLK and RX.*
9. *Test is done only for last BIT = 1, which is worst case.*
10. *Test done from 0.8 V of falling to 2.0 V of rising edge.*
11. *Test correlated to T_{TCH}.*
12. *Measured from 50% point of X1 driving the input in production test.*

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off" State

KS000010

SWITCHING WAVEFORMS



Notes:

A. Minimum Width > 45 ns.

B. RCLK = INTRCLK when $\overline{TEST}$ LOW.

C. RX undefined until bit time 5 (1st decoded bit).

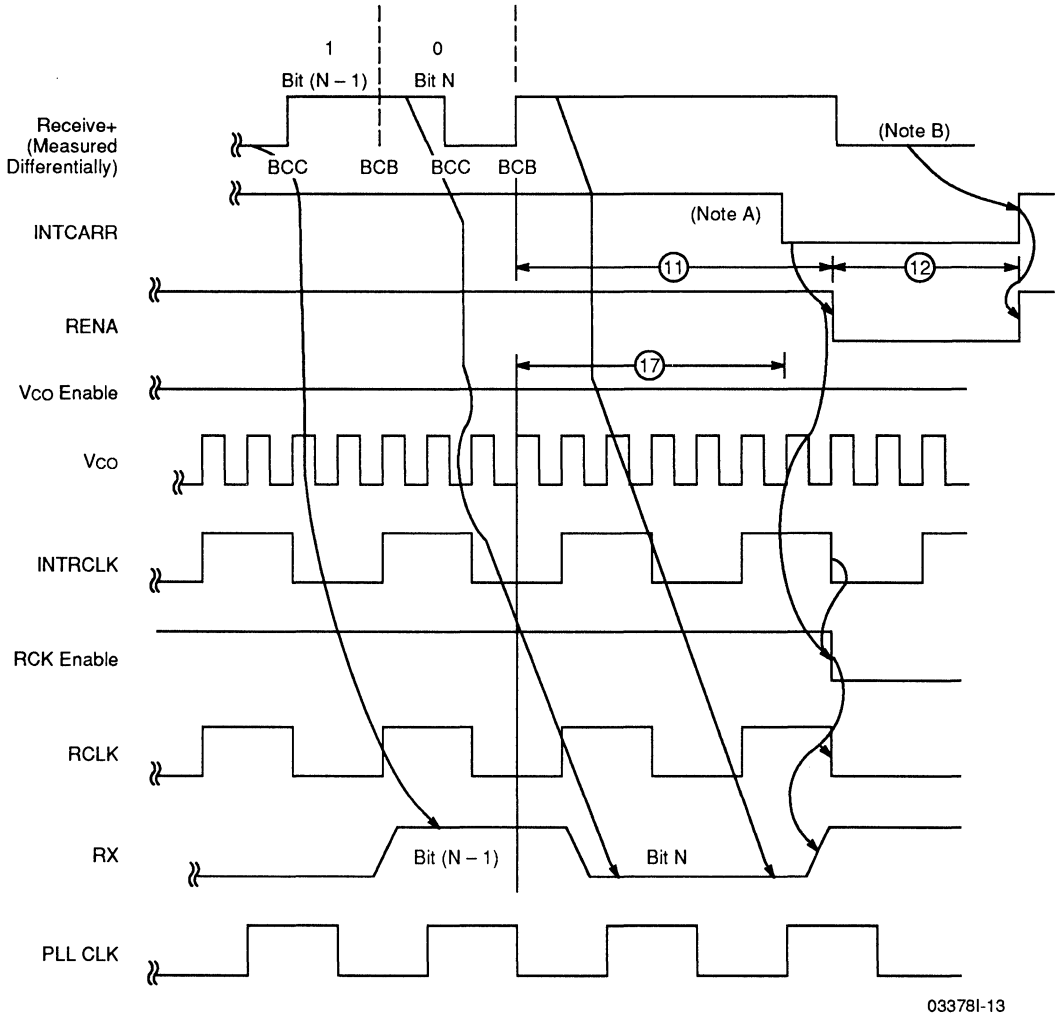
D. Oscillator Interrupt may occur at 2nd INTRCLK after Bit 2 Clock Transition.

E. Timing Diagram does not include Internal Propagation Delays.

F. First valid data at RX (Bit 5).

Receive Timing – Start of Reception Clock Acquisition

SWITCHING WAVEFORMS



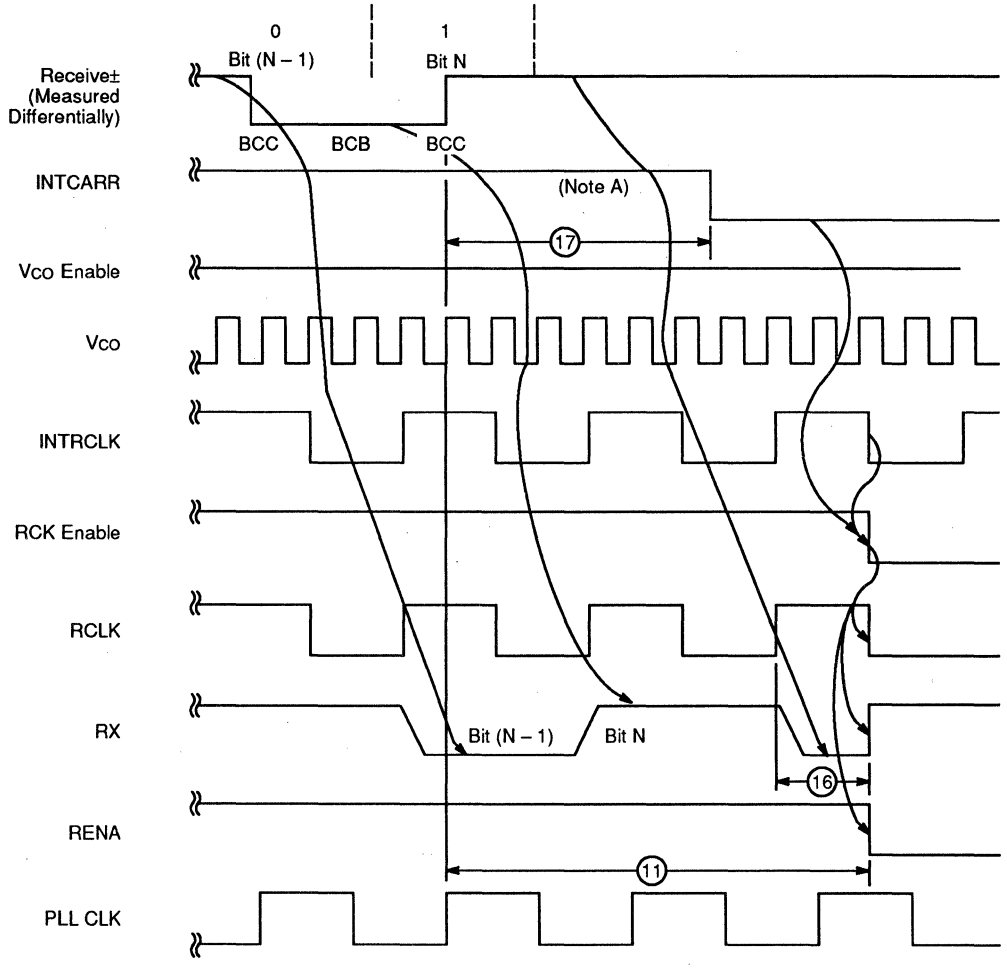
Notes:

A. INTCARR deasserts 1.55 bit times after last Receive± Rising Edge.

B. Start of Next Packet.

Receive Timing – End of Reception (Last Bit = 0)

SWITCHING WAVEFORMS

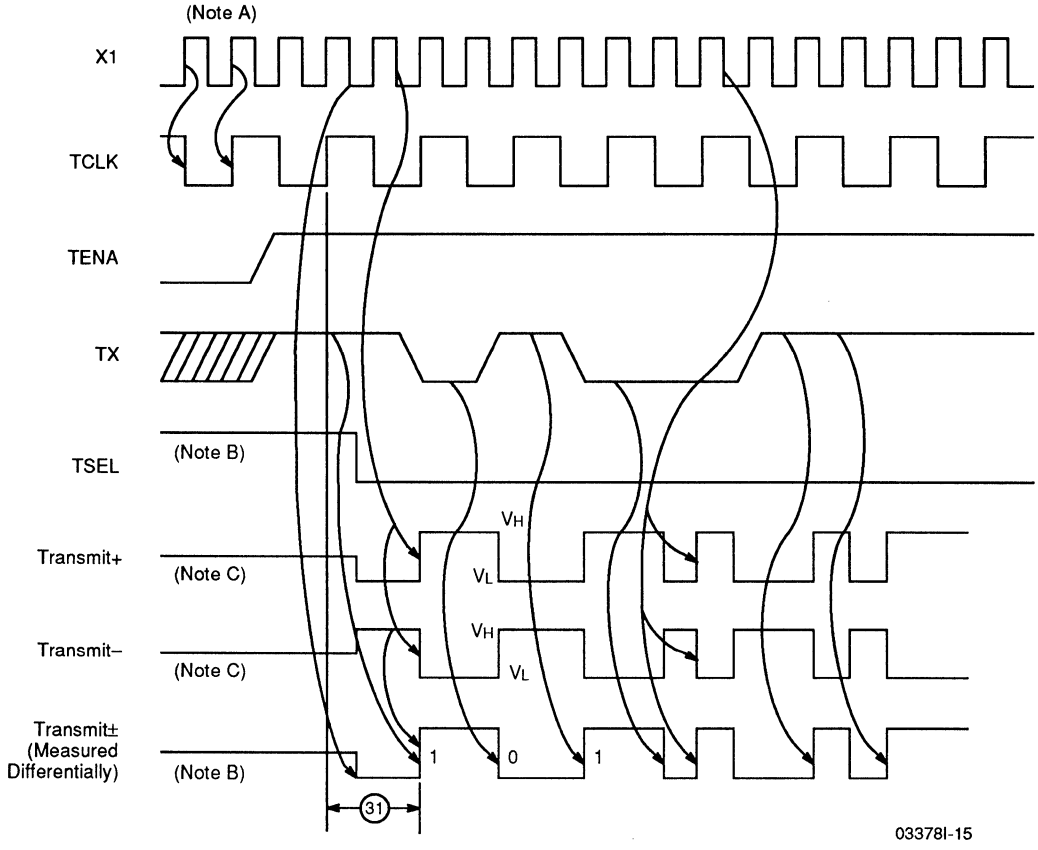


Note:

A. INTCARR deasserts 1.55 bit times after last Receive± Rising Edge.

Receive Timing – End of Reception (Last Bit = 1)

SWITCHING WAVEFORMS



Notes:

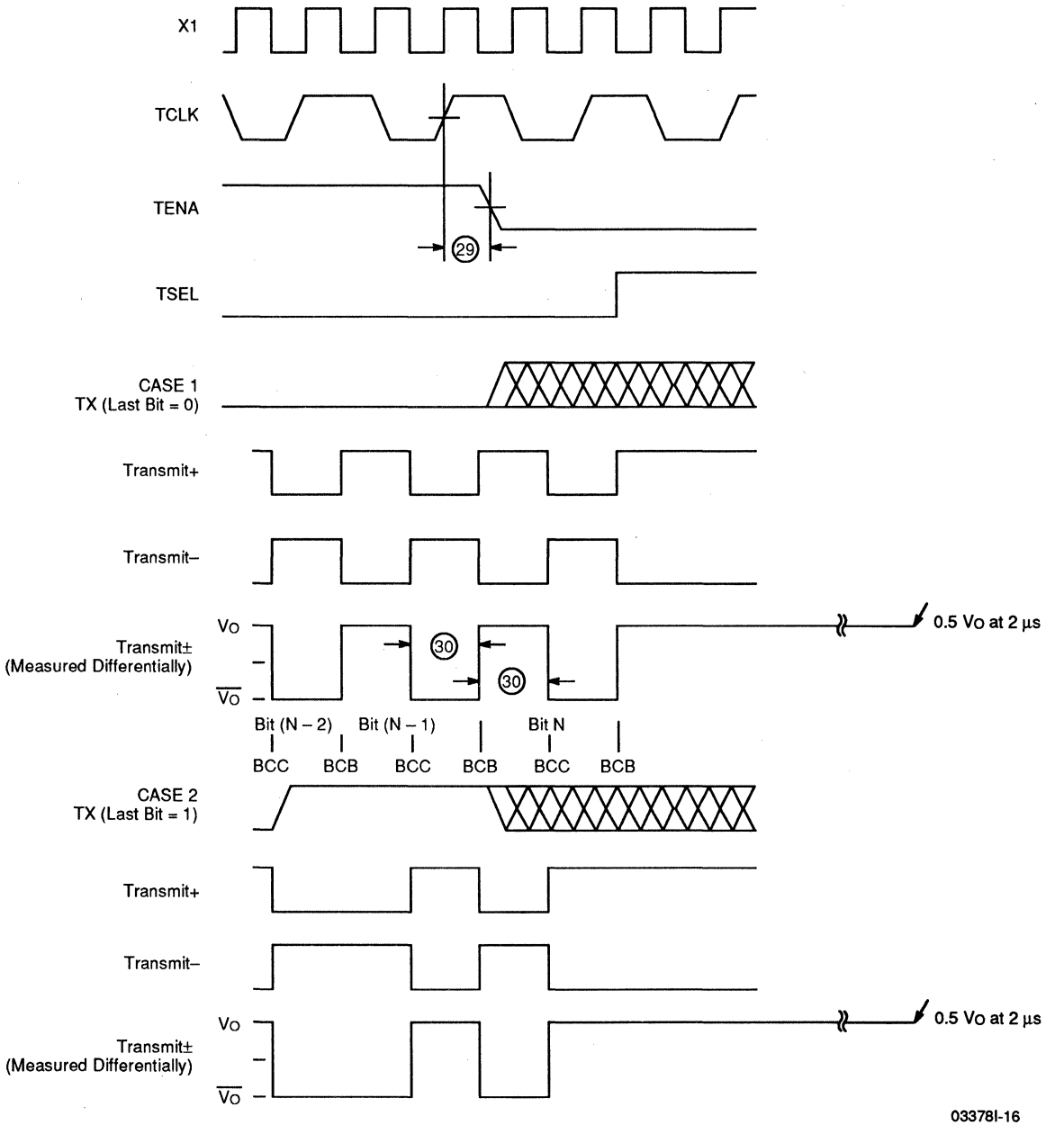
A. X1 20 MHz Sine Wave from Crystal Oscillator or driven with X1 driven from External Source Waveform.

B. TSEL connected as shown in Figure 2B. For Figure 2A, Transmit+ is HIGH when TENA is LOW.

C. When Idle Transmit± Zero Differential is 1/2 (V_H + V_L).

Transmit Timing – Start of Packet

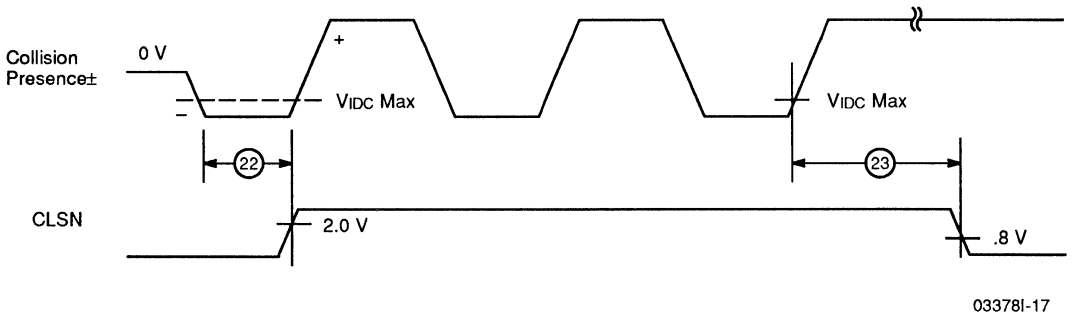
SWITCHING WAVEFORMS



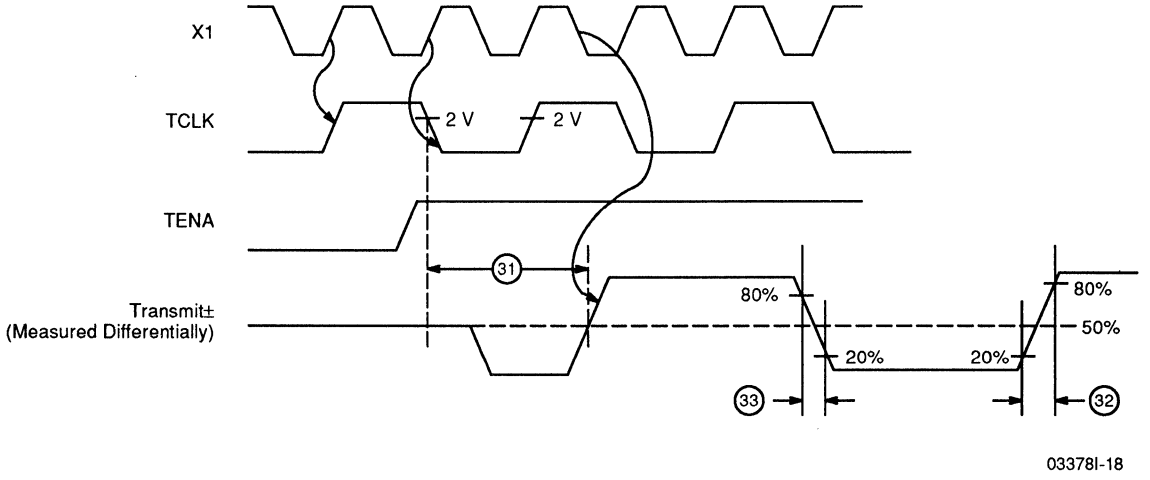
Transmit Timing – End of Transmission*

**TSEL Components (see Figure 2B).*
See Typical Performance Curve for Response at End of Transmission with Inductive Loads.

SWITCHING WAVEFORMS

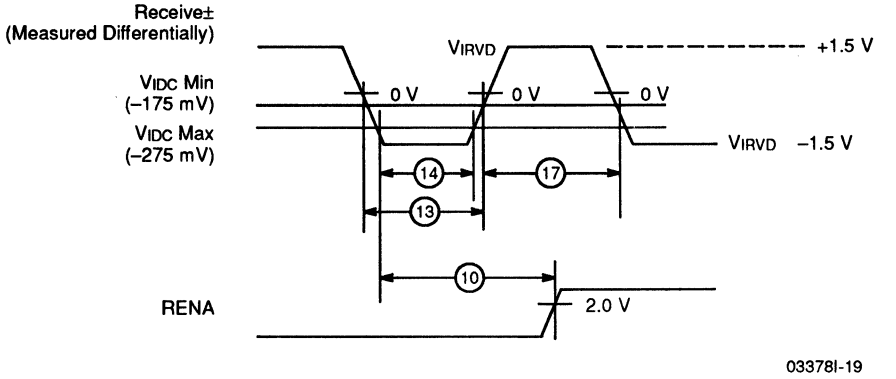


Collision Timing

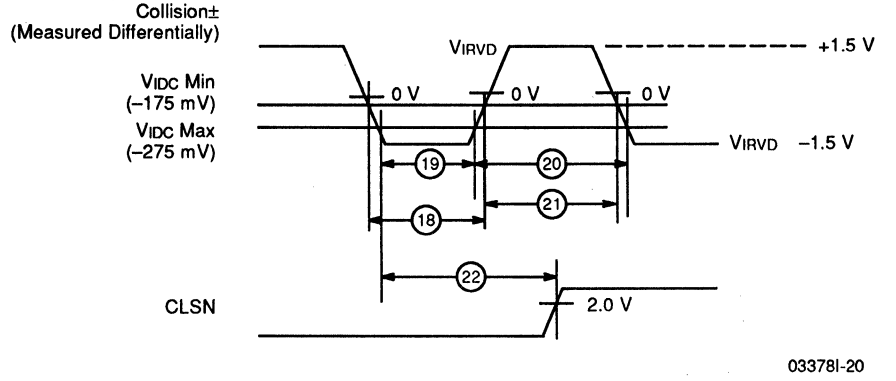


Transmit Timing (at start of packet)

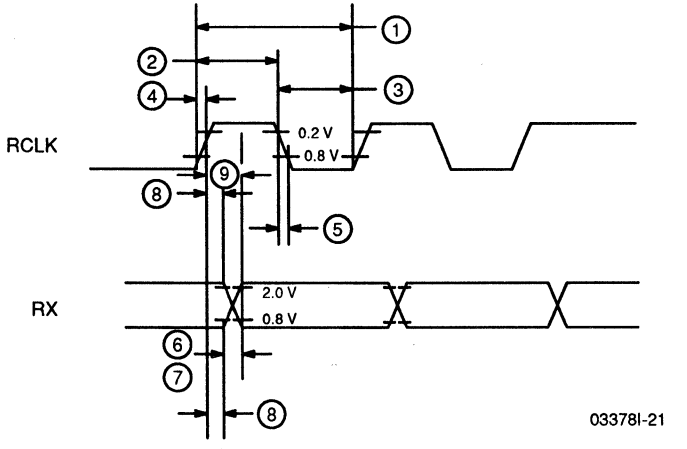
SWITCHING WAVEFORMS



Receive± Input Pulse Width Timing

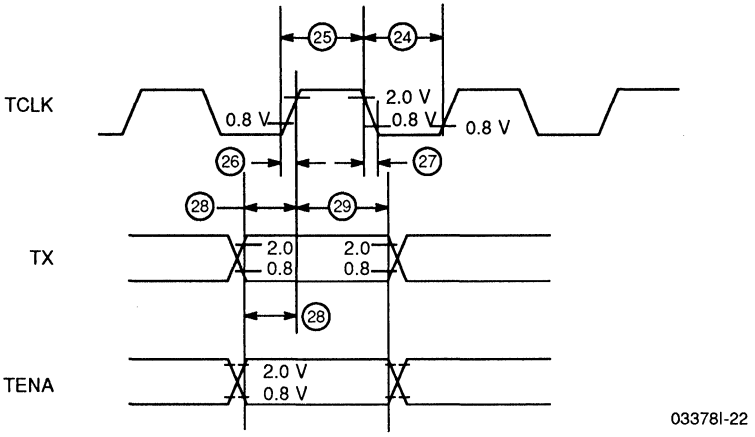


Collision± Input Pulse Width Timing

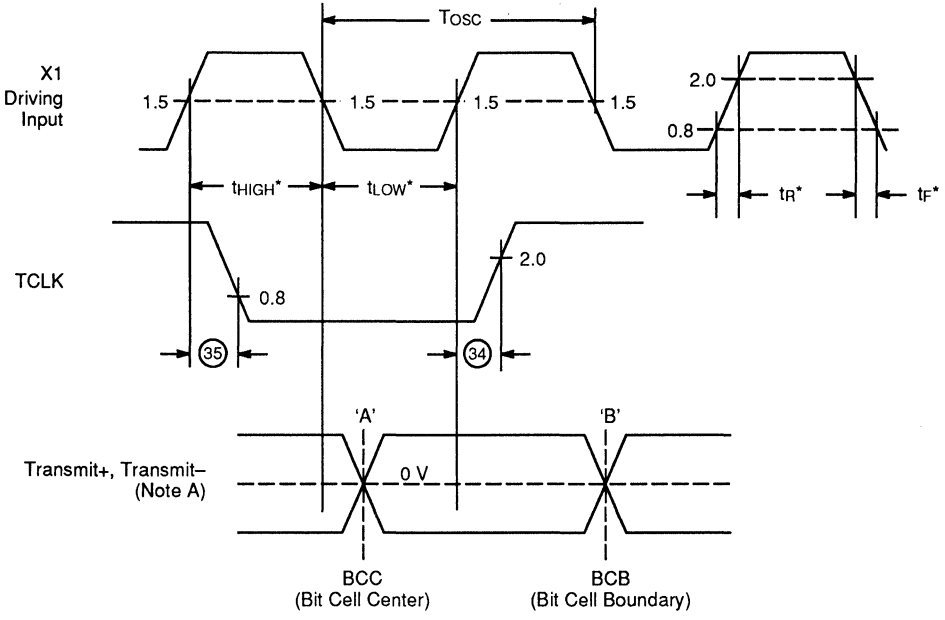


RCLK and RX Timing

SWITCHING WAVEFORMS



TCLK and TX Timing



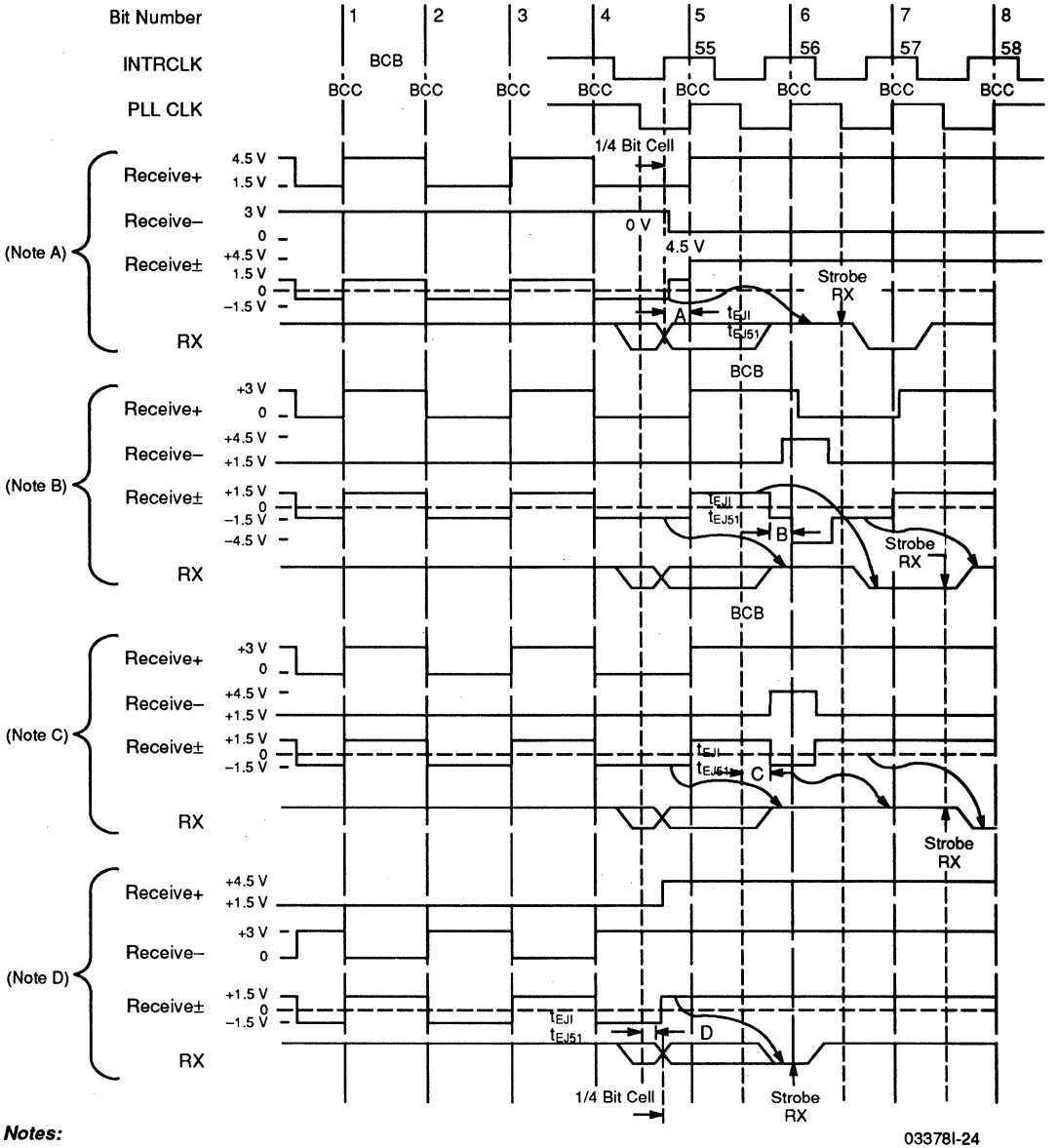
Note:

A. Encode Manchester clock transition (BCC) at Point 'A' and bit cell edge (BCB) at point 'B'.

**See Specification for External TTL Level in Functional Description section.*

X1 Driven from External Source

SWITCHING WAVEFORMS



03378I-24

Notes:

A. ***Case 1, 5 Data Bit Pattern 0, 1***
Rising clock edge moved toward 1/4 bit cell RCLK data strobe. Case 1 uses bit 5, Case 5 uses bit 55.

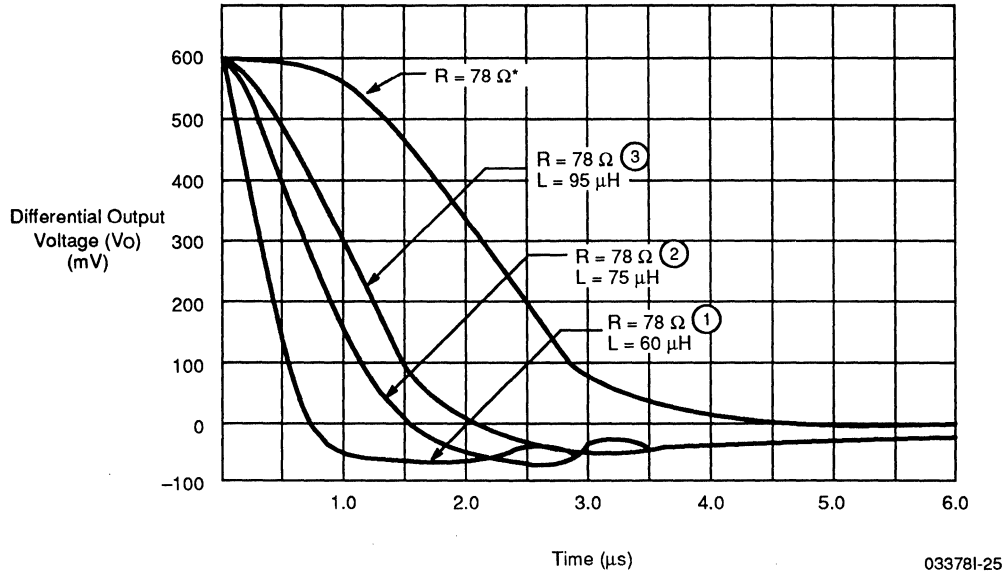
B. ***Case 2, 6 Data Bit Pattern 1, 0***
Falling clock edge moved toward 1/4 bit cell RCLK data strobe. Case 2 uses bit 6, Case 6 uses bit 56.

C. ***Case 3, 7 Data Bit Pattern 1, 1***
Falling bit cell edge moved toward 1/4 bit cell RCLK data strobe. Case 3 uses bit 6, Case 7 uses bit 56.

D. ***Case 4, 8 Data Bit Pattern X, 0***
Rising bit cell edge moved toward 1/4 bit cell RCLK data strobe. Case 4 uses bit 5, Case 8 uses bit 55.

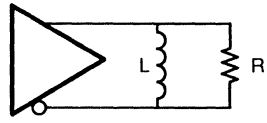
Input Jitter Timing

TYPICAL PERFORMANCE CURVE



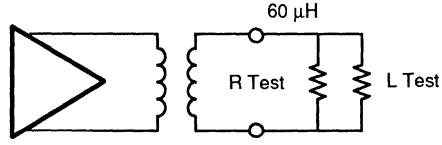
End of Transmission – Differential Output Voltage*

**Equivalent Load:*

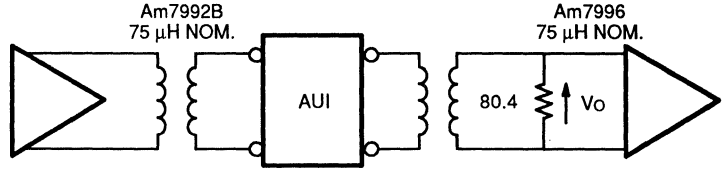


Notes:

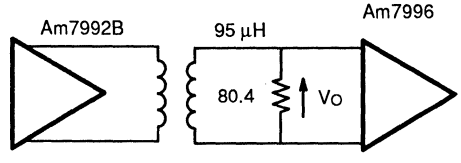
1. 802.3 Test Load:



2. 802.3 10BASE5 Network Connection:

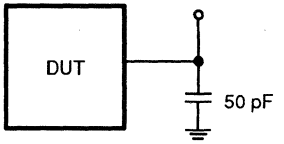


3. 802.3 10BASE2 Network Connection:



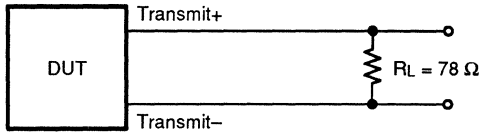
03378I-26

SWITCHING TEST CIRCUITS



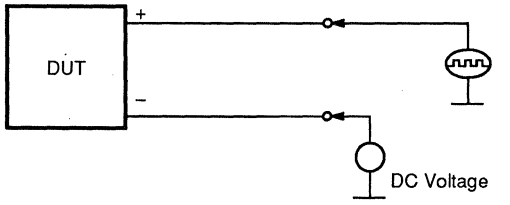
03378I-27

A. Test Load for RX, RENA, RCLK, TCLK, CLSN



03378I-28

B. Transmit± Output



03378I-29

C. Receive± and Collision± Input

FINAL

Am7996

IEEE 802.3/Ethernet/Cheapernet Transceiver

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Compatible with Ethernet Version 2 and IEEE 802.3 10BASE5 and 10BASE2 specifications**
- **Pin-selectable SQE Test (Heartbeat) option**
- **Internal Jabber Controller prevents excessive transmission time**
- **Noise rejection filter ensures only valid data is transmitted onto network**
- **Collision detection on both transmit and receive data**
- **Collision detect threshold levels adjustable for other networking applications**

GENERAL DESCRIPTION

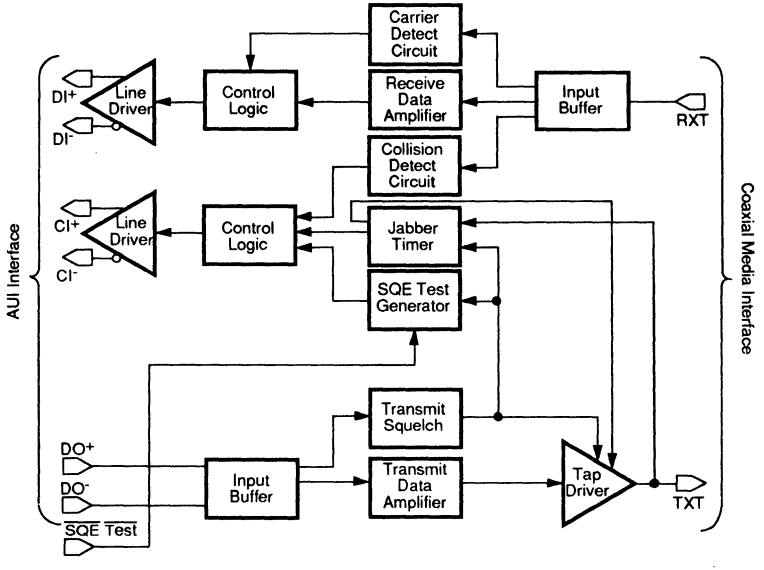
The Am7996 IEEE 802.3/Ethernet/Cheapernet Transceiver supports Ethernet Version 2, IEEE 802.3 (10BASE5), and IEEE 802.3 (10BASE2—Cheapernet) transceiver applications. Transmit, receive, and collision detect functions at the coaxial media interface to the Data Terminal Equipment (DTE) are all performed by this single device.

In an IEEE 802.3 (10BASE5)/Ethernet application, the Am7996 interfaces the coaxial (0.4″ diameter) media to the DTE through an isolating pulse transformer and the 78 Ω Attachment Unit Interface (AUI) cable. In IEEE 802.3 10BASE2—Cheapernet applications, the Am7996 typically resides inside the DTE with its signals to the DTE isolated and the coaxial (0.2″ diameter) media directly connected to the DTE. Transceiver power and ground in both applications are isolated from that of the DTE.

The Am7996's Tap Driver provides controlled skew and current drive for data signalling onto the media. The Jabber Controller prevents the node from transmitting excessively. While transmitting, collisions on the media are detected if one or more additional stations are transmitting.

The Am7996 features an optional SQE Test function that provides a signal on the *CI* pair at the end of every transmission. The SQE Test indicates the operational status of the *CI* pair to the DTE. It can also serve as an acknowledgement to the node that packet transmission onto the coax was completed.

BLOCK DIAGRAM



07506E-1

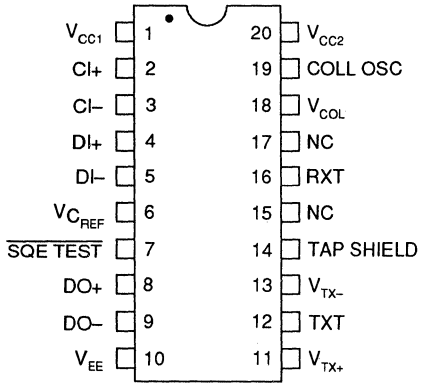
Publication# 07506 Rev. E Amendment /0
Issue Date: May 1994

RELATED PRODUCTS

Part No.	Description
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted Pair Ethernet Transceiver Plus (TPEX+)
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA+ Single-Chip Ethernet Controller (with Microsoft® Plug n' Play® Support)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller (for 386DX, 486 and VL buses)
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems

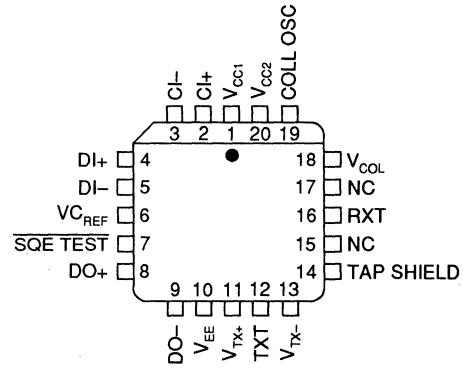
CONNECTION DIAGRAMS

DIP



07506E-2

PLCC



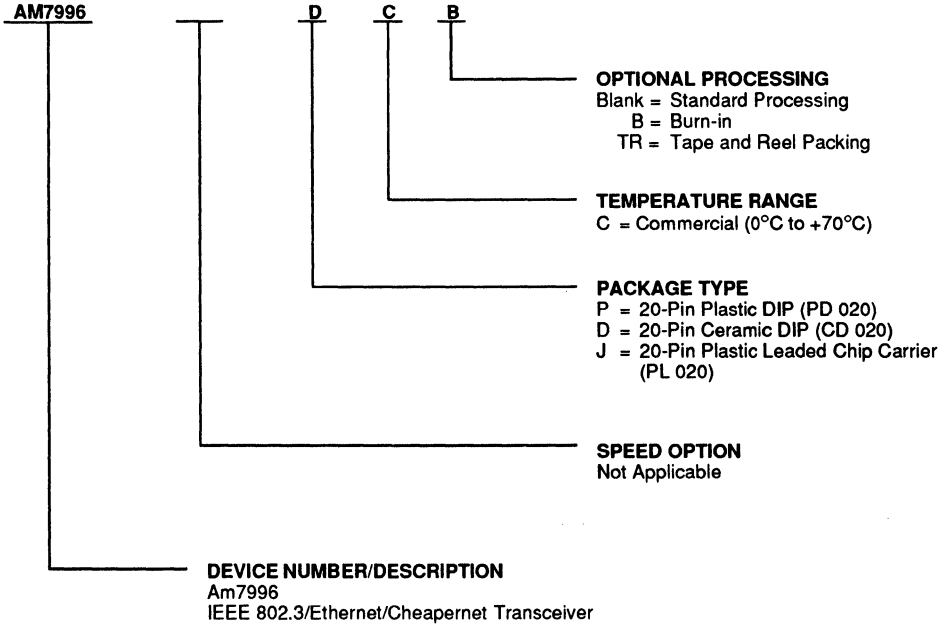
07506E-3

Notes:
Pin 1 is marked for orientation.
NC = No Connection

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of the following elements:



Valid Combinations	
AM7996	PC, PCB, DC, DCB, JC, JCTR

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

PIN DESCRIPTION

Attachment Unit Interface (AUI)

DI+, DI–
Receive Line Output (Differential Outputs)

This pair is intended to operate into terminated 78 Ω transmission lines. Signals at RXT meeting bandwidth requirements and carrier sense levels are outputted at DI±. Signaling at DI± meets requirements of IEEE 802.3, Rev. D.

CI+, CI–
Collision Line Output (Differential Outputs)

This pair is intended to operate into terminated 78 Ω transmission lines. Signal Quality Error (SQE), detected at DO± inputs (excessive transmissions) or RXT input (during a collision), outputs the 10 MHz internal oscillator signal to the AUI interface. For proper component values at COLL OSC, signaling at CI± meets requirements of IEEE 802.3, Rev. D.

DO+, DO–
Transmit Input (Differential Inputs)

A pair of internally biased line receivers consisting of a squelch detect receiver with offset and noise filtering, and a data receiver with zero offset for data signal processing. Signals meeting squelch requirements are waveshaped and output at TXT.

Coaxial Media Interface (TAP)

RXT
Media Signal Receiver Input (Input)

RXT connects to the media through a 4:1 attenuator of 100 kΩ total resistance (25 kΩ and 75 kΩ in series). Return for the attenuator is V_{COL}. RXT is an analog input with internal AC coupling for Manchester data signals and direct coupling for Carrier Detect and SQE average level detection. Signals at RXT meeting carrier squelch, enable data to the DI± outputs. Data signals are AC coupled to DI± with a 150-ns time constant, high-pass filter. Signals meeting SQE levels enable COLL OSC frequency to CI± outputs.

TXT
Tap Node Driver (Input/Output)

A controlled bandwidth current source and sense amplifier. This I/O port is to be connected to the media through an isolation network and a low-pass filter. Signals meeting DO± squelch and jabber timing requirements are output at TXT as a controlled rise and fall time current pulse. When operated into a double terminated 50 Ω transmission line, signaling meets IEEE 802.3, Rev. D recommendations for amplitude, pulse-width distortion, rise and fall times and harmonic content. The sense amplifier monitors TXT faults and inhibits transmission.

Global Signals

VC_{REF}
Timing Reference Set (Input)

VC_{REF} is a compensated voltage reference input with respect to V_{EE}. When a resistor is connected between VC_{REF} and V_{EE}, then internal transmit and receive squelch timing, SQE oscillator frequency, and receive and SQE output drive levels are set. SQE frequency set is also determined by components connected between V_{CC1} and COLL OSC.

$\overline{\text{SQE TEST}}$
Signal Quality Error Test Enable (Input)

The SQE Test function is enabled by connecting the $\overline{\text{SQE TEST}}$ pin to V_{EE} and disabled by connecting to V_{CC}.

V_{TX+}, V_{TX-}
Tap Node Driver Current Set (Inputs)

A reference input for transmission level and external redundant jabber. Transmit level is set by an external resistor between V_{TX+} and V_{TX-} (for an 80 mA peak level, R = 9.09 Ω). V_{TX-} may be operated between V_{EE} and V_{EE} + 1 V. When the voltage at V_{TX-} goes more positive than V_{EE} + 2 V, TXT is disabled and SQE message is output at the CI pair.

TAP SHIELD
Low-Noise Media Cable Return (Input)

This input is the return for V_{COL} reference and the receive signal from the media. External connection is to positive power supply.

V_{COL}
SQE Reference Voltage (Bias Supply)

SQE sense voltage and RXT input amplifier reference. An internally set analog reference for SQE level and data signal set at –1.600 V nominal with a source resistance of 150 Ω nominal. This reference should be filtered with respect to TAP SHIELD (see Applications section for adjusting threshold levels for other applications).

COLL OSC
SQE Timing Set (Input)

Timing input for SQE oscillator. For a properly set input at VC_{REF}, SQE oscillator period is set at 2.1RC. For a 10 MHz SQE oscillator frequency, R should be 1 kΩ and C, 47 pF including interconnect and device capacitance.

V_{CC1}
Positive Logic Supply

V_{CC2}
SQE Timing Reference (Positive Supply Voltage)

Timing reference return for SQE oscillator and analog signal ground.

V_{EE}
Negative Logic Supply and IC Substrate

FUNCTIONAL DESCRIPTION

The Am7996 IEEE 802.3/Ethernet/Cheapernet Transceiver consists of four sections: 1) Transmit—receives signals from DTE and sends it to the coaxial medium. 2) Receive—obtains data from media and sends it to DTE. 3) Collision Detect—indicates to DTE any collision on the media. and 4) Jabber—guards medium from node transmissions that are excessive in length.

Transmit

The Am7996 receives differential signals from the DTE (in the case of Am7990 Family applications, from the Am7992—Serial Interface Adapter-SIA). For IEEE (10BASE5)/Ethernet applications, this signal is received through the AUI cable and isolation transformer. In IEEE 802.3 10BASE2—Cheapernet applications, the AUI cable is optional.

Data is received through a noise rejection filter that rejects signals with pulse widths less than 7 ns (negative going), or with levels less than 175 mV peak. Only signals greater than –275 mV peak from the DTE are enabled. This minimizes false starts due to noise and ensures no valid packets are missed.

The Am7996's Tap Driver provides the driving capability to ensure adequate signal level at the end of the maximum length network segment (500 meters) under the worst case number of connections (100 nodes). Required rise and fall times of data transmitted on the network are maintained by the Am7996 Tap Driver. The Tap Driver's output is connected to the media through external isolating diodes. To safeguard network integrity, the driver is disabled whenever power falls below the minimum operation voltage.

During transmission, the Am7996 Jabber Controller monitors the duration that the Transmit Tap Driver is active and disables the driver if the jabber time is exceeded. This prevents network tie-up due to a "babbling" transceiver. Once disabled, the driver is not reset until 400 ms after the DO pair is idle and there is no fault on TXT. During the disable time, an SQE signal is sent on the CI pair to the DTE.

When $\overline{\text{SQE TEST}}$ is tied to V_{EE}, the Am7996 generates an SQE message at the end of every transmission. This signal is a self-test indication to the DTE that the Media Access Unit (MAU) collision pair is operational.

Receive and Carrier Detect

Signal is acquired from the tap through a high-impedance (100 kΩ) resistive divider. A high input-impedance (low capacitance, high bandwidth, low noise) DC-coupled input amplifier in the Am7996 receives the signal. The received signal passes through a high-pass filter to minimize inter-symbol distortion, and then through a data slicer. The Am7996 Carrier Detect compares received signals to a reference. Signals meeting carrier squelch requirements enable data to the differential line driver within five bit times from the start of packet.

Received data is transmitted from the DI pair through an isolation transformer to the AUI cable (Ethernet/IEEE 802.3—10BASE5). In IEEE 802.3 10BASE2—Cheapernet, the AUI cable is optional. Following the last transition of the packet, the DI pair is held HIGH for two bit times and then decreases to idle level within twenty bit times.

Collision Detect

The Am7996 detects collisions on Transmit if one or more additional stations are transmitting on the network.

Received signals are compared against the collision threshold reference. If the level is more negative than the reference, an enable signal is generated to the CI pair. The collision threshold can be modified by external components.

The Collision Oscillator is a 10 MHz oscillator which drives the differential CI pair to the DTE through an isolation transformer.

This signal is gated to the CI pair whenever there is a collision, the SQE Test is in progress, or the Jabber Controller is activated. The oscillator is also utilized in counting time for the Jabber Timer and SQE Test.

The CI ± output meets the drive requirements for the AUI interface. The output stays HIGH for two bit times at end of packet, decreasing to the idle level within twenty bit times.

Jabber Function

The Am7996 Jabber Timer monitors the activity on the DO pair and senses TXT faults. It inhibits transmission if the Tap Driver is active for longer than the jabber time (26 ms). An SQE message (10 MHz collision signal), is enabled on the CI pair for the fault duration.

After the fault is removed, the Jabber Timer counts the unjab time of 400 ms before it enables the driver.

If desired, a redundant Jabber function can be implemented externally, and the output driver disabled by removing the driver supply at V_{TX-}. The Am7996 senses this condition and forces an SQE message on the CI pair, during the disable time.

SQE Test

An SQE Test will occur at the end of every transmission if the $\overline{\text{SQE TEST}}$ pin is tied to V_{EE}. The SQE Test signal is a gated 10 MHz signal to the CI pair. The SQE Test ensures that the twisted pair assigned for collision notification to the DTE is intact and operational. The SQE Test starts eight bit times after the last transition of the transmitted signal and lasts for a duration of eight bit times.

The SQE Test can be disabled by connecting the $\overline{\text{SQE TEST}}$ pin to V_{CC}.

APPLICATIONS

The Am7996 is compatible with Ethernet Version 2 and IEEE 802.3 10BASE5 and 10BASE2 applications. (See Figure 1).

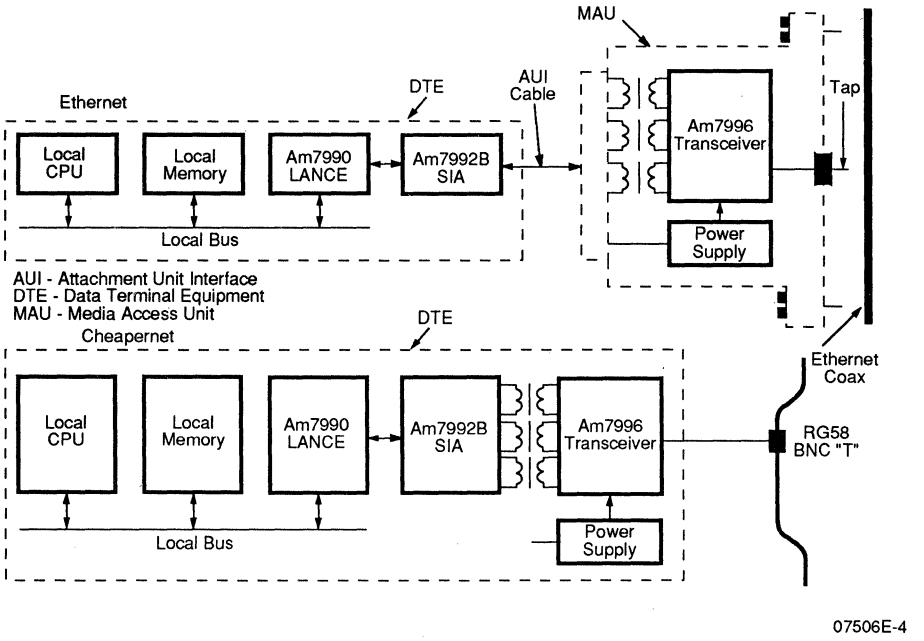


Figure 1. Typical Ethernet Node

Table 1. Transmit Mode Collision Detect Function Table

MAU Mode of Operation	Number of Transmitters		
	< 2	= 2	> 2
Transmitting	No	Yes	Yes
Not Transmitting	No	May	Yes

Table 2. IEEE 802.3 Recommended Transmit Mode Collision Detect Thresholds

IEEE 802.3	Threshold Voltage Level	
	No Detect	Must Detect
10BASE5, Ethernet	–1.492 V	–
10BASE2, Cheapernet	–1.404 V	–1.782 V

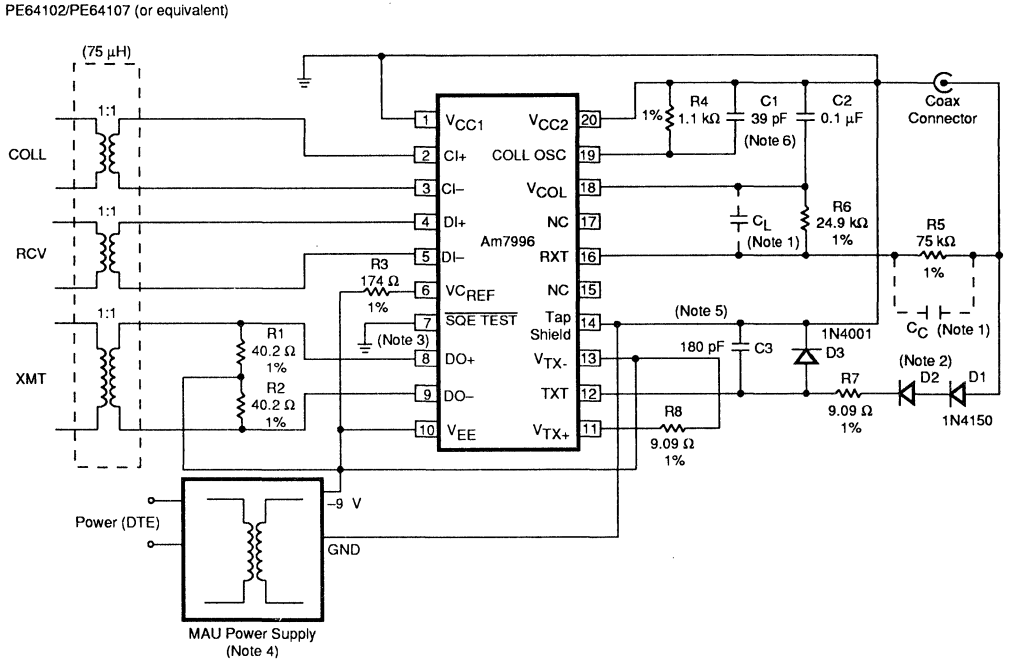
Table 3. Receive Mode Collision Detect Function Table

MAU Mode of Operation	Number of Transmitters		
	< 2	= 2	> 2
Transmitting	No	Yes	Yes
Not Transmitting	No	Yes	Yes

Table 4. IEEE 802.3 Recommended Receive Mode Collision Detect Thresholds

IEEE 802.3	Threshold Voltage Level	
	No Detect	Must Detect
10BASE5, Ethernet	–1.492 V	–1.629 V
10BASE2, Cheapernet	–1.404 V	–1.581 V

Figure 2 is an external component diagram showing how to implement the transmit mode collision detect levels recommended by IEEE 802.3. Figure 3 on the following page shows how to implement the receive mode collision detect levels recommended by IEEE 802.3. Receive mode collision detect threshold levels of the Am7996 are implemented by adding R9, R10 and C4. For the values of the components shown in Figure 3, a nominal receive mode collision detect threshold of –1.5 V, for a –1.404 V to –1.581 V window, is achieved.



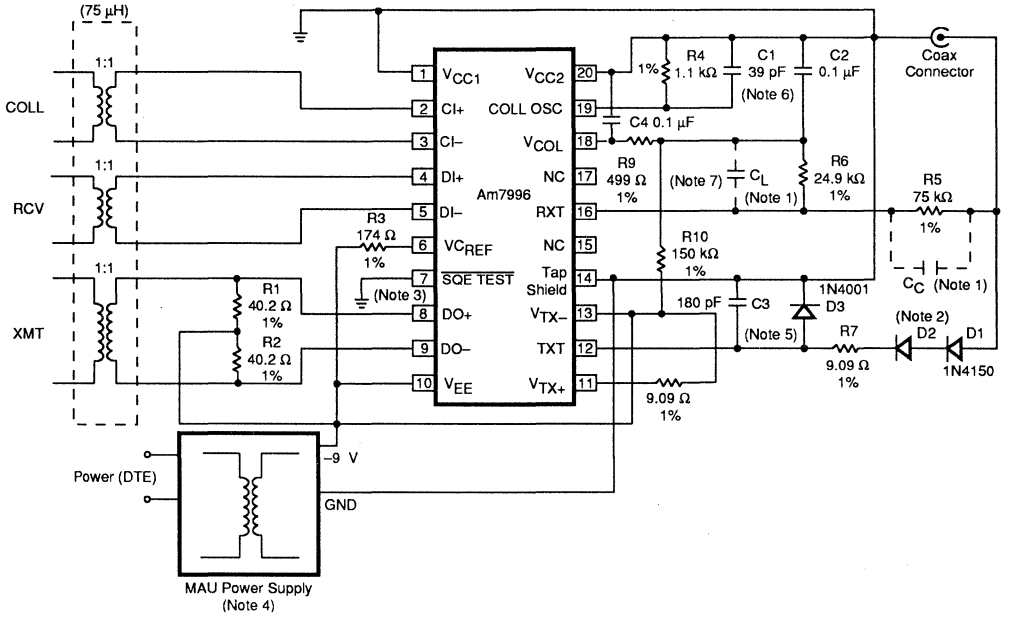
07506E-5

Notes:

1. *C_L is the effective load capacitance across R6; C_C is the compensation capacitance (C_C = 1/3 C_L).*
2. *D2 can be eliminated in Cheapernet (IEEE 802.3, 10BASE2) applications.*
3. *Shown with SQE Test disabled.*
4. *Discrete Power Supply or Hybrid-Hybrid DC-DC Converter Manufacturers include:*
   Ethernet (IEEE 802.3, 10BASE5)
   Reliability: 2E12R9
   Valor Electronics: PM1001
   Cheapernet (IEEE 802.3, 10BASE2)
   Reliability Inc: 2VP5U9
   Valor Electronics: PM7102
5. *The capacitance of C3, Am7996 package, D3 and the printed circuit board should add up to 180 pF ± 20%.*
6. *The capacitance of C1, Am7996 package and the printed circuit board should add up to 39 pF.*
7. *Figure 2 used for production testing of all parameters that are tested.*

Figure 2. Am7996 External Component Diagram for Transmit Mode Collision Detect

PE64102/PE64107 (or equivalent)



07506E-6

Notes:

1. *C_L is the effective load capacitance across R6; C_C is the compensation capacitance (C_C = 1/3 C_L).*
2. *D2 can be eliminated in Cheapernet (IEEE 802.3, 10BASE2) applications.*
3. *Shown with SQE Test disabled.*
4. *Discrete Power Supply or Hybrid-Hybrid DC-DC Converter Manufacturers include:*
   Ethernet (IEEE 802.3, 10BASE5)
   Reliability: 2E12R9
   Valor Electronics: PM1001
   Cheapernet (IEEE 802.3, 10BASE2)
   Reliability Inc: 2VP5U9
   Valor Electronics: PM7102
5. *The capacitance of C3, Am7996 package, D3 and the printed circuit board should add up to 180 pF ± 20%.*
6. *The capacitance of C1, Am7996 package and the printed circuit board should add up to 39 pF.*
7. *R9, R10 and C4 are for Receive Mode Collision detection only.*

Figure 3. Am7996 External Component Diagram with Collision Threshold Modified for Receive Mode Collision Detect

LAYOUT CONSIDERATIONS

To protect the transceiver from the environment and to achieve optimum performance, the Am7996 is designed to be used with two sets of external components: the transmitter circuit consisting of components D1, D2, D3, R7, R8, and C3, and the receiver circuit consisting of components R5, R6, C_L, and C_C, (C_L is a parasitic capacitance rather than a discrete component). These two circuits are shown in both Figure 2 and in Figure 3 respectively. The resistor tolerances for these circuits are specified as 1% for temperature stability.

The only layout restriction for the transmitter circuit is that the longest current path from the TXT pin (Pin 12) to the coaxial cable's center conductor must be no longer than 4 inches.

The layout of the receiver circuit, however, is critical. To minimize parasitic capacitance that can degrade the received signal, the external receiver circuit should be isolated from power and ground planes. There must be no power or ground plane under the area of the PC board that includes pins 15 through 20, R5, R6, and the connector for the coaxial cable. If a power or ground plane extends under this area, the receiver will not function properly due to excessive crosstalk and under- or over-compensation of the R5, R6 attenuator. Also, the RXT pin (Pin 16) should be as close to the coaxial cable connector as possible.

Since there are no severe layout restrictions on the transmitter circuit, the layout can be simplified by omitting power and ground planes from the whole area on the right side of the Am7996 as shown in Figure 4-1.

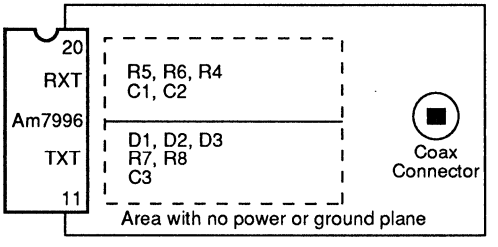


Figure 4-1.

If the above layout rules are followed, the parasitic capacitance in parallel with R6 will be about 6 pF. This parasitic capacitance is shown in the schematics as C_L (Note that C_L is a parasitic capacitance. Do not add a discrete capacitor in parallel with R6). The capacitor labeled C_C in the schematics is the total capacitance in parallel with R5 including parasitic capacitance. The parasitic component of C_C will be about 1 pF. For optimum performance, the ratio of C_L to C_C should be the same as the ration of R5 to R6, which is 3 to 1. This means that an additional 1 pF of capacitance must be added in parallel with R5.

This additional capacitance can easily be added by building a parallel-plate capacitor for PC traces right under resistor R5. This capacitor can consist of a 0.200 in. by 0.200 in. square of conductor on each side of the board as shown in Figure 4-2 (These dimensions assume that the PC board is made from 0.060 in. thick G-10 material). The top plate of the capacitor should be connected to one lead of R5, and the bottom plate should be connected to the other lead. Figure 4-3 shows an example of this suggested layout for a four layer printed circuit board. Note that the component labeling used in Figure 4-3 is not intended to correspond with the component labeling used in Figure 2 and Figure 3.

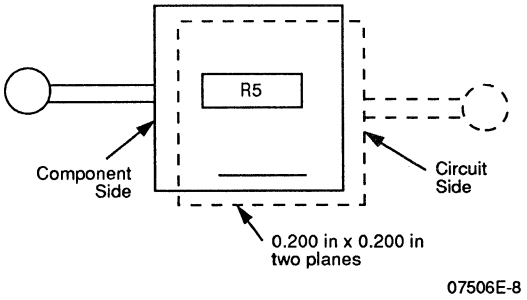
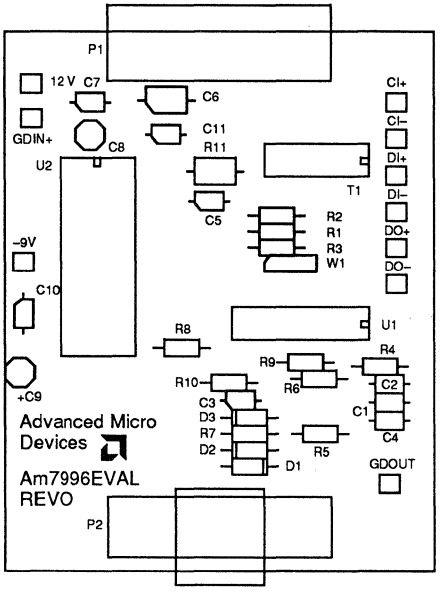
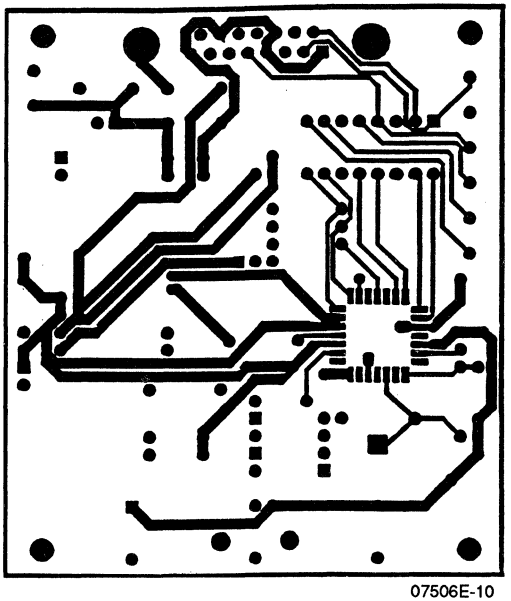


Figure 4-2.



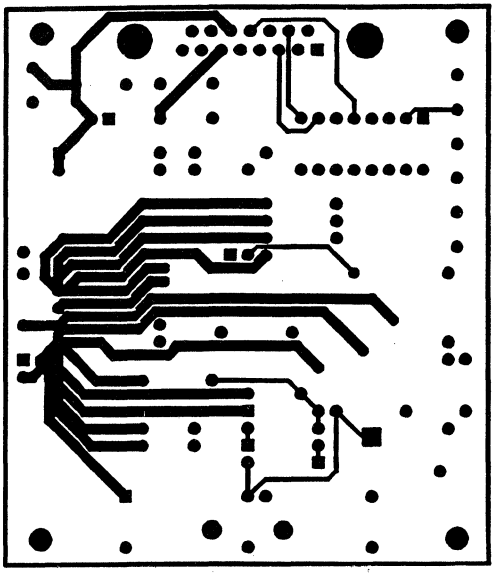
07506E-9

Component Side Silkscreen



07506E-10

Component Side



07506E-11

Solder Side

C11	CAP-0.01 μF
C2, C4, C7, C10	CAP-0.1 μF
C8	CAP-4.7 μF
C3	CAP-150 pF
C6	GAP CAP-0.001 μF
C1	CAP-39 pF
C5	CAP-1000 pF
D1, D2	DIODE 1N4153
D3	DIODE MUR120
P2	BNC
P1	15-Pin D Shell
R11	RES-1M
R4	RES-1.1K
R1	RES-40.2
R2	RES-40.2
R3	RES-174
R9	RES-499
R10	RES-150K
R6	RES-24.9K
R5	RES-75K with Trace Cap
R7	RES-9.09
R8	RES-9.09

Figure 4-3. Suggested Printed Circuit Board Layout for a Four Layer PCB Application

ABSOLUTE MAXIMUM RATINGS

Storage Temperature . . . . . –65°C to +150°C

Ambient Temperature
Under Bias . . . . . 0°C to +70°C

Supply Voltages (V_{EE}, V_{TX-}) . . . . . –12.0 V to +0.5 V

DC Input Voltage (D0+, D0–) . . . . –12.0 V to +0.5 V

DC Input Voltage (RXT) . . . . . –6 V to +0.5 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A) . . . . . 0°C to +70°C

Supply Voltage (V_{EE}) . . . . . –8.1 V to –9.9 V

Operating ranges define those limits between which the functionality of the device is guaranteed

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description		Test Conditions (Note 10)	Commercial Min	Commercial Typ	Commercial Max	Unit
Transmit Signals							
V_{TXTH}	Transmit Output HIGH Voltage (Note 1)		R_{LX} = 25 Ω	0	–0.05	–0.425	V
V_{TXTL}	Transmit Output LOW Voltage (Note 1)		R_{LX} = 25 Ω	–1.625	–2.0	–2.2	V
V_{TXT}	Transmit Average DC Voltage with 50% Duty-Cycle into DO+, DO– (Note 1)		R_{LX} = 25 Ω	–0.925	–1.0	–1.1	V
V_{ICM}	DO+, DO– Common Mode Bias Voltage		I_{IN} = 0	V_{EE} + 1.2	V_{EE} + 1.5	V_{EE} + 1.8	V
V_{IDC}	Differential Input Squelch Threshold (DO+, DO–) (Note 9)			–175	–225	–275	mV
I_{TXTL}	Transmit Current (Note 9)		V_{TXT} = –5.5 V	–65		–88	mA
I_{ILD}	Input Current (DO+, DO–)	V_{EE} = Max	V_{IN} = V_{EE} Max			–2.0	mA
I_{IHD}			V_{IN} = 0			2.5	
R_{IDF}	Differential Input Resistance (DO+, DO–)		V_{IN} = 0 to V_{EE}	6	8		kΩ
R_{ICM}	Common-Mode Input Resistance (DO+, DO–)		V_{IN} = 0 to V_{EE}	1.5	2		kΩ
Receive/Collision Signals							
V_{OD}	Differential Output Voltage (DI+, DI–; CI+, CI–)	R_L = 78 Ω	V_{OD+}	+550	+670	+850	mV
			V_{OD-}	–550	–670	–850	
V_{CMT}	Common-Mode Output (DI+, DI–; CI+, CI–)		R_L = 78 Ω	–1.0	–2.0	–3.0	V
V_{ODI}	Differential Output Voltage Imbalance (DI+, DI–; CI+, CI–) $\|\|V_{OD}\| - \|\overline{V}_{OD}\|\|$ (Note 6)		R_L = 78 Ω		5	20	mV
V_{OD} OFF	Differential Output Idle Voltage (DI+, DI–; CI+, CI–)		R_L = 78 Ω, V_{EE} = Max	–20	0	+20	mV
V_{CAT}	Carrier Sense Threshold		V_{IN} = 5 MHz Preamble	–400	–500	–600	mV
V_{COT}	Collision Sense Threshold (Note 5)			–1515	–1600	–1700	mV
I_{RXT}	RXT Input Bias Current		V_{IN} = 1 V to –2.5 V; V_{EE} = Max	–0.5	0	+0.5	µA
I_{OD} OFF	Differential Output Idle Current (DI+, DI–; CI+, CI–)		R_L = 0	–0.5	0	+0.5	mA
Global							
I_{EE}	Supply Current–Non-Transmitting		R_{LX} = 25 Ω (Note 4)		–88	–105	mA
	Supply Current–Transmitting				–128	–155	

CAPACITANCE* (T_A = 25°C; V_{EE} = 0; Pins 15, 17—No Connections)

Parameter Symbol	Parameter Description	Test Conditions	Min	Typ	Max	Unit
C_{RXT}	RXT Input Capacitance	Ceramic DIP		1.7		pF
		Plastic DIP/PLCC		1.1		

Notes:

See notes following Switching Characteristics section.

**Parameters are not "Tested."*

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified

No	Parameter Symbol	Parameter Description	Test Conditions	Commercial			Unit
				Min	Typ	Max	
Receiver Specification							
1	t_{PWREJ}	DO± Input Pulse Width to Reject ($DO\pm \geq V_{IDC}$, Max)	(Note 1)		15	7	ns
2	t_{PWTON}	DO± Input Pulse Width to Turn On ($DO\pm > V_{IDC}$, Max)	(Note 1)	20	15		ns
3	t_{PWSON}	DO± Input Pulse Width to Stay On ($DO\pm \geq V_{IDC}$, Max)	(Note 1)			105	ns
4	t_{PWOFF}	DO± Input Pulse Width to Turn Off ($DO\pm \geq V_{IDC}$, Max)	(Note 1)	160			ns
5	t_{TON}	Transmit Driver Turn-On Delay	(Note 1)			200	ns
7	t_{TSD}	Transmit Static Delay (Zero Crossing to 50% Point to Coax)	(Note 1)		30	50	ns
8	t_{TXTR}	Transmit Driver Rise Time	(Notes 1, 7)	20	25	30	ns
9	t_{TXTF}	Transmit Driver Fall Time	(Notes 1, 7)	20	25	30	ns
10	t_{DRF}	Difference in Driver Rise and Fall Times $\lvert t_{TXTR}-t_{TXTF}\rvert$	(Notes 1, 7)			1.0	ns
11	t_{SKEW}	Output Driver Skew—Transmit Data Symmetry	(Note 1)	–2.0		+2.0	ns
12	t_{JCT}	Jabber Control Time	(Note 1)	20	26	35	ms
13	t_{JRT}	Jabber Reset Time	(Note 1)	340	419	500	ms
14	t_{JREC}	Jabber Recovery Time	(Note 1)			1.0	µs
Receive/Collision Specification							
15	t_{RON}	Receiver Turn-On Delay	$V_{tap} > V_{CAT}$ Max		250	500	ns
16	t_{ROFF}	Receiver Turn-Off Delay	$V_{tap} < V_{CAT}$ Min			1000	ns
17	t_{RSD}	Receiver Static Delay	50% Point at RXT at Zero Crossing at DI± Outputs			50	ns
18	t_{RS}	Receive Data Symmetry		–2		+2	%
19	t_{RR}	DI± and CI± Rise Time	20%–80%, R_L = 78 Ω			7	ns
20	t_{RF}	DI± and CI± Fall Time	80%–20%, R_L = 78 Ω			7	ns
21	t_{CON}	CI± Turn-On Delay	$V_{tap} > V_{COT}$ Max			900	ns
22	t_{COFF}	CI± Turn-Off Delay	$V_{tap} < V_{COT}$ Min			2000	ns
23	t_{CL}	CI± LOW Time		35	50	70.5	ns
24	t_{CH}	CI± HIGH Time		35	50	70.5	ns
25	f_{CI}	Collision Frequency	(Note 8)	8.5	10.0	11.5	MHz
26	t_{STD}	SQE Test Delay Time	F_{CI} = 10.0 MHz	600		1000	ns
27	t_{STL}	SQE Test Length	F_{CI} = 10.0 MHz	600	800	1000	ns

Notes:

1. *Parameters are measured at coax tap. In production test, parameters are measured across at 25 Ω load equivalent to the coax tap.*
2. *For conditions shown as Min or Max, use the appropriate value specified under Operating Range for the applicable device type.*
3. *Typical values are at V_{EE} = –9.0 V, 25°C ambient.*
4. *V_{TX-} wired to V_{EE}.*
5. *This threshold can be modified externally (see Figure 3).*
6. *Parameter not tested.*
7. *Tested on a 5 Mbps preamble (continuous 1010 pattern) measured between 20% and 80% points, test limits correlated to 10% and 90% data sheet limits shown.*
8. *Determined by Am7966 External Component Diagrams values for R4 and C1.*
9. *In production test, input signal applied thru transformer to DO± inputs.*
10. *Figure 2 used for production testing of all parameters.*

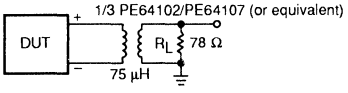
**Notes listed correspond to the respective references made in DC Characteristics and Switching Characteristics tables.*

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off" State

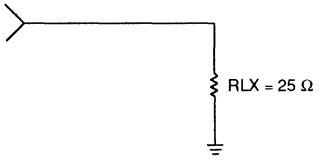
KS000010

SWITCHING TEST CIRCUIT



07506E-12

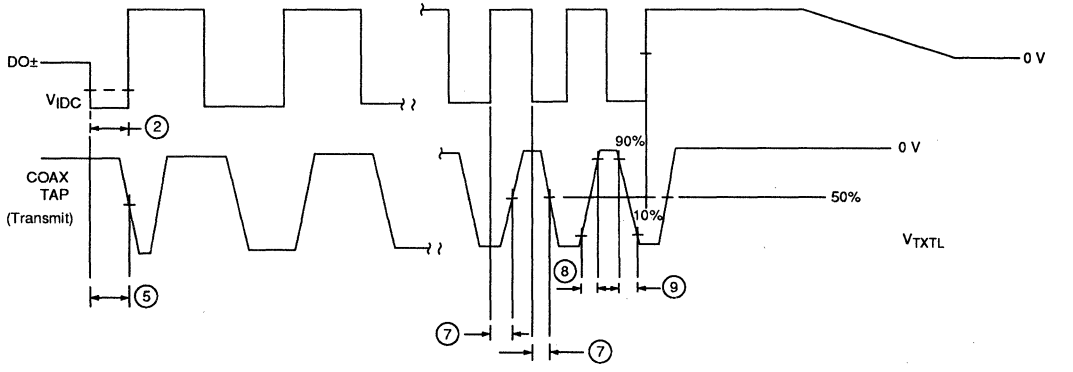
A. AUI Transmit (DI+, DI–,; CI+, CI–)



07506E-13

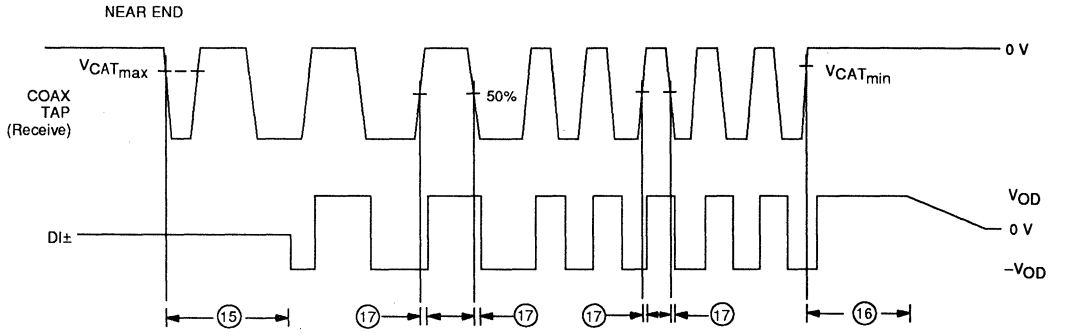
B. Test Load (TXT)

SWITCHING WAVEFORMS



07506E-14

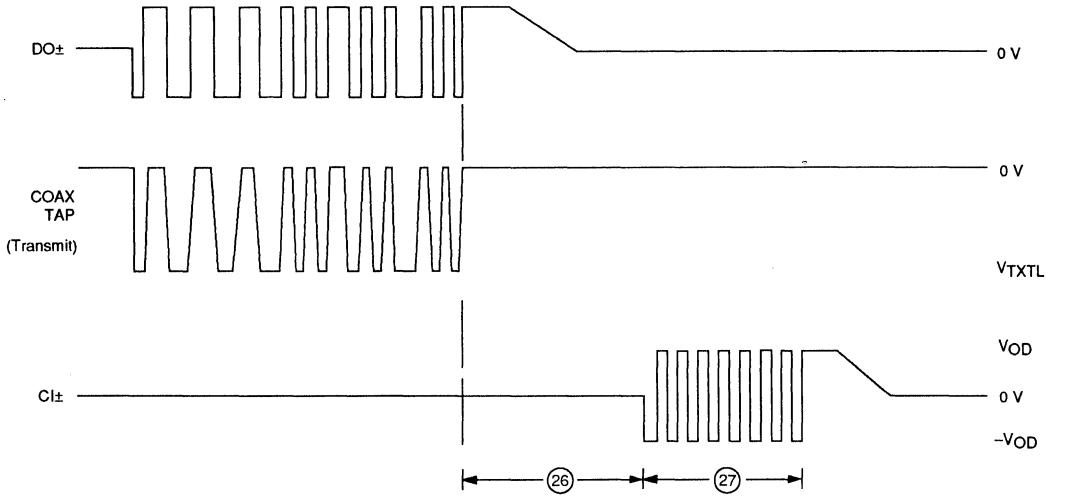
Transmit Function



07506E-15

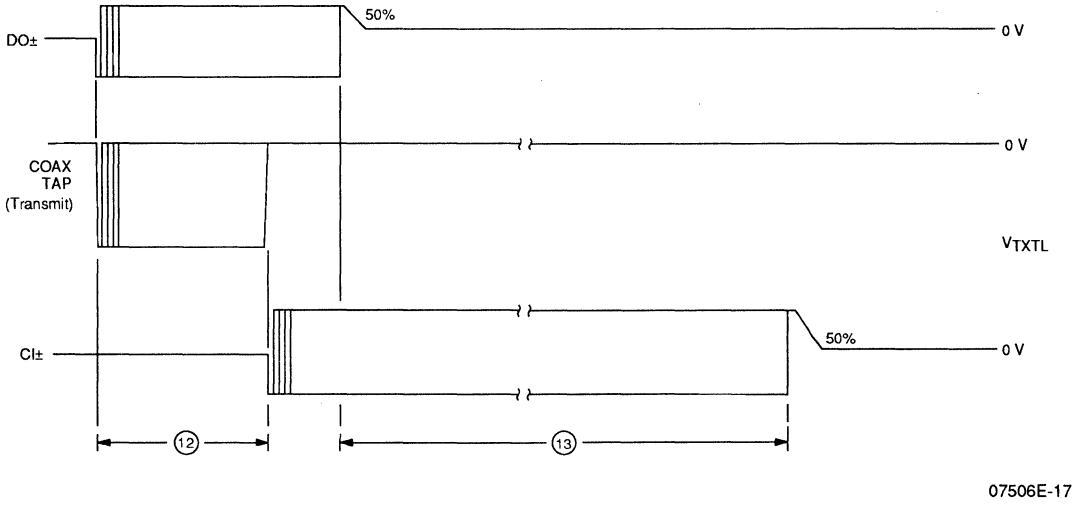
Receiver Function

SWITCHING WAVEFORMS



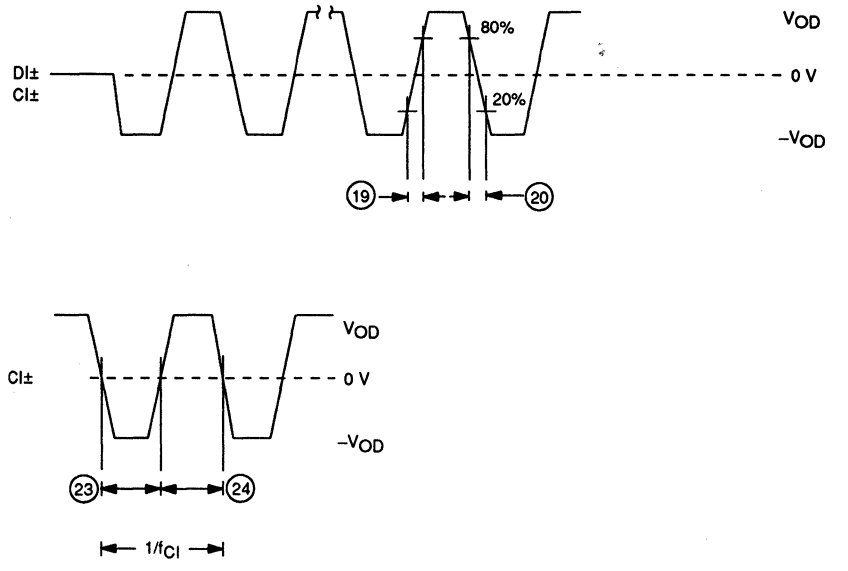
SQE Test*

**SQE TEST* pin connected to V_{EE}*

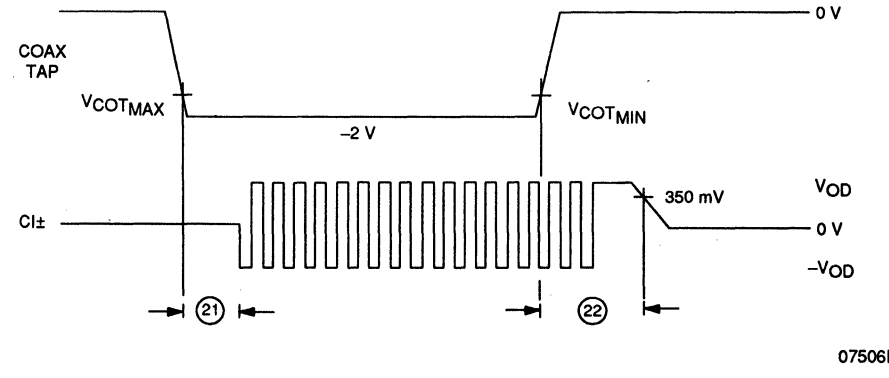


Jabber Function

SWITCHING WAVEFORMS



DI±/CI± Parameters



Collision Detect Timing

Note:

This signal is used for test purposes. It represents the average value of the signal that might be seen on the coax tap when a collision occurs.

FINAL

Am79C98

Twisted Pair Ethernet Transceiver (TPEX)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **CMOS device provides compliant operation and low operating current from single +5 V supply**
- **Power-Down mode provides reduced power consumption for battery-powered applications. Reset capability allows use in remote MAU applications.**
- **Pin-selectable Twisted Pair receive polarity detection and automatic inversion of the receive signal. Polarity indication output pin can directly drive a LED.**
- **Pin-selectable Twisted Pair Link Integrity Test capability conforming to the IEEE 802.3 standard for 10BASE-T. Link status pin can directly drive a LED.**
- **Internal Twisted Pair transmitter digital pre-distortion circuit reduces medium induced jitter and ensures compliance with the 10BASE-T transmit and receive waveform requirements**
- **Pin-selectable SQE Test (Heartbeat) enable**
- **Transmit and Receive status indication are available on separate, dedicated pins**
- **AUI loop-back, Jabber Control, and SQE Test functions comply with the 10BASE-T Standard IEEE Std 802.3i-1990**

GENERAL DESCRIPTION

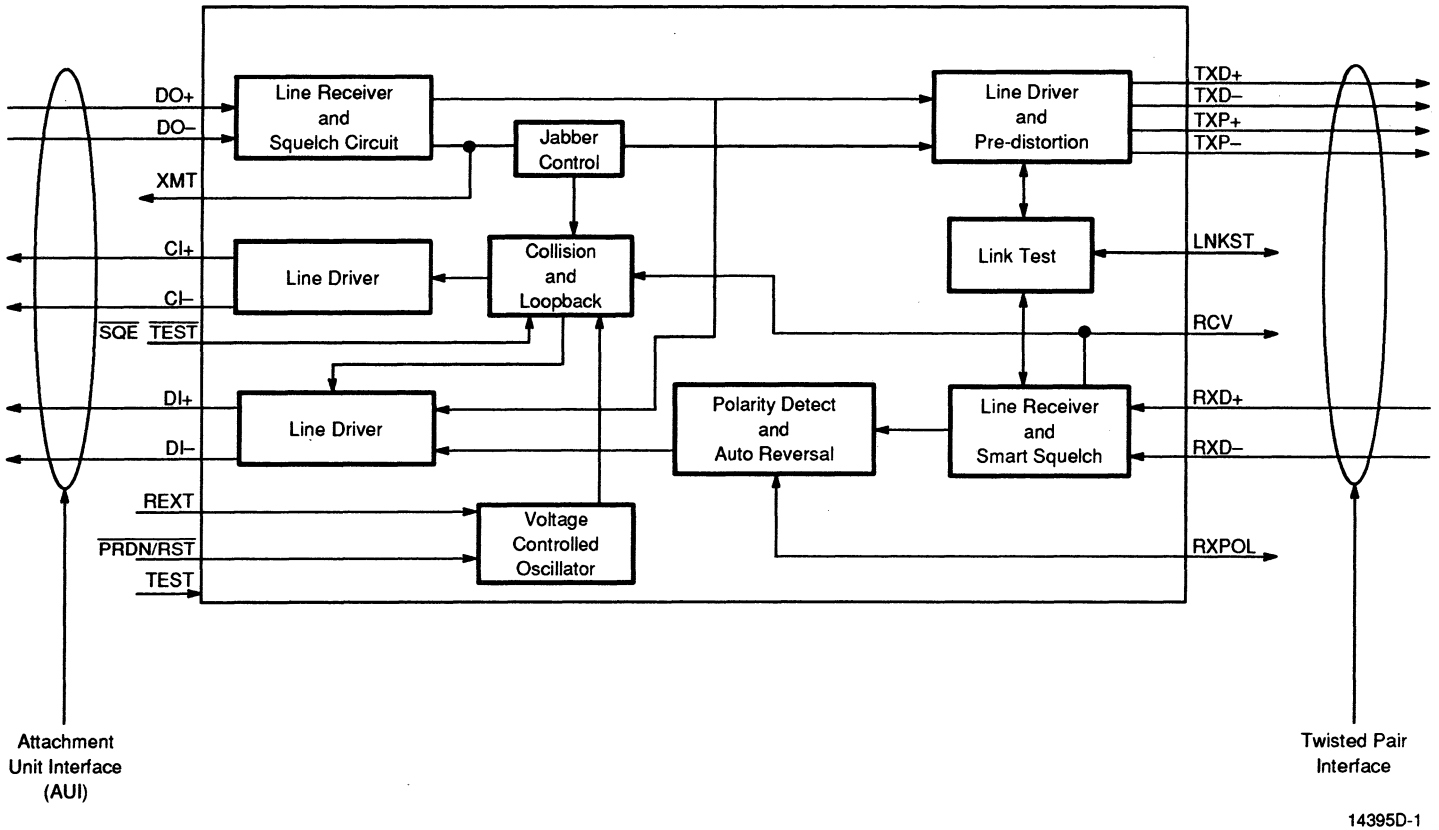
The Am79C98 Twisted Pair Ethernet Transceiver (TPEX) is an integrated circuit that implements the Medium Attachment Unit (MAU) functions for the Twisted Pair Medium, as specified by the IEEE 802.3 standard (Type 10BASE-T). This device provides the necessary electrical and functional interface between the IEEE 802.3 standard Attachment Unit Interface (AUI) and the Twisted Pair cable.

A network based on the 10BASE-T standard can use unshielded twisted pair cables, therefore providing an economical solution to networking by allowing the use of existing telephone wiring. The Am79C98 provides a minimal component count and cost effective solution to the design and implementation of 10BASE-T standard networks.

TPEX provides Twisted Pair driver and receiver circuits, including on-board transmit digital pre-distortion, receiver squelch, and an AUI port with pin selectable SQE Test enable. The device also provides a number of additional features including pin selectable Twisted Pair Receive Polarity Detection and Automatic Polarity Reversal, Link Status indication, Link Test disable function, and transmit and receive status. The Twisted Pair Polarity and Link status pins can be used to drive LEDs directly.

The Am79C98 is fabricated in CMOS technology and requires a single +5 V supply. The device is available in 24-pin SKINNYDIP® Plastic Dual-in-Line and 28-pin Plastic Leaded Chip Carrier (PLCC).

Publication# **14395** Rev. **D** Amendment /**0**
Issue Date: **May 1994**


DO+
DO–
XMT
CI+
CI–
SQE TEST
DI+
DI–
REXT
PRDN/RST
TEST
Line Receiver and Squelch Circuit
Jabber Control
Line Driver
Collision and Loopback
Line Driver
Voltage Controlled Oscillator
Polarity Detect and Auto Reversal
Line Driver and Pre-distortion
Link Test
Line Receiver and Smart Squelch
TXD+
TXD–
TXP+
TXP–
LNKST
RCV
RXD+
RXD–
RXPOL
Attachment Unit Interface (AUI)
Twisted Pair Interface

14395D-1

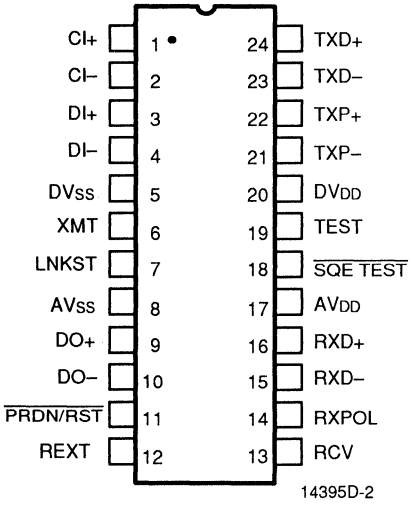
RELATED AMD PRODUCTS

Part No.	Description
Am7996	IEEE-802.3/Ethernet/Cheapernet Tap Transceiver
Am79C100	Twisted-Pair Ethernet Transceiver Plus (TPEX+)
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA+ Single-Chip Ethernet Controller (with Microsoft® Plug n' Play support)
Am79C965	PCnet-32 Single-Chip Ethernet Controller (for 386DX, 486 and VL buses)
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)

CONNECTION DIAGRAM

Top View

DIP

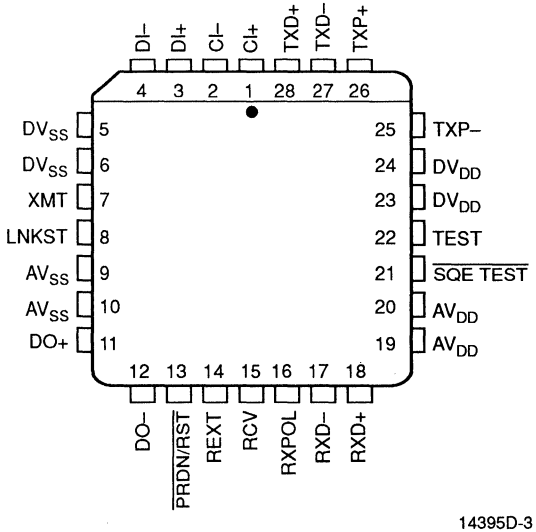


14395D-2

Note:

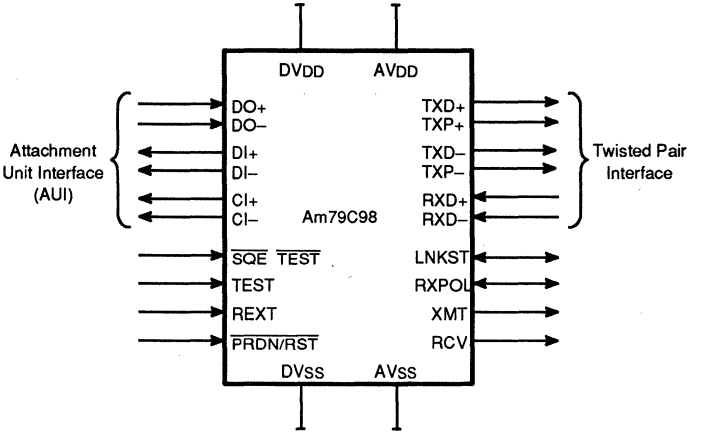
Pin 1 is marked for orientation

PLCC



14395D-3

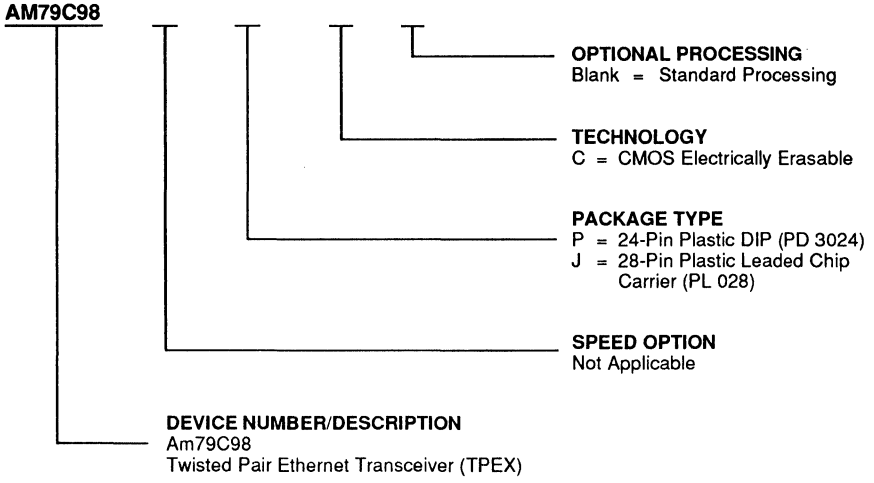
LOGIC SYMBOL



14395D-4

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM79C98	PC, JC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and check on newly released combinations.

PIN DESCRIPTION

AVDD
Analog Power

This pin supplies the +5 V to analog portions of TPEX circuitry.

AVSS
Analog Ground

This pin is the ground reference for analog portions of TPEX circuitry.

CI+,CI–
Control In
Output

AUI port differential driver.

DI+,DI–
Data In
Output

AUI port differential driver.

DO+,DO–
Data Out
Input

AUI port differential receiver.

DVDD
Digital Power

This pin supplies the +5 V to digital portions of TPEX circuitry.

DVSS
Digital Ground

This pin is the ground reference for digital portions of TPEX circuitry.

LNKST
Link Status
Open Drain, Input-Output

When this pin is tied LOW, the internal Link Test Receive function is disabled and the Transmit and Receive functions will remain active irrespective of arriving idle Link Test pulses and data. TPEX continues to generate idle Link Test pulses irrespective of the status of this pin.

As an output, this pin is driven LOW if the link is identified as functional. However, if the link is determined to be nonfunctional, due to missing idle Link Test pulses or data packets, then this pin is not driven. In the LOW output state, the pin is capable of sinking a maximum of 16mA and can be used to drive an LED.

This pin is internally pulled HIGH when inactive.

$\overline{\text{PRDN}}/\overline{\text{RST}}$
Power Down/Reset
Input, Active LOW

Driving this input LOW resets the internal logic of TPEX and places the device in a special Power Down mode. In the Power Down/Reset mode, all output drivers are placed in their inactive state.

RCV
Receive
Output

This pin is driven HIGH while TPEX is receiving data on the RXD pins and is transferring the received signal onto the AUI DI pair. The RCV and XMT pins are simultaneously driven HIGH during Collision.

REXT
External Resistor
Input

An external precision resistor is connected between this pin and AV_{DD}, in order to provide a voltage reference for the internal Voltage Controlled Oscillator (VCO).

RXD+,RXD–
Receive Data
Input

10BASE-T port differential receivers.

RXPOL
Receive Polarity

Open Drain, Input-Output

The twisted pair receiver is capable of detecting a receive signal with reversed polarity (wiring error). RXPOL pin is normally in the LOW state, indicating correct polarity of the received signal. If the receiver detects reversed polarity, then this pin is not driven (goes HIGH) and the polarity of subsequent packets is inverted. In the LOW output state, this pin can sink up to a maximum of 16 mA and is therefore capable of driving an LED.

This feature can be disabled by strapping this pin LOW. In this case the Receive Polarity correction circuit is disabled and the internal receive signal remains non-inverted, irrespective of the received signal.

This pin is internally pulled HIGH when inactive.

$\overline{\text{SQE TEST}}$
Signal Quality Test (Heartbeat) Enable
Input, Active LOW

The SQE test function is enabled by tying this input LOW.

This input is internally pulled HIGH when inactive.

TEST
Test
Input, Active HIGH

This pin should be tied LOW for normal operation. If this pin is driven HIGH, TPEX will enter Loopback Test mode. The type of loopback is determined by the state of the $\overline{\text{SQE TEST}}$ pin. If this pin is in the LOW state (Station MAU), TPEX transfers data independently from DO to the TXD/TXP circuit and from RXD to the DI circuit. If the $\overline{\text{SQE TEST}}$ is in the HIGH state (Repeater MAU), then data on the RXD circuit is transmitted back onto the TXD/TXP circuit and data on the DO circuit is transmitted onto the DI pair.

TXD+,TXD–
Transmit Data.
Output

10BASE-T port differential drivers.

TXP+,TXP–
Transmit Pre-Distortion
Output

Transmit waveform Pre-Distortion Control.

XMT
Transmit
Output

This pin is driven HIGH while TPEX is receiving data on the AUI DO pair and is transmitting data on the TXD/TXP pins. The XMT and RCV pins are simultaneously driven HIGH during Collision.

FUNCTIONAL DESCRIPTION

The Twisted Pair Ethernet Transceiver (TPEX) complies with the requirements specified by the IEEE 802.3 standard for the Attachment Unit Interface (AUI) and the standard for 10BASE-T Medium Attachment Unit (MAU). TPEX also implements a number of features in addition to the IEEE 802.3 standard. An outline of functions implemented by the Am79C98 are given below:

Attachment Unit Interface (DO+/–,DI+/–,CI+/–)

The AUI electrical and functional characteristics comply with that specified by the IEEE 802.3, sections 7 and 14 (drafted).The AUI pins can be wired directly to the isolation transformer, for a remote MAU application, or to another device (e.g. Am7992 Serial Interface Adapter). The end-of-packet SQE Test function (Heartbeat) can be disabled to allow the device to be employed in a Repeater application.

Twisted Pair Transmit Function

Data transmission to the 10BASE-T medium occurs when valid AUI signals appear on the DO+/- differential pair. This data stream is routed to the differential driver circuitry in the TXD+/– pins.The driver circuitry provides necessary electrical driving capability and pre-distortion control for transmitting signals over maximum length Twisted Pair cable, as specified by the IEEE 802.3 10BASE-T standard. The transmit function meets the propagation delays and jitter specified by the standard. During transmission, the XMT pin is driven HIGH and can be used for status information.

Twisted Pair Receive Function

The receiver complies with the receiver specifications of the IEEE 802.3 10BASE-T standard, including noise immunity and received signal rejection criteria ("Smart Squelch"). Signals meeting this criteria appearing at the RXD+/- differential input pair are routed to the DI+/- outputs. The receiver function meets the propagation delays and jitter requirements specified by the standard. Receiver squelch level drops to approximately half its threshold value after unsquelch to allow reception of minimum amplitude signals and to offset carrier fade in the event of worst case signal attenuation and crosstalk noise conditions. During receive, the RCV pin is driven HIGH and can be used for status information.

Link Test Function

The Link Test function is implemented as specified by the IEEE 802.3 10BASE-T standard. During periods of transmit pair inactivity, Link Test pulses will be periodically sent over the twisted pair medium to allow constant monitoring of medium integrity. When the Link Test function is enabled, the absence of Link Test pulses on the RXD+/– pair will cause the TPEX to go into a link fail state. In link fail state, data transmission, data reception, and the collision detection functions are disabled, and remain disabled until valid data or >2 consecutive Link Test pulses appear on the RXD+/– pair. During link fail, the LNKST pin is internally pulled HIGH. When the link is identified as functional, the LNKST pin is driven LOW, and is capable of directly driving a "link OK" LED. In order to interoperate with systems which do not implement Link Test, this function can be disabled by grounding the LNKST pin. When disabled, the driver and receiver function remain enabled irrespective of the presence or absence of data or Link Test pulses on the RXD+/– pair. The transmitter continues to generate Link Test pulses in the absence of transmit data even if the Link Test function is disabled.

Polarity Detection and Reversal

The TPEX receive function includes the ability to invert the polarity of the signals appearing at the RXD± pair if the polarity of the received signal is reversed (such as in the case of a wiring error). This feature allows data packets received from a reverse wired RXD± input pair to be corrected in the TPEX prior to transfer to the DTE via the AUI interface (DI±). The polarity detection function is activated following reset or Link Fail, and will reverse the receive polarity based on both the polarity of any previous Link Test pulses and the polarity of subsequent packets with a valid End Transmit Delimiter (ETD).

When in the Link Fail state, TPEX will recognize Link Test pulses of either positive or negative polarity. Exit from the Link Fail state is caused by the reception of five to six consecutive Link Test pulses of identical polarity. On entry to the Link Pass state, the polarity of the last five Link Test pulses is used to determine the initial receive polarity configuration and the receiver is reconfigured to subsequently recognize only Link Test pulses of the previously established polarity. This link pulse algorithm is employed only until ETD polarity determination is made as described later in this section.

Positive Link Test pulses are defined as received signals with a positive amplitude greater than 520 mV with a pulse width of 60 ns to 200 ns. This positive excursion may be followed by a negative excursion. This definition is consistent with the expected received signal at a correctly wired receiver, when a Link Test pulse which fits the template of Figure 14–12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

Negative Link Test pulses are defined as received signals with a negative amplitude greater than 520 mV with a pulse width of 60 ns to 200 ns. This negative excursion may be followed by a positive excursion. This definition is consistent with the expected received signal at a reverse wired receiver, when a Link Test pulse which fits the template of Figure 14–12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

The polarity detection/correction algorithm will remain "armed" until two consecutive packets with valid ETD of identical polarity are detected. When "armed", the receiver is capable of changing the initial or previous polarity configuration based on the most recent ETD polarity.

On receipt of the first packet with valid ETD following reset or Link Fail, TPEX will utilize the inferred polarity information to configure its RXD± input, regardless of its previous state. On receipt of a second packet with a valid ETD with correct polarity, the detection/correction algorithm will "lock-in" the initial polarity. If the second (or subsequent) packet is not detected as confirming the previous polarity decision, the most recently detected ETD polarity will be used as the new default. Note that packets with invalid ETD have no effect on updating the previous polarity decision. Once two consecutive packets with valid ETD have been received, TPEX will disable the detection/correction algorithm until either a Link Fail condition occurs or $\overline{\text{PRDN/RST}}$ is asserted.

During polarity reversal, the RXPOL pin is internally pulled HIGH. During normal polarity conditions, the RXPOL pin is driven LOW, and is capable of directly driving a "Polarity OK" LED using an integrated 16 mA driver. If desired, the Polarity Reversal function can be disabled by grounding the RXPOL pin.

Twisted Pair Interface Status

Two outputs (XMT and RCV) indicate whether TPEX is transmitting (AUI to Twisted Pair) or Receiving (Twisted Pair to AUI). Both signals are asserted during a collision. In link fail mode, RCV is disabled. In jabber detect mode, XMT is disabled. Both signals are active HIGH.

Collision Detect Function

Simultaneous Carrier Sense (presence of valid data signals) by both the AUI DO+/– pair and the RXD+/– pair constitutes a collision, thereby causing a 10 MHz signal to be asserted on the CI+/– output pair. The CI+/– output meets the drive requirements for the AUI interface. This 10 MHz signal will remain on the CI+/– pair until one of the two colliding states changes from active to idle. The CI+/– output pair stays HIGH for two bit times at the end of a collision, decreasing to the idle level within eighty bit times after the last Low-to-High transition. Both the XMT and RCV pins are driven HIGH during collision.

Signal Quality Error (SQE) Test (Heartbeat) Function

When the $\overline{\text{SQE TEST}}$ pin is driven LOW, TPEX will routinely exercise the collision detection circuitry by generating an SQE message at the end of every transmission. This signal is a self-test indication to the DTE that the MAU collision circuitry is functional. An SQE message consists of a 10 MHz signal on the CI+/– pair with a duration of 8 bit times (800 ns). When enabled, a SQE Test will occur at the end of every transmission, starting eight bit times (800 ns) after the last transition of the transmitted signal. For repeater applications, the SQE Test function can be disabled by tying the $\overline{\text{SQE TEST}}$ pin HIGH or by leaving it disconnected.

Jabber Function

The Jabber function inhibits the twisted pair transmit function of TPEX if the DO+/– circuit is active longer than the time permitted to transmit the maximum length 802.3/Ethernet data packet (50 ms nominal). This prevents any one node from disrupting the network due to a "stuck-on" or faulty transmitter. If this maximum transmit time is exceeded, TPEX transmitter circuitry is disabled and a 10 MHz signal is driven onto the CI+/– pair. Once the transmit data stream is removed from the DO+/– pair of inputs, an "unjab" time of 250 ms to 750 ms will elapse before TPEX removes the 10 MHz signal from the CI+/– pair and re-enables the Transmit path.

Power Down

In addition to onboard power-on-reset circuitry, the $\overline{\text{PRDN/RST}}$ pin is used as the master reset for TPEX. $\overline{\text{PRDN/RST}}$ must be driven LOW for a minimum of two microseconds for reset to occur. The $\overline{\text{PRDN/RST}}$ pin can also be used to put the TPEX into an inactive state, causing the device to consume less power. This feature is useful in battery powered or low duty cycle systems. Driving $\overline{\text{PRDN/RST}}$ LOW resets the internal logic of TPEX, and places the device into idle mode. In this mode, the Twisted Pair driver pins (TXD+/–,TXP+/–) are driven LOW, the AUI pins (CI+/–, DI+/–) are driven HIGH, the LNKST and RXPOL pins are in the inactive state, and XMT and RCV are LOW. TPEX will remain in IDLE as long as $\overline{\text{PRDN/RST}}$ is asserted. Following the rising edge of the signal on $\overline{\text{PRDN/RST}}$, TPEX will remain in the reset state for 10 µs.

Test Modes

TPEX implements two types of loopback test modes suitable for Station (DTE) or Repeater applications. The Test mode is entered by driving the TEST pin HIGH. The two types of test modes available are:

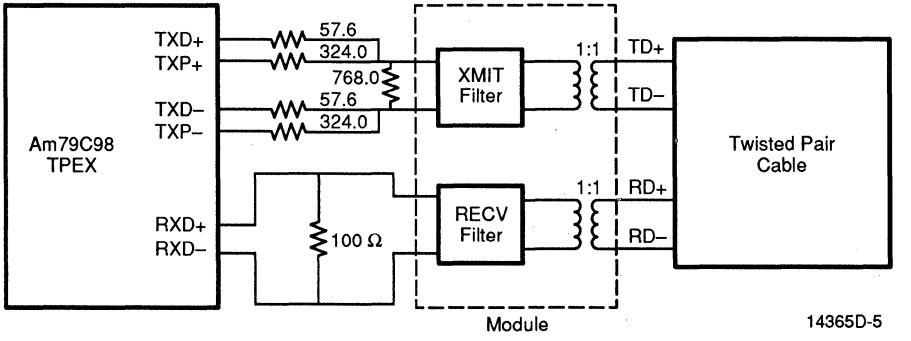
1. **Station (DTE):** $\overline{\text{SQE EST}}$ pin LOW. Data on DO+/– pair is transmitted onto the TXD+/– and TXP+/– pairs and data on the RXD+/– input pair is transmitted onto the DI+/– output pair. The jabber function and collision detection function are disabled.

2. **Repeater:** $\overline{\text{SQE TEST}}$ pin HIGH. Data on DO+/– pair is looped back onto the DI+/– pair and data on the RXD+/– pair is re-transmitted on the Twisted Pair drivers (TXD+/– and TXP+/– pairs).

In both modes the jabber circuitry, collision detection, and collision oscillator functions are disabled, and the AUI and RXD+/– squelch circuits are active.

TPEX External Components

Figure 1 shows a typical twisted pair port external components schematic. The resistors used should have a ±1% tolerance to ensure interoperability with 10BASE-T compliant networks. Filters and pulse transformers are necessary devices that have a major influence on the performance and compliance of a TPEX-based MAU. Specifically, the transmitted waveforms are heavily influenced by filter characteristics and the twisted pair receivers employ several criteria to continuously monitor the incoming signal's amplitude and timing characteristics to determine when and if to assert the internal carrier sense. For these reasons, it is crucial that the values and tolerances of the external components be as specified. Several manufacturers produce a module that combines the functions of the transmit and receive filters and the pulse transformers into one package.



The Filter/Transformer Module shown is available from the following manufacturers:

Belfuse	*TDK*
Pulse Engineering	*PCA*
Valor Electronics	*Nano Pulse*

Figure 1. Typical TP Port External Components

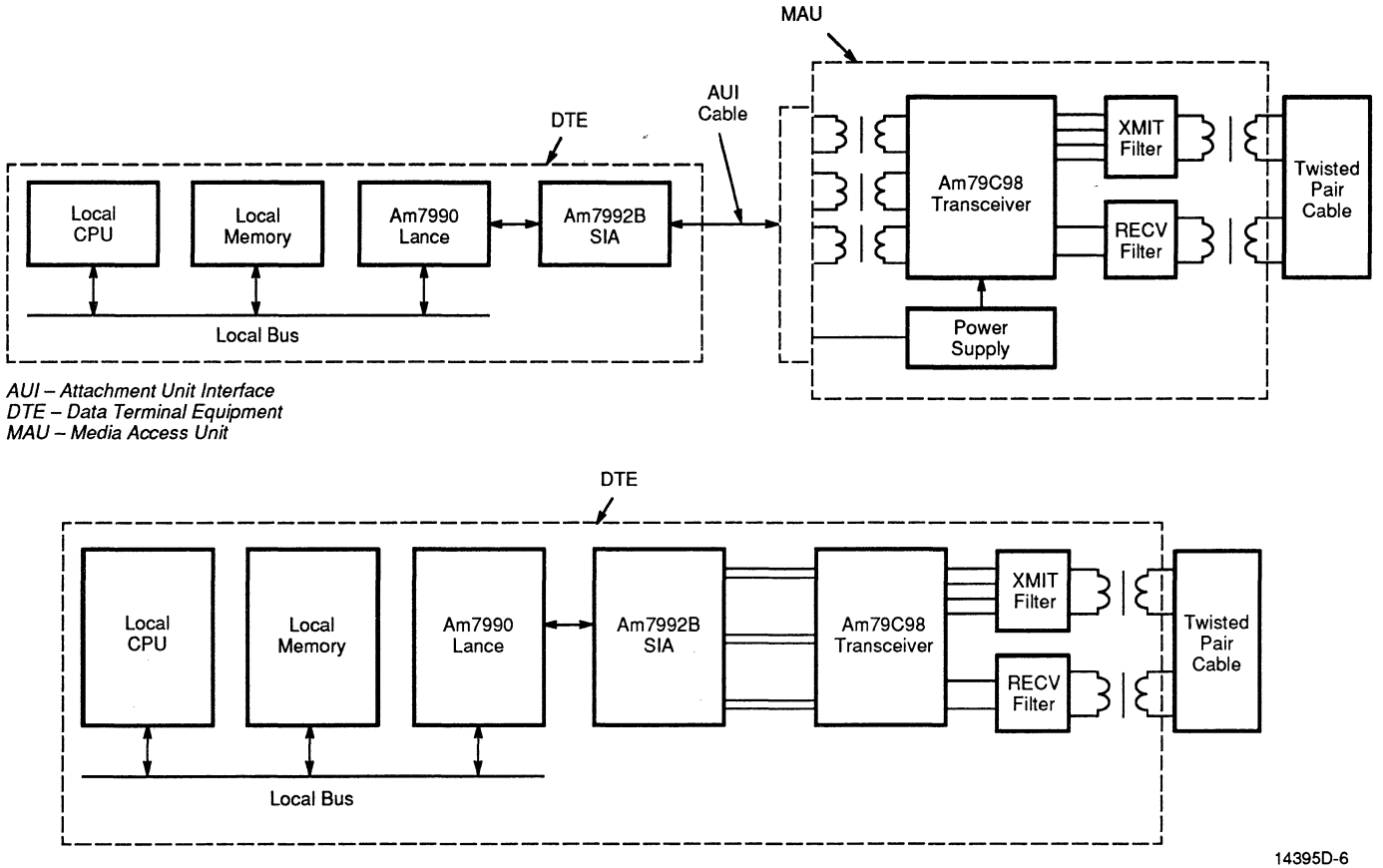
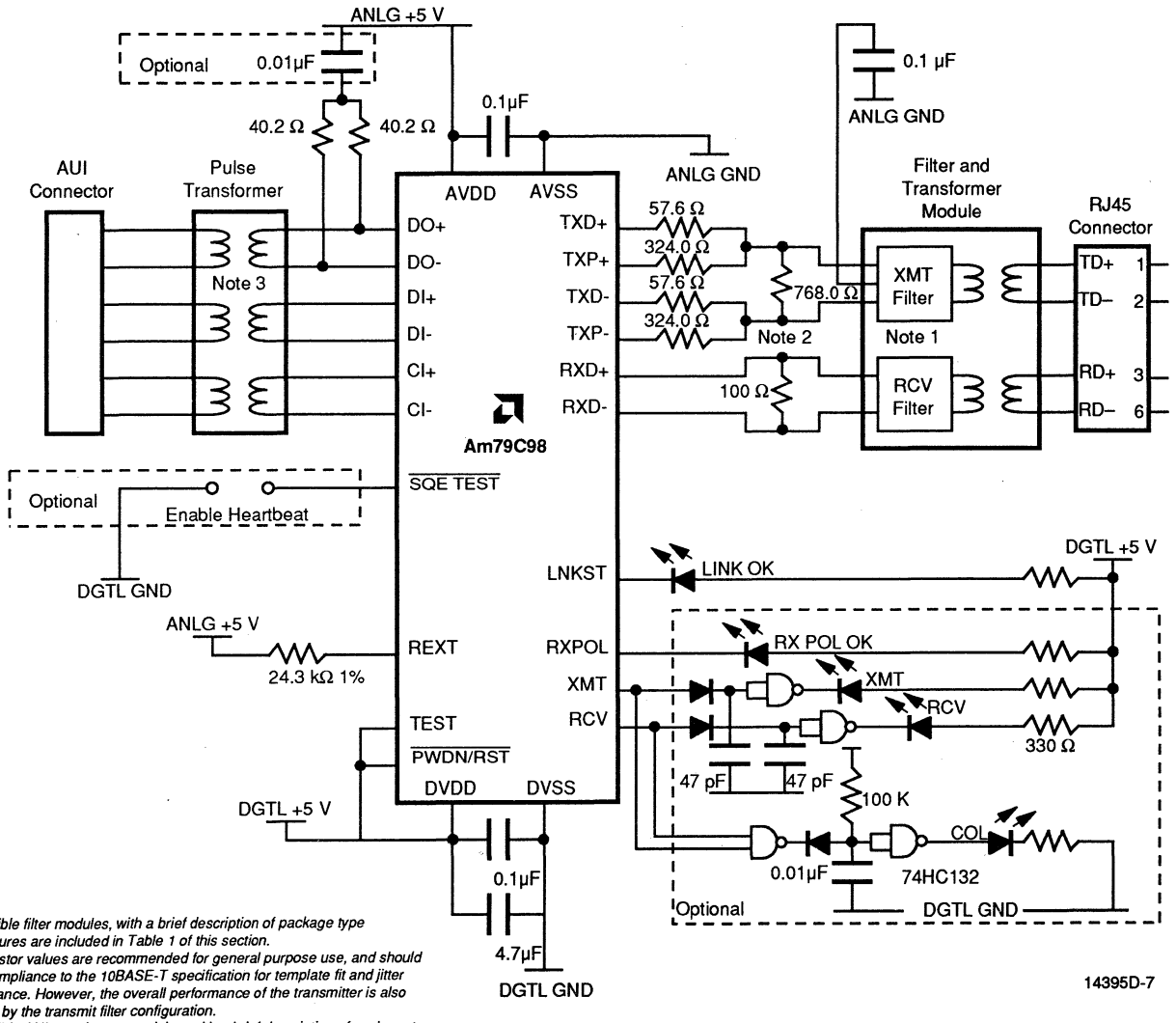


Figure 2. Typical Twisted Pair Ethernet Node



Notes:

1. *Compatible filter modules, with a brief description of package type and features are included in Table 1 of this section.*
2. *The resistor values are recommended for general purpose use, and should allow compliance to the 10BASE-T specification for template fit and jitter performance. However, the overall performance of the transmitter is also affected by the transmit filter configuration.*
3. *Compatible AUI transformer modules, with a brief description of package type and features are included in Table 2 of this section.*

Figure 3. Typical TPEX System Application

Table 1. TPEX Compatible Media Interface Modules

Manufacturer	Part #	Package	Description
Bel Fuse	A556-2006-DE	16-pin 0.3" DIL	Transmit and receive filters and transformers
Bel Fuse	0556-2006-00	14-pin SIP	Transmit and receive filters and transformers
Bel Fuse	0556-2006-01	14-pin SIP	Transmit and receive filters, transformers and common mode chokes
Valor Electronics	PT3877	16-pin 0.3" DIL	Transmit and receive filters and transformers
Valor Electronics	PT3983	8-pin 0.3" DIL	Transmit and receive common mode chokes
Valor Electronics	FL1012	16-pin 0.3" DIL	Transmit and receive filters and transformers, transmit common mode choke
Nano pulse	NP6612	16-pin 0.3" DIL	Transmit and receive filters, transformers and common mode chokes
Nano pulse	NP6581	8-pin 0.3" DIL	Transmit and receive common mode chokes
Nano pulse	NP6696	24-pin 0.6" DIL	Transmit and receive filters, transformers and common mode chokes
TDK	TLA 470	14-pin SIP	Transmit and receive filters and transformers
TDK	HIM3000	24-pin 0.6" DIL	Transmit and receive filters, transformers and common mode chokes
Pulse Engineering	PE65421	16-pin 0.3" DIL	Transmit and receive filters and transformers
Pulse Engineering	SUPRA 1.1	16-pin 0.5" DIL	Transmit and receive filters and transformers, transmit common mode choke
Bel Fuse	0556-6392-00	16-pin 0.5" DIL	Transmit and receive filters, transformers, and common mode chokes

Table 2. Am79C98 TPEX Compatible AUI Transformers

Manufacturer	Part #	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3" DIL	50 µH
Valor Electronics	LT6031	16-pin 0.3" DIL	50 µH
TDK	TLA 100-3E	16-pin 0.3" DIL	100 µH
Pulse Engineering	PE64106	16-pin 0.3" DIL	50 µH

ABSOLUTE MAXIMUM RATINGS

Storage Temperature: ..... -65°C to +150°C

Ambient Temperature Under Bias: .... 0°C to +70°C

Supply Voltage to AVss or DVss (AV_{DD}, DV_{DD}): ..... −0.3 V to +6 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A): ..... 0°C to +70°C

Supply Voltages (AV_{DD}, DV_{DD}): ..... +5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating range unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Digital Input Voltage					
V_{IL}	Input LOW Voltage		DV_{SS}-0.5	0.8	V
V_{IH}	Input HIGH Voltage		2.0	0.5 + DV_{DD}	V
Digital Output Voltage					
V_{OL}	Output LOW Voltage	I_{OL1} = 16 mA (Open drain) I_{OL2} = 4.0 mA		0.4	V
V_{OH}	Output HIGH Voltage	I_{OH} = -0.4 mA	2.4		V
Digital Input Leakage Current					
I_{ILL}	Input Leakage Current	0 < V_{IN} < DV_{DD} + 0.5 V		10	µA
I_{ILD}	Input Leakage Current (Open drain pins, output inactive)	0 < V_{IN} < DV_{DD} + 0.5 V		500	µA
AUI					
I_{IAXD}	Input Current at DO+, DO-	-1 < Vin < AV_{DD} + 0.5 V	-500	500	µA
V_{AICM}	DO+/- Open Circuit Input Common Mode Voltage (Bias)	I_{IN} = 0 V	AV_{DD} - 3.0	AV_{DD} - 1.0	V
V_{AIDV}	Differential Mode Input Voltage Range (DO+/-)	AV_{DD} = 5 V	-2.5	+2.5	V
V_{ASQ}	DO+/- Squelch Threshold		-160	-275	mV
V_{ATH}	DO+/- Switching Threshold	(Note 1)	-35	+35	mV
V_{AOD}	Differential Output Voltage |(DI+) - (DI-)| OR |(CI+) - (CI-)|	R_L = 78 Ω	620	1100	mV
V_{AODI}	DI+/- & CI+/- Differential Output Voltage Imbalance	R_L = 78 Ω (Note 1)	-25	+25	mV
V_{AODOFF}	DI+/- & CI+/- Differential Idle Output Voltage	R_L = 78 Ω	-40	+40	mV
I_{AODOFF}	DI+/- & CI+/- Differential Idle Output Current	R_L = 78 Ω (Note 1)	-1	1	mA
V_{AOCM}	DI+/- & CI+/- Common Mode Output Voltage	R_L = 78 Ω	2.5	AV_{DD}	V

DC CHARACTERISTICS (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Twisted Pair Interface					
I_{IRXD}	Input Current at RXD+/–	$AV_{SS} < V_{IN} < AV_{DD}$	–500	500	µA
R_{RXD}	RXD+/– Differential Input Resistance	(Note 1)	10		KΩ
V_{TIVB}	RXD+, RXD– Open Circuit Input Voltage (Bias)	I_{IN} = 0 mA	AV_{DD} - 3.0	AV_{DD} - 1.5	V
V_{TIDV}	Differential Mode Input Voltage Range (RXD+/–)	AV_{DD} = +5 V	–3.1	3.1	V
V_{TSQ+}	RXD Positive Squelch Threshold (Peak)	Sinusoid 5 MHz < f< 10 MHz	300	520	mV
V_{TSQ-}	RXD Negative Squelch Threshold (Peak)	Sinusoid 5 MHz < f< 10 MHz	–520	–300	mV
V_{THS+}	RXD Post-Squelch Positive Threshold (Peak)	Sinusoid 5 MHz < f< 10 MHz	150	293	mV
V_{THS-}	RXD Post-Squelch Negative Threshold (Peak)	Sinusoid 5 MHz < f< 10 MHz	–293	–150	mV
V_{RXDTH}	RXD Switching Threshold	(Note 1)	–60	60	mV
V_{TXH}	TXD+/– and TXP+/– Output HIGH Voltage	(Note 2) DV_{SS} = 0 V	DV_{DD} - 0.6	DV_{DD}	V
V_{TXL}	TXD+/– and TXP+/– Output LOW Voltage	(Note 2) DV_{DD} = +5 V	DV_{SS}	DV_{SS} + 0.6	V
V_{TXI}	TXD+/– and TXP+/– Differential Output Voltage Imbalance		–40	+40	mV
V_{TXOFF}	TXD+/– and TXP+/– Differential Idle Output Voltage	DV_{DD} = +5 V	–40	+40	mV
R_{TX}	TXD+/– and TXP+/– Differential Driver Output Impedance	(Note 1)		40	Ω
I_{IREXT}	Input Current at REXT Pin	R_{EXT} = 24.3K Ω ± 1% AV_{DD} = +5 V		120	µA
Power Supply Current					
I_{DD}	Power Supply Current (Transmitting 10 MHz Data) (Typical TP load)	$\overline{PRDN/RST}$ = HIGH		115	mA
	Power Supply Current (Transmitting 10 MHz Data) (No TP load)	$\overline{PRDN/RST}$ = HIGH		90	mA
I_{DDPRDN}	Power Supply Current in Power Down Mode	$\overline{PRDN/RST}$ = LOW		4	mA

SWITCHING CHARACTERISTICS over COMMERCIAL

Parameter Symbol	Parameter Description		Min	Max	Unit
Transmit Timing					
t_{PWODO}	DO Pulse Width Accept/Reject Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 3)	15	35	ns
t_{PWKDO}	DO Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 4)	105	200	ns
t_{TON}	Transmit Start Up Delay			300	ns
t_{TSD}	Transmit Static Propagation Delay (DO to TXD)			120	ns
t_{DODION}	DO to DI Startup Delay			300	ns
t_{DODISD}	DO to DI Static Propagation Delay			100	ns
t_{TETD}	Transmit End of Transmission		250	450	ns
t_{TR}	Transmitter Rise Time (10% to 90%)			10	ns
t_{TF}	Transmitter Fall Time (90% to 10%)			10	ns
t_{TM}	Transmitter Rise and Fall Time Mismatch			4	ns
t_{THD}	DO L–>H to TXD+ L–>H and TXD- H->L Delay	Steady State (Note 1)	$t_{TSD} - 1.0$	$t_{TSD} + 1.0$	ns
t_{TLD}	DO H–>L to TXD+ H–>L and TXD– L–>H Delay	Steady State (Note 1)	$t_{TSD} - 1.0$	$t_{TSD} + 1.0$	ns
t_{THDP}	DO L–>H to TXP+ H–>L and TXP- L->H Delay	Steady State (Note 1)	$t_{TSD} + 40$	$t_{TSD} + 60$	ns
t_{TLDP}	DO H–>L to TXP+ L–>H and TXP– H->L Delay	Steady State (Note 1)	$t_{TSD} + 40$	$t_{TSD} + 60$	ns
t_{XMTON}	XMT Asserted Delay			100	ns
t_{XMTOFF}	XMT De-asserted Delay			300	ns
t_{PERLP}	Idle Signal Period		8	24	ms
t_{PWLP}	Idle Link Test Pulse Width	(Note 1)	75	120	ns
t_{PWPLP}	Predistortion Idle Link Test Pulse Width	(Note 1)	40	60	ns
t_{JA}	Transmit Jabber Activation Time		20	150	ms
t_{JR}	Transmit Jabber Reset Time		250	750	ms
t_{JREC}	Transmit Jabber Recovery Time (Minimum time gap between transmitted packets to prevent jabber activation)		1.0	–	µs

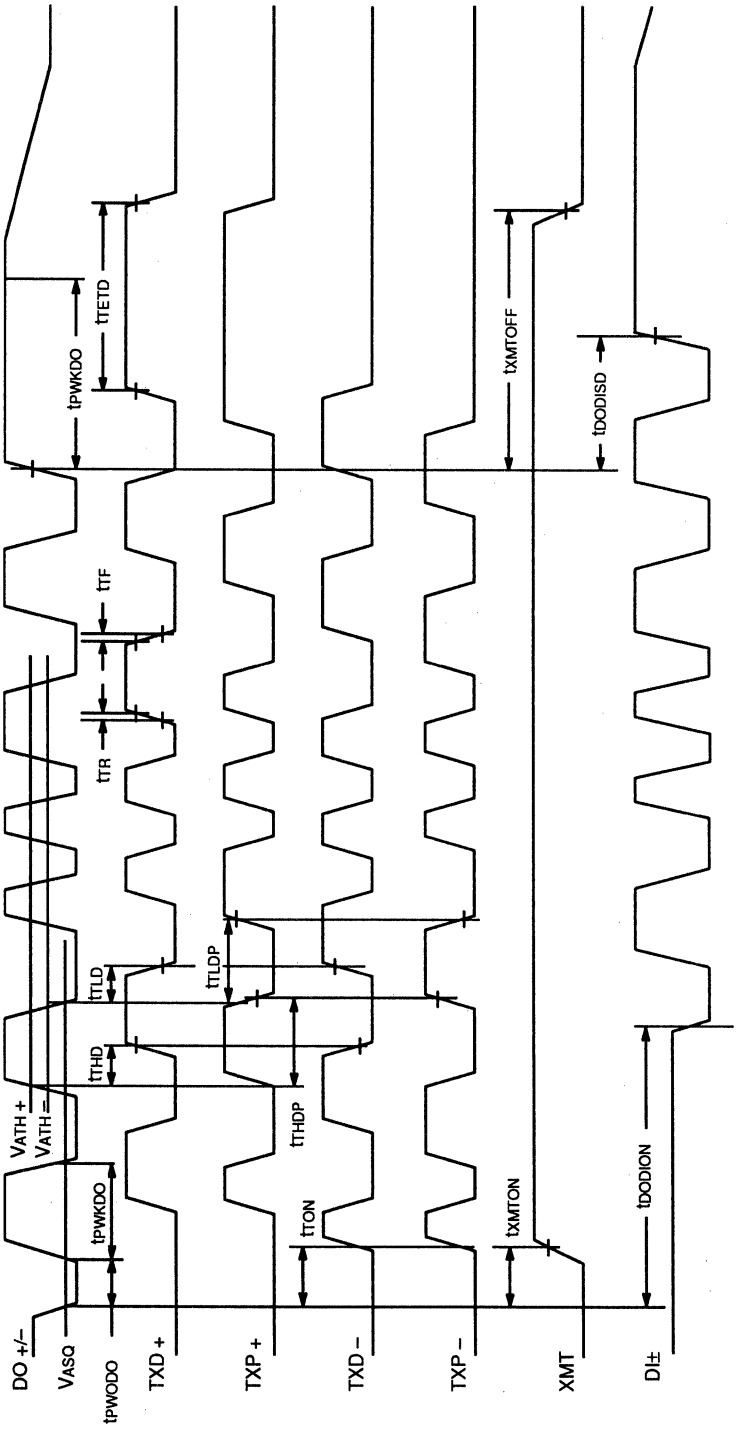
SWITCHING CHARACTERISTICS (continued)

Parameter Symbol	Parameter Description		Min	Max	Unit
Receive Timing					
t_{PWKRD}	RXD Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|V_{THS}\|$ (Note 5)	136	200	ns
t_{RON}	Receiver Start Up Delay (RXD to DI+/–)	5 MHz Sinusoid	200	400	ns
t_{RVB}	First Validly Timed Bit on DI+/– (RXD to DI)			t_{RON} + 100	ns
t_{RSD}	Receiver Static Propagation Delay (RXD to DI)			70	ns
t_{RETD}	DI End of Transmission		200		ns
t_{RHD}	RXD L–>H to DI+ L–>H and DI– H–>L Delay	(Note 1)	t_{RSD} – 2.5	t_{RSD} + 2.5	ns
t_{RLD}	RXD H–>L to DI+ H–>L and DI- L->H Delay	(Note 1)	t_{RSD} – 2.5	t_{RSD} + 2.5	ns
t_{RR}	DI+, DI–, CI+, CI– Rise Time (10% to 90%)			5.0	ns
t_{RF}	DI+, DI–, CI+, CI– Fall Time (10% to 90%)			5.0	ns
t_{RM}	DI+/– & CI+/– Rise and Fall Time Mismatch ($\|t_{RR} - t_{RF}\|$)			2.0	ns
t_{RCVON}	RCV Asserted Delay		t_{RON} – 50	t_{RON} + 100	ns
t_{RCVOFF}	RCV De-asserted Delay			t_{RSD} + 250	ns
Collision Detection and SQE Test					
t_{CON}	Collision Turn-On Delay (CI+/–)			500	ns
t_{COFF}	Collision Turn-Off Delay (CI+/–)			500	ns
t_{PER}	Collision Period (CI+/–)		87	117	ns
t_{CPW}	Collision Output Pulse Width (CI+/-)		40	60	ns
t_{SQED}	SQE Test Delay Time		600	1600	ns
t_{SQEL}	SQE Test Length		500	1500	ns

Notes:

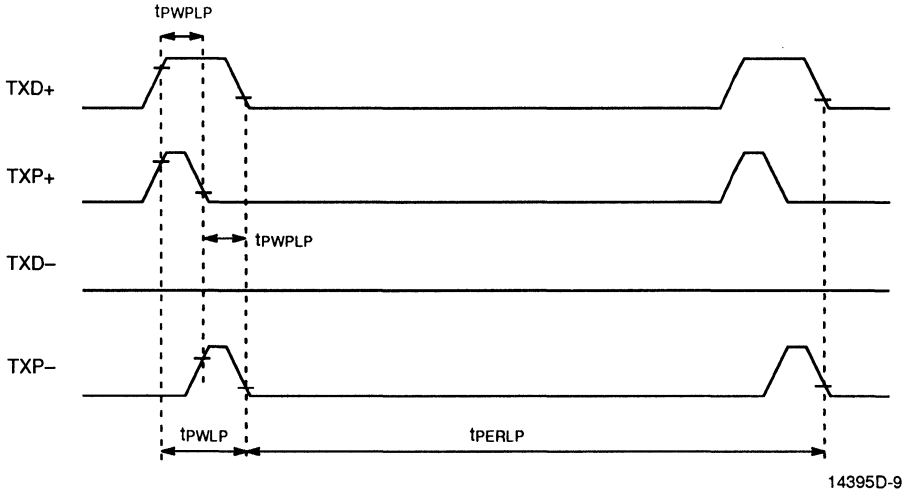
1. *Parameter not tested.*
2. *Uses switching test load.*
3. *DO pulses narrower than t_{PWODO} (min) will be rejected; pulses wider than t_{PWODO} (max) will turn internal DO carrier sense on.*
4. *DO pulses narrower than t_{PWKDO} (min) will maintain internal DO carrier sense on; pulses wider than t_{PWKDO} (max) will turn internal DO carrier sense off.*
5. *RXD pulses narrower than t_{PWKRD} (min) will maintain internal RXD carrier sense on; pulses wider than t_{PWKRD} (max) will turn internal RXD carrier sense off.*

SWITCHING WAVEFORMS



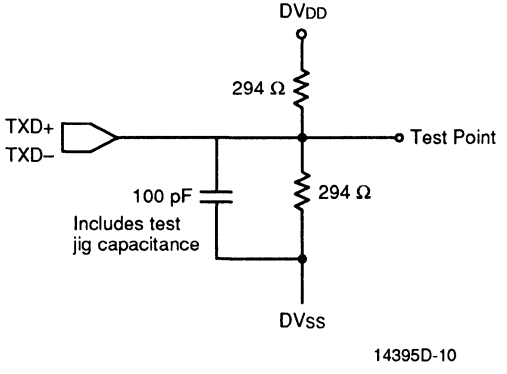
Transmit Timing

SWITCHING WAVEFORMS

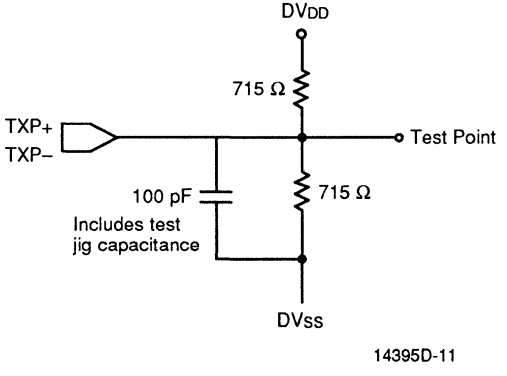


TP Idle Link Test Pulse

SWITCHING TEST CIRCUITS

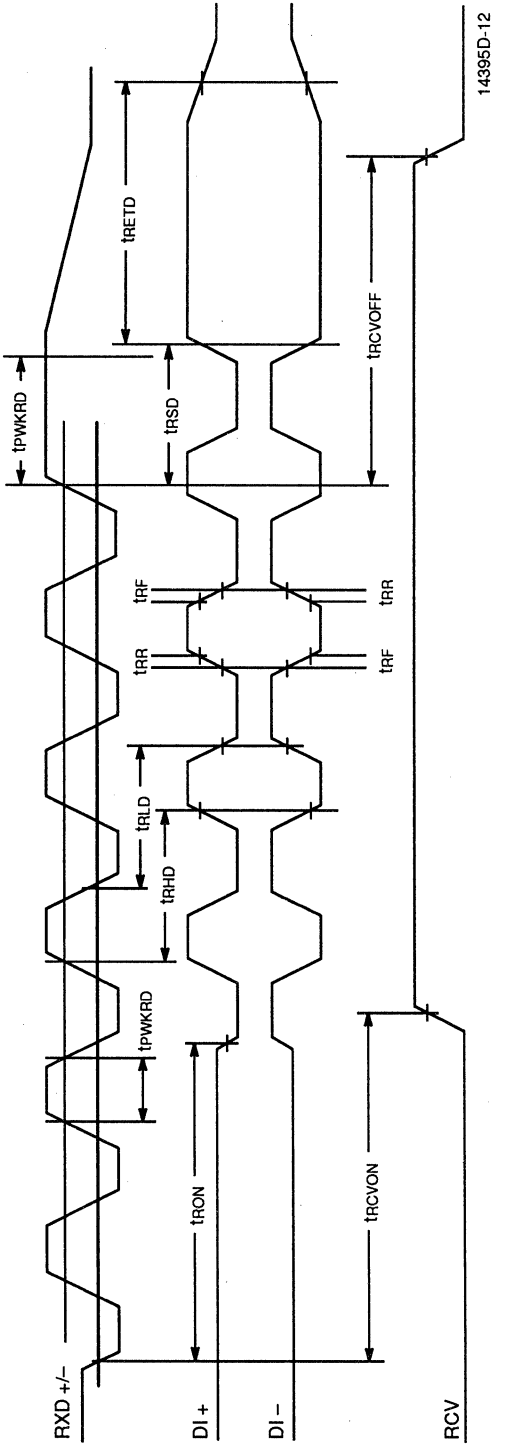


TXD Switching Test Circuit



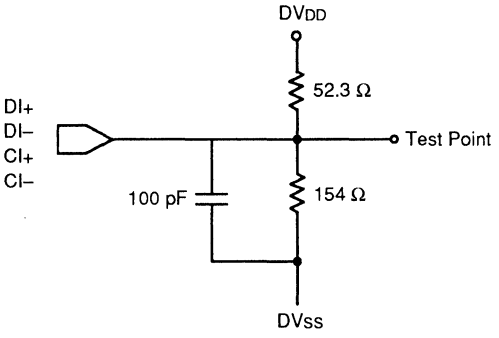
TXP Switching Test Circuit

SWITCHING WAVEFORMS



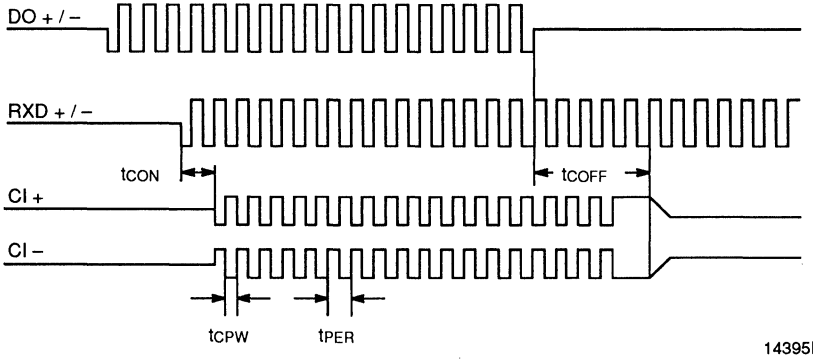
Receive Timing

RECEIVE TEST CIRCUIT



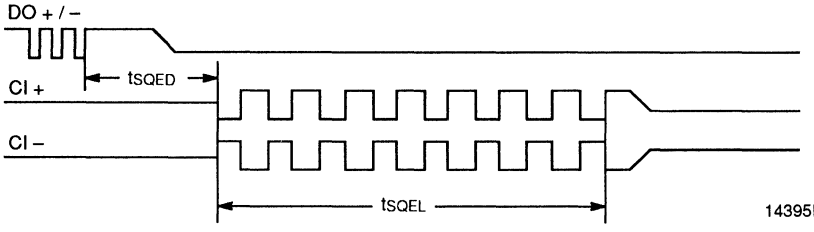
14395D-13

AUI DI, CI Switching Test Circuit



14395D-14

Collision Timing



14395D-15

SQE Test Timing ($\overline{\text{SQE Test}}$ Pin Connected to V_{SS})

FINAL

Am79C100

Twisted Pair Ethernet Transceiver Plus (TPEX Plus)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **CMOS device provides IEEE 802.3 compliant operation and low operating current from a single +5 V supply**
- **Power Down mode for reduced power consumption in battery powered applications**
- **Automatic Twisted Pair Link Integrity**
- **Pin-selectable Twisted Pair receive polarity detection and automatic inversion of the receive signal. Polarity indication output pin can directly drive a LED.**
- **Pin-selectable Twisted Pair Link Integrity Test capability conforming to the IEEE 802.3 standard. Link status pin can directly drive a LED.**
- **Transmit, Receive and Collision status indications available on separate, dedicated pins. Outputs can directly drive LEDs with pulses stretched to ensure LED visibility.**
- **Internal Twisted Pair transmitter digital predistortion circuit to reduce medium induced jitter**
- **Pin-selectable SQE Test (Heartbeat) enable**
- **AUI loop-back, Jabber Control, and SQE Test functions comply with the 10BASE-T Standard**
- **User selectable loopback operations**
- **Pin selectable Twisted Pair receive threshold programming for extended distance line lengths**

GENERAL DESCRIPTION

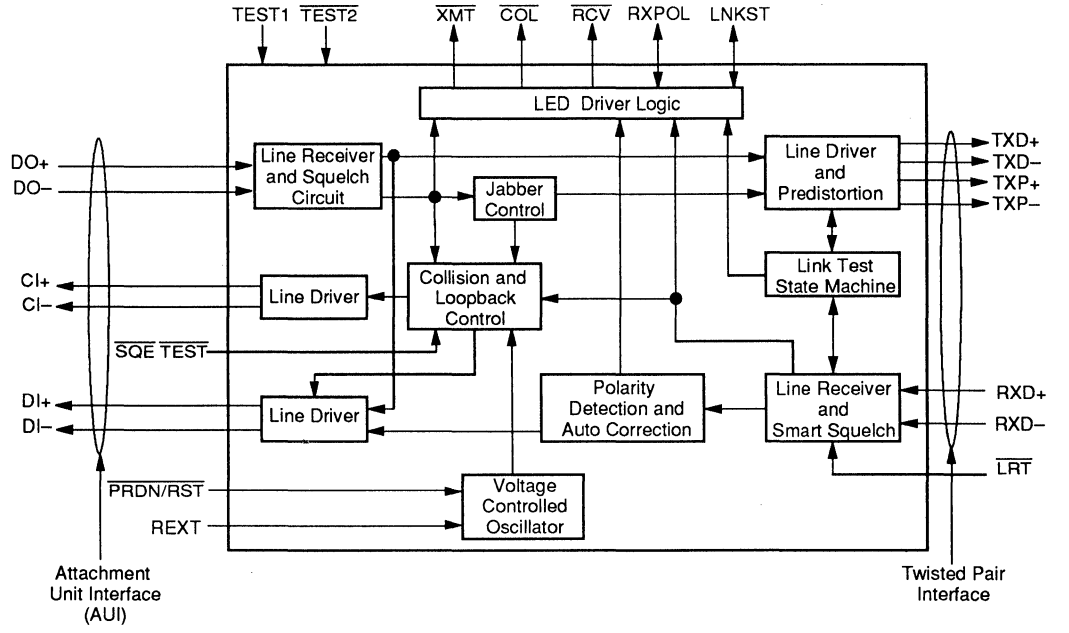
The Am79C100 Twisted Pair Ethernet Transceiver Plus (TPEX Plus) is an integrated circuit that implements the Medium Attachment Unit (MAU) functions for the Twisted Pair Medium, as specified by the supplement to IEEE 802.3 standard (Type 10BASE-T). This device provides the necessary electrical and functional interface between the IEEE 802.3 standard Attachment Unit Interface (AUI) and the Twisted Pair cable.

A network based on the 10BASE-T standard can use unshielded twisted pair cables, therefore providing an economical solution to networking by allowing the use of existing telephone wiring. The Am79C100 provides a minimal component count and cost effective solution to the design and implementation of 10BASE-T standard networks.

TPEX Plus provides twisted pair driver and receiver circuits, including on-board transmit digital predistortion, receiver squelch, and an AUI port with pin selectable SQE Test enable. The device provides a number of additional features including Link Status indication with Automatic Twisted Pair Receive Polarity Detection/Correction and indication; pin selectable receive threshold programming for extended distance line lengths; and Receive Carrier Sense, Transmit Active and Collision Present indication. The device provides separate Twisted Pair Link Status, Polarity Status, Receive, Transmit and Collision outputs to drive LEDs directly.

Publication# 16511 Rev. B Amendment/0
Issue Date: May 1994

BLOCK DIAGRAM



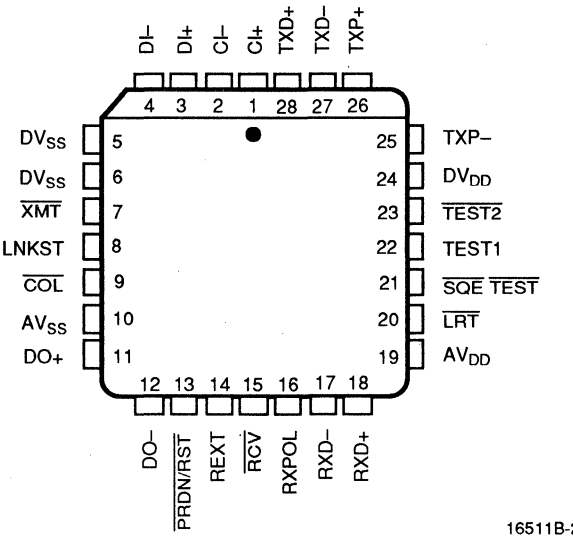
16511B-1

RELATED AMD PRODUCTS

Part No.	Description
Am7996	IEEE-802.3/Ethernet/Cheapernet Tap Transceiver
Am79C90	CMOS Local Area Network Controller for Ethernet™ (C-LANCE)
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA Single-Chip Ethernet Controller (with Microsoft® Plug n' Play support)
Am79C965	PCnet-32 Single-Chip Ethernet Controller (for 386DX, 486 and VL buses)
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems
Am79C98	Twisted-Pair Ethernet Transceiver (TPEX)
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)

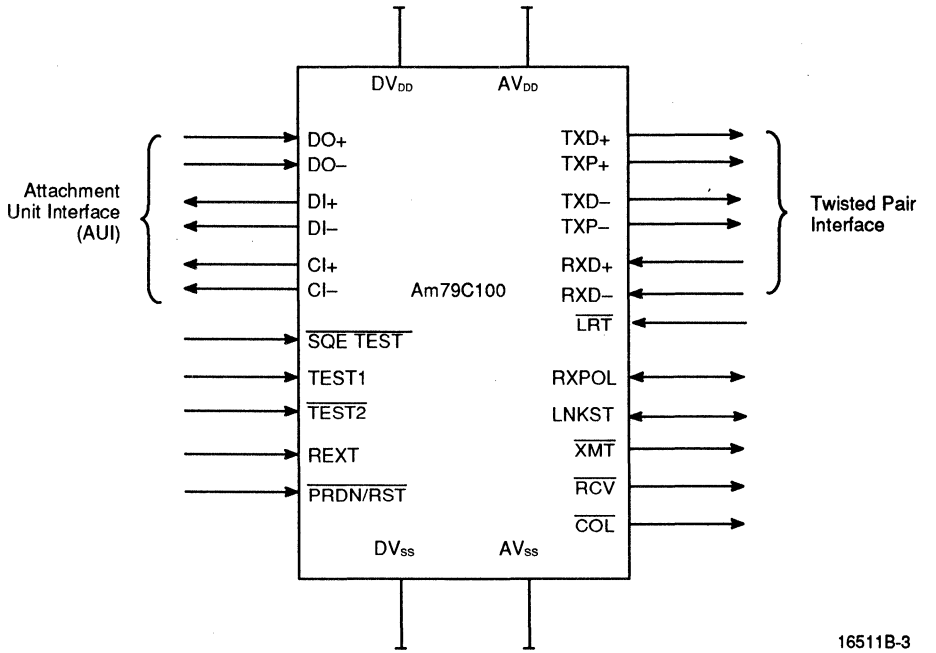
CONNECTION DIAGRAM

PLCC



16511B-2

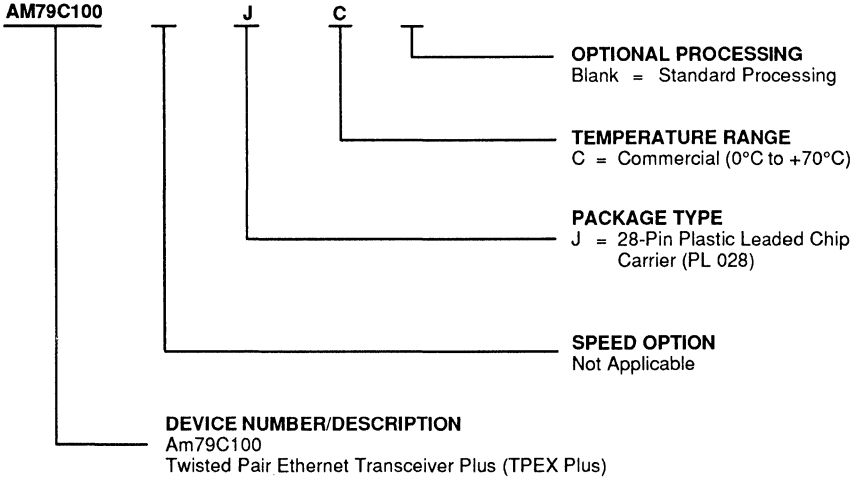
LOGIC SYMBOL



16511B-3

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM79C100	JC

Valid Combinations

The Valid Combinations table lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and check on newly released combinations.

PIN DESCRIPTION

AVDD
Analog Power

This pin supplies the +5 V to analog portions of TPEX Plus circuitry.

AVSS
Analog Ground

This pin is the ground reference for analog portions of TPEX Plus circuitry.

CI+, CI-
Control In
Output

AUI port differential driver.

$\overline{COL}$
Collision
Output,Open Drain

This pin is driven LOW while the TPEX Plus is simultaneously receiving data on the AUI DO pins and the twisted pair RXD pins, indicating a collision condition exists. It is also driven if TPEX Plus enters the jabber condition due to excessive length of activity on the DO pair. In this case TPEX Plus will wait for a period of inactivity on DO for the "unjab" time of 250 to 750 ms, before the 10 MHz pattern on the CI pair is removed and $\overline{COL}$ returns inactive. $\overline{COL}$ will not be driven during SQE Test activity on the AUI CI pair. In the LOW output state, the pin is capable of sinking a maximum of 12 mA and can be used to drive an LED. The $\overline{COL}$ output is pulse stretched for 20 to 62 ms after the end of collision, to ensure LED visibility.

DI+, DI–
Data In
Output

AUI port differential driver.

DO+, DO-
Data Out
Input

AUI port differential receiver.

DVDD
Digital Power

This pin supplies the +5 V to digital portions of TPEX Plus circuitry, including all transmit drivers.

DVSS
Digital Ground

Two pins provide the ground reference for digital portions of TPEX Plus circuitry, including all transmit drivers and the status indication LED drivers.

LNKST
Link Status
Input/Output, Open Drain

When this pin is tied LOW, the internal Link Test Receive function is disabled, and the Transmit and Receive functions will remain active regardless of arriving idle link pulses and data. TPEX Plus continues to generate idle link pulses irrespective of the status of this pin.

As an output, this pin is driven LOW if the link is identified as functional. However, if the link is determined to be nonfunctional, due to missing idle link pulses or data packets, then this pin is not driven (internally pulled HIGH). In the LOW output state, the pin is capable of sinking a maximum of 12 mA and can be used to drive an LED.

In the absence of external drive, the pin is internally pulled HIGH when inactive.

$\overline{LRT}$
Low Receive Threshold
Input, Active LOW

When this pin is tied LOW, the internal twisted pair receive thresholds are reduced by 4.5 dB from their original values (approximately 3/5 of the normal 10BASE-T value). With $\overline{LRT}$ in the HIGH state, the unsquelch threshold for the RXD± circuit will be 300 mV to 520 mV peak. With $\overline{LRT}$ in the LOW state, the unsquelch threshold for the RXD± circuit will be 180 mV to 312 mV peak. In either case, the RXD± circuit post unsquelch threshold will be approximately one half of the initial unsquelch threshold.

$\overline{PRDN}/\overline{RST}$
Power Down/Reset
Input, Active LOW

Driving this input LOW resets the internal logic of TPEX Plus and places the device in a special Power Down mode. In the Power Down/Reset mode, all output drivers are placed in their inactive state.

REXT
External Resistor
Input

An external precision resistor is connected between this pin and AVDD, in order to provide a current reference for the internal Voltage Controlled Oscillator (VCO).

$\overline{RCV}$
Receive
Output,Open Drain

This pin is driven LOW while TPEX Plus is receiving data on the twisted pair RXD pins and is transferring the received signal onto the AUI DI pair. The output is LOW

during Collision simultaneously with the $\overline{COL}$ pin. In the LOW output state, the pin is capable of sinking a maximum of 12 mA and can be used to drive an LED. The $\overline{RCV}$ output is pulse stretched for 20 ms to 62 ms after the end of reception, to ensure LED visibility.

RXD+, RXD-
Receive Data
Input

10BASE-T port differential receiver.

RXPOL
Receive Polarity
Input/Output, Open Drain

The twisted pair receiver is capable of detecting a receive signal with reversed polarity (wiring error). The RXPOL pin is normally in the LOW state, indicating correct polarity of the received signal. If the receiver detects a received packet with reversed polarity, then this pin is not driven (goes HIGH) and the polarity of subsequent packets is inverted. In the LOW output state, this pin can sink up to a maximum of 12 mA and is therefore capable of driving an LED.

This feature can be disabled by strapping this pin LOW. In this case the Receive Polarity correction circuit is disabled and the internal receive signal remains non-inverted, irrespective of the received signal.

In the absence of external drive, the pin is internally pulled HIGH when inactive.

$\overline{SQE\ TEST}$
Signal Quality Test (Heartbeat) Enable
Input, Active LOW

The SQE Test function is enabled by tying this input LOW. When enabled, TPEX Plus will send a 10 MHz burst (heartbeat) on the CI± lines after DO± has become inactive, indicating integrity of the collision detection and AUI circuitry. $\overline{SQE\ TEST}$ should be disabled for repeater applications.

In the absence of external drive, the pin is internally pulled HIGH when inactive.

TEST1
Test
Input, Active HIGH

This pin should be tied LOW for normal operation. TEST1 permits system level diagnostics to be performed. If TEST1 is driven HIGH (while $\overline{TEST2}$ is maintained HIGH), TPEX Plus will enter the Loopback Test mode. The type of loopback is determined by the state of the $\overline{SQE\ TEST}$ pin. If $\overline{SQE\ TEST}$ is in the LOW state (Station MAU), TPEX Plus transfers data independently from DO to the TXD/TXP circuits and from RXD to the DI circuit. If the $\overline{SQE\ TEST}$ is in the HIGH state (Repeater MAU), then data on the RXD circuit is transmitted back onto the TXD/TXP circuit and data on the DO circuit is transmitted onto the DI pair.

During either test mode, the collision detection and SQE Test functions are disabled, and CI± will remain idle. Link beat pulses will continue to be generated normally in the absence of TXD/TXP output activity, and the Link Test Receive State Machine will be forced into the Link Pass state. The $\overline{COL}$ pin will be driven LOW whenever a Link Beat pulse or transmit data activity commences, and remain low during the output activity. The receive squelch will continue to operate on both the RXD± and DO± input circuits.

In the absence of external drive, the pin is internally pulled LOW.

$\overline{TEST2}$
Test
Input, Active LOW

This pin should be tied HIGH for normal operation. $\overline{TEST2}$ is reserved for factory testing, and should be permanently tied HIGH.

In the absence of external drive, the pin is internally pulled HIGH.

TXD+, TXD-
Transmit Data
Output

10BASE-T port differential drivers.

TXP+, TXP-
Transmit Pre-Distortion
Output

Transmit wave form differential driver for pre-distortion.

$\overline{XMT}$
Transmit
Output, Open Drain

This pin is driven LOW while TPEX Plus is receiving data on the AUI DO pair and is transmitting data on the TXD/TXP pins. The output is LOW during collision simultaneously with the $\overline{COL}$ pin. In the LOW output state, the pin is capable of sinking a maximum of 12 mA and can be used to drive an LED. The $\overline{XMT}$ output is pulse stretched for 20 to 62 ms after the end of transmission, to ensure LED visibility.

FUNCTIONAL DESCRIPTION

The Twisted Pair Ethernet Transceiver Plus (TPEX Plus) complies with the requirements specified by the IEEE 802.3 standard for the Attachment Unit Interface (AUI) and the 10BASE-T Standard for a twisted pair Medium Attachment Unit (MAU). TPEX Plus also implements a number of features in addition to the IEEE 802.3 standard. An outline of functions implemented by the Am79C100 are given below.

Attachment Unit Interface (DO±, DI±, CI±)

The AUI electrical and functional characteristics comply with those specified within the IEEE 802.3 documents, sections 7 and 14. The AUI pins can be wired to an isolation transformer, for a remote MAU application, or directly to another device (e.g. Am7992B Serial Interface Adapter), in the case of a local DTE application. The end-of-packet SQE Test function (Heartbeat) can be disabled to allow the device to be employed in a Repeater application.

Twisted Pair Transmit Function

Data transmission to the 10BASE-T medium occurs when valid AUI signals appear on the DO± differential pair. This data stream is routed to the differential driver circuitry in the TXD± and TXP± pins. The driver circuitry provides the necessary electrical driving capability and the pre-distortion control for transmitting signals over maximum length Twisted Pair cable, as specified by the IEEE 802.3 10BASE-T Standard. During transmission, data is looped back to the DI± differential circuit, indicating normal operation. The transmit function for data output and loopback operations meets the propagation delays and jitter specified by the standard. During normal transmission, and providing that TPEX Plus is not in a Link Fail or jabber state, the $\overline{\text{XMT}}$ pin will be driven LOW, and can be used to drive a status LED directly.

Twisted Pair Receive Function

The receiver complies with the receiver specifications of the IEEE 802.3 10BASE-T standard, including noise immunity and received signal rejection criteria ("Smart Squelch"). Signals meeting this criteria appearing at the RXD± differential input pair are routed to the DI± outputs. The receiver function meets the propagation delays and jitter requirements specified by the standard. The receiver squelch level drops to approximately half its threshold value after unsquelch to allow reception of minimum amplitude signals and to mitigate carrier fade in the event of worst case signal attenuation and crosstalk noise conditions. During receive, the $\overline{\text{RCV}}$ pin is driven LOW and can be used to drive a status LED directly.

Note that the 10BASE-T standard defines the receive input amplitude at the external Media Dependent Interface (MDI). Filter and transformer loss are not specified. The TPEX Plus receiver squelch levels are defined to account for a 1 dB insertion loss at 10 MHz, which is typical for the type of receive filters/transformers recommended (see also Table 1).

Normal 10BASE-T compatible receive thresholds are employed when the $\overline{\text{LRT}}$ pin is inactive (HIGH). When the $\overline{\text{LRT}}$ pin is externally pulled LOW, the Low Receive Threshold option is invoked, and the sensitivity of the TPEX Plus receiver is increased. This allows longer line lengths to be employed, exceeding the 100 m target distance of normal 10BASE-T (assuming typical 24 AWG cable). The additional cable distance attributes directly to increased signal attenuation and reduced signal amplitude at the TPEX Plus receiver. However, from a system perspective, making the receiver more sensitive means that it is also more susceptible to extraneous noise, primarily caused by coupling from co-resident services (crosstalk). For this reason, it is recommended that when using the Low Receive Threshold option that the service should be installed on 4-pair cable only. Multi-pair cables within the same outer sheath have lower crosstalk attenuation, and may allow noise emitted from adjacent pairs to couple into the receive pair, and be of sufficient amplitude to falsely unsquelch the TPEX Plus.

Link Test Function

The link test function is implemented as specified by 10BASE-T standard. During periods of transmit pair inactivity, "Link Beat" pulses will be periodically sent over the twisted pair medium to allow constant monitoring of medium integrity.

When the link test function is enabled, the absence of Link Beat pulses and receive data on the RXD± pair will cause the TPEX Plus to go into a Link Fail state. In the Link Fail state, data transmission, data reception, data loopback and the collision detection functions are disabled, and remain disabled until valid data or >5 consecutive link pulses appear on the RXD± pair. During Link Fail, the LNKST pin is internally pulled HIGH. When the link is identified as functional, the LNKST pin is driven LOW, and is capable of directly driving a "Link OK" LED. In order to inter-operate with systems which do not implement link test, this function can be disabled by grounding the LNKST pin. With link test disabled, the data driver, receiver and loopback functions as well as collision detection remain enabled irrespective of the presence or absence of data or link pulses on the RXD± pair.

Polarity Detection and Reversal

The TPEX Plus receive function includes the ability to invert the polarity of the signals appearing at the RXD± pair if the polarity of the received signal is reversed (such as in the case of a wiring error). This feature allows data packets received from a reverse wired RXD± input pair to be corrected in the TPEX Plus prior to

transfer to the DTE via the AUI interface (DI±). The polarity detection function is activated following reset or Link Fail, and will reverse the receive polarity based on both the polarity of any previous Link Beat pulses and the polarity of subsequent packets with a valid End Transmit Delimiter (ETD).

When in the Link Fail state, TPEX Plus will recognize Link Beat pulses of either positive or negative polarity. Exit from the Link Fail state is caused by the reception of 5 to 6 consecutive Link Beat pulses of identical polarity. On entry to the Link Pass state, the polarity of the last 5 Link Beat pulses is used to determine the initial receive polarity configuration and the receiver is reconfigured to subsequently recognize only Link Beat pulses of the previously recognized polarity. This link pulse algorithm is employed only until SFD polarity determination is made as described later in this section.

Positive Link Beat pulses are defined as received signal with a positive amplitude greater than 520 mV ($\overline{LRT}$ = HIGH) with a pulse width of 60 ns to 200 ns. This positive excursion may be followed by a negative excursion. This definition is consistent with the expected received signal at a correctly wired receiver, when a Link Beat pulse which fits the template of Figure 14–12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

Negative Link Beat pulses are defined as received signals with a negative amplitude greater than 520 mV ($\overline{LRT}$ = HIGH) with a pulse width of 60 ns to 200 ns. This negative excursion may be followed by a positive excursion. This definition is consistent with the expected received signal at a reverse wired receiver, when a Link Beat pulse which fits the template of Figure 14–12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

The polarity detection/correction algorithm will remain "armed" until two consecutive packets with valid ETD of identical polarity are detected. When "armed", the receiver is capable of changing the initial or previous polarity configuration based on the most recent ETD polarity.

On receipt of the first packet with valid ETD following reset or Link Fail, TPEX Plus will utilize the inferred polarity information to configure its RXD± input, regardless of its previous state. On receipt of a second packet with a valid ETD with correct polarity, the detection/correction algorithm will "lock-in" the received polarity. If the second (or subsequent) packet is not detected as confirming the previous polarity decision, the most recently detected ETD polarity will be used as the default. Note that packets with invalid ETD have no effect on updating the previous polarity decision. Once two consecutive packets with valid ETD have been received, TPEX Plus will disable the detection/correction algorithm until either a Link Fail condition occurs or $\overline{PRDN/RST}$ is asserted.

During polarity reversal, the RXPOL pin is internally pulled HIGH. During normal polarity conditions, the RXPOL pin is driven LOW, and is capable of directly driving a "Polarity OK" LED using an integrated 12 mA driver. If desired, the Polarity Reversal function can be disabled by grounding the RXPOL pin.

Twisted Pair Interface Status

Three outputs ($\overline{XMT}$, $\overline{RCV}$ and $\overline{COL}$) indicate whether the TPEX Plus is transmitting (AUI to Twisted Pair), receiving (Twisted Pair to AUI), or in a collision state with both functions active simultaneously.

The TPEX Plus will power up in the Link Fail state. The normal algorithm will apply to allow it to enter the Link Pass state. On power up, the $\overline{XMT}$, $\overline{RCV}$, and $\overline{COL}$ LED drivers activate for 20 ms to 62 ms as a lamp test feature, and will then go to their inactive state until TPEX Plus enters the Link Pass state.

In the Link Pass state, transmit or receive activity which passes the pulse width/amplitude requirements of the DO± or RXD± inputs will be indicated by the $\overline{XMT}$ or $\overline{RCV}$ pin respectively going active. $\overline{XMT}$, $\overline{RCV}$, and $\overline{COL}$ are all asserted during a collision.

In the Link Fail state, $\overline{XMT}$, $\overline{RCV}$, and $\overline{COL}$ are disabled.

In jabber detect mode, TPEX Plus will activate the $\overline{COL}$ driver, disable the $\overline{XMT}$ driver (regardless of DO± activity), and allow the $\overline{RCV}$ driver to indicate the current state of the RXD± pair. If there is no receive activity on RXD±, only $\overline{COL}$ will be active during jabber detect. If there is RXD± activity, both $\overline{COL}$ and $\overline{RCV}$ will be active.

All three outputs are active LOW and incorporate 12 mA drive capability with 20 ms to 62 ms pulse stretch circuitry, to extend the event to ensure LED visibility.

Collision Detect Function

Simultaneous Carrier Sense (presence of valid data signals) by both the AUI DO± pins and the twisted pair RXD± pins constitutes a collision, thereby causing a 10 MHz signal to be asserted on the CI± output pair, and the $\overline{COL}$ output to be activated. The CI± output meets the drive requirements for the AUI interface. This 10 MHz signal will remain on the CI± pair until one of the two colliding states changes from active to idle. During the collision condition, data presented on the DI± pair will be sourced from the RXD± input. At the end of collision, the data presented on the DI± pair will be sourced from the last remaining active input, either RXD± or DO±. The CI± output pair stays HIGH for 2 bit times at the end of a collision, decreasing to the idle level within 80 bit times after the last transition. The $\overline{XMT}$, $\overline{RCV}$, and $\overline{COL}$ pins are driven LOW during collision.

Signal Quality Error (SQE) Test (Heartbeat) Function

When the $\overline{SQE\ TEST}$ pin is driven LOW, TPEX Plus will routinely exercise the collision detection circuitry by generating an SQE Test message at the end of every transmission. This signal is a self-test indication to the DTE that the MAU collision circuitry is functional and the AUI cable/connection is intact. An SQE Test message consists of a 10 MHz signal on the CI± pair with a duration of 5 to 15 bit times (500 ns to 1500 ns). When enabled, a SQE Test will occur at the end of every transmission, starting 6 to 16 bit times (600 ns to 1600 ns) after the last transition of the transmitted signal. For repeater applications, the SQE Test function can be disabled by tying the $\overline{SQE\ TEST}$ pin HIGH or by leaving it disconnected. The $\overline{COL}$ output will remain inactive during the SQE Test message on CI±.

Jabber Function

The Jabber function inhibits the twisted pair transmit function of TPEX Plus if the DO± circuit is active for an excessive period (20 ms to 150 ms). This prevents any one node from disrupting the network due to a "stuck-on" or faulty transmitter. If this maximum transmit time is exceeded, the TPEX Plus transmitter circuitry is disabled and a 10 MHz signal is driven onto the CI± pair. Once the transmit data stream is removed from the DO± input pair, an "unjab" time of 250 ms to 750 ms will elapse before the TPEX Plus removes the 10MHz signal from the CI± pair and re-enables the transmit circuitry

When jabber is detected, TPEX Plus will activate the $\overline{COL}$ driver, disable the $\overline{XMT}$ driver (regardless of DO± activity), and allow the $\overline{RCV}$ driver to indicate the current state of the RXD± pair. If there is no receive activity on RXD±, only $\overline{COL}$ will be active during jabber detect. If there is RXD± activity, both $\overline{COL}$ and $\overline{RCV}$ will be active.

Power Down

In addition to on board power-on-reset circuitry, the $\overline{PRDN/RST}$ pin is used as the master reset for TPEX Plus. $\overline{PRDN/RST}$ must be driven LOW for a minimum of 2 µs for reset to occur. The $\overline{PRDN/RST}$ pin can also be used to put the TPEX Plus into an inactive or "sleep" state, causing the device to consume less power. This feature is useful in battery powered or low duty cycle systems. Driving $\overline{PRDN/RST}$ LOW resets the internal logic of TPEX Plus, and places the device into idle mode. In this mode, the Twisted Pair driver pins (TXD±,TXP±) are driven LOW, the AUI pins (CI±, DI±) are pulled to AV_{DD}, the LNKST and RXPOL pins are in the inactive state, and the $\overline{XMT}$, $\overline{RCV}$, and $\overline{COL}$ pins are in the high impedance state. TPEX Plus will remain in idle mode as long as $\overline{PRDN/RST}$ is asserted.

Following the rising edge of the signal on $\overline{PRDN/RST}$, TPEX Plus will remain in the reset state for up to 10 µs. Immediately after the reset condition is removed, TPEX Plus will drive the $\overline{XMT}$, $\overline{RCV}$ and $\overline{COL}$ outputs low for 20 ms to 62 ms as a lamp test feature, and will be forced into the Link Fail state. TPEX Plus will move to the Link Pass state only after 5 to 6 Link Beat pulses and/or a single received message is detected on the RXD± pair.

Test Modes

TPEX Plus implements two types of loopback test modes suitable for Station (DTE) or Repeater applications. The Test Mode is entered by driving the TEST1 pin HIGH. The $\overline{TEST2}$ pin is intended for factory test only and should be tied HIGH for Test Mode or normal operation. The two available Test Modes are:

1. **Station (DTE):** $\overline{SQE\ TEST}$ pin LOW. Data received on the DO± input pair is transmitted onto the TXD± and TXP± output pairs, and data received on the RXD± input pair is transmitted onto the DI± output pair.
2. **Repeater:** $\overline{SQE\ TEST}$ pin HIGH. Data received on the DO± input pair is looped back onto the DI± output pair, and data received on the RXD± pair is looped back and re-transmitted on the twisted pair drivers (TXD± and TXP± pairs).

In both modes TPEX Plus will be forced into the Link Pass state, and will not enter the Link Fail state regardless of RXD± inactivity. The following functions are disabled: jabber circuit, collision detection, and collision oscillator. The functions which remain enabled are: the DO± and RXD± squelch circuits, $\overline{XMT}$ and $\overline{RCV}$ outputs, Link Beat pulse generation and polarity detection/correction. In addition, in both modes, the $\overline{COL}$ pin (not used to indicate collision during Test Modes) will go active for the duration of any transmit activity on the TXD±/TXP± pairs, providing a leading high-to-low edge indicating the start of packet transmission or Link Beat pulse generation.

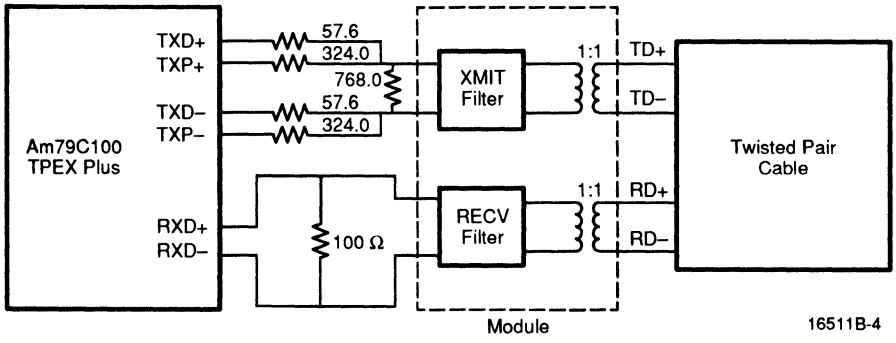
Upon exiting either of the Test Modes, the Link Test State Machine will be forced into the Link Fail state.

RXPOL may be pulled LOW and receive polarity correction will be disabled.

TPEX Plus External Components

Figure 1 shows a typical twisted pair port external components schematic. The resistors used should have a ±1% tolerance to ensure interoperability with 10BASE-T compliant networks. The filters and pulse transformers are necessary devices that have a major influence on the performance and compliance of a TPEX Plus based MAU. Specifically, the transmitted waveforms are heavily influenced by filter characteristics and the twisted pair receivers employ several criteria to continuously monitor the incoming signal's amplitude and timing characteristics to determine when and if to assert the internal carrier sense. For these reasons, it is crucial that the values and tolerances of the external components be as

specified. Several manufacturers produce a module that combines the functions of the transmit and receive filters and the pulse transformers into one package.



The Filter/Transformer Module shown is available from the following manufacturers:

Belfuse	*TDK*
Pulse Engineering	*PCA*
Valor Electronics	*Nano Pulse*

Figure 1. Typical Twisted Pair Port External Components

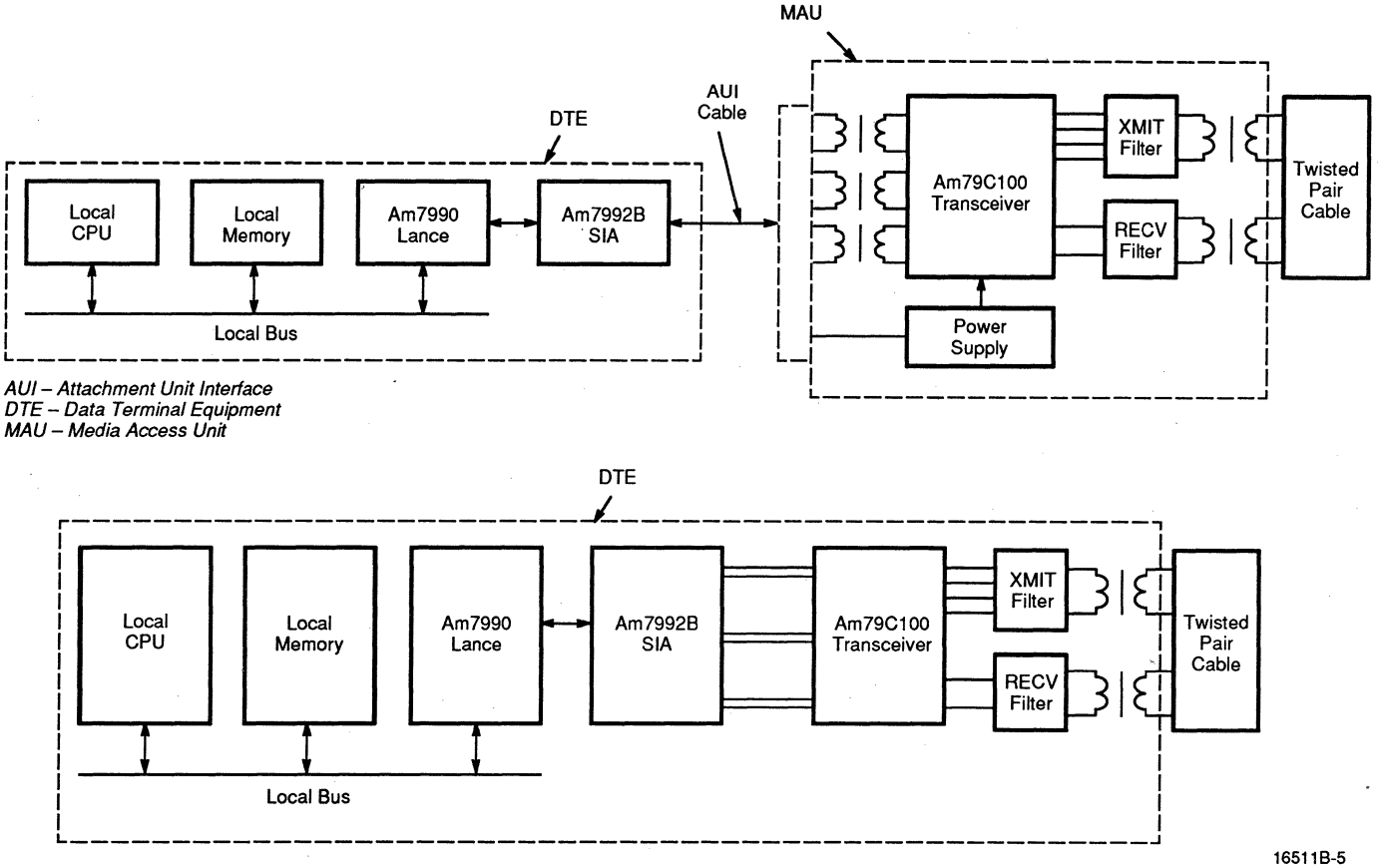
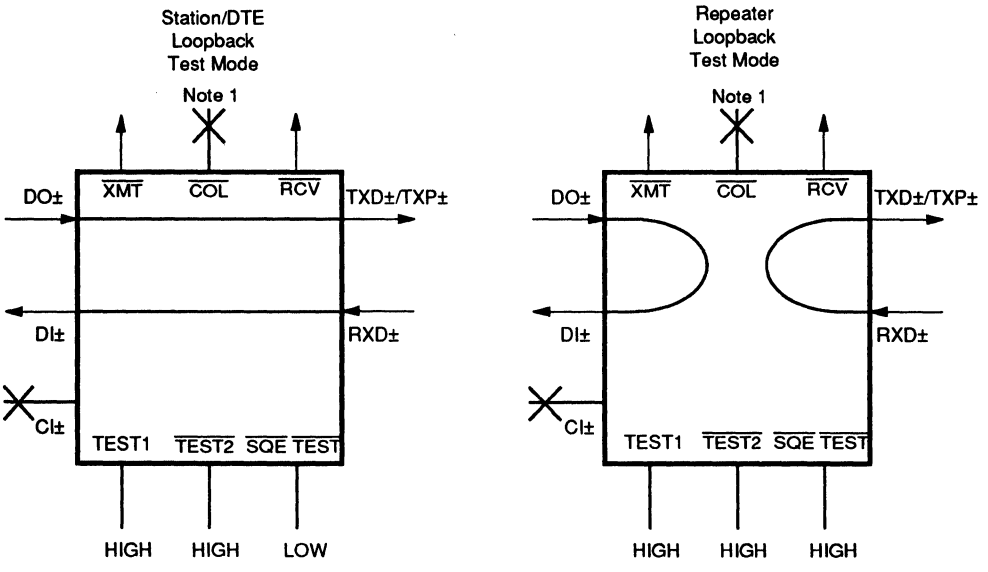


Figure 2. Typical Twisted Pair Ethernet Node

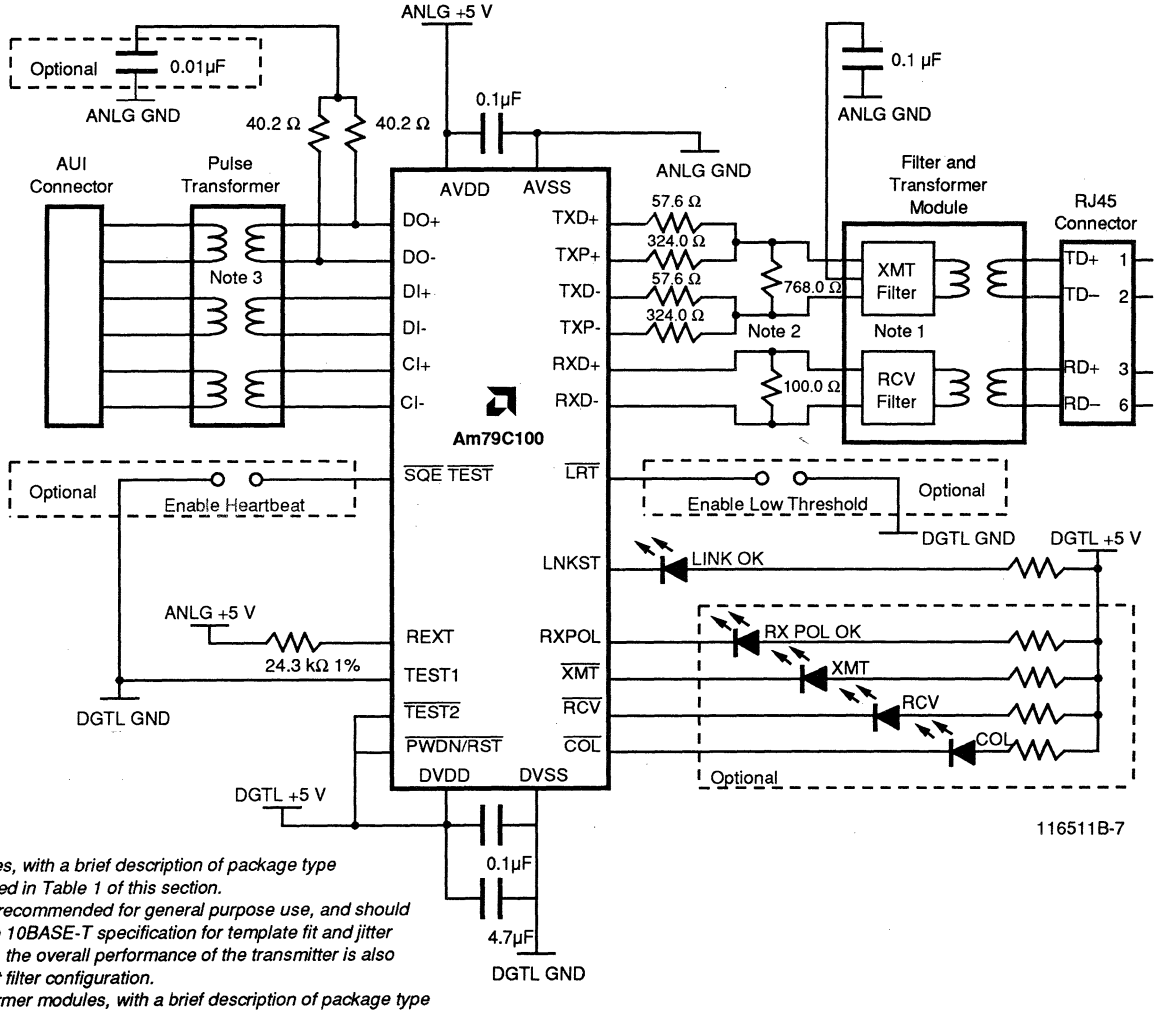


16511B-6

Note:

1. *During Loopback, the $\overline{\text{COL}}$ pin does not indicate collision, but instead provides indication of TXD±/TXP± activity. For details, refer to the section titled "Test Modes."*

Figure 3. Am79C100 TPEX Plus Loopback Operation



Notes:

1. *Compatible filter modules, with a brief description of package type and features are included in Table 1 of this section.*
2. *The resistor values are recommended for general purpose use, and should allow compliance to the 10BASE-T specification for template fit and jitter performance. However, the overall performance of the transmitter is also affected by the transmit filter configuration.*
3. *Compatible AUI transformer modules, with a brief description of package type and features are included in Table 2 of this section.*

Figure 4. Am79C100 Stand Alone MAU System Application

Table 1. TPEX Plus Compatible Media Interface Modules

Manufacturer	Part #	Package	Description
Bel Fuse	A556-2006-DE	16-pin 0.3" DIL	Transmit and receive filters and transformers
Bel Fuse	0556-2006-00	14-pin SIP	Transmit and receive filters and transformers
Bel Fuse	0556-2006-01	14-pin SIP	Transmit and receive filters, transformers and common mode chokes
Valor Electronics	PT3877	16-pin 0.3" DIL	Transmit and receive filters and transformers
Valor Electronics	PT3983	8-pin 0.3" DIL	Transmit and receive common mode chokes
Valor Electronics	FL1012	16-pin 0.3" DIL	Transmit and receive filters and transformers, transmit common mode choke
Nano pulse	NP6612	16-pin 0.3" DIL	Transmit and receive filters, transformers and common mode chokes
Nano pulse	NP6581	8-pin 0.3" DIL	Transmit and receive common mode chokes
Nano pulse	NP6696	24-pin 0.6" DIL	Transmit and receive filters, transformers and common mode chokes
TDK	TLA 470	14-pin SIP	Transmit and receive filters and transformers
TDK	HIM3000	24-pin 0.6" DIL	Transmit and receive filters, transformers and common mode chokes
Pulse Engineering	PE65421	16-pin 0.3" DIL	Transmit and receive filters and transformers
Pulse Engineering	SUPRA 1.1	16-pin 0.5" DIL	Transmit and receive filters and transformers, transmit common mode choke
Bel Fuse	0556-6392-00	16-pin 0.5" DIL	Transmit and receive filters, transformers, and common mode chokes

Table 2. Am79C100 TPEX Plus Compatible AUI Transformers

Manufacturer	Part #	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3" DIL	50 µH
Valor Electronics	LT6031	16-pin 0.3" DIL	50 µH
TDK	TLA 100-3E	16-pin 0.3" DIL	100 µH
Pulse Engineering	PE64106	16-pin 0.3" DIL	50 µH

ABSOLUTE MAXIMUM RATINGS

Storage Temperature: . . . . . −65°C to +150°C

Ambient Temperature Under Bias: . . . . 0°C to +70°C

Supply Voltage to AVss or DVss
(AV_{DD}, DV_{DD}): . . . . . −0.3 V to +6 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A): . . . . . 0°C to +70°C
Supply Voltages (AV_{DD}, DV_{DD}): . . . . . +5 V ± 5%
All inputs within the range:
AV_{SS}–0.5 V ≤ V_{IN} ≤ AV_{DD} + 0.5 V, or
DV_{SS}–0.5 V≤ V_{IN} ≤ DV_{DD} +0.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating range unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Digital Input Voltage					
V_{IL}	Input LOW Voltage			0.8	V
V_{IH}	Input HIGH Voltage		2.0		V
Digital Output Voltage					
V_{OL}	Output LOW Voltage ($\overline{XMT}$, $\overline{RCV}$, $\overline{COL}$, LNKST and RXPOL)	I_{OL} = 12 mA (Open Drain)		0.4	V
Digital Input Leakage Current					
I_{ILL}	Input Leakage Current (PRDN/RST)	DV_{SS} < V_{IN} < DV_{DD}		10	µA
I_{ILD}	Input Leakage Current (LNKST/RXPOL, output inactive)	DV_{SS} < V_{IN} < DV_{DD}		500	µA
Digital Output Leakage Current					
I_{OLD}	Output Leakage Current ($\overline{XMT}$, $\overline{RCV}$, $\overline{COL}$)	DV_{SS} < V_{IN} < DV_{DD}		10	µA
AUI					
I_{IAXD}	Input Current at DO+, DO–	AV_{SS} < Vin < AV_{DD}	–500	500	µA
V_{AICM}	DO± Open Circuit Input Common Mode Voltage (Bias)	I_{IN} = 0 V	AV_{DD} –3.0	AV_{DD} –1.0	V
V_{AIDV}	Differential Mode Input Voltage Range (DO±)	AV_{DD} = +5 V	–2.5	+2.5	V
V_{ASQ}	DO± Squelch Threshold		–160	–275	mV
V_{ATH}	DO± Switching Threshold	(Note 1)	–35	+35	mV
V_{AOD}	Differential Output Voltage |(DI+)–(DI–)| OR |(CI+)–(CI–)|	R_L = 78 Ω	620	1100	mV
V_{AODI}	DI± & CI± Differential Output Voltage Imbalance	R_L = 78 Ω (Note 1)	–25	+25	mV
V_{AODOFF}	DI± & CI± Differential Idle Output Voltage	R_L = 78 Ω	–40	+40	mV
I_{AODOFF}	DI± & CI± Differential Idle Output Current	R_L = 78 Ω (Note 1)	–1	1	mA
V_{AOCM}	DI± & CI± Common Mode Output Voltage	R_L = 78 Ω	2.5	AV_{DD}	V

DC CHARACTERISTICS (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Twisted Pair Interface					
I_{IRXD}	Input Current at RXD±	$AV_{SS} < V_{IN} < AV_{DD}$	–500	500	uA
R_{RXD}	RXD± Differential Input Resistance	(Note 1)	10		KΩ
V_{TIVB}	RXD+, RXD– Open Circuit Input Voltage (Bias)	I_{IN} = 0 mA	AV_{DD} –3.0	AV_{DD} –1.5	V
V_{TIDV}	Differential Mode Input Voltage Range (RXD±)	AV_{DD} = +5 V	–3.1	3.1	V
V_{TSQ+}	RXD Positive Squelch Threshold (Peak)	Sinusoid 5 MHz < f < 10 MHz	300	520	mV
V_{TSQ-}	RXD Negative Squelch Threshold (Peak)	Sinusoid 5 MHz < f < 10 MHz	–520	–300	mV
V_{THS+}	RXD Post-Squelch Positive Threshold (Peak)	Sinusoid 5 MHz < f < 10 MHz	150	293	mV
V_{THS-}	RXD Post-Squelch Negative Threshold (Peak)	Sinusoid 5 MHz < f < 10 MHz	–293	–150	mV
V_{LTSQ+}	RXD Positive Squelch Threshold (Peak)	$\overline{LRT}$ = LOW	180	312	mV
V_{LTSQ-}	RXD Negative Squelch Threshold (Peak)	$\overline{LRT}$ = LOW	–312	–180	mV
V_{LTHS+}	RXD Post-Squelch Positive Threshold (Peak)	$\overline{LRT}$ = LOW	90	175	mV
V_{LTHS-}	RXD Post-Squelch Negative Threshold (Peak)	$\overline{LRT}$ = LOW	–175	–90	mV
V_{RXDTH}	RXD Switching Threshold	(Note 1)	–60	60	mV
V_{TXH}	TXD± and TXP± Output HIGH Voltage	DV_{SS} = 0 V (Note 2)	DV_{DD} –0.6	DV_{DD}	V
V_{TXL}	TXD± and TXP± Output LOW Voltage	DV_{SS} = +5 V (Note 2)	DV_{SS}	DV_{SS} + 0.6	V
V_{TXI}	TXD± and TXP± Differential Output Voltage Imbalance		–40	40	mV
V_{TXOFF}	TXD± and TXP± Idle Output Voltage	DV_{DD} = +5 V	–40	40	mV
R_{TX}	TXD± and TXP± Differential Driver Output Impedance	(Note 1)		40	Ω
I_{IREXT}	Input Current at REXT Pin	R_{EXT} = 24.3 kΩ ±1% AV_{DD} = +5 V		120	µA
Power Supply Current					
I_{DD}	Power Supply Current (Idle)	$\overline{PRDN/RST}$ = HIGH $DV_{DD} = AV_{DD}$ = +5 V		40	mA
	Power Supply Current (Transmitting—No TP load)	$\overline{PRDN/RST}$ = LOW		95	mA
	Power Supply Current (Transmitting—with TP load)	$\overline{PRDN/RST}$ = HIGH $DV_{DD} = AV_{DD}$ = +5 V		150	mA
I_{DDPRDN}	Power Supply Current in Power Down Mode	$\overline{PRDN/RST}$ = LOW		4	mA

Notes:

1. *Parameter not tested.*
2. *Uses switching test load.*

SWITCHING CHARACTERISTICS over COMMERCIAL operating ranges

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Transmit Timing					
t_{PWODO}	DO Pulse Width Accept/ Reject Threshold	V_{DO} > |V_{ASQ} max| (Note 3)	15	35	ns
t_{PWKDO}	DO Pulse Width Maintain/ Turn-Off Threshold	V_{DO} > |V_{ASQ} max| (Note 4)	105	200	ns
t_{TON}	Transmit Start Up Delay			300	ns
t_{TSD}	Transmit Static Propagation Delay (DO± to TXD±)			120	ns
t_{TETD}	Transmit End Transmit Delimiter		250	450	ns
t_{TR}	Transmitter Rise Time (10% to 90%)			10	ns
t_{TF}	Transmitter Fall Time (90% to 10%)			10	ns
t_{TM}	Transmitter Rise and Fall Time Mismatch			4	ns
t_{THD}	DO ↑ to TXD+ ↑ and TXD– ↓ Delay	Steady State (Note 1)	t_{TSD} – 1.0	t_{TSD} + 1.0	ns
t_{TLD}	DO ↓ to TXD+ ↓ and TXD– ↑ Delay	Steady State (Note 1)	t_{TSD} – 1.0	t_{TSD} + 1.0	ns
t_{THDP}	DO ↑ to TXP+ ↓ and TXP– ↑ Delay	Steady State (Note 1)	t_{TSD} + 40	t_{TSD} + 60	ns
t_{TLDP}	DO ↓ to TXP+ ↑ and TXP– ↓ Delay	Steady State (Note 1)	t_{TSD} + 40	t_{TSD} + 60	ns
t_{XMTON}	XMT Asserted Delay			100	ns
t_{XMTOFF}	XMT De-asserted Delay		20	62	ms
t_{PERLP}	Idle Signal Period		8	24	ms
t_{PWLP}	Link Beat Pulse Width	(Note 1)	75	120	ns
t_{PWPLP}	Predistortion Idle Link Beat Width	(Note 1)	40	60	ns
t_{JA}	Transmit Jabber Activation Time		20	150	ms
t_{JR}	Transmit Jabber Reset Time		250	750	ms
t_{JREC}	Transmit Jabber Recovery Time (Minimum time gap between transmitted packets to prevent jabber activation)	(Note 1)	1.0	–	µs
t_{DODION}	DO to DI Startup Delay			300	ns
t_{DODISD}	DO to DI Static Propagation Delay			100	ns

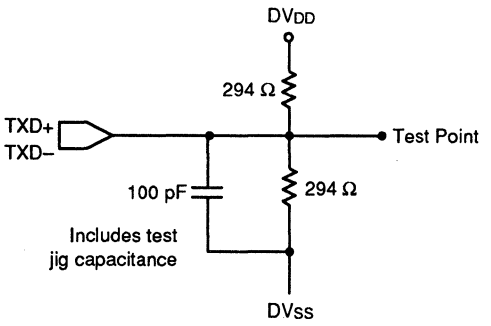
SWITCHING CHARACTERISTICS (Continued)

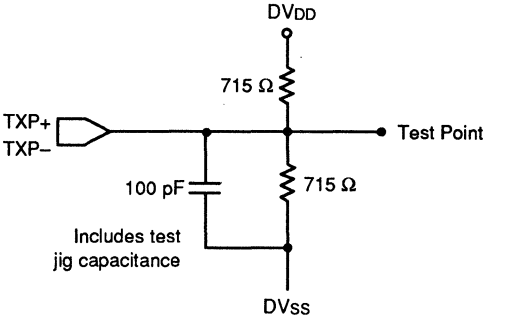
Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Receive Timing					
t_{PWKRD}	RXD Pulse Width Maintain/ Turn-Off Threshold	$V_{IN} > V_{THS}$ min (Note 5)	136	200	ns
t_{RON}	Receiver Start Up Delay (RXD to DI±)	Tested with 5 MHz Sinusoid	200	400	ns
t_{RVB}	First Validly Timed Bit on DI±			t_{RON} + 100	ns
t_{RSD}	Receiver Static Propagation Delay (RXD± to DI±)			70	ns
t_{RETD}	DI End of Transmission		200		ns
t_{RHD}	RXD ± ↑ to DI+ ↑ and DI– ↓ Delay	(Note 1)	t_{RSD} – 2.5	t_{RSD} + 2.5	ns
t_{RLD}	RXD ± ↓ to DI+ ↓ and DI– ↑ Delay	(Note 1)	t_{RSD} – 2.5	t_{RSD} + 2.5	ns
t_{RR}	DI+, DI–, CI+, CI– Rise Time (10% to 90%)			5	ns
t_{RF}	DI+, DI–, CI+, CI– Fall Time (10% to 90%)			5	ns
t_{RM}	DI± and CI± Rise and Fall Time Mismatch ($t_{RR} - t_{RF}$)			2	ns
t_{RCVON}	RCV Asserted Delay		t_{RON} – 50	t_{RON} + 100	ns
t_{RCVOFF}	RCV De-asserted Delay		20	62	ms
Collision Detection and SQE Test					
t_{CON}	Collision Turn-On Delay (CI±)			500	ns
t_{COFF}	Collision Turn-Off Delay (CI±)			500	ns
t_{PER}	Collision Period (CI±)		87	117	ns
t_{CPW}	Collision Output Pulse Width (CI±)		40	60	ns
t_{SQED}	SQE Test Delay Time		600	1600	ns
t_{SQEL}	SQE Test Length		500	1500	ns
t_{COLON}	COL Asserted Delay		t_{CON} – 50	t_{CON} + 100	ns
t_{COLOFF}	COL De-asserted Delay		20	62	ms

Notes:

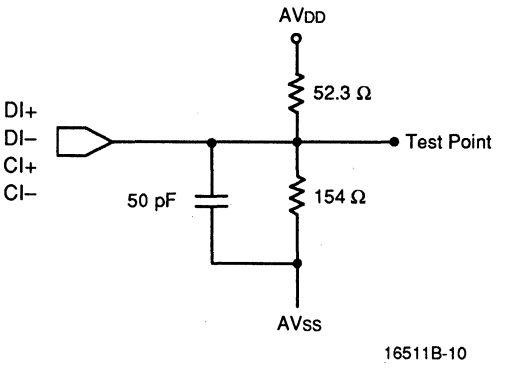
1. *Parameter not tested.*
2. *Uses switching test load.*
3. *DO pulses narrower than t_{PWODO} (min) will be rejected; pulses wider than t_{PWODO} (max) will turn internal DO carrier sense on.*
4. *DO pulses narrower than t_{PWKDO} (min) will maintain internal DO carrier sense on; pulses wider than t_{PWKDO} (max) will turn internal DO carrier sense off.*
5. *RXD pulses narrower than t_{PWKRD} (min) will maintain internal RXD carrier sense on; pulses wider than t_{PWKRD} (max) will turn internal RXD carrier sense off.*

SWITCHING TEST CIRCUITS



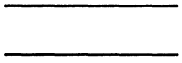
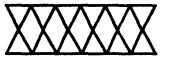
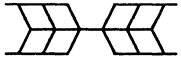


Twisted Pair Transmit Test Circuit



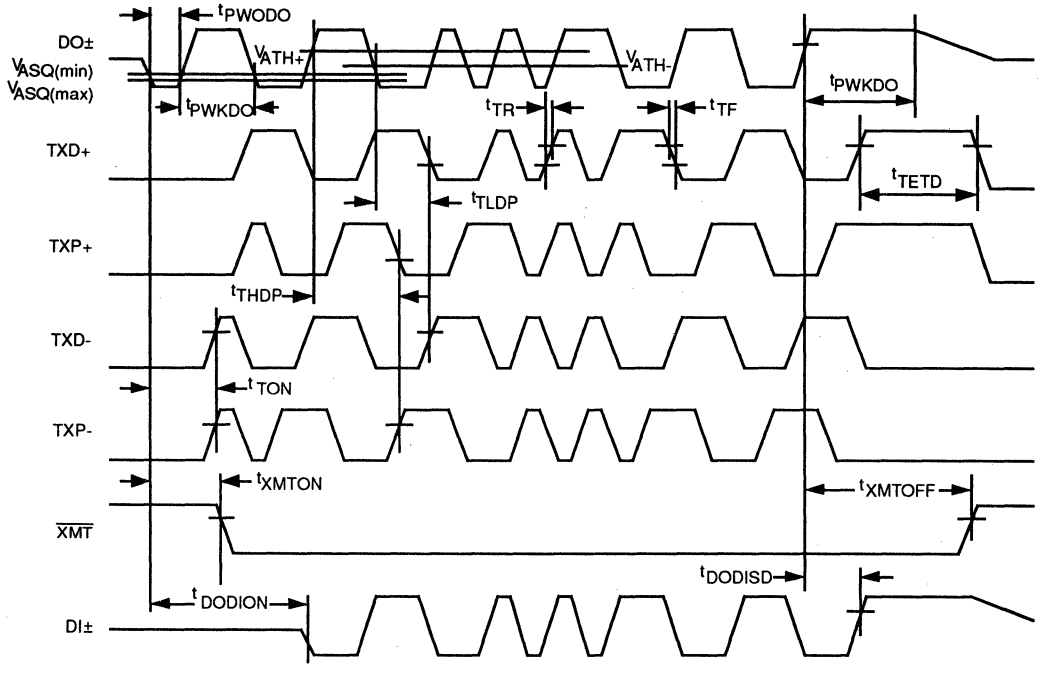
AUI Transmit Test Circuit

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off" State

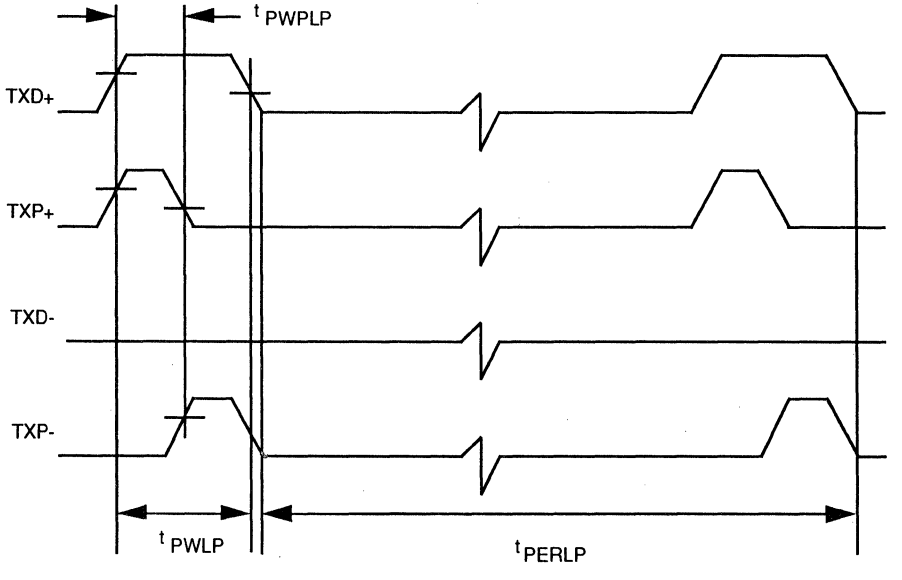
KS000010

SWITCHING WAVEFORMS



16511B-11

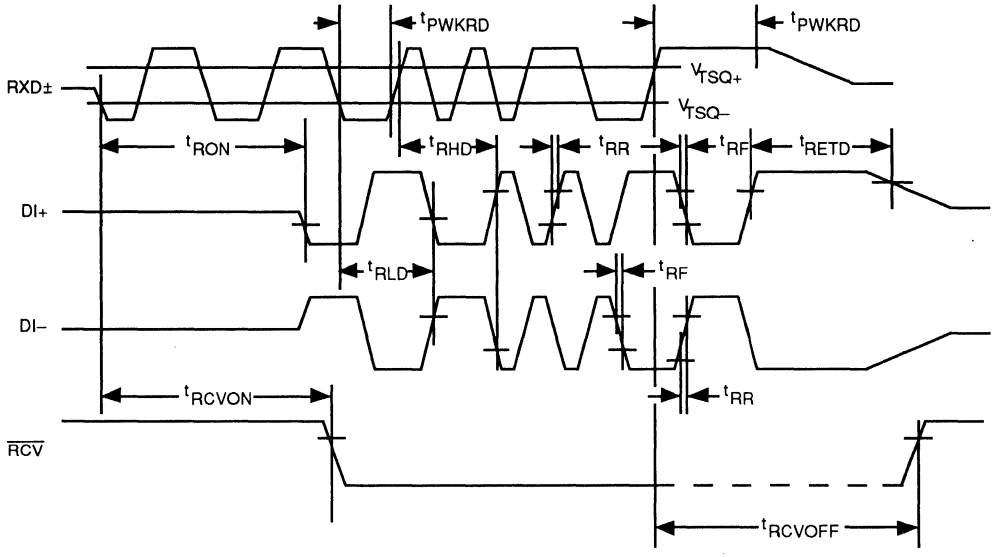
Transmit Timing



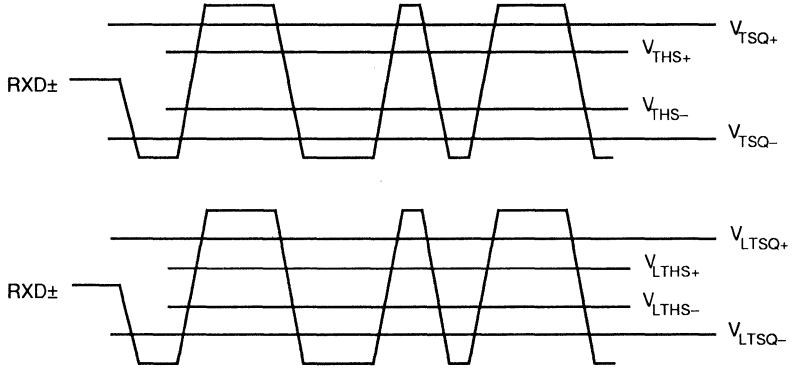
Transmit Link Beat Pulse

16511B-12

SWITCHING WAVEFORMS



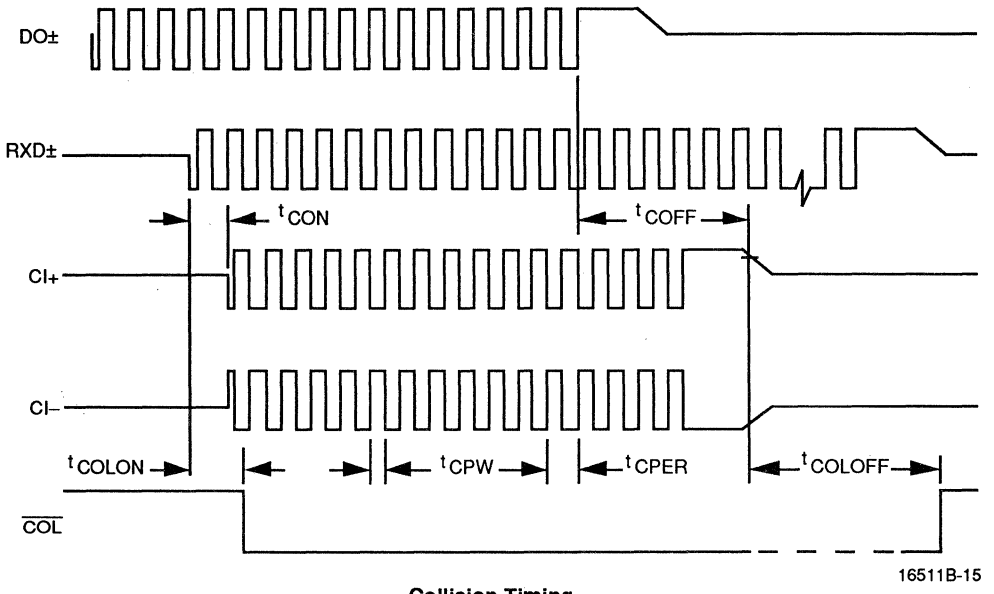
Receive Timing 16511B-13



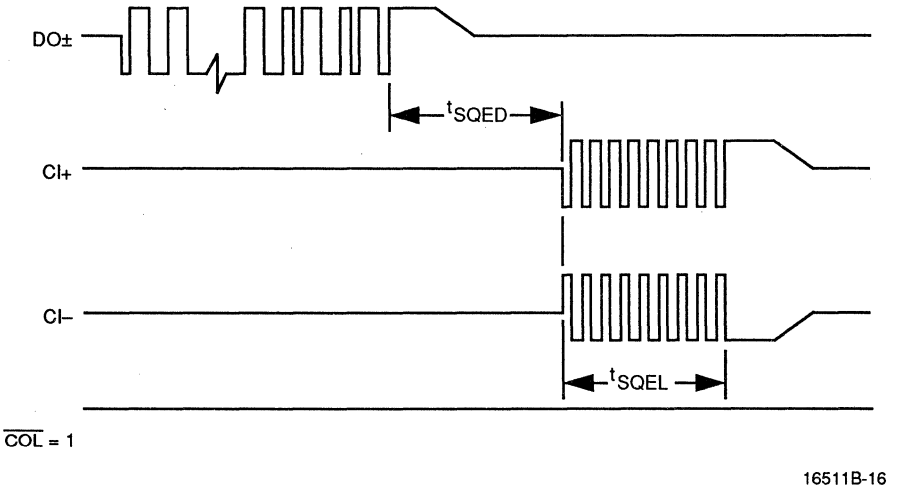
16511B-14

Receive Thresholds

SWITCHING WAVEFORMS



Collision Timing



SQE Test Timing

PRELIMINARY

Am79C981

Integrated Multiport Repeater Plus™ (IMR+™)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Enhanced version of AMD's Am79C980 Integrated Multiport Repeater™ (IMR™) chip with the following enhancements:**
  - Additional Management Port features
  - **Minimum Mode** provides support for an extra four LED outputs per port for additional status in non-intelligent repeater designs
  - Pin/socket compatible with the Am79C980 IMR chip
  - Fully backward compatible with existing IMR device designs
- **Interfaces directly with the Am79C987 HIMIB™ device to build a fully Managed Multiport Repeater**
- **CMOS device features high integration and low power with a single +5 V supply**
- **Repeater functions comply with IEEE 802.3 Repeater Unit specifications**
- **Eight integral 10BASE-T transceivers utilize the required pre-distortion transmission technique**
- **Attachment Unit Interface (AUI) port allows connectivity with 10BASE5 (Ethernet) and 10BASE2 (Cheapernet) networks, as well as 10BASE-F and/or Fiber Optic Inter-repeater Link (FOIRL) segments**
- **On board PLL, Manchester encoder/decoder, and FIFO**
- **Expandable to increase number of repeater ports**
- **All ports can be separately isolated (partitioned) in response to excessive collision conditions or fault conditions**
- **Network management and optional features are accessible through a dedicated serial management port**
- **Twisted Pair Link Test capability conforming to the 10BASE-T standard. The receive Link Test Function can be optionally disabled through the management port to facilitate interoperability with devices that do not implement the Link Test Function**
- **Programmable option of Automatic Polarity Detection and Correction permits automatic recovery due to wiring errors**
- **Full amplitude and timing regeneration for retransmitted waveforms**
- **Preamble loss effects eliminated by deep FIFO**

GENERAL DESCRIPTION

The Integrated Multiport Repeater Plus (IMR+) chip is a VLSI circuit that provides a system level solution to designing a compliant 802.3 repeater incorporating 10BASE-T transceivers. The device integrates the Repeater functions specified by section 9 of the IEEE 802.3 Standard and Twisted Pair Transceiver functions complying to the 10BASE-T Standard. The Am79C981 provides eight integral Twisted Pair Medium Attachment Units (MAUs) and an Attachment Unit Interface (AUI) port in an 84-pin Plastic Leaded Chip Carrier (PLCC).

A network based on the 10BASE-T standard uses unshielded twisted pair cables, therefore providing an economical solution to networking by allowing the use of low cost unshielded twisted pair (UTP) cable or existing telephone wiring.

The total number of ports per repeater unit can be increased by connecting multiple IMR+ devices through their expansion ports, hence minimizing the total cost per repeater port. Furthermore, a general purpose Attachment Unit Interface (AUI) provides connection capability to 10BASE5 (Ethernet) and 10BASE2 (Cheapernet) coaxial networks, as well as 10BASE-F and/or Fiber Optic Inter-Repeater Link (FOIRL) fiber segments. Network management and test functions are provided through TTL compatible I/O pins.

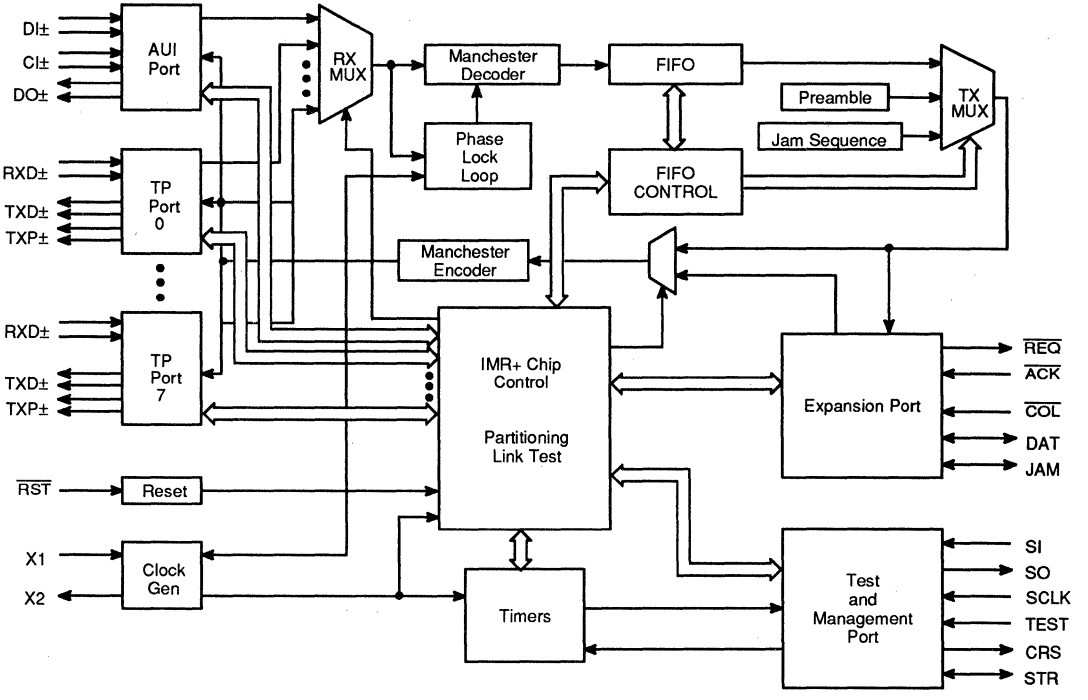
The IMR+ device interfaces directly with the AMD's Am79C987 Hardware Implemented Management Information Base™ (HIMIB) chip to build a fully managed multiport repeater as specified by the IEEE 802.3 (Layer Management for 10 Mb/s Baseband Repeaters) Standard. When the IMR+ and HIMIB devices are interconnected, complete repeater and per port statistics are maintained, and can be accessed on demand using a simple 8-bit parallel interface.

Publication# 17306 Rev. B Amendment/0
Issue Date: April 1994

For application examples on building a fully managed repeater using the IMR+ and HIMIB devices, refer to AMD's IEEE 802.3 Repeater Technical Manual (PID#17314A) and the ISA-HUB™ User Manual (PID # 17642A).

The device is fabricated in CMOS technology and requires a single +5 V supply.

BLOCK DIAGRAM



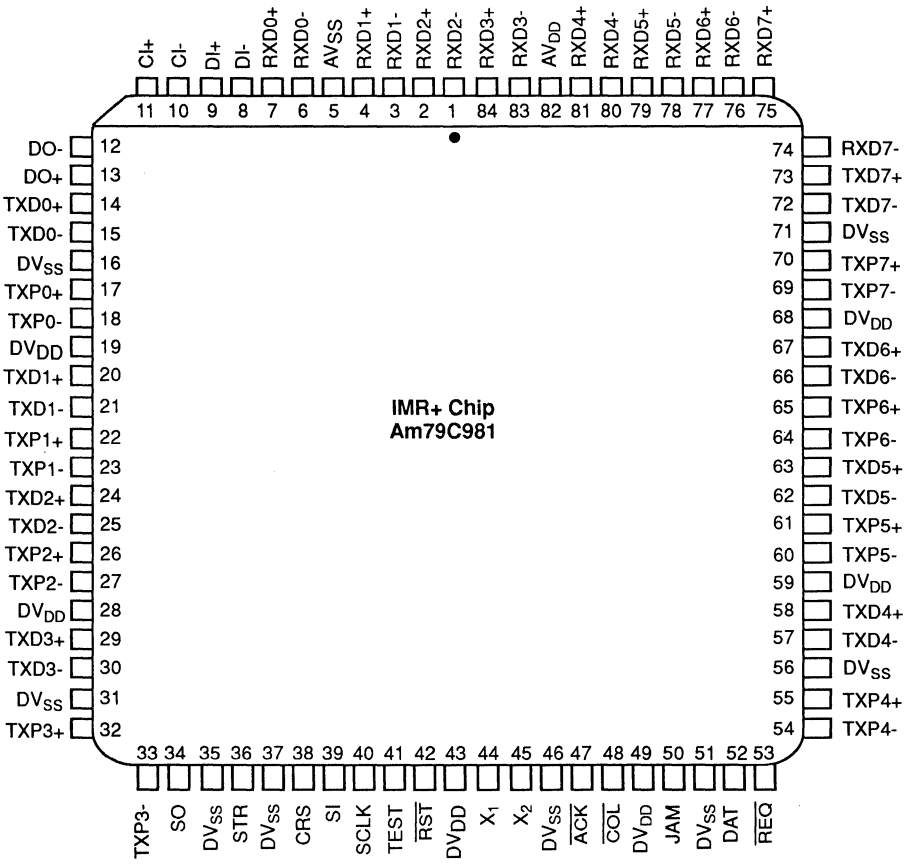
17306B-1

RELATED AMD PRODUCTS

Part No.	Description
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted Pair Ethernet Transceiver Plus (TPEX+)
Am7996	IEEE 802.3/Ethernet/Cheapernet Transceiver
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA+ Single-Chip Ethernet Controller for ISA (with Microsoft® Plug n' Play® Support)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems

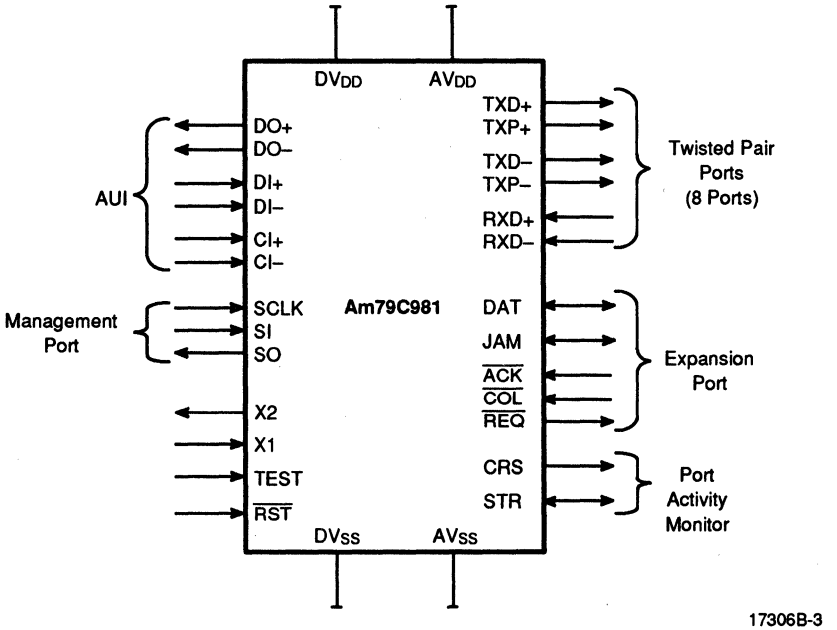
CONNECTION DIAGRAM

PLCC



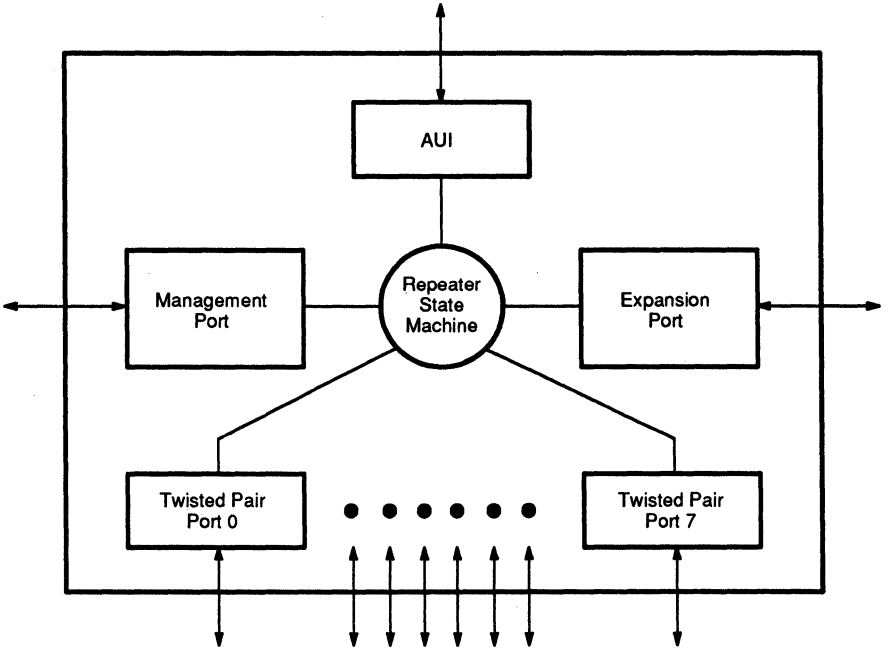
17306B-2

LOGIC SYMBOL



17306B-3

LOGIC DIAGRAM

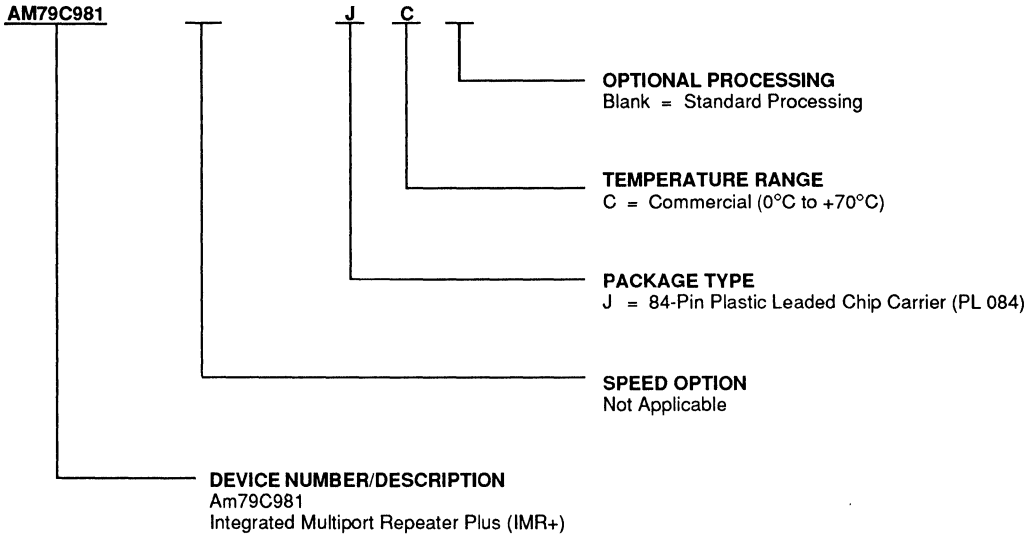


17306B-4

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM79C981	JC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

$\overline{ACK}$
Acknowledge

Input, Active LOW

When this input is asserted, it signals to the requesting IMR+ device that it may control the DAT and JAM pins. If the IMR+ chip is not requesting control of the DAT line ($\overline{REQ}$ pin HIGH), then the assertion of the $\overline{ACK}$ signal indicates the presence of valid collision status on the JAM or valid data on the DAT line.

AV_{DD}
Analog Power

Power Pin

These pins supply the +5 V to the RXD+/- receivers, the DI+/- and CI+/- receivers, the DO+/- drivers, the internal PLL, and the internal voltage reference of the IMR+ device. These power pins should be decoupled and kept separate from other power and ground planes.

AV_{SS}
Analog Ground

Ground Pin

These pins are the 0 V reference for AV_{DD}.

$\overline{COL}$
Expansion Collision

Input, Active LOW

When this input is asserted by an external arbiter, it signifies that more than one IMR+ device is active and that each IMR+ device should generate the Collision Jam Sequence independently.

CI+, CI-
Control In

Input

AUI port differential receiver. Signals comply with IEEE 802.3, Section 7.

CRS
Carrier Sense

Output

The states of the internal carrier sense signals for the AUI port and the eight twisted pair ports are serially output on this pin continuously. The output serial bit stream is synchronized to the X_1 clock.

DAT
Data

Input/Output/3-State

In non-collision conditions, the active IMR+ device will drive DAT with NRZ data, including regenerated preamble. During collision, when JAM = HIGH, DAT is used to signal a multiport (DAT = 0) or single port (DAT = 1) condition.

When $\overline{ACK}$ is not asserted, DAT is in high impedance. If $\overline{REQ}$ and $\overline{ACK}$ are both asserted, then DAT is an output. If $\overline{ACK}$ is asserted and $\overline{REQ}$ not asserted, then DAT is an input.

This pin needs to be either pulled up or pulled down through a high value resistor.

DI+, DI-
Data In

Input

AUI port differential receiver. Signals comply with IEEE 802.3, Section 7.

DO+, DO-
Data Out

Output

AUI port differential driver. Signals comply with IEEE 802.3, Section 7.

DV_{DD}
Digital Power

Power Pin

These pins supply the +5 V to the logic portions of the IMR+ chip and the TXP+/-, TXD+/-, and DO+/- line drivers.

DV_{SS}
Digital Ground

Ground Pin

These pins are the 0 V reference for DV_{DD}.

DV_{DD} Pin #	DVss Pin #	Function
19	16	TP ports 0 & 1 drivers
28	31	TP ports 2 & 3 drivers
43, 49	35, 37, 46, 51	Core logic and expansion and control pins
59	56	TP ports 4 & 5 drivers
68	71	TP ports 6 & 7 drivers

JAM
Jam

Input/Output/3-State

When JAM is asserted, the state of DAT will indicate either a multiport (DAT = 0) or single port (DAT = 1) collision condition.

When $\overline{ACK}$ is not asserted, JAM is in high impedance. If $\overline{REQ}$ and $\overline{ACK}$ are both asserted, then JAM is an output. If $\overline{ACK}$ is asserted and $\overline{REQ}$ not asserted, then JAM is an input.

This pin needs to be either pulled up or pulled down through a high value resistor.

This pin needs to be either pulled up or pulled down through a high value resistor.

$\overline{REQ}$
Request

Output, Active LOW

This pin is driven LOW when the IMR+ chip is active. An IMR+ chip is active when it has one or more ports receiving or colliding or is in the state where it is still transmitting data from the internal FIFO. The assertion of this signal signifies that the IMR+ device is requesting the use of the DAT and JAM lines for the transfer of repeated data or collision status to other IMR+ devices.

$\overline{RST}$
Reset

Input, Active LOW

Driving this pin LOW resets the internal logic of the IMR+ device. Reset should be synchronized to the X_1 clock if either expansion or port activity monitor is used.

$RXD+_{0-7}$, $RXD-_{0-7}$
Receive Data

Input

10BASE-T port differential receive inputs (8 ports).

SCLK
Serial Clock

Input

In normal operating mode, serial data (input or output) is clocked (in or out) on the rising edge of the signal on this pin. SCLK is asynchronous to X1 and can operate up to 10 MHz. In Minimum Mode, this pin, together with the SI pin, controls which information is output on the SO pin.

SI
Serial In

Input

In normal operating mode, the SI pin is used for test/management serial input port. Management commands are clocked in on this pin synchronous to the SCLK input. In Minimum Mode, this pin, together with the SCLK pin, controls which information is output on the SO pin.

In Minimum Mode, the state of SI at the deassertion of $\overline{RST}$ signal determines the programming of automatic polarity detection/correction for 10BASE-T ports.

SO
Serial Out

Output

In normal operating mode, the SO pin is used for test/management serial output port. Management results are clocked out on this pin synchronous to the SCLK input. In Minimum Mode, the SO pin is used to serially output the various status information based on the state of the SI and SCLK pins.

SCLK	SI	SO Output
0	0	TP Ports Receive Polarity Status + AUI SQE Test Error Status.
0	1	Bit Rate Error (all ports).
1	0	TP Ports Link Status + AUI LoopBack Status.
1	1	Port Partitioning Status (all ports).

STR
Store

Input/Output

As an output, this pin goes HIGH for two X_1 clock cycle times after the nine carrier sense bits are output on the CRS pin. Note that the carrier sense signals arriving from each port are latched internally, so that an active transition is remembered between samples. The accuracy of the carrier sense signals produced in this manner is 10 bit times (1 µs).

When used in conjunction with the HIMIB device, the STR pin will be configured as an input automatically after a hardware reset. The HIMIB device uses this input to communicate with the IMR+ device. When used with the HIMIB chip, this pin must be pulled up via a high value resistor.

TEST
Test Pin

Input, Active HIGH

This pin should be tied LOW for normal operation. If this pin is driven HIGH, then the IMR+ device can be programmed for Loopback Test Mode. Also, if this pin is HIGH when the $\overline{RST}$ pin is deasserted, the IMR+ device will enter the Minimum Mode. An inverted version of the $\overline{RST}$ signal can be used to program the device into the Minimum Mode.

Test	SI	Functions
0	0	Normal Management Mode
0	1	Normal Management Mode
1	0	Minimum Mode, Receive Polarity Correction disabled
1	1	Minimum Mode, Receive Polarity Correction enabled

$TXD+_{0-7}$, $TXD-_{0-7}$
Transmit Data

Output

10BASE-T port differential drivers (8 ports).

$TXP+_{0-7}$, $TXP-_{0-7}$
Transmit Pre-distortion

Output

10BASE-T transmit waveform pre-distortion control differential outputs (8 ports).

X_1
Crystal 1

Crystal Connection

The internal clock generator uses a 20 MHz crystal attached to pins X_1 and X_2. Alternatively, an external 20 MHz CMOS clock signal can be used to drive this pin.

X_2
Crystal 2

Crystal Connection

The internal clock generator uses a 20 MHz crystal attached to pins X_1 and X_2. If an external clock source is used, this pin should be left unconnected.

FUNCTIONAL DESCRIPTION

The Am79C981 Integrated Multiport Repeater Plus device is a single chip implementation of an IEEE 802.3/Ethernet repeater (or hub). In addition to the eight integral 10BASE-T ports plus one AUI port comprising the basic repeater, the IMR+ chip also provides the hooks necessary for complex network management and diagnostics. The IMR+ device is also expandable, enabling the implementation of high port count repeaters based on several IMR+ devices.

The IMR+ device interfaces directly with AMD's Am79C987 Hardware Implemented Management Information Base (HIMIB) device to allow a fully managed multiport repeater to be implemented as specified by the Layer Management for 10 Mb/s Baseband Repeaters Standard. When the IMR+ and HIMIB devices are used as a chip set, the HIMIB device maintains complete repeater and per port statistics which can be accessed on demand by a microprocessor through a simple 8-bit parallel port.

The IMR+ chip complies with the full set of repeater basic functions as defined in section 9 of ISO 8802.3 (ANSI/IEEE 802.3c). These functions are summarized below.

Repeater Function

If any single network port senses the start of a valid packet on its receive lines, then the IMR+ device will retransmit the received data to all other enabled network ports. The repeated data will also be presented on the DAT line to facilitate multiple-IMR+ device repeater applications.

Signal Regeneration

When re-transmitting a packet, the IMR+ device ensures that the outgoing packet complies with the 802.3 specification in terms of preamble structure, voltage amplitude, and timing characteristics. Specifically, data packets repeated by the IMR+ chip will contain a minimum of 56 preamble bits before the Start of Frame Delimiter. In addition, the voltage amplitude of the repeated packet waveform will be restored to levels specified in the 802.3 specification. Finally, signal symmetry is restored to data packets repeated by the IMR+ device, removing jitter and distortion caused by the network cabling.

Jabber Lockup Protection

The IMR+ chip implements a built-in jabber protection scheme to ensure that the network is not disabled due to transmission of excessively long data packets. This protection scheme will automatically interrupt the transmitter circuits of the IMR+ device for 96-bit times if the IMR+ device has been transmitting continuously for more than 65,536-bit times. This is referred to as MAU Jabber Lockup Protection (MJLP). The MJLP status for the IMR+ chip can be read through the Management Port using the Get MJLP Status command (M bit returned).

Collision Handling

The IMR+ chip will detect and respond to collision conditions as specified in 802.3. A multiple-IMR+ device repeater implementation also complies with the 802.3 specification due to the inter-IMR+ chip status communication provided by the expansion port. Specifically, a repeater based on one or more IMR+ devices will handle the transmit collision and one-port-left collision conditions correctly as specified in Section 9 of the 802.3 specification.

Fragment Extension

If the total packet length received by the IMR+ device is less than 96 bits, including preamble, the IMR+ chip will extend the repeated packet length to 96 bits by appending a Jam sequence to the original fragment.

Auto Partitioning/Reconnection

Any of the integral TP ports and AUI port can be partitioned under excessive duration or frequency of collision conditions. Once partitioned, the IMR+ device will continue to transmit data packets to a partitioned port, but will not respond (as a repeater) to activity on the partitioned port's receiver. The IMR+ chip will monitor the port and reconnect it once certain criteria indicating port 'wellness' are met. The criteria for reconnection are specified by the 802.3 standard. In addition to the standard reconnection algorithm, the IMR+ device implements an alternative reconnection algorithm which provides a more robust partitioning function for the TP ports and/or the AUI port. Each TP port and the AUI port are partitioned and/or reconnected separately and independently of other network ports.

Either one of the following conditions occuring on any enabled IMR+ device network port will cause the port to partition:

a. A collision condition exists continuously for a time between 1024- to 2048-bit times (AUI port—SQE signal active; TP port—simultaneous transmit and receive)

b. A collision condition occurs during each of 32 consecutive attempts to transmit to that port.

Once a network port is partitioned, the IMR+ device will reconnect that port if the following is met:

a. Standard reconnection algorithm—A data packet longer than 512-bit times (nominal) is transmitted or received by the partitioned port without a collision.

b. Alternate reconnection algorithm—A data packet longer than 512-bit times (nominal) is transmitted by the partitioned port without a collision.

The reconnection algorithm option (standard or alternate) is a global function for the TP ports, i.e. all TP ports use the same reconnection algorithm. The AUI reconnection algorithm option is programmed independently of the TP port reconnection option.

Link Test

The integral TP ports implement the Link Test function as specified in the 802.3 10BASE-T standard. The IMR+ device will transmit Link Test pulses to any TP port after that port's transmitter has been inactive for more than 8 to 17 ms. Conversely, if a TP port does not receive any data packets or Link Test pulses for more than 65 to 132 ms and the Link Test function is enabled for that port then that port will enter link fail state. A port in link fail state will be disabled by the IMR+ chip (repeater transmit and receive functions disabled) until it receives either four consecutive Link Test pulses or a data packet. The Link Test receive function itself can be disabled via the IMR+ chip management port on a port-by-port basis to allow the IMR+ device to interoperate with pre-10BASE-T twisted pair networks that do not implement the Link Test function. This interoperability is possible because the IMR+ device will not allow the TP port to enter link fail state, even if no Link Test pulses or data packets are being received. Note however that the IMR+ chip will always transmit Link Test pulses to all TP ports regardless of whether or not the port is enabled, partitioned, in link fail state, or has its Link Test receive function disabled.

Polarity Reversal

The TP ports have the optional (programmable) ability to invert (correct) the polarity of the received data if the TP port senses that the received data packet waveform polarity is reversed due to a wiring error. This receive circuitry polarity correction allows subsequent packets to be repeated with correct polarity. This function is executed once following reset or link fail, and has a programmable enable/disable option on a port-by-port basis. This function is disabled upon reset and can be enabled via the IMR+ chip Management Port.

Reset

The IMR+ device enters reset state when the $\overline{RST}$ pin is driven LOW. After the initial application of power, the $\overline{RST}$ pin must be held LOW for a minimum of 150 µs (3000 X1 clock cycles). If the $\overline{RST}$ pin is subsequently asserted while power is maintained to the IMR+ device, a reset duration of only 4 µs is required. The IMR+ chip continues to be in the reset state for 10 X1 clocks (0.5 µs) following the rising edge of $\overline{RST}$. During reset, the output signals are placed in their inactive states. This means that all analog signals are placed in their idle states, bidirectional signals (except STR signal) are not driven, active LOW signals are driven HIGH, and all active HIGH signals and the STR pin are driven LOW.

An internal circuit ensures that a minimum reset pulse is generated for all internal circuits. For a $\overline{RST}$ input with a slow rising edge, the input buffer threshold may be crossed several times due to ripple on the input waveform.

In a multiple IMR+ chip repeater the $\overline{RST}$ signal should be applied simultaneously to all IMR+ devices and should be synchronized to the external X1 clock. Reset synchronization is also required when accessing the PAM (Port Activity Monitor).

The SI signal should be held HIGH for at least 500 ns following the rising edge of $\overline{RST}$.

Table 1 summarizes the state of the IMR+ chip following reset.

Table 1. IMR+ Chip After Reset

Function	State After Reset	Pull Up/Pull Down
Active LOW outputs	HIGH	No
Active HIGH outputs	LOW	No
SO Output	HIGH	No
DAT, JAM	HI-IMPEDANCE	Either
STR	LOW	Pull-Up*
Transmitters (TP and AUI)	IDLE	No
Receivers (TP and AUI)	ENABLED	Terminated
AUI Partitioning/Reconnection Algorithm	STANDARD ALGORITHM	N/A
TP Port Partitioning/Reconnection Algorithm	STANDARD ALGORITHM	N/A
Link Test Function for TP Ports	ENABLED, TP PORTS IN LINK FAIL	N/A
Automatic Receiver Polarity Reversal Function	DISABLED	N/A

**Only when used with the HIMIB device.*

Expansion Port

The IMR+ chip Expansion Port is comprised of five pins; two are bi-directional signals (DAT and JAM), two are input signals ($\overline{ACK}$ and $\overline{COL}$), and one is an output signal ($\overline{REQ}$). These signals are used when a multiple-IMR+ device repeater application is employed. In this configuration, all IMR+ chips must be clocked synchronously with a common clock connected to the X1 inputs of all IMR+ devices. Reset needs to be synchronized to X1 clock.

The IMR+ device expansion scheme allows the use of multiple IMR+ chips in a single board repeater or a modular multiport repeater with a backplane architecture. The DAT pin is a bidirectional I/O pin which can be used to transfer data between the IMR+ devices in a multiple-IMR+ chip design. The data sent over the DAT line is in NRZ format and is synchronized to the common clock. The JAM pin is another bidirectional I/O pin that is used by the active IMR+ chip to communicate its internal status to the remaining (inactive) IMR+ devices. When JAM is asserted HIGH, it indicates that the active IMR+ device has detected a collision condition and is generating Jam Sequence. During this time when JAM is asserted HIGH, the DAT line is used to indicate whether the active IMR+ chip is detecting collision on one port only or on more than one port. When DAT is driven HIGH by the IMR+ chip (while JAM is asserted by the IMR+ chip), then the active IMR+ device is detecting a collision condition on one port only. This 'one-port-left' signaling is necessary for a multiple-IMR+ device repeater to function correctly as a single multiport repeater unit. The IMR+ chip also signals the 'one port left' collision condition in the event of a runt packet or collision fragment; this signal will continue for one expansion port bus cycle (100 ns) before deasserting $\overline{REQ}$.

The arbitration for access to the bussed bi-directional signals (DAT and JAM) is provided by one output ($\overline{REQ}$) and two inputs ($\overline{ACK}$ and $\overline{COL}$). The IMR+ chip asserts the $\overline{REQ}$ pin to indicate that it is active and wishes to drive the DAT and JAM pins. An external arbiter senses the $\overline{REQ}$ lines from all the IMR+ devices and asserts the $\overline{ACK}$ line when one and only one IMR+ chip is asserting its $\overline{REQ}$ line. If more than one IMR+ chip is asserting its $\overline{REQ}$ line, the arbiter must assert the $\overline{COL}$ signal, indicating that more than one IMR+ device is active. More than one active IMR+ device at a time constitutes a collision condition, and all IMR+ devices are notified of this occurence via the $\overline{COL}$ line of the Expansion Port.

Note that a transition from multiple IMR+ devices arbitrating for the DAT and JAM pins (with $\overline{COL}$ asserted, $\overline{ACK}$ deasserted) to a condition when only one IMR+ chip is arbitrating for the DAT and JAM pins (with $\overline{ACK}$ asserted, $\overline{COL}$ deasserted) involves one expansion port bus cycle (100 ns). During this transitional bus cycle, $\overline{COL}$ is deasserted, $\overline{ACK}$ is asserted, and the DAT and JAM pins are not driven. However, each IMR+ device will remain in the collision state (transmitting jam sequence) during this transitional bus cycle. In subsequent expansion port bus cycles ($\overline{REQ}$ and $\overline{ACK}$ still asserted), the IMR+ devices will return to the 'master and slaves' condition where only one IMR+ device is active (with collision) and is driving the DAT and JAM pins. An understanding of this sequence is crucial if non-IMR+ devices (such as an Ethernet controller) are connected to the expansion bus. Specifically, the last device to back off of the Expansion Port after a multi-IMR+ chip collision must assert the JAM line until it too drops its request for the Expansion Port.

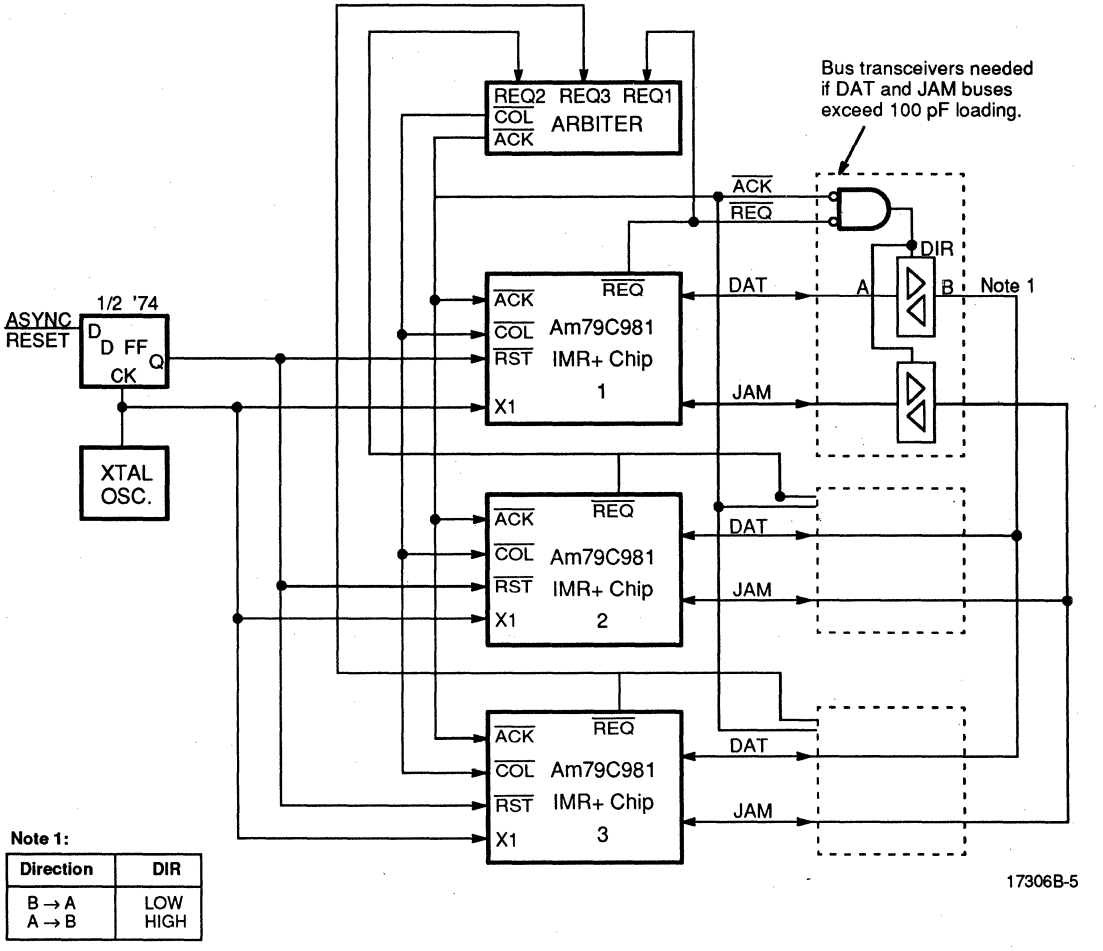
External Arbiter

A simple arbitration scheme is required when multiple IMR+ devices are connected together to increase the total number of repeater ports. The arbiter should have one input ($\overline{REQ1}$...$\overline{REQn}$) for each of the n IMR+ devices to be used, and two global outputs ($\overline{COL}$ and $\overline{ACK}$). This function is easily implemented in a PAL® device, with the following logic equations:

$$\begin{aligned}
\overline{ACK} &= \overline{REQ1} \,\&\, \overline{REQ2} \,\&\, \overline{REQ3} \,\&\, ... \overline{REQn} \\
&+ \overline{REQ1} \,\&\, REQ2 \,\&\, \overline{REQ3} \,\&\, ... \overline{REQn} \\
&\quad \bullet \\
&\quad \bullet \\
&\quad \bullet \\
&+ \overline{REQ1} \,\&\, \overline{REQ2} \,\&\, \overline{REQ3} \,\&\, ... \; REQn \\
\overline{COL} &= \overline{ACK} \,\&\, (REQ1 + REQ2 + REQ3 + ... \; REQn)
\end{aligned}$$

Above equations are in positive logic, i.e., a variable is true when asserted.

A single PALCE16V8 will perform the arbitration function for a repeater based on several IMR+ devices.



Note 1:

Direction	DIR
B → A	LOW
A → B	HIGH

Figure 1. Multiple IMR+ Devices

Modular Repeater Design

The expansion port of the IMR+ chip also allows for modular expansion. By sharing the arbitration duties between a backplane bus architecture and several separate repeater modules one can build an expandable repeater based on modular 'plug-in' cards. Each repeater module performs the local arbitration function for the IMR+ devices on that module, and provides signals to the backplane for use by a global arbiter.

For more detailed information, see AMD's IEEE 802.3 Repeater Technical Manual, PID# 17314A.

Repeater MAC Interconnection

Because all repeated data in the IMR+ chip or multi-IMR+ chip design is available on the Expansion Port, all network traffic can be monitored by an external Media Access Controller (MAC) device such as the Am7990, Am79C900, Am79C940, or Am79C960. A repeater with such a controller is capable of providing extensive hub management functions, as well as being addressable as a network node. The MAC device can gather statistics and data concerning the state of the hub and the network, and the network addressability allows a remote management station to monitor this statistical data and to request actions to be performed by the repeater (i.e. port enable/disable).

Figure 2 shows how to interface a repeater based on multiple IMR+ devices to an Ethernet controller such as the Am79C900 ILACC or the Am7990 LANCE. For more information on this design, refer to AMD's IEEE 802.3 Repeater Technical Manual, PID# 17314A.

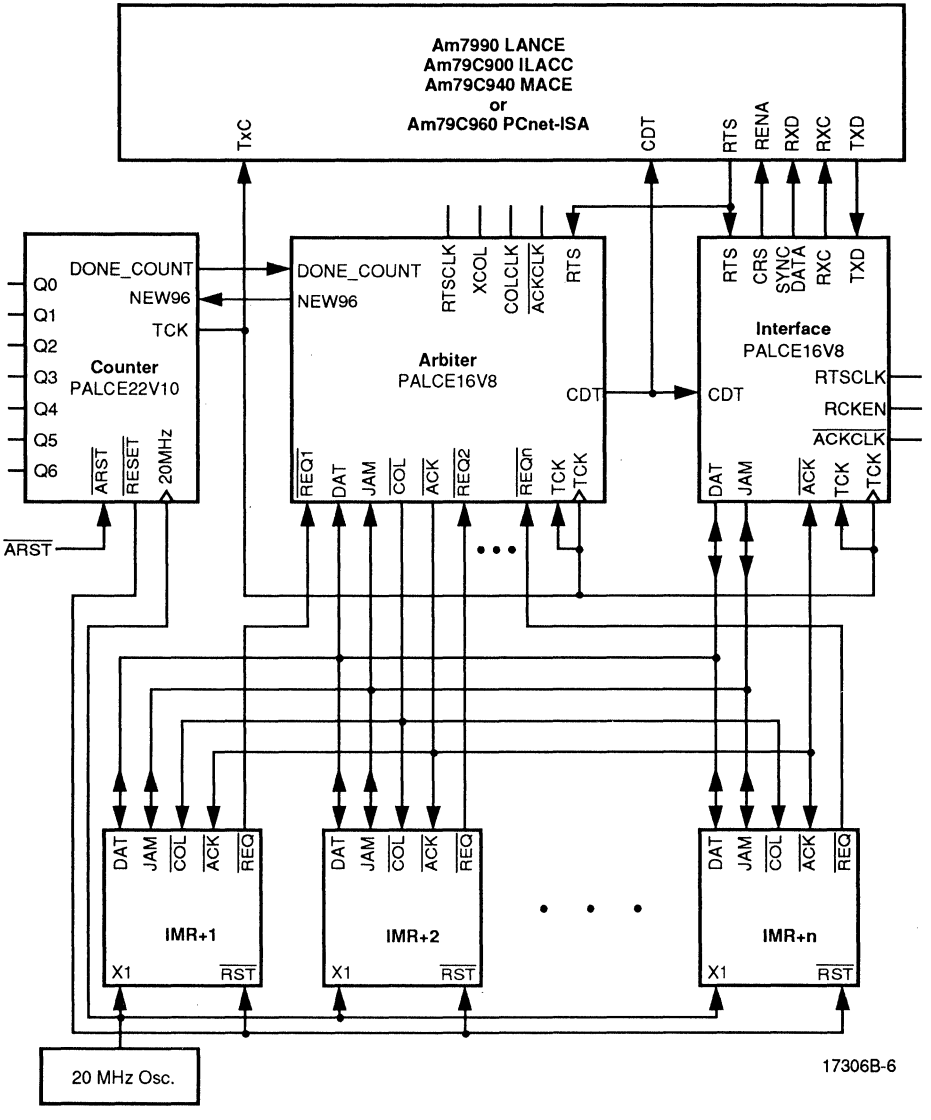


Figure 2. Expandable Modular Repeater

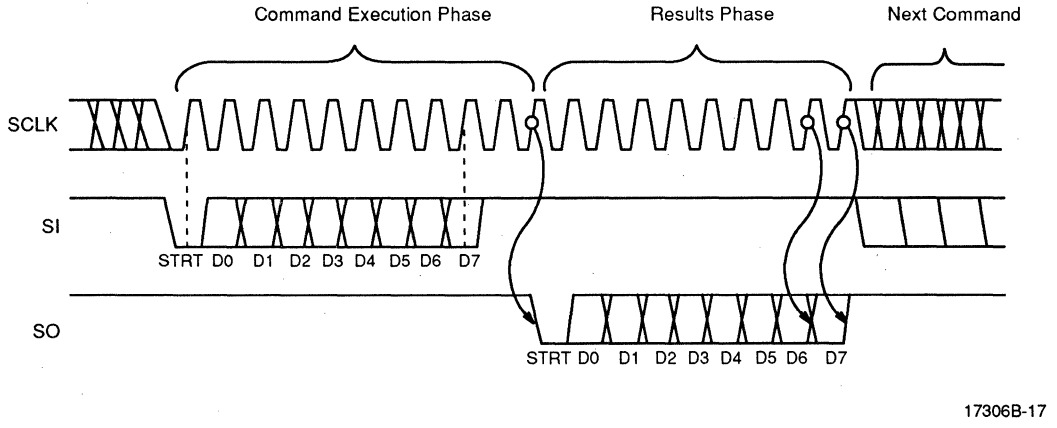
Management Port

The IMR+ device management functions are enabled when the TEST pin is tied LOW. The management commands are byte oriented data and are input serially on the SI pin. Any responses generated during execution of a management command are output serially in a byte-oriented format by the IMR+ device on the SO pin. Both the input and output data streams are clocked with the rising edge of the SCLK pin. The serial command data stream and any associated results data stream are structured in a manner similar to the RS232 serial data format, i.e., one Start Bit followed by eight Data Bits.

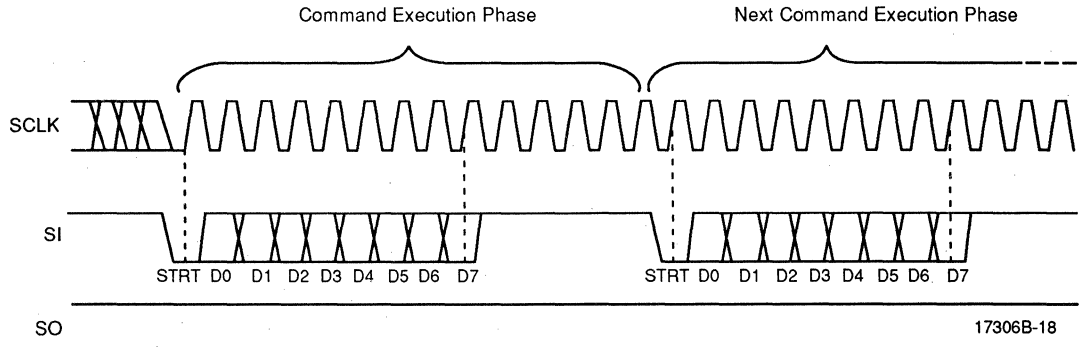
The externally generated clock at the SCLK pin can be either a free running clock synchronized to the input bit patterns or a series of individual transitions meeting the setup and hold times with respect to the input bit pattern. If the latter method is used, it is to be noted that 20 SCLK clock transitions are required for proper execution of management commands that produce SO data, and that 14 SCLK clock transitions are needed to execute management commands that do not produce SO data.

Management Commands

The following section details the operation of each management command available in the IMR+ chip. In all cases, the individual bits in each command byte are shown with the MSB on the left and the LSB on the right. Data bytes are received and transmitted LSB first and MSB last. See Table 2 for a summary of the management commands.



Management Command/Response Timing



Management Command Timing with No Response

Table 2. Management Port Command Summary

Commands	SI Data	SO Data
Set (Write) Opcodes		
IMR+ Chip Programmable Options	0000 1CSA	
Alternate AUI Partitioning Algorithm	0001 1111	
Alternate TP Partitioning Algorithm	0001 0000	
AUI Port Disable	0010 1111	
AUI Port Enable	0011 1111	
TP Port Disable	0010 0###	
TP Port Enable	0011 0###	
Disable Link Test Function (per TP port)	0100 0###	
Enable Link Test Function (per TP port)	0101 0###	
Disable Automatic Receiver Polarity Reversal (per TP port)	0110 0###	
Enable Automatic Receiver Polarity Reversal (per TP port)	0111 0###	
Get (Read) Opcodes		
AUI Port Status (B, S, L Cleared)	1000 1111	PBSL 0000
TP Port Partitioning Status	1000 0000	C7...C0
Bit Rate Status of TP ports	1010 0000	E7...E0
Link Test Status of TP ports	1101 0000	L7...L0
Receive Polarity Status of all TP ports	1110 0000	P7...P0
MJLP Status	1111 0000	M000 0000
Version	1111 1111	XXXX 0001
AUI Port Status (S, L Cleared)	1000 1011	PBSL 0000
AUI Port Status (B Cleared)	1000 1101	PBSL 0000
AUI Port Status (None Cleared)	1000 1001	PBSL 0000

Notes:

1. *Unused opcodes are reserved for future use.*
2. *### is the port number (000 to 111 for TP0 to TP7)*

SET (Write) Opcodes

IMR+ Chip Programmable Options

SI data: 0000 1CSA

SO data: None

IMR+ Chip Programmable Options can be enabled (disabled) by setting (resetting) the appropriate bit in the command string. The three programmable bits are: **C**—CI Reporting; **S**—AUI SQE Test Mask, and **A**—Alternative Port Activity Monitor (PAM) Function. These options can be enabled (disabled) by setting (resetting) the appropriate bit in the command string.

When writing to this register through the Am79C987 HIMIB device, the A and C bits should not be changed (A=0, C=1).

C—CI Reporting

Setting this bit alters the function of the STR pin. In this mode, the STR pin becomes an input in response to the AMD's Am79C987 HIMIB device. Upon deassertion of $\overline{RST}$, the HIMIB automatically sets this bit following IMR/IMR+ device type detection.

When this mode is selected, the CRS output bit string format is modified to include CI carrier bit (in addition to AUI carrier). This bit occupies the bit position immediately preceding the AUI bit in the CRS bit string (10 bits) output. Note that the AUI bit gets asserted if either the CI or DI signal pairs are active.

S—AUI SQE Test Mask

Setting this bit allows the IMR+ chip to ignore activity on the CI signal pair, in the SQE Test Window, following a transmission on the AUI port. This event occurs when the attached MAU has the SQE Test option enabled, therefore generating a burst of CI activity following every transmission. This is interpreted by the IMR+ device as a collision, causing the IMR+ device to generate a full Jam pattern. Although the MAU attached to a repeater is required not to have its SQE test function active, this is a common installation error, causing difficulty in diagnosing network throughput problems.

The SQE Test Window, as defined by the IEEE 802.3 (Section 7.2.2.2.4), is from 6-bit times to 34-bit times (0.6 µs to 3.4 µs). This includes delay introduced by a 50 m AUI. CI activity that occurs outside this window is not ignored and is treated as true collision.

Note that enabling this function does not prevent the reporting of this condition by the IMR+ device and the two functions operate independently.

A—Alternative Port Activity Monitor (PAM) Function

Setting the Alternative Port Activity Monitor Function allows the PAM function to be altered such that the Carrier Sense data is presented unmodified. In default operation the PAM output (Carrier Sense bits in the CRS bit stream) are masked if the port is either disabled or partitioned. This does not allow the Repeater Management software to sense activity on all segments at all times. The ability to monitor partitioned or disabled ports allows fault tolerance to be built into the Repeater Management software.

Alternate AUI Port Partitioning Algorithm

SI data: 00011111
SO data: None

The AUI port Partitioning/Reconnection scheme can be programmed for the alternate (transmit only) reconnection algorithm by invoking this command. To return the AUI back to the standard (transmit or receive) reconnection algorithm, it is necessary to reset the IMR+ device. Standard partitioning algorithm is selected upon reset.

Alternate TP Ports Partitioning Algorithm

SI data: 00010000
SO data: None

The TP ports Partitioning/Reconnection scheme can be programmed for the alternate (transmit only) reconnection algorithm by invoking this command. All TP ports are affected as a group by this command. To return the TP ports back to the standard (transmit or receive) reconnection algorithm, it is necessary to reset the IMR+ device. The standard partitioning algorithm is selected upon reset.

AUI Port Disable

SI data: 00101111
SO data: None

The AUI port will be disabled upon receiving this command. Subsequently, the IMR+ chip will ignore all inputs (Carrier Sense and SQE) appearing at the AUI port and will not transmit any data or Jam Sequence on the AUI port. Issuing this command will also cause the AUI port to have its internal partitioning state machine forced to its idle state. Therefore, a Partitioned Port may be reconnected by first disabling and then re-enabling the port.

AUI Port Enable

SI data: 00111111
SO data: None

This command enables a previously disabled AUI port. Note that a partitioned AUI port may be reconnected by first disabling (AUI Port Disable Command) and then re-enabling the port with this command.

All ports are enabled upon reset.

TP Port Disable

SI data: 00100###
SO data: None
(### is TP port number)

The TP port designated in the command byte will be disabled upon receiving this command. Subsequently, the IMR+ device will ignore all inputs appearing at the disabled port's receive pins and will not transmit any data or JAM Sequence on that port's transmit pins. Issuing this command will also cause a TP port to have its partitioning state machine returned to its Idle State (Port Reconnected). Therefore, a partitioned port may be reconnected by first disabling and then re-enabling the port. The disabled port will continue to report correct Link Test Status.

TP Port Enable

SI data: 00110###
SO data: None
(### is TP port number)

This command enables a previously disabled TP port. Re-enabling a disabled port causes the port to be placed into Link Test Fail state. This ensures that packet fragments received on the port are not repeated to the rest of the network. Note that to force a TP port into the Link Fail state and/or to reconnect a partitioned TP port, the port should first be disabled (TP Port Disable Command) and then re-enabled with this command. All ports are enabled upon reset.

Disable Link Test Function of a TP Port

SI data: 01000###
SO data: None
(### is TP port number)

This command disables the Link Test Function at the TP port designated in the command byte, i.e., the TP port will no longer be disconnected due to Link Fail. A TP port which has its Link Test Function disabled will continue to transmit Link Test Pulses. If a twisted pair port has Link Test disabled, then reading the Link Test Status indicates it being in Link Test Pass.

Enable Link Test Function of a TP Port

SI data: 01010###
SO data: None
(### is TP port number)

This command re-enables the Link Test Function in the TP port designated in the command byte. This command executes only if the designated TP port has had the Link Test Function disabled by the Disable Link Test Function command. Otherwise, the command is ignored. Link Test is enabled upon reset.

Disable Automatic Receiver Polarity Reversal

SI data: 01100###
SO data: None
(### is TP port number)

This command disables the Automatic Receiver Polarity Reversal Function for the TP port designated in the command byte. If this function is disabled on a TP port with reverse polarity (due to a wiring error), then the TP port will fail Link Test due to the reversed polarity of the Link Pulses. If the Link Test Function is also disabled on the TP port, then the received reversed polarity packets would be repeated to all other network ports in the IMR+ chip as inverted data. Automatic Polarity reversal is disabled upon reset.

Enable Automatic Receiver Polarity Reversal

SI data: 01110###
SO data: None
(### is TP port number)

This command enables the Automatic Receiver Polarity Reversal Function for the TP port designated in the command byte. If enabled in a TP port, the IMR+ chip will automatically invert the polarity of that TP port's receiver circuitry if the TP port is detected as having reversed polarity (due to a wiring error). After reversing the receiver polarity, the TP port could then receive subsequent (reverse polarity) packets correctly.

GET (Read) Opcodes

AUI Port Status

SI data: 10001111
SO data: PBSL0000

The combined AUI status allows a single instruction to be used for monitoring AUI port. The four status bits reported are:

P Partitioning Status. This bit is 0 if the AUI port is partitioned and 1 if connected.

B Bit Rate Error. This bit is set to 1 if there has been an instance of FIFO Overflow or Underflow, caused by data received at the AUI port. This bit is cleared when the status is read.

S SQE Test Status. This bit is set to 1 if SQE Test is detected by the IMR+ chip. This bit is cleared when the status is read. A MAU attached to a repeater must have SQE Test disabled. This bit is set even if the AUI port is disabled or partitioned.

L Loop Back Error. The MAU attached to the AUI is required to loopback data transmitted to DO onto the DI circuit. If loopback carrier is not detected by the IMR+ device, then this bit is set to 1 to report this condition. This bit is cleared when the status is read. For a repeater this is the only indication of a broken or missing MAU.

Alternate AUI Port Status

SI data: 10001111
SO data: PBSL0000

There are three further variations of the above command, allowing selective clearing of a combination of B, S, and L bits. They are primarily included for use by the HIMIB chip. These are:

Alternative 1.

SI data: 10001011
SO data: PBSL0000

B is not cleared. S and L are cleared.

Alternative 2.

SI data: 10001101
SO data: PBSL0000

S and L are not cleared. B is cleared.

Alternative 3.

SI data: 10001001
SO data: PBSL0000

None of S, B and L are cleared.

TP Port Partitioning Status

SI data: 10000000
SO data: P7.....P0

Pn = 0 TP port n partitioned

Pn = 1 TP port n connected

The partitioning Status of all eight TP ports are accessed by this command. If a port is disabled, reading it partitioning status will indicate that it is connected.

Bit Rate Error Status of TP Ports

SI data: 10100000
SO data: E7.....E0

This allows a single command to be used to report Bit Rate Error condition (FIFO Overflow or Underflow) of all Twisted Pair ports. The 8 bits of the output pattern correspond to each of the 8 TP ports, with least significant bit corresponding to port 0.

The status bit for a port is set to 1 if there has been an instance when data received from that port has caused a FIFO error.

All status bits stay set until the status is read.

Link Test Status of TP Ports

SI data: 11010000
SO data: L7.....L0

Ln = 0 TP Port n in Link Test Fail
Ln = 1 TP Port n in Link Test Pass

The Link Test Status of all eight TP ports are accessed by this command. A disabled port continues to report correct Link Test Status. Re-enabling a disabled port causes the port to be placed into Link Test Fail state. This ensures that packet fragments received on the port are not repeated to the rest of the network.

Receive Polarity Status of TP Ports

SI data: 11100000
SO data: P7.....P0

Pn = 0 TP Port n Polarity Correct
Pn = 1 TP Port n Polarity Reversed

The statuses of all eight TP port polarities are accessed with this command. The IMR+ chip has the ability to detect and correct reversed polarity on the TP ports' RXD+/– pins. If the polarity is detected as reversed for a TP port, then the IMR+ chip will set the appropriate bit in this command's result byte only if the Polarity Reversal Function is enabled for that port.

MJLP Status

SI data: 11110000
SO data: M00000000

Each IMR+ chip contains an independent MAU Jabber Lock Up Protection Timer. The timer is designed to inhibit the IMR+ device transmit function, if it has been transmitting continuously for more than 65536 Bit Times. The MJLP Status bit (M) is set to 1 if this happens. This bit remains set and is only cleared when the MJLP status is read by using this command.

Version

SI data: 11111111
SO data: XXXX0001

This command (1111 1111) can be used to determine the device version.

The IMR+ chip responds by the bit pattern: XXXX 0001

The IMR chip (Am79C980) responds by the bit pattern: XXXX 0000

Minimum Mode

The Minimum Mode reconfigures the IMR+ device Management Port and is intended to provide support for the low end, non-managed repeaters, requiring minimal external logic to provide LED indication of:

- Twisted Pair Ports Link Status indication and AUI Loopback Status
- Port Partitioning Status
- Twisted Pair Ports Receiver Polarity Status and AUI SQE Test Error Status
- Port Bit Rate Error Status

The Minimum Mode is selected by controlling the state of the TEST pin while $\overline{RST}$ is asserted. If TEST is High (asserted), while reset is active ($\overline{RST}$ LOW), then Minimum Mode is selected. The state of SI pin, at the deassertion of the $\overline{RST}$ signal, determines whether the IMR+ chip is to be programmed for Automatic Polarity Detection/Correction.

When entering the Minimum Mode, the TEST input has to be deasserted on the rising edge of reset. A maximum delay of 100 ns is allowed to account for slow devices. The following table summarizes the different modes available.

Test	SI	Functions
0	0	Normal Management Mode
0	1	Normal Management Mode
1	0	Minimum Mode, Receive Polarity Correction disabled
1	1	Minimum Mode, Receive Polarity Correction enabled

In Minimum Mode, the SO pin is used to serially output the various status information based on the state of the SI and SCLK pins. A summary of the status information is provided in the following table.

SCLK	SI	SO Output
0	0	TP Ports Receive Polarity Status + AUI SQE Test Error Status.
0	1	Bit Rate Error (all ports).
1	0	TP Ports Link Status + AUI LoopBack Status
1	1	Port Partitioning Status (all ports)

When SI = 0 then SO will output the related AUI status bits (LoopBack or SQE), followed by the 8 TP status bits (Link or Polarity), starting with the TP port 0.

When SI = 1, the Port Partitioning Status or Port Bit Rate Error Status are scanned out with the AUI first and TP ports following. TP Port 0 is scanned out first.

Note that the Bit Rate Error, AUI Loopback, and AUI SQE Test Error status bits stay set until they are scanned out.

The state of SI and SCLK inputs is checked at the end of every STR cycle. The rising edge of the X1 clock, occurring before falling edge of STR, is used to strobe in the state of the SI and SCLK pins.

In this Minimum Mode, the Management Port mode is not active. To exit the Minimum mode, the IMR+ device must reset into the normal Management Port mode.

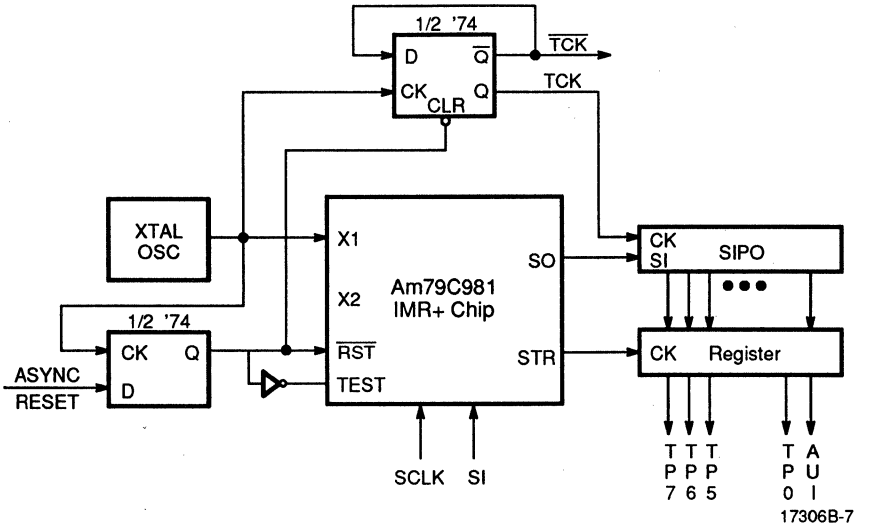
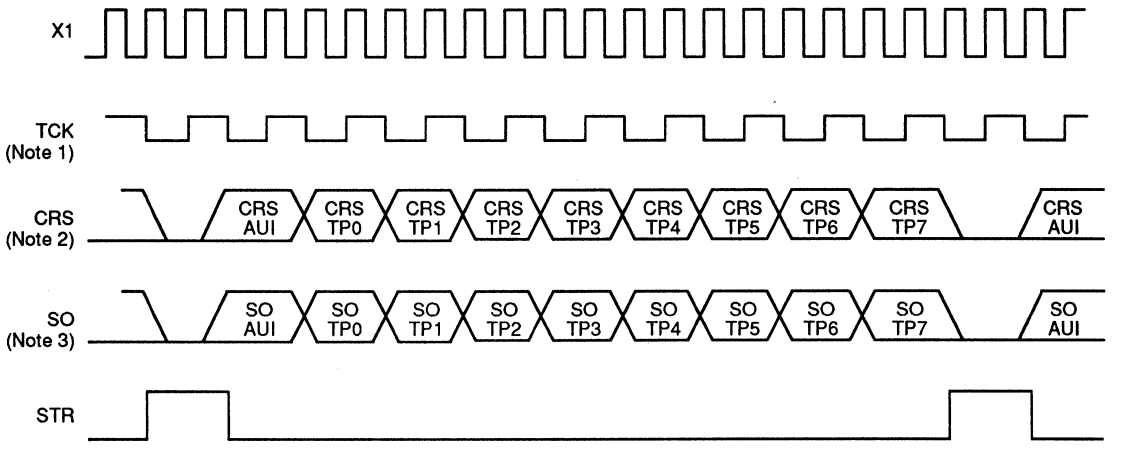


Figure 3. Minimum Mode, Non-Intelligent Repeater Example



Notes:

1. *Externally generated signal illustrates internal IMR+ chip clock phase relationship.*
2. *CRS timing with the C-bit cleared (IMR+ Chip Programmable Options)*
3. *For Minimum Hub Mode*

Figure 4. Management Port Minimum Mode and Port Activity Monitor Signal Relationship

Port Activity Monitor

Two pins, CRS and STR, are used to serially output the state of the internal Carrier Sense signals from the AUI and the eight TP ports. This function together with external hardware and/or software can be used to monitor repeater receive and/or collision activity.

The following diagram shows typical external hardware employed to convert the serial bit stream into parallel form. The accuracy of the CRS signals is 10 Bit Times (BT) (1 µs). Specifically, a transition to active state by any of the internal carrier sense bits that lasts for less than 10BT is latched internally and is used to set the appropriate bit during the next sample period.

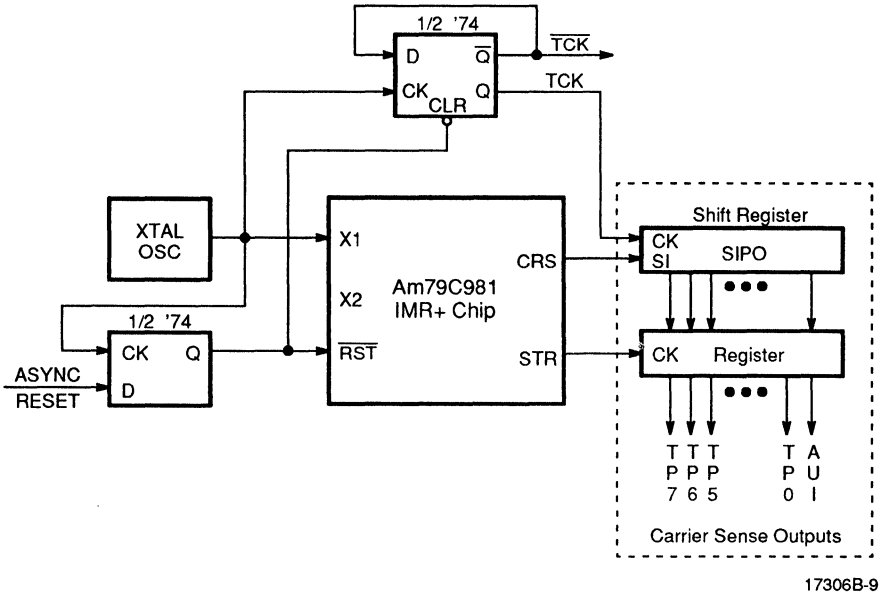
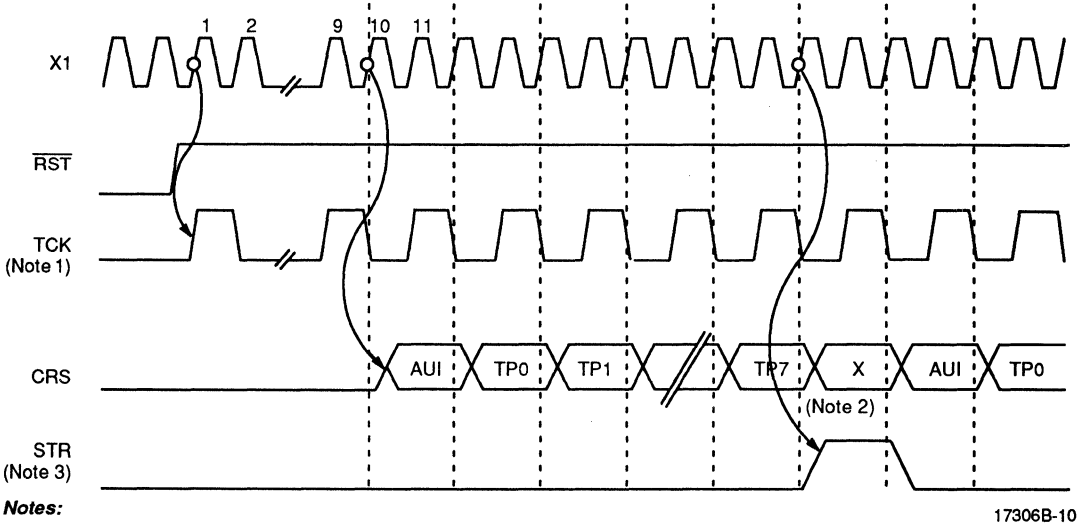


Figure 5a. Port Activity Monitor Implementation



Notes:

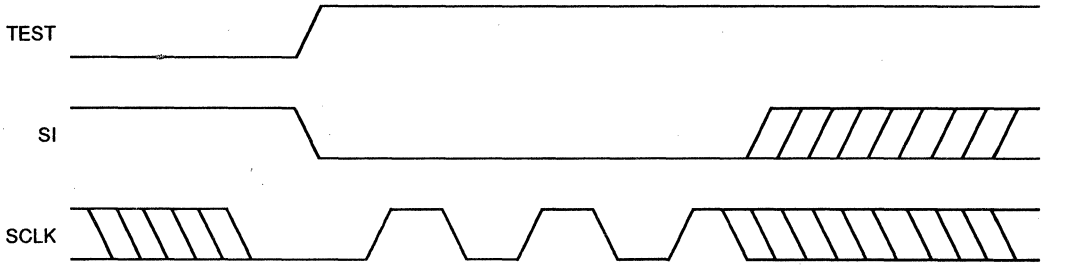
1. *Externally generated signal illustrates internal IMR+ chip clock phase relationship.*
2. *IMR+ chip standalone, X will be low. When attached to a HIMIB device, X reflects the state of the CI pair.*
3. *STR signal is not available when the IMR+ chip is attached to HIMIB device, and must be generated externally.*

Figure 5b. Port Activity Monitor Implementation (Continued)

Loopback Test Mode

The IMR+ chip can be programmed to enter Loopback Mode on all network ports. This is accomplished by first driving the TEST pin HIGH, then clocking (using the SCLK pin) a minimum of three 0s into the SI pin. This causes the IMR+ chip to loop all received data on each port back to each port's corresponding transmit outputs. Specifically, the AUI DI input is passed unaltered to the AUI DO output, and each RXD input on the twisted pair ports is passed (unaltered) to the respective TXD and TXP outputs. Only receive data that passes the required amplitude squelch criteria is looped back to the transmit outputs. Note that the data is looped back unaltered, meaning that no signal retiming or regeneration takes place. Therefore, any signal distortion present on the receive data paths will be retransmitted.

In Minimum Mode, the Loopback Test Mode cannot be accessed. The IMR+ device will return to normal operation when the TEST pin is again driven LOW.

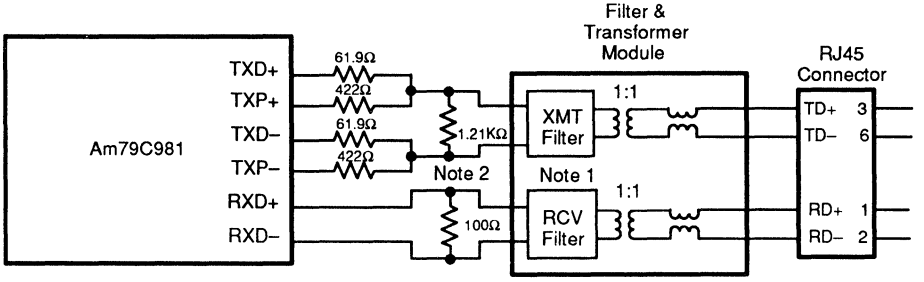


17306B-11

Figure 6. Programming the IMR+ Device for Loopback Mode

IMR+ Chip External Components

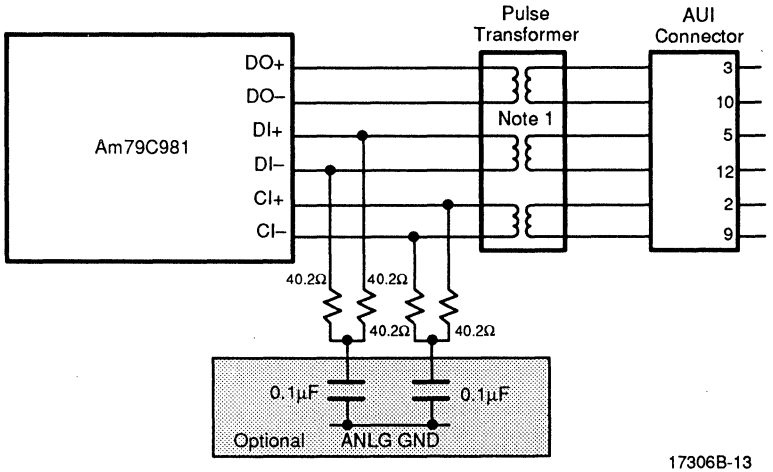
Figure 6 shows a typical twisted pair port external components schematic. The resistors used should have a 1% tolerance to ensure interoperability with 10BASE-T compliant networks. The filters and pulse transformers are necessary devices that have a major influence on the performance and compliance of the 10BASE-T ports of the repeater. Specifically, the transmitted waveforms are heavily influenced by the filter characteristics and the twisted pair receivers employ several criteria to continuously monitor the incoming signal's amplitude and timing characteristics to determine when and if to assert the internal carrier sense. For these reasons, it is crucial that the values and tolerances of the external components be as specified. Several manufacturers produce a module that combines the functions of the transmit and receive filters and the pulse transformers into one package.



Notes:

1. *Compatible filter modules, with a brief description of package type and features are included in the Appendix.*
2. *The resistor values are recommended for general purpose use and should allow compliance to the 10BASE-T specification for template fit and jitter performance. However, the overall performance of the transmitter is also affected by the transmit filter configuration.*

Figure 7a. Typical TP Port External Components



Notes:

1. *Compatible AUI transformer modules, with a brief description of package type and features are included in the Appendix.*
2. *The differential input DI± and the CI± pairs are externally terminated by two 40.2Ω ±1% resistors and one optional common-mode bypass capacitor. The differential input impedance, ZIDF, and the common-mode input impedance, ZICM, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. If SIP devices are used, 39 Ω is the nearest usable equivalent value.*

Figure 7b. Typical AUI Port Components

APPLICATIONS

A fully managed multiport repeater can be easily built by interfacing the IMR+ chip with the Hardware Implemented Management Information Base (HIMIB), Am79C987 device. The HIMIB device interfaces with all common Microprocessor System Busses with a minimum of external logic. Note that additional buffering of DAT and JAM are required for most applications. For more information, refer to AMD's IEEE 802.3 Repeater Technical Manual (PID# 17314A).

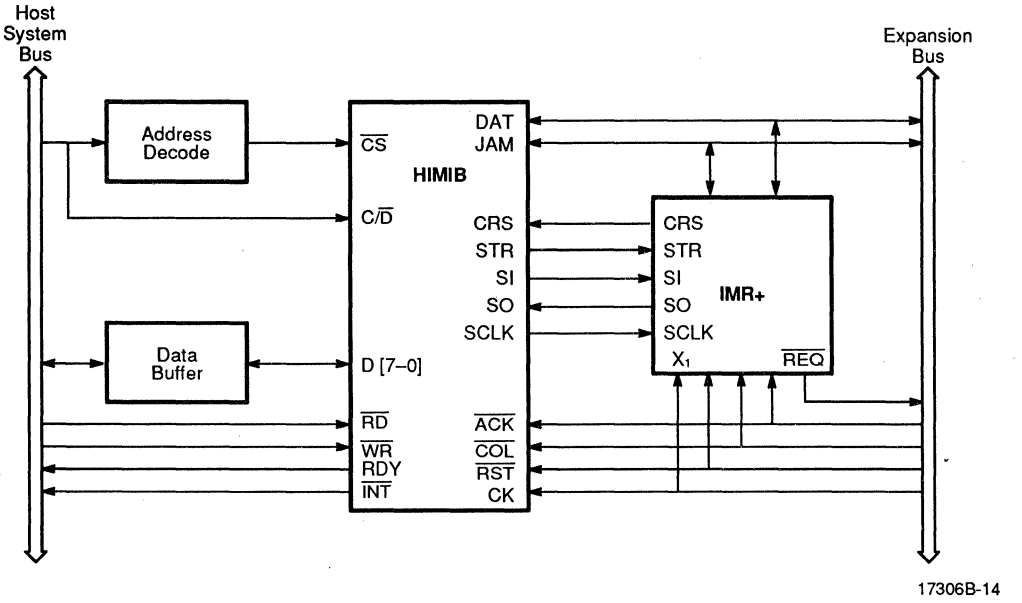


Figure 8. Simplified ISA-HUB Block Diagram

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C

Ambient Temperature Under Bias ..... 0 to 70°C

Supply Voltage referenced to
AV_{SS} or DV_{SS} (AV_{DD}, DV_{DD}) ..... −0.3 to +6 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A) ..... 0 to +70°C

Supply Voltage (AV_{DD}, DV_{DD}) ..... 5 V to ±5%

Operating ranges define those limits between which the funtionality of the device is guaranteed

DC CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Digital I/O					
V_{IL}	Input LOW Voltage	DV_{SS} = 0.0 V	−0.5	0.8	V
V_{IH}	Input HIGH Voltage		2.0	DV_{DD}+0.5	V
V_{OL}	Output LOW Voltage	I_{OL} = 4.0 mA	–	0.4	V
V_{OH}	Output HIGH Voltage	I_{OH} = −0.4 mA	2.4	–	V
I_{IL}	Input Leakage Current (also DAT and JAM as inputs)	DV_{SS}<V_{IN}<DV_{DD}	–	10	µA
V_{ILX}	X_1 Crystal Input LOW Voltage	DV_{SS} = 0.0 V	−0.5	1.0	V
V_{IHX}	X_1 Crystal Input HIGH Voltage	DV_{SS} = 0.0 V	3.8	DV_{DD}+0.5	V
I_{ILX}	Crystal Input LOW Current	V_{IN} = DV_{SS}	–	10	µA
I_{IHX}	Crystal Input HIGH Current	V_{IN} = DV_{DD}	–	10	µA
AUI Port					
I_{IAXD}	Input Current at DI+/- and CI+/- pairs	AV_{SS} < V_{IN} < AV_{DD}	−500	+500	µA
V_{AICM}	DI+,DI-,CI+,CI- Open Circuit Input Common Mode Voltage (bias)	I_{IN} = 0A, AV_{SS} = 0 V	AV_{DD} − 3.0	AV_{DD}−1.0	V
V_{AIDV}	Differential Mode Input Voltage Range (DI, CI)	AV_{DD} = 5.0 V	−2.5	+2.5	V
V_{ASQ}	DI, CI Squelch Threshold		−275	−160	mV
V_{ATH}	DI Switching Threshold	(Note 1)	−35	+35	mV
V_{AOD}	Differential Output Voltage |(DO+) − (DO-)|	R_L = 78 Ω	620	1100	mV
V_{AODI}	DO Differential Output Voltage Imbalance	R_L = 78 Ω	−25	+25	mV
V_{AODOFF}	DO Differential Idle Output Voltage	R_L = 78 Ω	−40	+40	mV
I_{AODOFF}	DO Differential Idle Output Current	R_L = 78 Ω (Note 1)	−1	+1	mA
V_{AOCM}	DO+/- Common Mode Output Voltage	R_L = 78 Ω	2.5	AV_{DD}	V

DC CHARACTERISTICS (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Twisted Pair Ports					
I_{IRXD}	Input current at RXD+/-	$AV_{SS}<V_{IN}<AV_{DD}$	–500	+500	µA
R_{RXD}	RXD differential input resistance	(Note 1)	10	–	KΩ
V_{TIVB}	RXD+,RXD- open circuit input voltage (bias)	I_{IN} = 0 mA	AV_{DD}–3.0	AV_{DD}–1.5	V
V_{TID}	Differential Mode input voltage range (RXD)	AV_{DD} = 5.0 V	–3.1	+3.1	V
V_{TSQ+}	RXD positive squelch threshold (peak)	Sinusoid 5 MHz <f< 10 MHz	300	520	mV
V_{TSQ-}	RXD negative squelch threshold (peak)	Sinusoid 5 MHz <f< 10 MHz	–520	–300	mV
V_{THS+}	RXD post-squelch positive threshold (peak)	Sinusoid 5 MHz <f< 10 MHz	150	293	mV
V_{THS-}	RXD post-squelch negative threshold (peak)	Sinusoid 5 MHz <f< 10 MHz	–293	–150	mV
V_{RXDTH}	RXD switching threshold	(Note 1)	–60	+60	mV
V_{TXH}	TXD+/- and TXP+/- output HIGH voltage	DV_{SS} = 0 V (Note 2)	DV_{DD}–0.6	DV_{DD}	V
V_{TXL}	TXD+/- and TXP+/- output LOW voltage	DV_{DD} = 5 V (Note 2)	DV_{SS}	DV_{SS}+0.6	V
V_{TXI}	TXD+/- and TXP+/- differential output voltage imbalance		–40	+40	mV
V_{TXOFF}	TXD+/- and TXP+/- differential idle output voltage	DV_{DD} = 5 V	–	40	mV
R_{TXD}	TXD+/- differential driver output impedance	(Note 1)	–	40	Ω
R_{TXP}	TXP+/- differential driver output impedance	(Note 1)	–	80	Ω
Power Supply Current					
I_{DD}	Power supply current (idle)	f_{X1} = 20 MHz	–	180	mA
	Power supply current (transmitting – no TP load)	f_{X1} = 20 MHz	–	300	mA
	Power supply current (transmitting – with TP load)	f_{X1} = 20 MHz	–	(Note 8)	mA

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Clock and Reset					
t_{X1}	X1 Clock Period		49.995	50.005	ns
t_{X1H}	X1 Clock HIGH		20	30	ns
t_{X1L}	X1 Clock LOW		20	30	ns
t_{X1R}	X1 Clock Rise Time		–	10	ns
t_{X1F}	X1 Clock Fall Time		–	10	ns
t_{PRST}	Reset pulse width after power on ($\overline{RST}$ pin LOW)		150	–	µs
t_{RST}	Reset pulse width ($\overline{RST}$ pin LOW)		4	–	µs
t_{RSTSET}	$\overline{RST}$ HIGH setup time with respect to X1 Clock		20	–	ns
t_{RSTHLD}	$\overline{RST}$ LOW hold time with respect to X1 Clock		0	–	ns
Management Port					
t_{SCLK}	SCLK Clock Period		100	–	ns
t_{SCLKH}	SCLK Clock HIGH		30	–	ns
t_{SCLKL}	SCLK Clock LOW		30	–	ns
t_{SCLKR}	SCLK Clock Rise Time		–	10	ns
t_{SCLKF}	SCLK Clock Fall Time		–	10	ns
t_{SISET}	SI input setup time with respect to SCLK rising edge		10	–	ns
t_{SIHLD}	SI input hold time with respect to SCLK rising edge		10	–	ns
t_{SODLY}	SO output delay with respect to SCLK rising edge	C_L = 100 pF	–	40	ns
t_{X1HCRS}	X1 rising edge to CRS valid	C_L = 100 pF	5	40	ns
t_{X1HSTH}	X1 rising edge to STR HIGH	C_L = 100 pF	–	40	ns
t_{X1HSTL}	X1 rising edge to STR LOW	C_L = 100 pF	–	40	ns
$t_{TESTSET}$	TEST input setup time with respect to SCLK rising edge		10	–	ns
$t_{TESTHLD}$	TEST input hold time with respect to SCLK rising edge		10	–	ns
t_{STRSET}	STR setup time		5	–	ns
t_{STRHLD}	STR hold time		12	–	ns
Expansion Port					
t_{X1HRL}	X1 rising edge to $\overline{REQ}$ driven LOW	C_L = 100 pF	14	40	ns
t_{X1HRH}	X1 rising edge to $\overline{REQ}$ driven HIGH	C_L = 100 pF	14	40	ns
t_{X1HDR}	X1 rising edge to DAT/JAM driven	C_L = 100 pF	14	40	ns
t_{X1HDZ}	X1 rising edge to DAT/JAM not driven	C_L = 100 pF	14	40	ns

SWITCHING CHARACTERISTICS (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Expansion Port (Continued)					
t_{DJSET}	DAT/JAM setup time		10	–	ns
t_{DJHOLD}	DAT/JAM hold time		14	–	ns
t_{CASET}	$\overline{COL/ACK}$ setup time		5	–	ns
t_{CAHOLD}	$\overline{COL/ACK}$ hold time		14	–	ns
t_{MHSET}	TEST setup time with respect to $\overline{RST}$ to enter Minimum Hub Mode		200	–	ns
t_{MHHLD}	TEST hold time with respect to $\overline{RST}$ to enter Minimum Hub Mode		0	100	ns
$t_{SCLKSET}$	SI, SCLK set up time with respect to X1		50	–	ns
$t_{SCLKHLD}$	SI, SCLK hold time with respect to X1		50	–	ns
AUI Port					
t_{DOTD}	X1 rising edge to DO toggle		–	30	ns
t_{DOTR}	DO+,DO- rise time (10% to 90%)		2.5	5.0	ns
t_{DOTF}	DO+,DO- fall time (90% to 10%)		2.5	5.0	ns
t_{DORM}	DO+,DO- rise and fall time mismatch		–	1.0	ns
t_{DOETD}	DO+/- End of Transmission		275	375	ns
t_{PWODI}	DI pulse width accept/reject threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 3)	15	45	ns
t_{PWKDI}	DI pulse width maintain/turn-off threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 4)	136	200	ns
t_{PWOCI}	CI pulse width accept/reject threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 5)	10	26	ns
t_{PWKCI}	CI pulse width maintain/turn-off threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 6)	90	160	ns
Twisted Pair Ports					
t_{TXTD}	X1 rising edge to TXD+,TXP+ TXD-,TXP- transition delay		–	50	ns
t_{TR}	TXD+,TXD-,TXP+,TXP- rise time		–	20	ns
t_{TF}	TXD+,TXD-,TXP+,TXP- fall time		–	20	ns
t_{TM}	TXD+,TXD-,TXP+,TXP- rise and fall time mismatch		–	6	ns
t_{TETD}	Transmit End of Transmission		275	375	ns
t_{PWKRD}	RXD pulse width maintain/turn-off threshold	$\|V_{IN}\| > \|V_{THS}\|$ (Note 7)	130	200	ns
t_{PERLP}	Idle signal period		8	24	ms
t_{PWLP}	Idle Link Test pulse width (TXD+)		75	120	ns
t_{PWPLP}	Idle Link Test pulse width (TXP+,TXP-)		40	60	ns

SWITCHING CHARACTERISTICS (continued)

Notes:

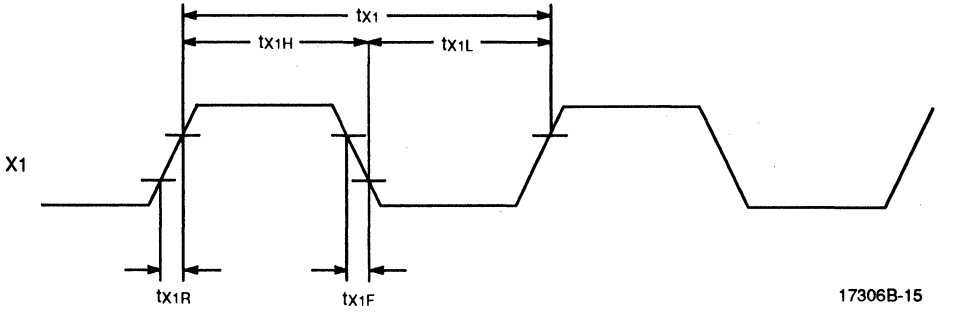
1. *Parameter not tested.*
2. *Uses switching test load.*
3. *DI pulses narrower than t_{PWODI} (min) will be rejected; pulses wider than t_{PWODI} (max) will turn internal DI carrier sense on.*
4. *DI pulses narrower than t_{PWKDI} (min) will maintain internal DI carrier sense on; pulses wider than t_{PWKDI} (max) will turn internal DI carrier sense off.*
5. *CI pulses narrower than t_{PWOCI} (min) will be rejected; pulses wider than t_{PWOCI} (max) will turn internal CI carrier sense on.*
6. *CI pulses narrower than t_{PWKCI} (min) will maintain internal CI carrier sense on; pulses wider than t_{PWKCI} (max) will turn internal CI carrier sense off.*
7. *RXD pulses narrower than t_{PWKRD} (min) will maintain internal RXD carrier sense on; pulse wider than t_{PWKRD} (max) will turn internal RXD carrier sense off.*
8. *For the typical twisted pair load as shown in Figure 7, using a 100 Ω cable, an additional 28 mA (max) of I_{DD} current is required for each twisted pair port used. Less than 18% of the power associated with this additional current is dissipated by the IMR+ chip; the remainder is dissipated externally in the twisted pair load and cable.*

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	Must Be Steady	Will Be Steady
	May Change from H to L	Will Be Changing from H to L
	May Change from L to H	Will Be Changing from L to H
	Don't Care Any Change Permitted	Changing State Unknown
	Does Not Apply	Center Line is High Impedance "Off" State

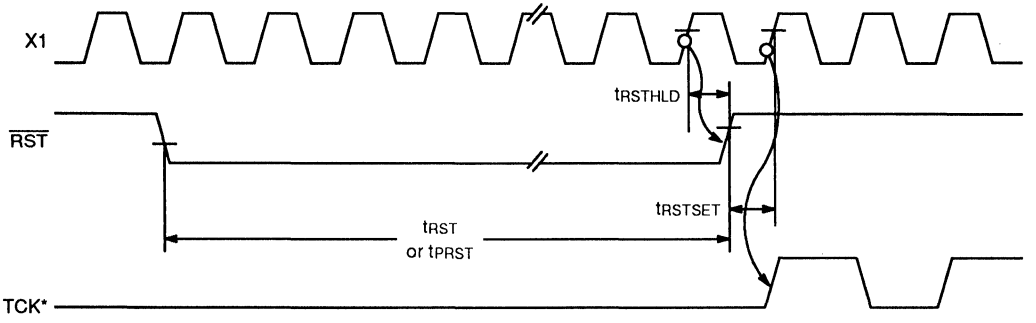
KS000010

SWITCHING WAVEFORMS



Clock Timing

SWITCHING WAVEFORMS



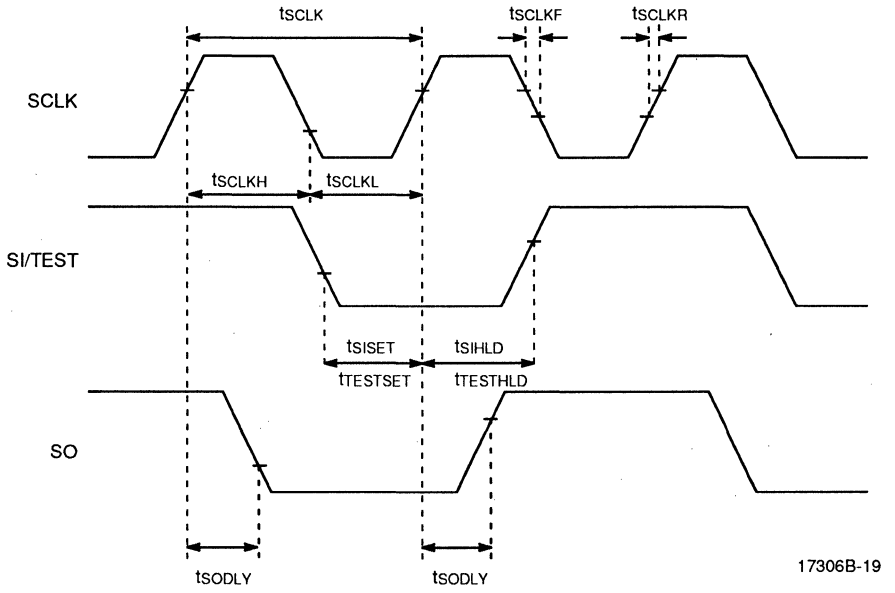
17306B-16

Notes:

tRSTSET refers to synchronous Reset Timing.

**Externally generated (Figure 4) signal illustrates internal IMR+ device clock phase relationships.*

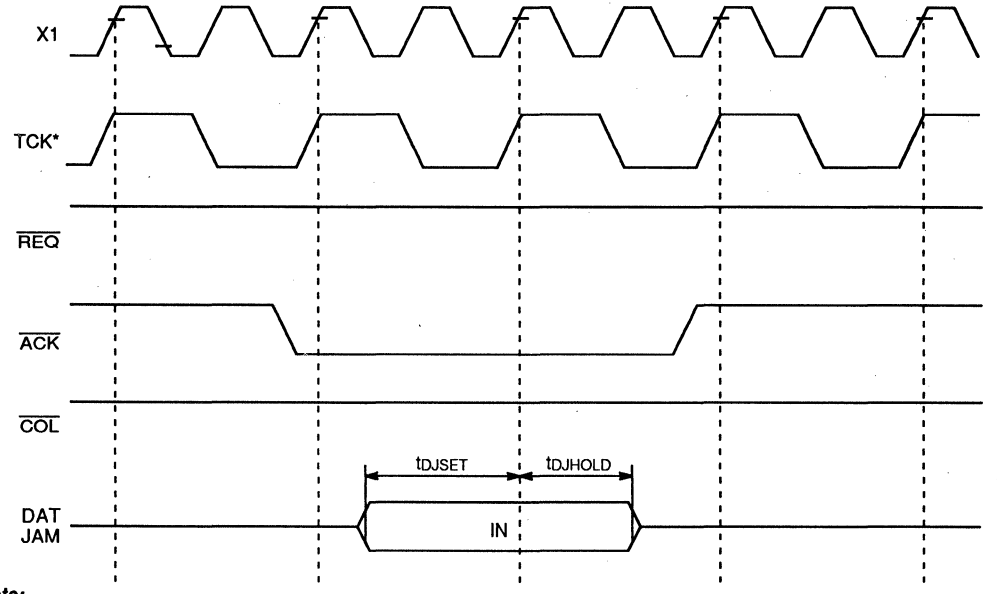
Reset Timing



17306B-19

Management Port Clock Timing

SWITCHING WAVEFORMS

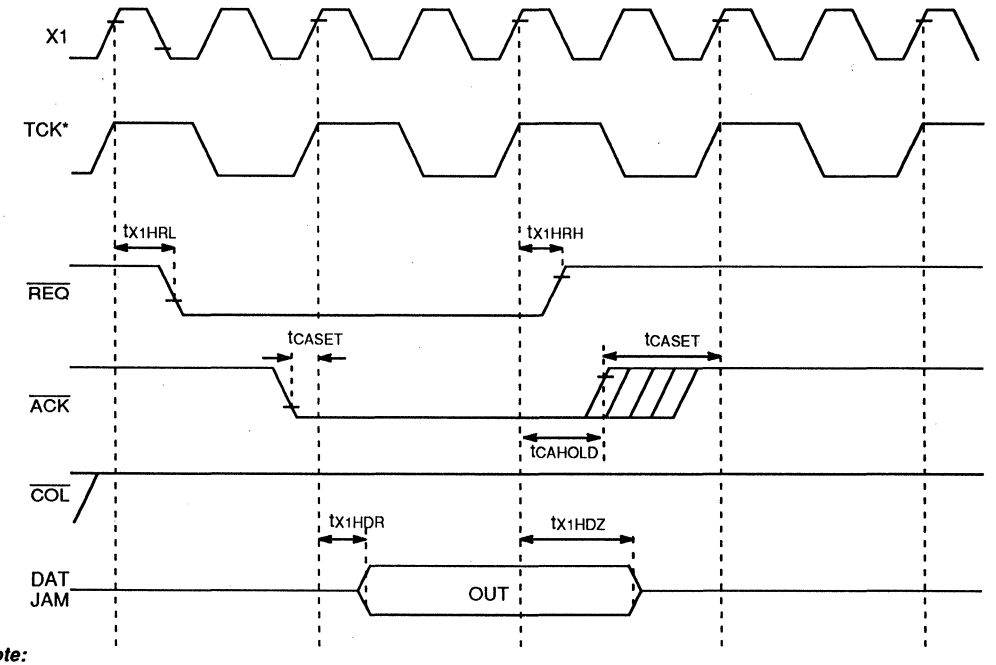


Note:

**Externally generated (Figure 4) signal illustrates internal IMR+ chip clock phase relationships.*

17306B-20

Expansion Port Input Timing



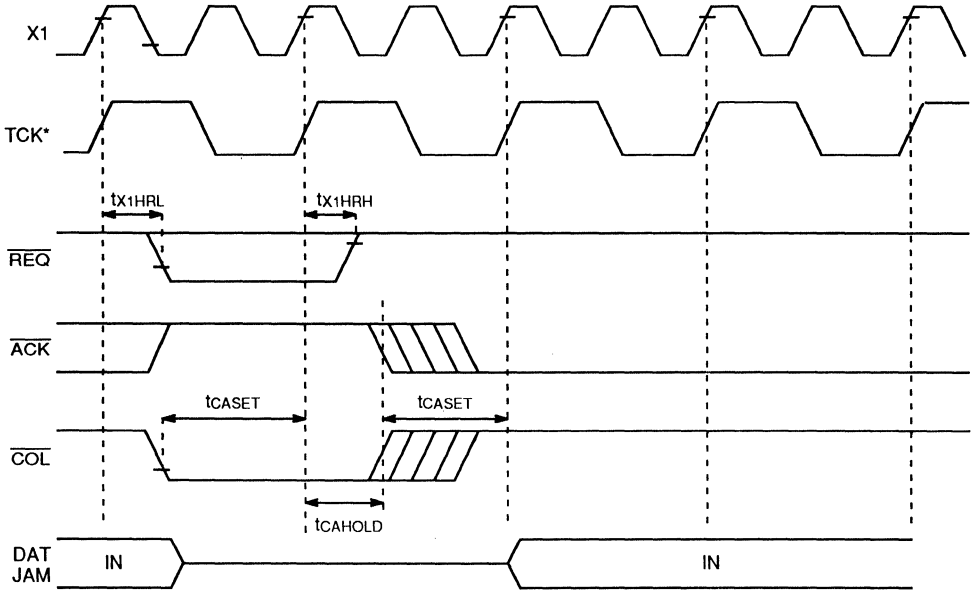
Note:

**Externally generated (Figure 4) signal illustrates internal IMR+ chip clock phase relationships.*

17306B-21

Expansion Port Output Timing

SWITCHING WAVEFORMS

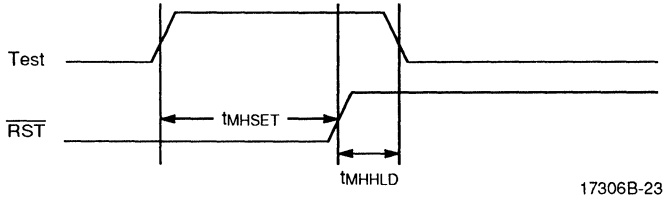


17306B-22

Note:

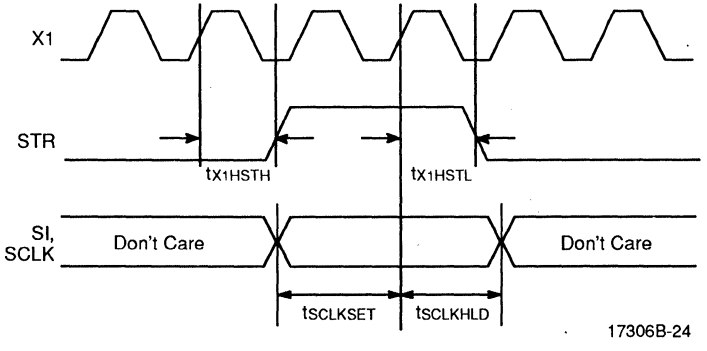
**Externally generated (Figure 4) signal illustrates internal IMR+ chip clock phase relationships.*

Expansion Port Collision Timing



17306B-23

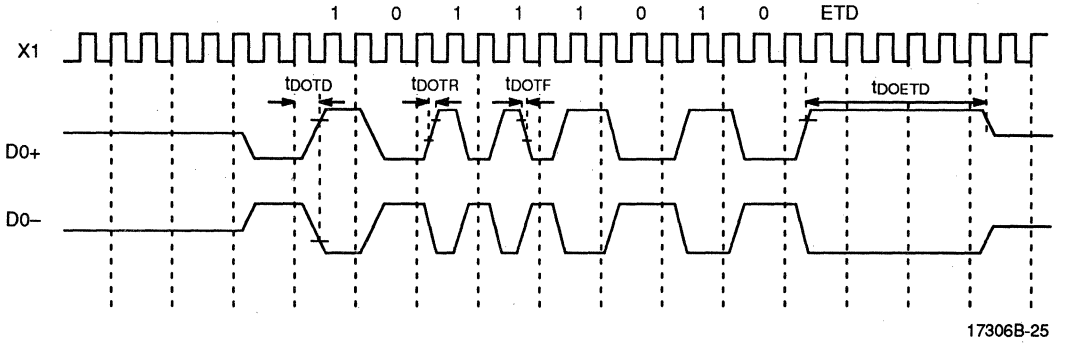
To Enter Minimum Mode



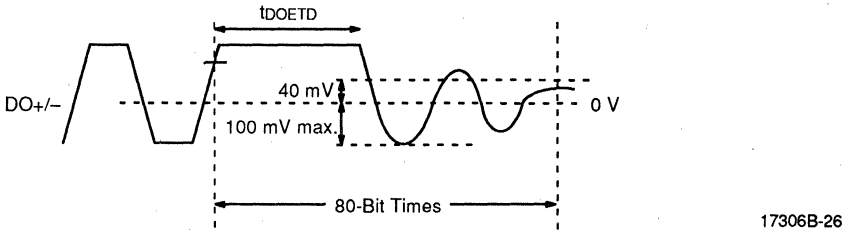
17306B-24

Minimum Mode

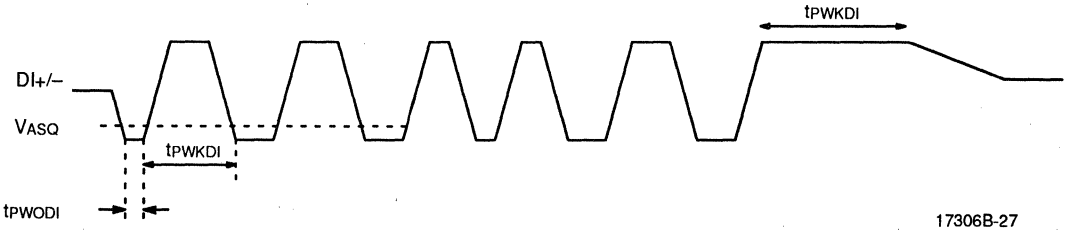
SWITCHING WAVEFORMS



AUI DO Timing Diagram

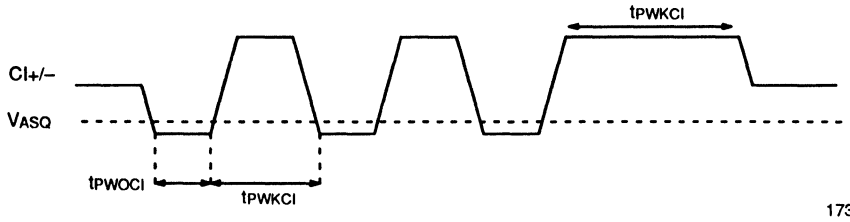


AUI Port DO ETD Waveform

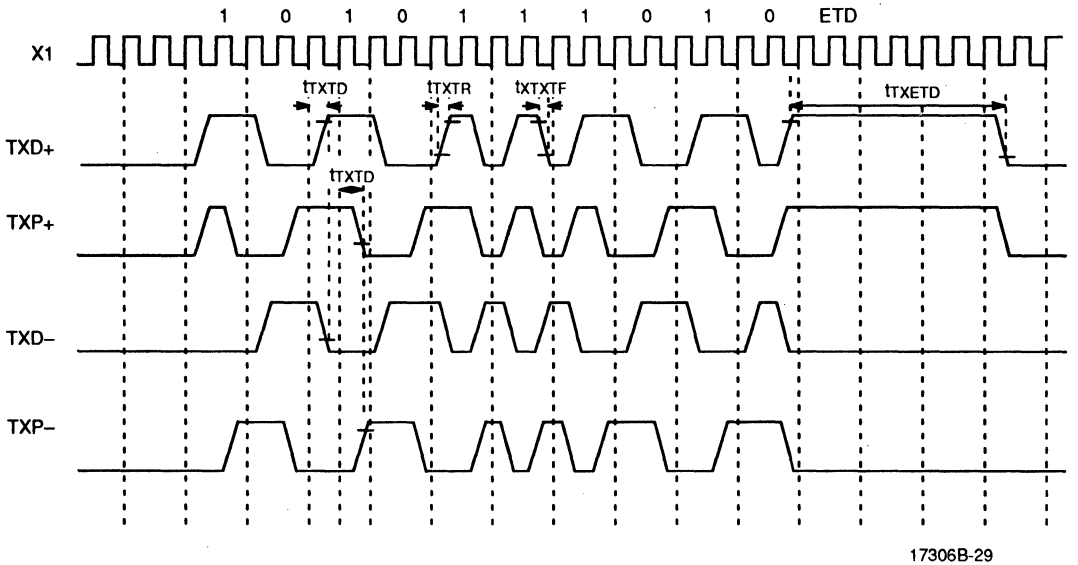


AUI Receive Timing Diagram

SWITCHING WAVEFORMS

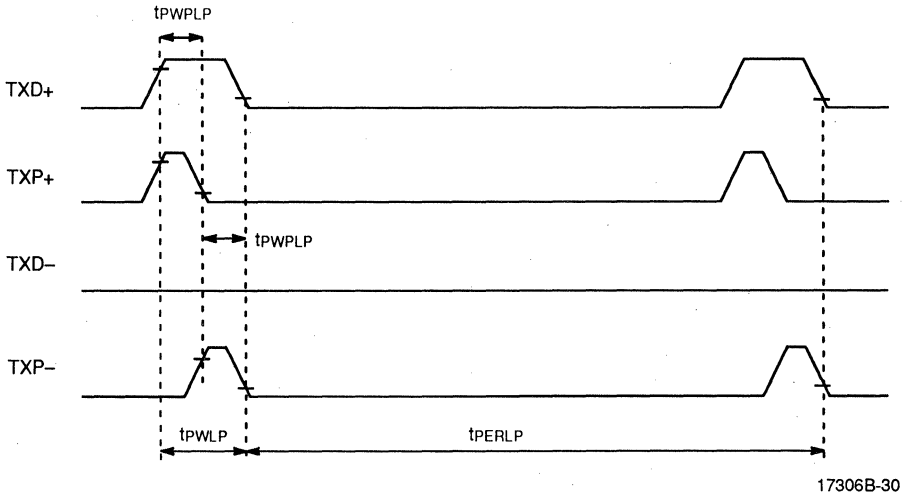


AUI Collision Timing Diagram

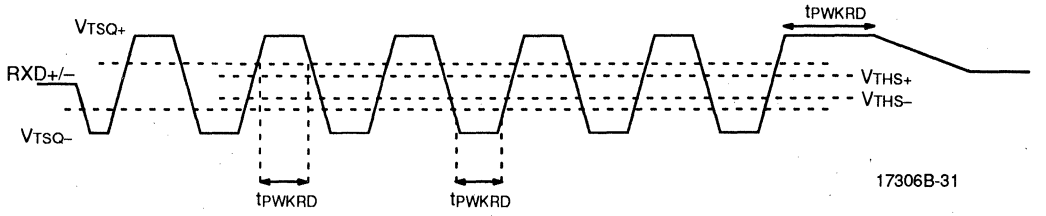


TP Ports Output Timing Diagram

SWITCHING WAVEFORMS

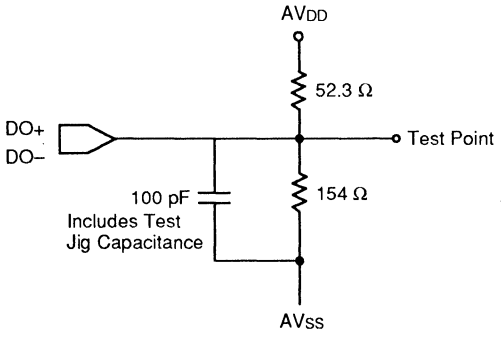


TP Idle Link Test Pulse

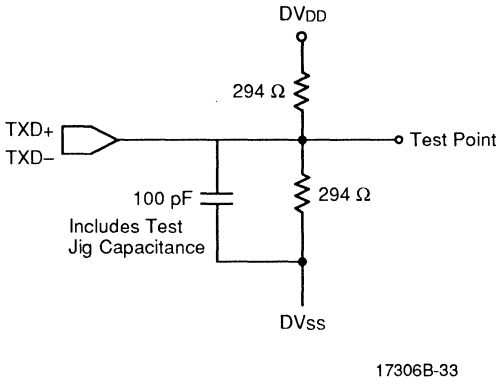


TP Receive Timing Diagram

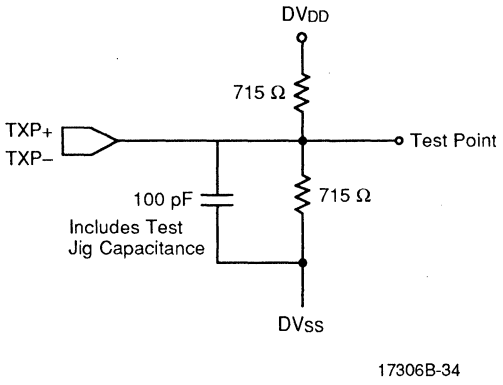
SWITCHING TEST CIRCUITS



AUI DO Switching Test Circuit



TXD Switching Test Circuit



TXP Outputs Test Circuit

APPENDIX A

10BASE-T Interface

The table below lists the recommended resistor values and filter and transformer modules for the IMR+ device.

IMR+ Device Compatible 10BASE-T Media Interface Modules

Manufacturer	Part #	Package	Description
Bel Fuse	A556-2006-DE	16-pin 0.3" DIL	Transmit and receive filters and transformers.
Bel Fuse	0556-2006-00	14-pin SIP	Transmit and receive filters and transformers.
Bel Fuse	0556-2006-01	14-pin SIP	Transmit and receive filters, transformers and common mode chokes.
Bel Fuse	0556-6392-00	16-pin 0.5" DIL	Transmit and receive filters, transformers and common mode chokes.
Halo Electronics	FD02-101G	16-pin 0.3" DIL	Transmit and receive filters and transformers.
Halo Electronics	FD12-101G	16-pin 0.3" DIL	Transmit and receive filters and transformers, transmit common mode choke.
Halo Electronics	FD22-101G	16-pin 0.3" DIL	Transmit and receive filters, transformers and common mode chokes.
Nano pulse	NP6612	16-pin 0.3" DIL	Transmit and receive filters, transformers and common mode chokes.
Nano pulse	NP6581	8-pin 0.3" DIL	Transmit and receive common mode chokes.
Nano pulse	NP6696	24-pin 0.6" DIL	Transmit and receive filters, transformers and common mode chokes.
PCA Electronics	EPA1990A	16-pin 0.3" DIL	Transmit and receive filters and transformers.
PCA Electronics	EPA2013D	16-pin 0.3" DIL	Transmit and receive filters and transformers, transmit common mode choke.
Pulse Engineering	PE-65434	10-pin SIP	Transmit and receive filters, transformers, and common mode choke.
Pulse Engineering	PE-65445	16-pin 0.3" DIL	Transmit and receive filters and transformers (for SMT use PE-65446)
Pulse Engineering	PE-65467	16-pin 0.3" DIL	Transmit and receive filters, transformers, common mode chokes, and AMD specified resistors.
Pulse Engineering	PE-65424	16-pin 0.3" DIL	Transmit and receive filters, transformers, and common mode chokes.
TDK	TLA 470	14-pin SIP	Transmit and receive filters and transformers.
TDK	HIM3000	24-pin 0.6" DIL	Transmit and receive filters, transformers and common mode chokes.
Valor Electronics	PT3877	16-pin 0.3" DIL	Transmit and receive filters and transformers.
Valor Electronics	PT3983	8-pin 0.3" DIL	Transmit and receive common mode chokes.
Valor Electronics	FL1012	16-pin 0.3" DIL	Transmit and receive filters and transformers, transmit common mode choke.

APPENDIX B

Glossary

Active Status

In a non-collision state, an IMR+ chip is considered active if it is receiving data on any one of its network ports, or is in the process of broadcasting (repeating) FIFO data from a recently completed data reception. In a collision state (the IMR+ device is generating Jam Sequence), an IMR+ device is considered active if any one or more network ports is receiving data. The IMR+ device asserts the $\overline{REQ}$ line to indicate that it is active.

Collision

In a carrier sense multiple access/collision detection (CSMA/CD) network such as Ethernet, only one node can successfully transfer data at any one time. When two or more separate nodes (DTEs or repeaters) are simultaneously transmitting data onto the network, a Collision state exists. In a repeater using one or more IMR+ devices, a Collision state exists when more than one network port is receiving data at any instant, or when any one or more network ports receives data while the IMR+ device is transmitting (repeating) data, or when the CI+/- pins become active (nominal 10 MHz signal) on the AUI port.

Jam Sequence

A signal consisting of alternating 1s and 0s that is generated by the IMR+ device when a Collision state is detected. This signal is transmitted by the IMR+ device to indicate to the network that one or more network ports in the repeater is involved in a collision.

Network Port

Any of the eight 10BASE-T ports or the AUI port present in the IMR+ device (i.e. not the Expansion Port or the Management Port).

Partitioning

A network port on a repeater has been partitioned if the repeater has internally 'disconnected' it from the repeater due to localized faults that would otherwise bring the entire network down. These faults are generally cable shorts and opens that tend to cause excessive collisions at the network ports. The partitioned network port will be internally re-connected if the network port starts behaving correctly again, usually when successful 'collisionless' transmissions and/or receptions resume.

Receive Collision

A network port is in a Receive Collision state when it detects collision and is not one of the colliding network 'nodes'. This applies mainly to a non-transmitting AUI port because a remote collision is clearly identified by the presence of a nominal 10 MHz signal on the CI+/- pins. However, any repeater port would be considered to be in a receive collision state if the repeater unit is receiving data from that port as the 'one-port-left' in the collision sequence.

Transmit Collision

A network port is in a Transmit Collision state when collision occurs while that port is transmitting. On the AUI port, Transmit Collision is indicated by the presence of a nominal 10 MHz signal on the CI+/- pins while the AUI port is transmitting on the DO+/- pins. On a 10BASE-T port, Transmit Collision occurs when incoming data appears on the RXD+/- pins while the 10BASE-T port is transmitting on the TXD+/- and TXP+/- pins.

PRELIMINARY

Am79C987

Hardware Implemented Management Information Base™ (HIMIB™) Device

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Provides Repeater Management functions, complying to all options detailed in the Layer Management for 10 Mbyte/s Baseband Repeaters (IEEE802.3k) Standard
- Fully compatible with the Novell Hub Management Interface (HMI) Specification
- Provides additional IEEE MAU Management functions (802.3p draft)
- Interfaces directly with AMD's Am79C981 Integrated Multiport Repeater Plus™ (IMR+™) device to build a fully managed repeater
- Multiple HIMIB/IMR+ devices can be used in a system
- 8-bit microprocessor interface allows attribute access, interrupt control, and management control
- Maskable interrupts for notification of status/error reporting
- Internal "receive only" MAC tracks all address information and monitors exception conditions
- Supports mapping of node source addresses to port numbers, through implementing source address match function
- Full 32-bit hardware implemented counters incur no additional software overhead to keep network statistics
- Pinout allows simple board layout between IMR+ and HIMIB devices
- 28-pin PLCC device in CMOS technology for low power with a single +5 V supply

GENERAL DESCRIPTION

The Am79C987 Hardware Implemented Management Information Base (HIMIB) device is a highly integrated chip that simplifies building fully managed multiport repeaters. The device integrates all the necessary counters, attributes, actions and notifications, specified by the Layer Management for 10 Mbyte/s Baseband Repeaters (IEEE802.3k) Standard as well as additional features and enhancements, including functions specific to 10BASE-T repeaters.

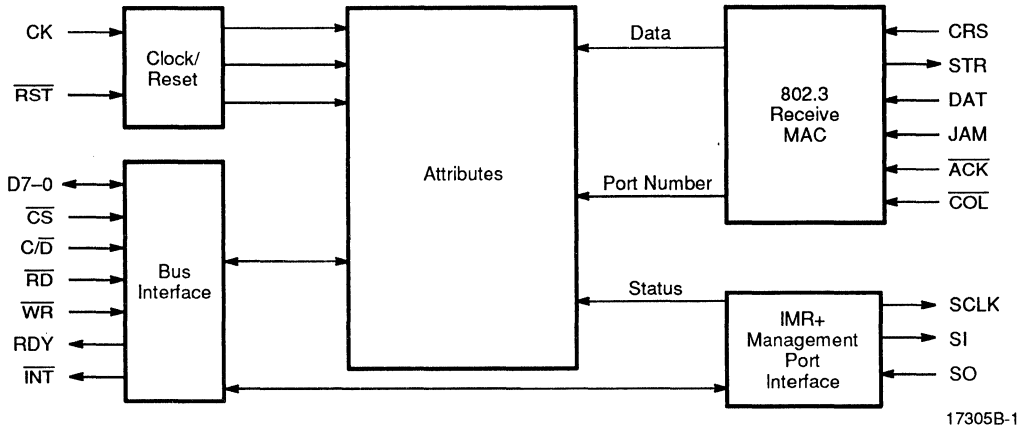
The HIMIB chip is designed to be used in conjunction with AMD's Integrated Multiport Repeater Plus (IMR+) device. When connected to an IMR+ (Am79C981) device, the HIMIB chip provides complete repeater and per port statistics on demand from an 8-bit parallel interface. No external processor is required to keep track of attributes locally as full 32-bit counters are provided.

The HIMIB device implements a simple 8-bit microprocessor interface, allowing multiple HIMIB devices to be used in a system. No additional logic is required for interfacing the HIMIB device to the IMR+ device.

The HIMIB chip is packaged in a 28-pin Plastic Leaded Chip Carrier (PLCC). The device is fabricated in CMOS technology and requires a single +5 V supply.

Publication# 17305 Rev. B Amendment/0
Issue Date: May 1994

BLOCK DIAGRAM

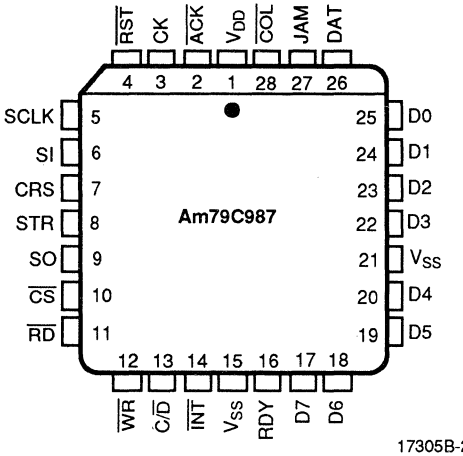


RELATED AMD PRODUCTS

Part No.	Description
Am79C98	Twisted-Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted-Pair Ethernet Transceiver Plus (TPEX+)
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA+ Single-Chip Ethernet Controller (with Microsoft® Plug n' Play support)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller (for 486 and VL buses)
Am79C970	PCnet-PCI Single-Chip Ethernet Controller for PCI Local Bus
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am7996	IEEE 802.3/Ethernet/Cheapernet Tap Transceiver

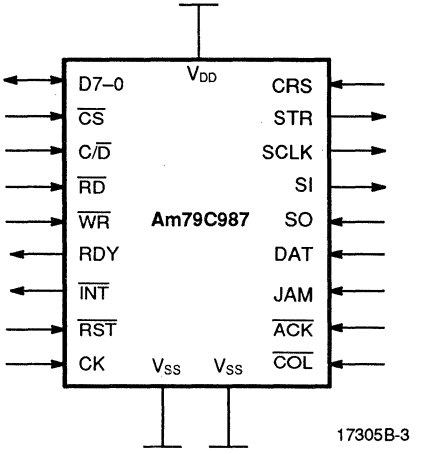
CONNECTION DIAGRAM

PLCC



17305B-2

LOGIC SYMBOL

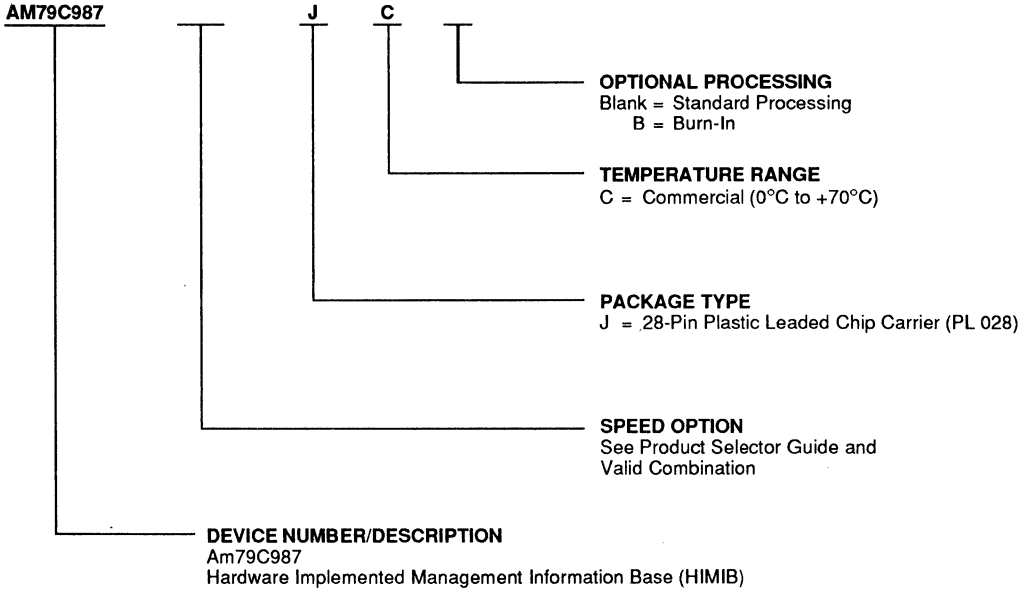


17305B-3

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM79C987	JC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

PIN DESCRIPTION

CK
Clock
Input

CK is the master 20 MHz clock. The IMR+ device X1 pin must also be clocked with the identical clock signal.

$\overline{RST}$
Reset
Input, Active Low

Driving this pin low resets the internal logic of the HIMIB. The HIMIB device must be reset with the identical synchronous $\overline{RST}$ signal of the IMR+ device.

Note:
None of the 32-bit and 48-bit attributes are cleared upon reset.

SI
Serial Input (to the IMR+ chip)
Output

The SI pin is used to output management port commands to the IMR+ device. This pin should be connected to the SI pin of the IMR+ chip.

SO
Serial Output (from the IMR+ chip)
Input

The SO pin is used to receive management port information from the IMR+ device. This pin should be connected to the SO pin of the IMR+ chip.

SCLK
Serial Clock
Output

10 MHz clock used to drive the IMR+ management port serial clock (SCLK).

CRS
Carrier Sense
Input

The CRS pin should be connected to the CRS pin of the IMR+ device. States of the internal carrier sense signals of the IMR+ AUI and twisted pair ports are serially input on this pin continuously.

STR
Store
Output, High Impedance

This pin should be connected to the STR pin of the IMR+ chip. This pin is an output when the HIMIB device is interfaced to an IMR+ device, otherwise it remains in High Impedance state.

$\overline{ACK}$
Acknowledge
Input, Active Low

When this input is asserted, it indicates that data on the DAT and JAM inputs are valid.

$\overline{COL}$
Expansion Collision
Input, Active Low

When this input is asserted, it indicates that there is a transmit collision because more than one IMR+ device is active (requesting access to the Expansion Port).

DAT
Expansion Port Data
Input

When $\overline{ACK}$ is asserted and JAM is LOW, the expansion port data consists of the NRZ received data. When $\overline{ACK}$ is not asserted, the state of DAT is ignored.

JAM
Jam
Input

When $\overline{ACK}$ is asserted and JAM is HIGH, an active IMR+ device is in a collision state. When JAM is asserted, the state of DAT will indicate either a multiport (DAT = 0) or single port (DAT = 1) collision condition. When $\overline{ACK}$ is not asserted, the state of JAM is ignored.

D7–0
Data
Input/Output, 3-State

Data Input/Output pins. These pins are in high impedance state if the HIMIB device is not selected.

C/$\overline{D}$
Command/Data
Input

This input pin allows selection of either the Command or Data Port in the HIMIB device. When this signal is HIGH, the Command Port is selected and, when it is LOW, the Data Port is selected. This pin is typically connected to the least significant bit of the address bus.

$\overline{WR}$
Write Strobe
Input, Active Low

When this pin is asserted, and the $\overline{CS}$ is active, a write operation is initiated.

$\overline{RD}$
Read Strobe
Input, Active Low

When this pin is asserted, and the $\overline{CS}$ is active, a read operation is initiated.

$\overline{CS}$
Chip Select
Input, Active Low

The chip-select input, when asserted, enables a read from or a write to the 8-bit parallel port of the HIMIB device.

RDY
Ready
Output, Open Drain

Ready is driven LOW at the start of every Read or Write cycle and is released when the HIMIB device is ready to complete the transaction.

$\overline{INT}$
Interrupt
Output, Active Low, Open Drain

Interrupt is driven LOW when any of the unmasked (enabled) interrupts occur.

V_{DD}
Power

This pin supplies +5 V to the device. Connect to DV_{DD} of the IMR+ device.

V_{SS}
Ground

These two pins are the 0 V reference for the device. Connect to DV_{SS} of the IMR+ device.

FUNCTIONAL DESCRIPTION

Overview

The functional specification of the HIMIB device is a superset of that defined by the Layer Management for 10 Mbyte/s Baseband Repeaters Standard (IEEE802.3k), commonly referred to as the "Repeater Management Standard." The HIMIB chip contains the complete set of repeater and port functions as defined in the standard. All mandatory and optional capabilities are supported. These are defined as the Basic Control, Performance Monitor and Address Tracking Capabilities. In addition, node address mapping and MAU management specific functions are implemented.

The HIMIB device keeps track of the IEEE 802.3k specified attributes by extracting data from the expansion port, management port, and port activity monitor (PAM) port of the IMR+ device. All attribute counts are held in 32 bit registers, as specified in the Repeater Management Standard. For more detailed information, refer to the IEEE 802.3 Layer Management for 10 Mbyte/s Baseband Repeaters Standard and AMD's IEEE 802.3 Repeater Technical Manual (PID #17314A).

The HIMIB chip supports the following Repeater Management functions:

Repeater Attributes:

Transmit Collisions – 32-bit counter
Total Octets – 32-bit counter

Port Attributes:

Auto Partition State – from IMR+ chip
Readable Frames – 32-bit counter
Readable Octets – 32-bit counter
Frame Check Sequence Errors – 32-bit counter
Alignment Errors – 32-bit counter
Frames Too Long – 32-bit counter
Short Events – 32-bit counter
Runts – 32-bit counter
Collisions – 32-bit counter
Late Events – 32-bit counter
Very Long Events – 32-bit counter
Data Rate Mismatches – 32-bit counter
Auto Partitions – 32-bit counter
Source Address Changes – 32-bit counter
Last Source Address – 48-bit register

Node ID to Port Address Map:
Source Address Match Register (48-bit register)

Port Actions:
Port Admin Control (Enable / Disable).

Note: *The HIMIB device executes this action by direct access to the IMR+ device Management Port.*

Individually maskable Interrupts are available for the following events:

- Change in the Port Partitioning Status
- Change in the Twisted Pair Ports Link Test State
- AUI Loop Back Error
- AUI SQE Test Error
- Source Address Changed
- Source Address Match
- IMR+ Interface Error

The HIMIB chip provides direct access to the management port of the IMR+ device for additional functions including twisted pair port automatic receive polarity detection/correction state and enabling the alternate reconnection algorithm.

The HIMIB device's 8-bit microprocessor interface allows access to onboard registers. The interface is designed to be usable with a variety of available microprocessors and buses.

The HIMIB device can also be used to collect network statistics from a standard 802.3 MAC device. This mode is programmed by setting the MAC Interface Mode Enable bit in the Configuration Register. In this mode the HIMIB device can be interfaced with any Ethernet controller with a general purpose serial interface (GPSI). The HIMIB device will record various network events occurring at that node of the network, and assign these gathered statistics to the AUI port. All TP ports statistics are invalid in this mode.

Microprocessor Interface

Access to the HIMIB device's on-chip registers is made via its simple processor interface which is designed to be used by a variety of available microprocessors. The bus interface is designed to be asynchronous and can be easily adapted for different hardware interfaces.

The interface protocol is as follows:

- Assert $\overline{CS}$ (LOW) and C/$\overline{D}$ (HIGH to access Control and LOW to access Data)
- Assert $\overline{RD}$ (LOW) to start a Read cycle or $\overline{WR}$ (LOW) to start a Write cycle
- The HIMIB device forces RDY LOW in response to the falling edge of either of $\overline{RD}$ or $\overline{WR}$

Note: *$\overline{CS}$ is internally gated with $\overline{RD}$ and $\overline{WR}$, such that $\overline{CS}$ may be permanently grounded, if not required. The start of Read or Write cycle is the time when $\overline{CS}$ and either $\overline{RD}$ or $\overline{WR}$ strobes are both asserted (LOW).*

Write Cycle:

- Data is to be placed on the Data (D7–0) pins prior to rising edge of $\overline{WR}$
- The HIMIB device releases RDY (pulled high externally), indicating that it is ready to latch the data
- $\overline{WR}$ strobe is de-asserted (HIGH) in response to RDY. The HIMIB chip latches data internally on rising edge of $\overline{WR}$
- The processor can stop driving the Data pins after the rising edge of $\overline{WR}$

Read Cycle:

- The HIMIB device drives the Data (D7–0) pins
- The HIMIB device releases RDY (pulled high externally), indicating valid data
- $\overline{RD}$ strobe is de-asserted (HIGH) in response to RDY. The external device should latch the HIMIB chip's data on the rising edge of $\overline{RD}$.
- The HIMIB device stops driving the Data pins after the rising edge of $\overline{RD}$

Typically, Read and Write cycles take 500 ns (10 CK clock cycles) to complete.

Upon reset, the Interrupt pin ($\overline{INT}$) is not driven, all internal sources of interrupts are cleared and all interrupts are disabled (masked). Use of the $\overline{INT}$ pin requires explicit enabling by setting the appropriate enable bits. The $\overline{INT}$ pin is driven low when any of the enabled interrupts occur.

The $\overline{INT}$ pin will go inactive after the internal source(s) of the interrupt are cleared by reading the corresponding Status registers.

Register Access

All HIMIB internal registers are accessed by reading or writing to or from two externally visible ports. These are the Command Port (C Port) and the Data Port (D Port).

The C Port is accessed by asserting C/$\overline{D}$ pin HIGH during read or write accesses. The D Port is accessed by driving the C/$\overline{D}$ pin LOW during Read/Write access to the HIMIB device.

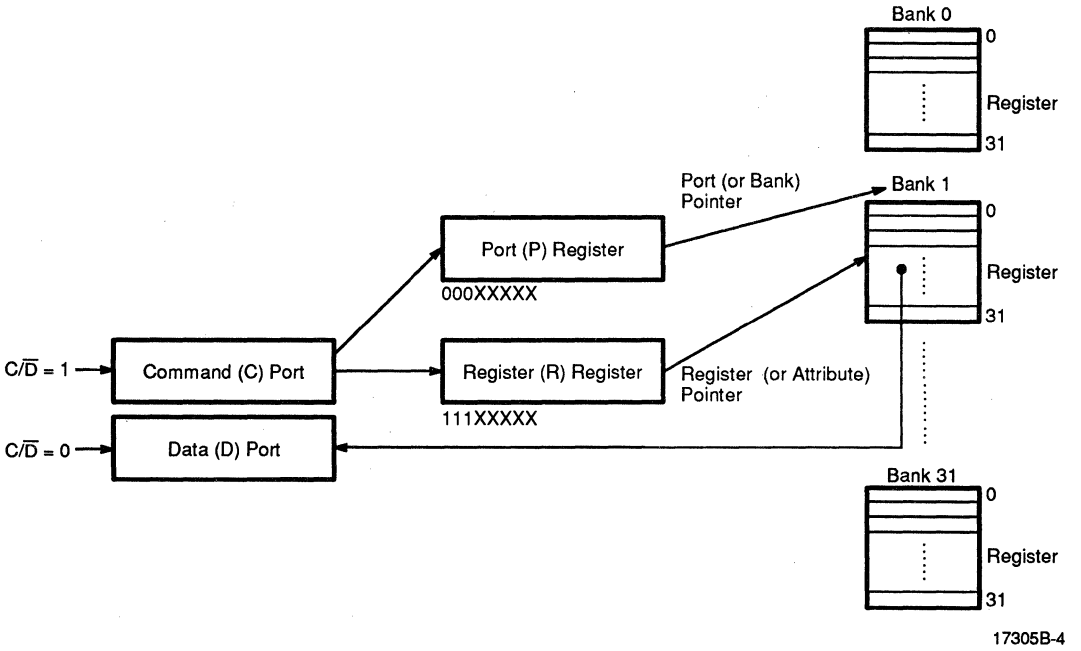
As the C/$\overline{D}$ pin is the only "address" line provided on the HIMIB device bus interface, the internal register to be accessed must be selected by writing its "address" into the Command Port.

The address appears to the programmer as two registers referred to as the P and R registers, both of which are accessed via the Command Port. The P register selects the register Port Number (or Bank Number), and is accessed by writing a byte with the three most significant bits set to zero into the C Port. The R register selects the Register Number (or Attribute Number), and is accessed by writing a byte with the three most significant bits set to one into the C Port.

Once the C Port is programmed with a valid Port (Bank) and Register (Attribute) Number, the entire 32-bit attribute is transferred to a holding register upon reading the first byte. Subsequent accesses to the D Port access the value in a least significant to most significant byte order. When reading, once the last byte is read, the attribute value is re-transferred to the holding register and the sequence can be restarted.

When the C Port is programmed for access to these multi-byte registers, reading the D Port causes the value of the register to be copied into the holding register. The data is then read out from the holding register. This sequence is repeated until the last byte is read and the D Port is accessed again. When the C Port is (re)programmed, the first byte read from the D Port will be the least significant byte.

Note that the P and R registers can be accessed in any sequence prior to accessing the D Port. If either P or R register is not written prior to accessing the D Port then the previous value of P or R register will be used.



17305B-4

Figure 1. Overview of HIMIB Register Definition

An exception to the normal Command/Data Port access scheme, is the Status Register which is read directly by reading only the C Port. This allows the Status Register to be read directly, without the need to write to the C Port.

Register Definition

In the following description, all bit fields are ordered such that the left most bit is the most significant bit. Unused Port and Register Numbers are reserved and should not be accessed as this may cause device malfunction.

When specifying the Port or Bank Number, the following command byte is written to the C Port:

C Port Write

0	0	0	P4	P3	P2	P1	P0

MSB LSB

P[4:0] represent the Register Bank or Port Number. These are organized as follows:

P = [P4P3P2P1P0]

Note that to access the P register the three most significant bits of this byte must be zero.

P [4:0]	Port/Register Bank
0	Repeater Registers
1	Port Status Registers
2	Port Control Registers
16 – 23	Twisted Pair Ports Attributes
31	AUI Port Attributes

When specifying the Register or Attribute to be accessed, the following command byte is written to the C Port.

C Port Write

1	1	1	R4	R3	R2	R1	R0

MSB LSB

R = [R4R3R2R1R0]

Note that to access the R register the three most significant bits of this byte must be one.

For P = 0 (Repeater Registers), the following registers are accessible:

R[4:0]	Register
10	Source Address Match (6-byte)
12	Total Octets (4-byte)
13	Transmit Collisions (4-byte)
16	Configuration Register
28	Version/Device ID
30	IMR+ Management Port Set Register
31	IMR+ Management Port Get Register

Register 12 and 13 are 4 bytes long and their contents are read in the least to most significant byte order.

Register 10 is 6 bytes long and can be read as well as written to in the least to most significant byte order.

Port Status Registers are organized as follows (P = 1):

R[4:0]	Register
0	TP (Twisted Pair) Ports Partition Status Change
1	AUI Port Partition Status Change
2	TP Link Status Change
3	AUI Loop Back Error
4	Reserved
5	AUI SQE Test Error
6	TP Source Address Change
7	AUI Source Address Change
8	TP Source Address Match Status
9	AUI Source Address Match Status

Port Control Registers are organized as follows (P = 2):

R[4:0]	Register
0	TP Partition Change Interrupt Enable
1	AUI Partition Change Interrupt Enable
2	TP Link Status Change Interrupt Enable
3	AUI Loop Back Error Interrupt Enable
4	Reserved
5	AUI SQE Test Error Interrupt Enable
6	TP Source Address Change Interrupt Enable
7	AUI Source Address Change Interrupt Enable

For other valid port numbers (P in the range 16...23 or 31), the following Attribute Registers are available:

R[4:0]	Register
0	Readable Frames
1	Readable Octets
2	Frame Check Sequence Errors
3	Alignment Errors
4	Frames Too Long
5	Short Events
6	Runts
7	Collisions
8	Late Events
9	Very Long Events
10	Data Rate Mismatches
11	Auto Partitions
12	Source Address Changes
13	Reserved
14	Last Source Address

Registers 0 through 12 are 4 bytes long and their contents are read in the least to most significant byte order.

Register 14 is 6 bytes long and can be read as well as written to in the least to most significant byte order.

Note that the contents of all attribute registers are maintained during an external reset. At power up, the values of all 4- and 6-byte attributes are random.

Table 1. Summary of All the HIMIB Device Registers

Register				Bytes	Access
Status Register		Note: Read the C Port for Status No Need to Specify the Port or Register Number		1	R
Port/Register Bank	**P[4:0]**	**Register**	**R[4:0]**	**Bytes**	**Access**
Repeater Registers	0	Source Address Match	10	6	R/W
		Total Octets	12	4	R
		Transmit Collisions	13	4	R
		Configuration Register	16	1	R/W
		Version/Device ID	28	1	R
		IMR+ Management Port Set Register	30	1	W
		IMR+ Management Port Get Register	31	1	R/W
Port Status Registers	1	TP Partition Status Change	0	1	R
		AUI Partition Status Change	1	1	R
		TP Link Status Change	2	1	R
		AUI Loop Back Error	3	1	R
		Reserved	4		
		AUI SQE Test Error	5	1	R
		TP Source Address Change	6	1	R
		AUI Source Address Change	7	1	R
		TP Source Address Match Status	8	1	R
		AUI Source Address Match Status	9	1	R
Port Control Registers	2	TP Partition Change Interrupt Enable	0	1	R/W
		AUI Partition Change Interrupt Enable	1	1	R/W
		TP Link Status Change Interrupt Enable	2	1	R/W
		AUI Loop Back Error Interrupt Enable	3	1	R/W
		Reserved	4		
		AUI SQE Test Error Interrupt Enable	5	1	R/W
		TP Source Address Change Interrupt Enable	6	1	R/W
		AUI Source Address Change Interrupt Enable	7	1	R/W
Attribute Registers	16–23, 31	Readable Frames	0	4	R
		Readable Octets	1	4	R
		Frame Check Sequence Errors	2	4	R
		Alignment Errors	3	4	R
		Frames Too Long	4	4	R
		Short Events	5	4	R
		Runts	6	4	R
		Collisions	7	4	R
		Late Events	8	4	R
		Very Long Events	9	4	R
		Data Rate Mismatches	10	4	R
		Auto Partitions	11	4	R
		Source Address Changes	12	4	R
		Reserved	13		
		Last Source Address	14	6	R/W

Note that all register locations listed as reserved and those which might be accessed by values or combinations of P and R which are not listed in the table above should not be accessed by the software. Read/write access to reserved registers may cause incorrect operation.

DETAILED REGISTER FUNCTIONS

Status Register

The HIMIB Status Register can be accessed at any time by reading the C Port.

The 8-bit quantity read has the following format:

C Port Read

I	E	S	X	X	X	X	X

MSB LSB

I Interrupt. This bit reflects the state of the $\overline{INT}$ output pin. If this bit is set to 1, then this HIMIB device is driving the $\overline{INT}$ pin. Note that the $\overline{INT}$ pin is an open drain output and multiple devices may share the same interrupt signal.

E Interface Error. This bit is set if the HIMIB device is unable to communicate with the IMR+ device. This bit is reset upon reading this register.

S Source Address Match. This bit is set if the interrupt is caused by a source address match of the incoming data packet. This bit remains set until the TP and/or AUI Source Address Match Status register(s) in the Port Status registers are read.

X Reserved. The values of reserved bits are indeterminate.

Repeater, Port Status, Port Control and Port Attribute Register Access

The bit pattern which must be written to the C Port in order to correctly set the value of the R register to access each of the registers is described in this section.

Repeater Register Bank

These registers are accessed by writing the bit pattern 0000 0000 to the C Port, i.e., P[4:0] = 0. Content of all attribute counters are indeterminate upon power up.

Source Address Match Register

P[4:0] = 0, R[4:0] = 10

D Port Read/Write

Byte 0	bit 7							bit 0
Byte 1								
Byte 2								
Byte 3								
Byte 4								
Byte 5	bit 47							bit 40

MSB LSB

This is a read/write register. The 6 bytes are read or written in Low byte to High byte order. The sequence is (re)started once the C Port is programmed for access to this register. This register may be used to track nodes within a LAN by reporting the port that received a packet with a specific Source Address (SA). The Source Address field of an incoming packet is always compared with the 48-bit quantity stored in this register. The initial value of this register is indeterminate.

A match is indicated by the HIMIB device by setting the corresponding bit in the TP or AUI Source Address Match Status register for the receiving port. If the corresponding Source Address Match Interrupt Enable bit is enabled, then the $\overline{INT}$ output pin is driven LOW. The set bit(s) in the TP/AUI Source Address Match Status Registers are cleared when these registers are read.

Note that once a write sequence is started, all 6 bytes must be written in order to change the contents of this register.

Total Octets

P[4:0] = 0, R[4:0] = 12

D Port Read

Byte 0	bit 7							bit 0
Byte 1								
Byte 2								
Byte 3	bit 31							bit 24

MSB LSB

This is a 4-byte attribute, read only register, whose contents are incremented while the repeater is repeating packet data. This counter is a truncated divide by 8 of the total number of bits transmitted by the repeater. The counter is incremented for non-collision packets with valid SFD (Start of Frame Delimiter). This attribute increments by same amount for all HIMIB devices connected to the same expansion bus in a repeater.

The 4 bytes in this attribute are sequentially accessed by reading the D Port, least significant byte first. Note that once the C Port is programmed for access to this attribute, reading the D Port causes the value of this register to be copied to the internal holding register. The data is then read from the holding register, without affecting this attribute. This sequence is repeated when the last byte is read and the D Port is accessed.

Transmit Collisions

P[4:0] = 0, R[4:0] = 13

D Port Read

Byte 0	bit 7							bit 0
Byte 1								
Byte 2								
Byte 3	bit 31							bit 24
	MSB							LSB

Transmit Collisions is a 4-byte read-only attribute that counts the number of transmit collisions this repeater has detected. The value of the Transmit Collisions attribute is a 32-bit counter with a minimum rollover time of 15 hours.

The 4 bytes in this attribute are sequentially accessed by reading the D Port, least significant byte first. Note that once the C Port is programmed for access to this attribute, reading the D Port causes the value of this register to be copied to the internal holding register. The data is then read from the holding register, without affecting this attribute. This sequence is repeated when the last byte is read and the D Port is accessed.

Configuration Register

P[4:0] = 0, R[4:0] = 16

This is a read/write register. The value read is the same as that written. Only zeros should be written into unused bits. All bits are cleared upon reset.

D Port Read/Write

I	E	S	M	0	0	0	0
MSB							LSB

I Enable Interrupts. When this bit is set to 0, all interrupts from this HIMIB device are masked (but not cleared) and the $\overline{INT}$ output pin is forced to inactive state (not driven).

E Interface Error Interrupt Enable. When this bit is set to 1, the HIMIB device generates an interrupt if the IMR+ interface is not functioning correctly.

S Source Address Match Interrupt Enable. When this bit is set, the HIMIB chip will generate an interrupt if the Source Address of the received packet matches that programmed into the Source Address Match Register (in the Repeater Register Bank).

M MAC Interface Mode Enable. When this bit is set to 1, the HIMIB device is assumed to be interfaced to an 802.3/Ethernet MAC Controller. In this mode only statistics for port 31 (AUI) are valid. The Expansion Port interface statistics are reported for port 31 (AUI). The HIMIB chip must be kept in this mode until an external reset occurs.

When the HIMIB chip is interfaced to a MAC device, such as AMD's LANCE (Am7990) and MACE (Am79C940) etc., the CRS pin from the MAC device should be connected to the CRS pin of the HIMIB chip. Also, the SO input pin of the HIMIB chip should be tied HIGH. Note that in this mode, the HIMIB chip will report an Interface Error in the Status Register since there is no connection to the Management Port. Therefore, it is recommended that the Interface Error Interrupt is left disabled. Certain attributes specific to the Repeater Management Standard, such as bit rate error, AUI loopback error etc., will have no meaning.

Note: *Once this bit is set by software, it should not be cleared again as this may cause incorrect device operation.*

Version and Device ID Register

P[4:0] = 0, R[4:0] = 28

This is a read only register. The 8-bit read has the following format:

D Port Read

V3	V2	V1	V0	D3	D2	D1	D0
MSB							LSB

V Version. These bits contain the HIMIB chip version code. Software may interrogate these bits to determine additional features that may be available with future versions of the device. The original version is 0000.

D Device ID. The HIMIB device detects the Repeater version upon reset. This field is updated to report the type of physical repeater attached to the HIMIB device.

D	Device
0	IMR chip (Does not support all attributes)
1	IMR+ chip
2–15	Reserved for future use

Note: *If the HIMIB chip detects an interface error upon reset, then this field may not contain valid data.*

IMR+ Management Port Set Register (S)

P[4:0] = 0, R[4:0] = 30

D Port Write

D7	D6	D5	D4	D3	D2	D1	D0
MSB							LSB

This is a write only register. This register is used for sending a Set command to the IMR+ device. When a byte is written to this register, the HIMIB chip will serialize and transfer this byte to the IMR+ Management port.

If a Get command is written to this register accidentally, the IMR+ device output will be retained in the Get register, however, the management Interface Error bit will be set in the Status Register. Writing to this register prior to execution (transfer) of the last command (Get or Set) causes the processor to be placed into the wait state.

IMR+ Management Port Get Register (G)

P[4:0] = 0, R[4:0] = 31

D Port Read/Write

D7	D6	D5	D4	D3	D2	D1	D0

MSB LSB

This is a read/write register. This register is used to transfer a Get command to the IMR+ device. This is performed by serializing and transferring the command placed into this register to the IMR+ device following the end of the processor write cycle that writes the Get command. The byte returned by the IMR+ chip is then placed in this register, overwriting its previous content. The microprocessor can read the byte result of the Get operation once the information has been transferred to the HIMIB device. If the read operation is started prior to completion of this transfer the HIMIB device will hold the RDY line inactive until the transfer is complete. In most applications this will insert wait states into the processor read cycle.

If a Set command is written to this register accidentally, the IMR+ device will receive the Set command. However, the management Interface Error bit will be set in the Status Register.

Note that reading the IMR+ Twisted Pair Bit Rate Error Status Registers using the Get command may affect accuracy of the Bit Rate Error attribute.

Port Status Registers

These registers are accessed by writing the bit pattern 0000 0001 to the C port, i.e., P[4:0] = 1. These registers are read only and are cleared to 0 upon reading.

TP and AUI Partition Status Change

Any port changing state from the partitioned to the reconnected state, or vice versa, causes the appropriate bit to be set to 1, in one of these two registers.

TP Ports

P[4:0] = 1, R[4:0] = 0

The format for the TP ports is:

D Port Read

T7	T6	T5	T4	T3	T2	T1	T0

MSB LSB

AUI Port

P[4:0] = 1, R[4:0] = 1

For the AUI port, only the most significant bit is used. Bits denoted as X are undefined.

D Port Read

A	X	X	X	X	X	X	X

MSB LSB

TP Link Status Change

P[4:0] = 1, R[4:0] = 2

A change in the Link Test state of a TP port (from Link Fail to Link Pass or vice versa), causes the appropriate bit to be set to 1 in this register:

D Port Read

T7	T6	T5	T4	T3	T2	T1	T0

MSB LSB

AUI Loop Back Error

P[4:0] = 1, R[4:0] = 3

This register is not valid for the IMR device (Am79C980). When the HIMIB chip is interfaced with the IMR+ device (Am79C981), the most significant bit (A) is set to 1 if the AUI port is connected to a MAU which does not loopback data from DO to DI during transmission. For the error to be detected, the network needs to be active and a packet transmitted from the AUI port. Bits denoted as X are undefined.

D Port Read

A	X	X	X	X	X	X	X

MSB LSB

Note that if the DO to DI loopback path is not operational, this bit will be set again when the next packet is transmitted via the AUI port.

AUI SQE Test Error

P[4:0] = 1, R[4:0] = 5

This register is not valid for the IMR device (Am79C980). When the HIMIB device is interfaced with the IMR+ chip (Am79C981), this bit is set to 1 if the AUI port is connected to a MAU with SQE Test enabled. For the error to be detected, the network needs to be active and a packet transmitted from the AUI port. Bits denoted as X are undefined.

D Port Read

A	X	X	X	X	X	X	X

MSB LSB

Note that if the error persists, once read, this bit will be set again when the next packet is transmitted via the AUI port.

TP and AUI Port Source Address Change Status

A change in the source address of a valid received frame from any port causes the appropriate bit to be set in these registers. The source address assigned to any port after power up is indeterminate, and the first packet received from any port will cause the SA changed status bit for that port to be set.

TP Ports

P[4:0] = 1, R[4:0] = 6

TP Ports Source Address Changed Status:

D Port Read

T7	T6	T5	T4	T3	T2	T1	T0

MSB LSB

AUI Port

P[4:0] = 1, R[4:0] = 7

AUI Port Source Address Changed Status:

D Port Read

A	X	X	X	X	X	X	X

MSB LSB

Note: *The Last Source Address attribute is programmable and can be used to store the expected Node ID for this port. If the appropriate interrupt is also enabled, then a change in the source address can be used to alert the network manager of an unauthorized access. This is particularly useful for segments that are supposed to be connected to a single station.*

TP and AUI Port Source Address Match Status

When the source address of the received packet from any port matches that programmed into the Source Address Match Register (in the Repeater Registers), then the appropriate bit will be set in the following registers:

TP Ports

P[4:0] = 1, R[4:0] = 8

D Port Read

T7	T6	T5	T4	T3	T2	T1	T0

MSB LSB

AUI Port

P[4:0] = 1, R[4:0] = 9

D Port Read

A	X	X	X	X	X	X	X

MSB LSB

Note: *This function is useful for mapping an individual Node ID to a specific port on the repeater.*

Port Control Registers

These registers are accessed by writing the bit pattern 0000 0010 to the C port, i.e., P[4:0] = 2. All are read/write registers. A set (1) control bit enables an interrupt or function for the corresponding port. All control registers are cleared upon reset.

TP and AUI Partition Status Change Interrupt Enable

These two registers are used to enable or mask interrupts caused by a change in the port partitioning status. All interrupts are disabled and all status bits are cleared upon hardware reset. Note that disabling an active interrupt source causes the $\overline{INT}$ output to be placed into an inactive state.

TP Ports

P[4:0] = 2, R[4:0] = 0

D Port Read/Write

T7	T6	T5	T4	T3	T2	T1	T0

MSB LSB

AUI Port

P[4:0] = 2, R[4:0] = 1

D Port Read/Write

A	X	X	X	X	X	X	X

MSB LSB

The AUI port only uses the most significant bit (A) and all other bits are reserved. Software should be designed to write 0s into unused bits.

TP Link State Change Interrupt Enable

P[4:0] = 2, R[4:0] = 2

Setting any of the bits in this register causes the $\overline{INT}$ pin to be driven when there is a change in the Link Test State of the corresponding TP port. The corresponding status bit in the TP Link Status Change register is set to 1.

D Port Read/Write

T7	T6	T5	T4	T3	T2	T1	T0
MSB							LSB

AUI Loop Back Error Interrupt Enable

P[4:0] = 2, R[4:0] = 3

Setting the A bit to 1 in this register causes the $\overline{INT}$ pin to be driven when the IMR+ chip senses a Loop Back Error condition at the AUI port.

D Port Read/Write

A	X	X	X	X	X	X	X
MSB							LSB

Note that the HIMIB device will continue generating interrupts every time a packet is transmitted by the AUI port while this condition exists. This does not necessarily indicate a problem as an unconnected AUI port will always report Loop Back Error.

AUI SQE Test Error Interrupt Enable

P[4:0] = 2, R[4:0] = 5

Setting the A bit to 1 in this register causes the $\overline{INT}$ pin to be driven when the IMR+ chip senses a SQE Test Error condition at the AUI port (attached MAU has SQE Test enabled).

Note that the HIMIB device will continue generating interrupts every time a packet is transmitted by the AUI port, while this condition exists and this interrupt is enabled.

D Port Read/Write

A	X	X	X	X	X	X	X
MSB							LSB

TP and AUI Source Address Change Interrupt Enable

These two registers are used to enable or mask interrupts caused by a change in the Source Address of a port. A TP port connected to another repeater or an AUI connected to a mixing (multiple DTEs) segment will have frequent source address changes.

A TP port connected to a single end station will only detect a change of address if the end station is physically changed to a different MAC address. The Last Source Address (LSA) register (in the Port Attribute Registers) of a port known to be connected to a single station can be programmed with the Node ID (48-bit MAC address) of the DTE. If the LSA is not programmed after power up it will be overwritten by the source address of the first packet received, and generate an interrupt if enabled.

TP Ports

P[4:0] = 2, R[4:0] = 6

D Port Read/Write

T7	T6	T5	T4	T3	T2	T1	T0
MSB							LSB

AUI Port

P[4:0] = 2, R[4:0] = 7

D Port Read/Write

A	X	X	X	X	X	X	X
MSB							LSB

The AUI port only uses the most significant bit (A) and all other bits are reserved. Software should be designed to write 0s into unused bits.

Port Attribute Registers

The Port Attribute Registers are accessed in the same fashion as the Repeater, Status or Control Registers by writing the appropriate Port Number and Register Number into the C Port. TP port number zero is accessed by writing 0001 0000, TP port number one by writing 00010001 and so on. The AUI port attributes are accessed by writing 00011111 to the C Port.

Except for the Last Source Address (LSA) register, all other registers are 4 bytes and read only. The (LSA) register is 6 bytes long and its contents can be written and read.

Once the C Port is programmed with a valid Port (Bank) and Register (Attribute) Number, the corresponding attribute is transferred to a holding register upon reading the first byte. Subsequent accesses to the D Port read the value in a least significant to most significant byte order. When reading, once the last byte is read, the attribute value is re-transferred to the holding register and the sequence can be restarted. When writing the LSA register, if the sequence is aborted prior to the sixth consecutive write cycle, the internally stored register value is not updated. The sequence (read or write) may be aborted and restarted by programming the C Port.

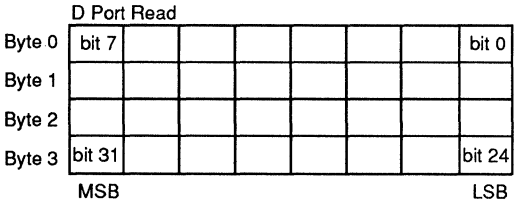
Note that the contents of all attribute registers are maintained during an external reset.

These attributes and their definitions comply with the IEEE 802.3k Layer Management for 10 Mbyte/s Baseband Repeaters Repeater Management Standard.

A brief summary of attribute description is included here for reference only. For detailed description, refer to the IEEE 802.3k document.

Readable Frames

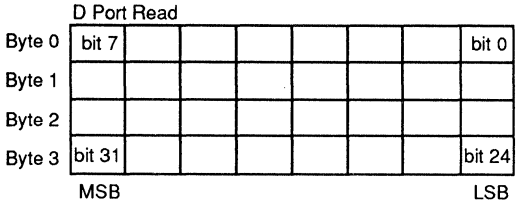
P[4:0] = 16–23, 31 , R[4:0] = 0



"Readable Frames" is a read-only attribute that counts the number of valid frames detected by the port. Valid frames are from 64 bytes to 1518 bytes in length, have a valid frame CRC and are received without a collision. This attribute is a 32-bit counter with a minimum rollover time of 80 hours.

Readable Octets

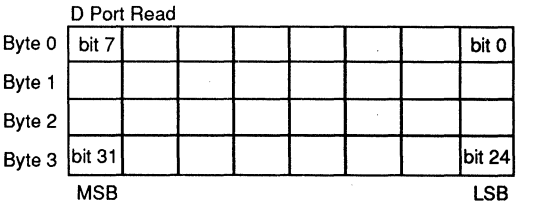
P[4:0] = 16–23, 31, R[4:0] = 1



"Readable Octets" is a read-only attribute that counts the number of octets received on each port. This number is determined by adding the frame length to this register at the completion of every valid frame. This attribute is a 32-bit counter with a minimum rollover time of 58 minutes.

Frame Check Sequence (FCS) Errors

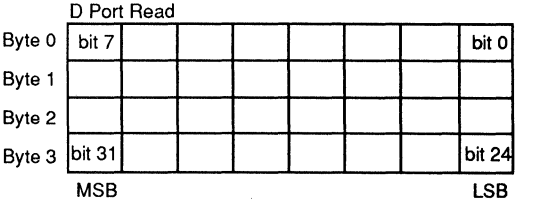
P[4:0] = 16–23, 31 , R[4:0] = 2



"Frame Check Sequence Errors" is a read-only attribute that counts the number of frames detected on each port with an invalid frame check sequence. This counter is incremented on each frame of valid length (64 bytes to 1518 bytes) that does not suffer a collision during the frame. This counter is incremented on each invalid frame, however it is not incremented for frames with both framing errors and frame check sequence errors. This attribute is a 32-bit counter with a minimum rollover time of 80 hours.

Alignment Errors

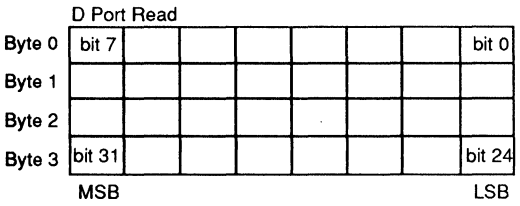
P[4:0] = 16–23, 31, R[4:0] = 3



"Alignment Errors" is a read-only attribute that counts the number of frames detected on each port with an FCS error and a framing error. This counter is incremented on each frame of valid length (64 bytes to 1518 bytes) that does not suffer a collision during the frame. Frames that have both framing errors and FCS errors are counted by this attribute, but not by the "Frame Check Sequence Errors" attribute. This attribute is a 32-bit counter with a minimum rollover time of 80 hours.

Frames Too Long

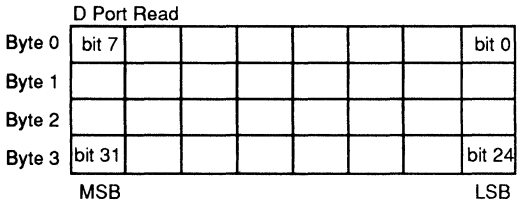
P[4:0] = 16–23, 31, R[4:0] = 4



"Frames Too Long" is a read-only attribute that counts the number of frames that exceed the maximum valid packet length of 1518 bytes. This attribute is a 32-bit counter with a minimum rollover time of 61 days.

Short Events

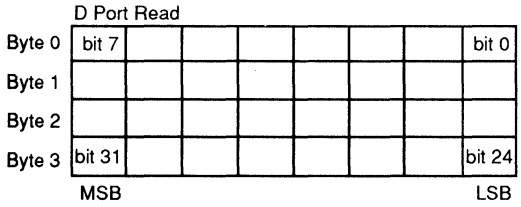
P[4:0] = 16–23, 31, R[4:0] = 5



"Short Events" is a read-only attribute that counts the number of instances where activity is detected with a duration less than the "ShortEventMaxTime" (74–82-bit times). This attribute is a 32-bit counter with a minimum rollover time of 16 hours.

Runs

P[4:0] = 16–23, 31, R[4:0] = 6

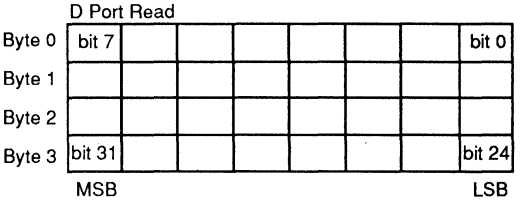


"Runts" is a read-only attribute that counts the number of instances where activity is detected with a duration greater than the "ShortEventMaxTime" (74–82-bit times), but less than the minimum valid frame time (512-bit times, or 64 bytes). This attribute is a 32-bit counter with a minimum rollover time of 16 hours.

Note: *Runts usually indicate collision fragments, a normal network event. In certain situation associated with large diameter networks a percentage of runts may exceed ValidPacketMinTime.*

Collisions

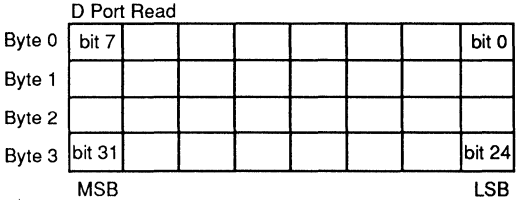
P[4:0] = 16–23, 31, R[4:0] = 7



"Collisions" is a read-only attribute that counts the number of instances where a carrier is detected on the port, and a collision is detected. This attribute is a 32-bit counter with a minimum rollover time of 16 hours.

Late Events

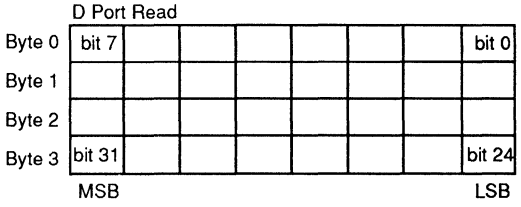
P[4:0] = 16–23, 31, R[4:0] = 8



"Late Events" is a read-only attribute that counts the number of instances where a collision is detected after the LateEventThreshold (480–565-bit times) in the frame. This event will be counted both by the "Late Events" attribute, as well as the "Collisions" attribute. This attribute is a 32-bit counter with a minimum rollover time of 81 hours.

Very Long Events

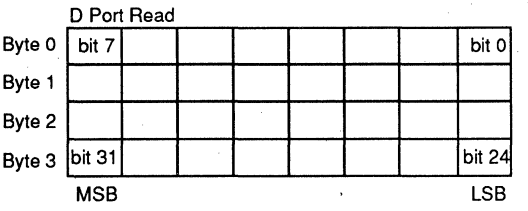
P[4:0] = 16–23, 31, R[4:0] = 9



"Very Long Events" is a read-only attribute that counts the number of times the transmitter is active in excess of the MAU Jabber Lockup Protection (MJLP) Timer (4 ms – 7.5 ms). This attribute is a 32-bit counter with a minimum rollover time of 198 days.

Data Rate Mismatches

P[4:0] = 16–23, 31, R[4:0] = 10

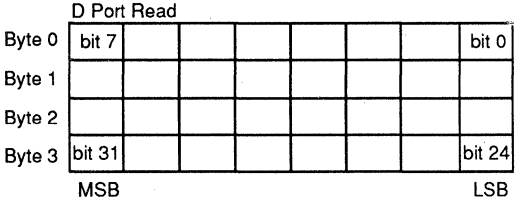


"Data Rate Mismatches" is a read-only attribute that counts the number of occurrences where the frequency, or data rate of the incoming signal is detectably different from the local transmit frequency. The attribute is a 32-bit counter that is incremented on each such event.

Note that the rate at which the "Data Rate Mismatches" attribute will increment, will depend on the magnitude of the difference between the received signal clock and the local transmit frequency.

Auto Partitions

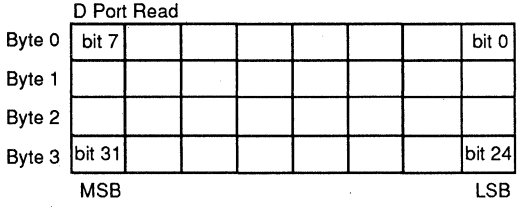
P[4:0] = 16–23, 31, R[4:0] = 11



"Auto Partitions" is a read-only attribute that counts the number of instances where the repeater has partitioned this port from the network. This attribute is a 32-bit counter that is incremented on each such event. The approximate minimum time between counter roll-overs is 20 days.

Source Address Changes

P[4:0] = 16–23, 31, R[4:0] = 12

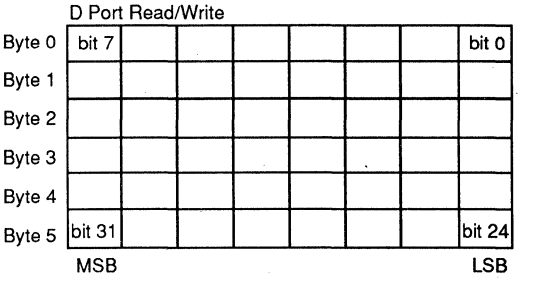


"Source Address Changes" is a read-only attribute that counts the number of times the Source Address field of valid frames received on a port changes. This attribute is a 32-bit counter with a minimum rollover of 81 hours.

Note: *This may indicate whether a port is connected to a single DTE or another multi-user segment.*

Last Source Address (LSA)

P[4:0] = 16–23, 31, R[4:0] = 14



"Last Source Address" is a read/write attribute that saves the value of the Source Address field of the last valid frame it received. This attribute is a 6-byte field.

This 6-byte register may be read or written. This feature allows the software to preset this attribute to the known Node ID, for a single node segment. A change in the contents of this register would then signal an anomaly. This will cause the Source Address Changes attribute to increment. Furthermore, setting the respective TP/AUI Port Source Address Change Interrupt Enable bit (in the Port Control Registers), can be used to generate a hardware interrupt to signal the software to automatically disable this port.

SYSTEMS APPLICATIONS

Typical System Interface

The block diagram on this page shows a typical system interface. A fully managed multiport repeater can be easily built by interfacing the HIMIB chip with the IMR+ chip (Am79C981). The HIMIB device interfaces with all common Microprocessor System Busses with a minimum of external logic. Note that additional buffering of DAT and JAM are required for most applications. For more information, refer to the AMD IEEE 802.3 Repeater Technical Manual.

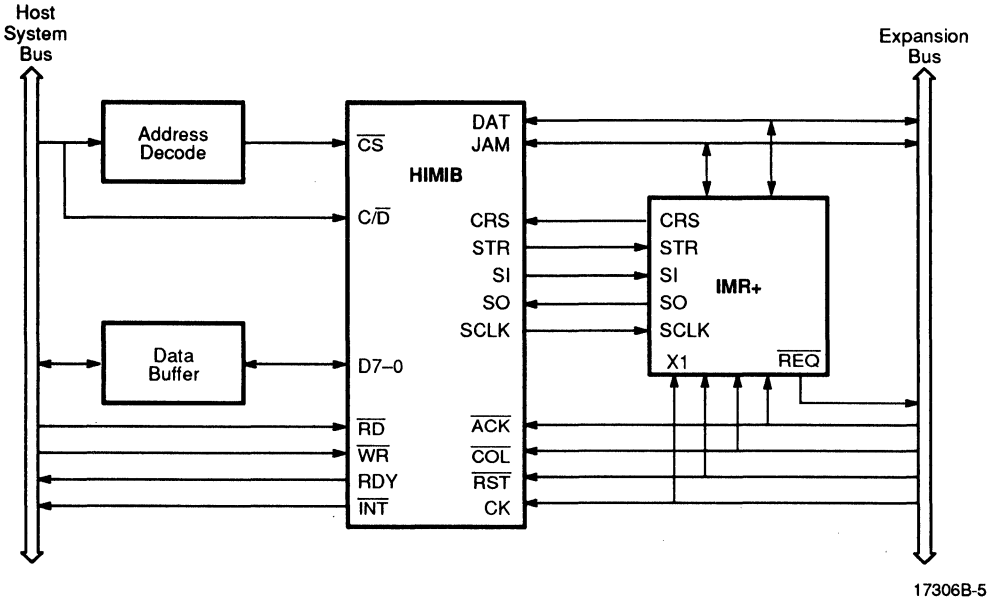


Figure 2. HIMIB Device Application Example

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C

Ambient Temperature
Under Bias ..... 0°C to +70°C

Supply Voltage ..... −0.3 V to +6.0 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Case Temperature (T_A) ..... 0°C to +70°C

Supply Voltages (V_{DD}) ..... 5 V ± 5%

All Inputs Within
the Range ..... V_{DD} + 0.5 V ≤ V_{IN} ≤ V_{SS} −0.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS

Symbol	Parameter Description	Test Condition	Min	Max	Unit
V_{IL}	Input LOW Voltage	V_{SS} = 0.0 V	−0.5	0.8	V
V_{IH}	Input HIGH Voltage		2.0	0.5 +V_{DD}	V
V_{OL}	Output LOW Voltage	I_{OL} = 4.0 mA		0.4	V
V_{OH}	Output High Voltage	I_{OH} = −0.4 mA	2.4		V
V_{OLOD}	Open Drain Output Low Voltage	I_{OLOD} = 12 mA		0.4	V
I_{IL}	Input Leakage Current	0 < V_{IN} and V_{IN} < V_{DD} +0.5 V		10	µA
V_{ILX}	CK Input LOW Voltage	V_{SS} = 0.0 V	−0.5	1.0	V
V_{IHX}	CK Input HIGH Voltage	V_{SS} = 0.0 V	3.8	0.5 +V_{DD}	V
I_{ILX}	CK Input LOW Current	V_{IN} = V_{SS}		10	µA
I_{IHX}	CK Input HIGH Current	V_{IN} = V_{DD}		10	µA
I_{DD}	Power Supply Current	f_{CK} = 20 MHz		40	mA

SWITCHING CHARACTERISTICS

Clock and Reset Timing					
Symbol	**Description**	**Test Condition**	**Min**	**Max**	**Unit**
t_{CK}	Clock Period		49.995	50.005	ns
t_{CKH}	Clock High		20	30	ns
t_{CKL}	Clock Low		20	30	ns
t_{CKR}	Clock Rise Time			10	ns
t_{CKF}	Clock Fall Time			10	ns
t_{RST}	Reset Pulse Width	(Note 1)	4		us
t_{RSTS}	Reset Input Setup Time with Respect to CK	(Note 1)	15		ns
t_{RSTH}	Reset Input Hold Time with Respect to CK	(Note 1)	0		ns
Expansion Port					
Symbol	**Description**	**Test Condition**	**Min**	**Max**	**Unit**
t_{DJSET}	DAT/JAM Setup Time		10		ns
t_{DJHOLD}	DAT/JAM Hold Time		9		ns
t_{CASET}	$\overline{COL}/\overline{ACK}$ Setup Time		5		ns
t_{CAHLD}	$\overline{COL}/\overline{ACK}$ Hold Time		9		ns
Management Port					
Symbol	**Description**	**Test Condition**	**Min**	**Max**	**Unit**
t_{SCKD}	SCLK Clock Delay with Respect to CK		9	45	ns
t_{SCKR}	SCLK Rise Time with Respect to CK	C_L = 50 pF		10	ns
t_{SCKF}	SCLK Fall Time with Respect to CK	C_L = 50 pF		10	ns
t_{SOS}	SO Input Setup Time with Respect to CK Rising Edge		10		ns
t_{SOH}	SO Input Hold Time with Respect to CK Rising Edge		9		ns
t_{SID}	SI Output Delay with Respect to CK Rising Edge	C_L = 50 pF	9	45	ns
Port Activity Monitor					
Symbol	**Description**	**Test Condition**	**Min**	**Max**	**Unit**
t_{CRSTS}	CRS Setup Time with Respect to CK Rising Edge		10		ns
t_{CRSTH}	CRS Hold Time with Respect to CK Rising Edge		5		ns

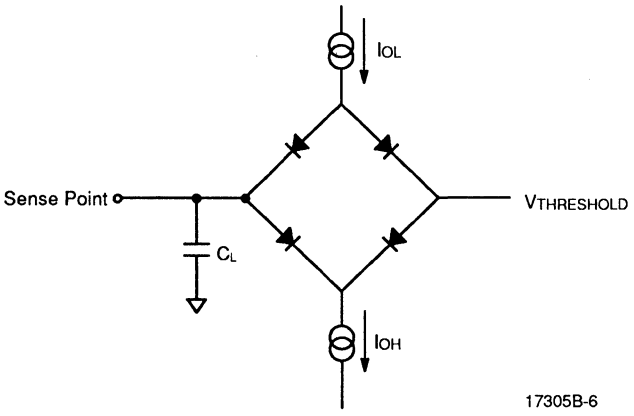
Note:

1. See IMR+ data sheet for reset.

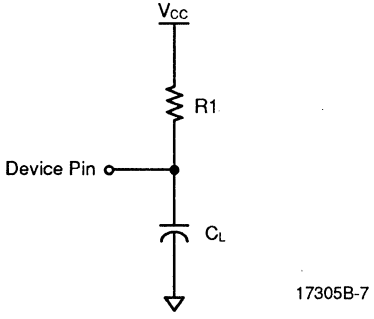
SWITCHING CHARACTERISTICS (continued)

Microprocessor Interface (MPI)					
Symbol	**Description**	**Test Condition**	**Min**	**Max**	**Unit**
t_{CDS}	C/$\overline{D}$ Setup Time with Respect to $\overline{RD}$/$\overline{WR}$ Falling Edge		10		ns
t_{CDH}	C/$\overline{D}$ Hold Time with Respect to $\overline{RD}$/$\overline{WR}$ Rising Edge		0		ns
t_{CSS}	$\overline{CS}$ Setup Time with Respect to $\overline{RD}$/$\overline{WR}$ Falling Edge		10		ns
t_{CSH}	$\overline{CS}$ Hold Time with Respect to $\overline{RD}$/$\overline{WR}$ Rising Edge		0		ns
t_{REST}	Rest Period between MPI Operations (Time between the Earliest $\overline{CS}$/$\overline{RD}$/$\overline{WR}$ Going HIGH to the Next $\overline{CS}$/$\overline{RD}$/$\overline{WR}$ Going LOW, whichever is the Latest)		150		ns
t_{RDYD}	RDY Leading Edge Delay	C_L = 100 pF		25	ns
t_{RDYH}	RDY High to $\overline{RD}$/$\overline{WR}$ ↑		0		ns
t_{DOUT}	Data Out Valid to RDY High	C_L = 100 pF	50		ns
t_{DOHLD}	Data Out Hold after $\overline{RD}$ High	C_L = 100 pF	10	45	ns
t_{DISET}	Data In Setup Time with Respect to $\overline{WR}$ Rising Edge		25		ns
t_{DIHLD}	Data in Hold after $\overline{WR}$ High		0		ns

SWITCHING TEST LOADS



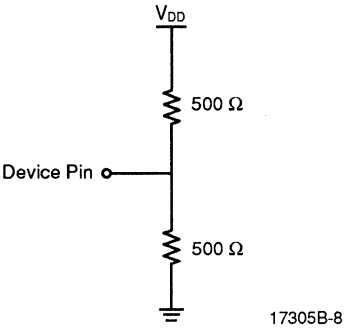
A. Normal and Three-State Outputs



B. Open-Drain Outputs ($\overline{RDY}$, $\overline{INT}$)

Test Output Loads			
Pin Name	**Test Circuit**	**R1**	**C_L (pF)**
All Outputs and I/O Pins except $\overline{RDY}$, $\overline{INT}$	A		100
$\overline{RDY}$, $\overline{INT}$	B	400	100

SWITCHING TEST LOADS (continued)



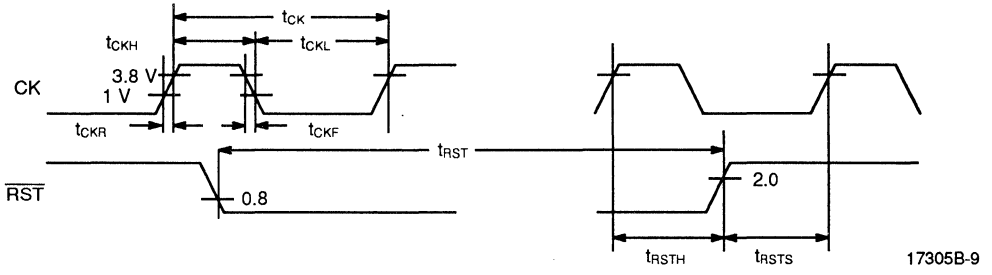
C. For Data Out (D7–0) Hold Only

KEY TO SWITCHING WAVEFORMS

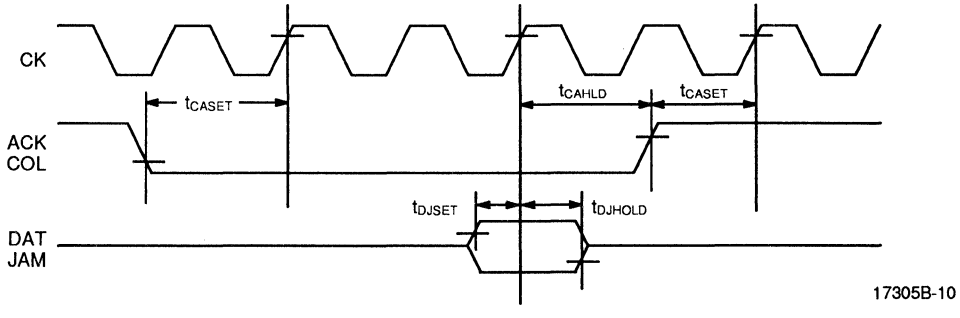
WAVEFORM	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off" State

KS000010

SWITCHING WAVEFORMS

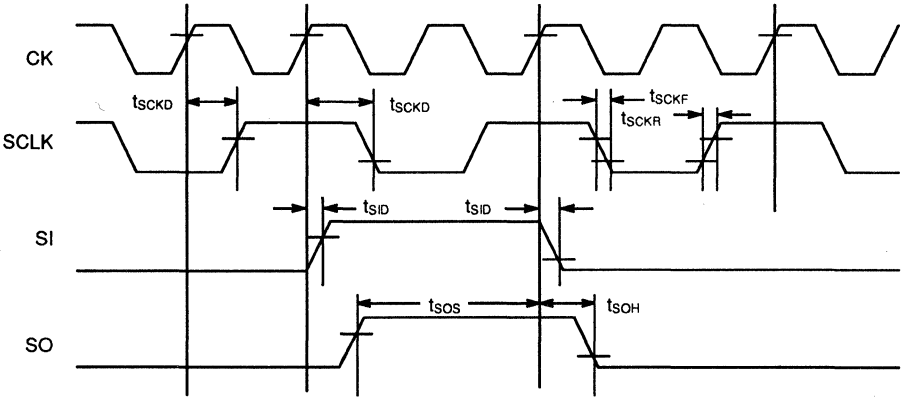


Clock and Reset Timing

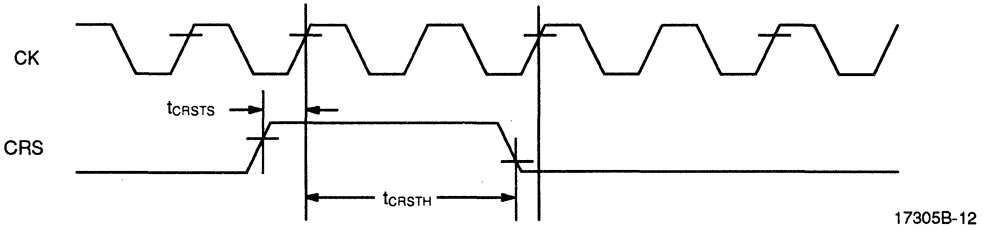


Expansion Port Timing

SWITCHING WAVEFORMS

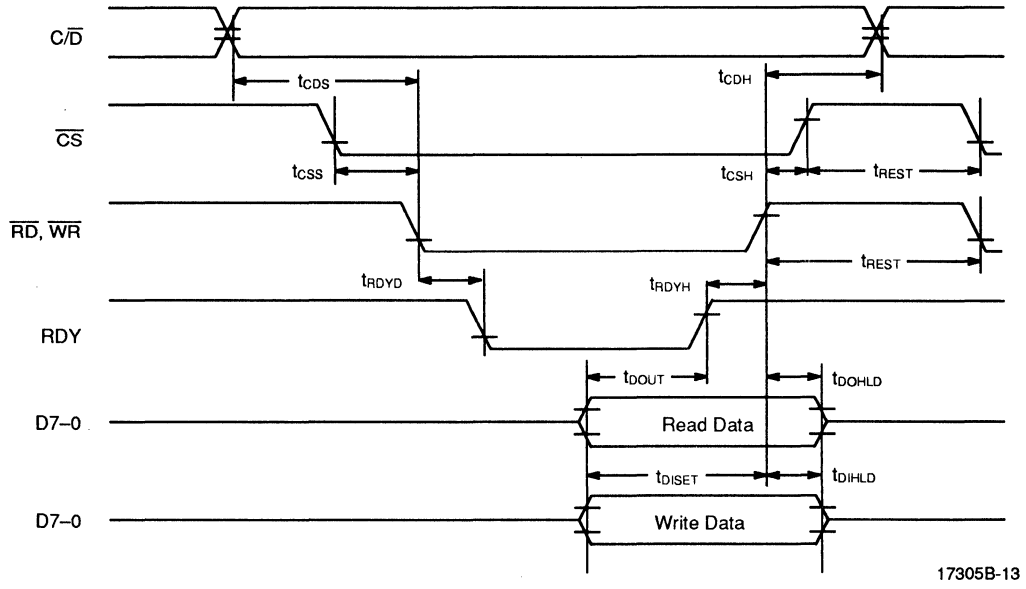


Management Port Timing



Port Activity Timing

SWITCHING WAVEFORMS



Bus Interface Timing

Note:

Refer to AMD's IEEE 802.3 Repeater Technical Manual (PID #17314A) for more detailed access timing.

APPENDIX A

IMR+/HIMIB Security Features

The Am79C981 Integrated Multiport Repeater Plus (IMR+) and the Am79C987 Hardware Implemented Management Information Base (HIMIB) Ethernet repeater chip-set is capable of providing physical network security features. AMD will only make these features available to customers who are under an IMR+/HIMIB security non disclosure agreement (NDA). A description of the security feature is summarized below. For more information, contact your local AMD sales office to generate an IMR+/HIMIB security NDA.

Security Features Summary

The HIMIB incorporates a feature to allow the destination address (DA) field of a received packet to be compared with the known MAC address connected to each port. The MAC address for each port is contained in the HIMIB Last Source Address (LSA) register, which can be programmed by the user or it will be "learnt" by the HIMIB device. On receipt of a packet on one port, all other ports have the contents of the LSA register compared with the DA field of the received frame. If there is a match on any port, the frame is repeated to that port normally. For those ports which have the security feature enabled and do not have a DA/LSA match, the repeated bit stream of the packet will be corrupted (frequently termed "eavesdrop protection"), and the port will transmit an alternating pattern of 1 and 0 following the 18th (approximate) bit of the Source Address field. This feature can be enabled/disabled on a port by port basis using a mask located in the HIMIB Port Control Registers. Any port with the security feature disabled (using the field in the Port Controls Registers) will repeat the packet normally. Note that multicast and broadcast packets are transmitted to all ports unmodified, regardless of the enable/disable state of the security function.

Ports that are connected to single stations can be secured by enabling the eavesdrop protection function and enabling the Last Source Address Change Interrupt. This prevents unauthorized eavesdropping by stations on the LAN who are not directly addressed by the sourcing node, hence the learning of valid source addresses and "snooping" on data is virtually impossible. In addition, this allows the management software to detect and possibly disable the port in real time if the HIMIB indicates via the hardware interrupt line that the Source Address has changed.

PRELIMINARY

Am79C960

PCnet™-ISA Single-Chip Ethernet Controller

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Single-chip Ethernet controller for the Industry Standard Architecture (ISA) and Extended Industry Standard Architecture (EISA) buses**
- **Supports IEEE 802.3/ANSI 8802-3 and Ethernet standards**
- **Direct interface to the ISA or EISA bus**
- **Software compatible with AMD's Am7990 LANCE register and descriptor architecture**
- **Low power, CMOS design with sleep mode allows reduced power consumption for critical battery powered applications**
- **Individual 136-byte transmit and 128-byte receive FIFOs provide packet buffering for increased system latency, and support the following features:**
  - — Automatic retransmission with no FIFO reload
  - — Automatic receive stripping and transmit padding (individually programmable)
  - — Automatic runt packet rejection
  - — Automatic deletion of received collision frames
- **Dynamic transmit FCS generation programmable on a frame-by-frame basis**
- **Single +5 V power supply**
- **Internal/external loopback capabilities**
- **Supports optional Boot PROM for diskless node applications**
- **Provides integrated Attachment Unit Interface (AUI) and 10BASE-T transceiver with 3 modes of port selection:**
  - — Automatic selection of AUI or 10BASE-T
  - — Software selection of AUI or 10BASE-T
  - — Jumper selection of AUI or 10BASE-T
- **Automatic Twisted Pair receive polarity detection and automatic correction of the receive polarity**
- **Supports bus-master and shared-memory architectures to fit in any PC application**
- **Supports edge and level-sensitive interrupts**
- **DMA Buffer Management Unit for reduced CPU intervention**
- **Integral DMA controller allows higher throughput by by-passing the platform DMA**
- **JTAG Boundary Scan (IEEE 1149.1) test access port interface for board level production test**
- **Integrated Manchester Encoder/Decoder**
- **Supports the following types of network interfaces:**
  - — AUI to external 10BASE2, 10BASE5, 10BASE-T or 10BASE-F MAU
  - — Internal 10BASE-T transceiver with Smart Squelch to Twisted Pair medium
- **Supports LANCE General Purpose Serial Interface (GPSI)**
- **120-pin PQFP package**

GENERAL DESCRIPTION

The PCnet-ISA controller, a single-chip Ethernet controller, is a highly integrated system solution for the PC-AT Industry Standard Architecture (ISA) architecture. It is designed to provide flexibility and compatibility with any existing PC application. This highly integrated 120-pin VLSI device is specifically designed to reduce parts count and cost, and addresses applications where higher system throughput is desired. The PCnet-ISA controller is fabricated with AMD's advanced low-power CMOS process to provide low stand by current for power sensitive applications.

The PCnet-ISA controller is a DMA-based device with a dual architecture that can be configured in two different operating modes to suit a particular PC application. In the Bus Master Mode all transfers are performed using the integrated DMA controller. This configuration enhances system performance by allowing the PCnet-ISA controller to bypass the platform DMA controller and directly address the full 24-bit memory space. The implementation of Bus Master Mode allows minimum parts count for the majority of PC applications. The PCnet-ISA controller can be configured to perform Shared Memory operations for compatibility with low-end machines, such as PC/XTs that do not support Bus Master and high-end machines that require local packet buffering for increased system latency.

Publication# 16907 Rev. B Amendment /0
Issue Date: May 1994

The PCnet-ISA controller is designed to directly interface with the ISA or EISA system bus. It contains an ISA bus interface unit, DMA Buffer Management Unit, IEEE 802.3 Media Access Control function, individual 136-byte transmit and 128-byte receive FIFOs, IEEE 802.3 defined Attachment Unit Interface (AUI), and a Twisted Pair Transceiver Media Attachment Unit. The PCnet-ISA controller is also register compatible with the LANCE (Am7990) Ethernet controller. The DMA Buffer Management Unit supports the LANCE descriptor software model. External remote boot and Ethernet physical address PROMs are also supported.

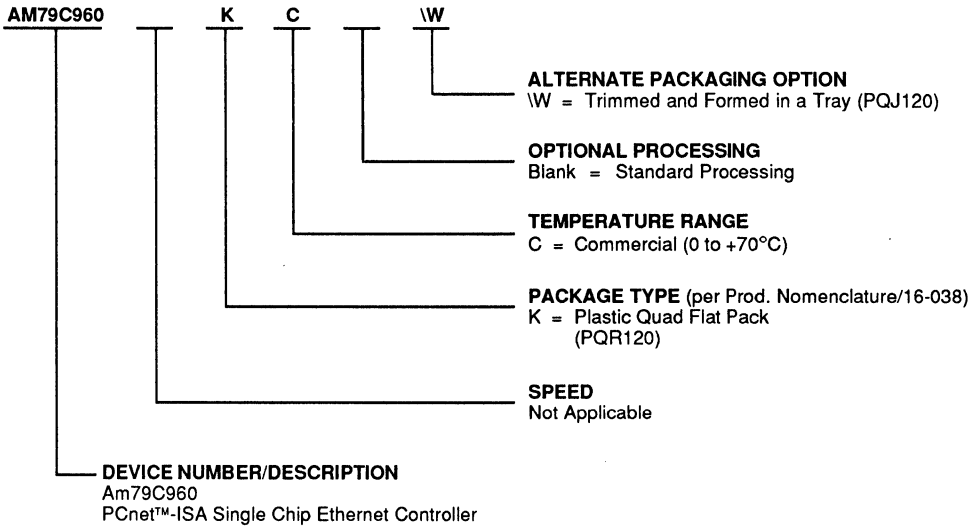
This advanced Ethernet controller has the built-in capability of automatically selecting either the AUI port or the Twisted Pair transceiver. Only one interface is active at any one time. The individual 136-byte transmit and 128-byte receive FIFOs optimize system overhead, providing sufficient latency during packet transmission and reception, and minimizing intervention during normal network error recovery. The integrated Manchester encoder/decoder eliminates the need for an external Serial Interface Adapter (SIA) in the node system. If support for an external encoding/decoding scheme is desired, the embedded General Purpose Serial Interface (GPSI) allows direct access to/from the MAC. In addition, the device provides programmable on-chip LED drivers for transmit, receive, collision, receive polarity, link integrity, or jabber status. The PCnet-ISA controller also provides an External Address Detection Interface™ (EADI™) to allow external hardware address filtering in internetworking applications.

RELATED PRODUCTS

Part No.	Description
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted Pair Ethernet Transceiver Plus (TPEX+)
Am7996	IEEE 802.3/Ethernet/Cheapernet Transceiver
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)
Am79C961	PCnet-ISA+ Single-Chip Ethernet Controller for ISA (with Microsoft® Plug n' Play® Support)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



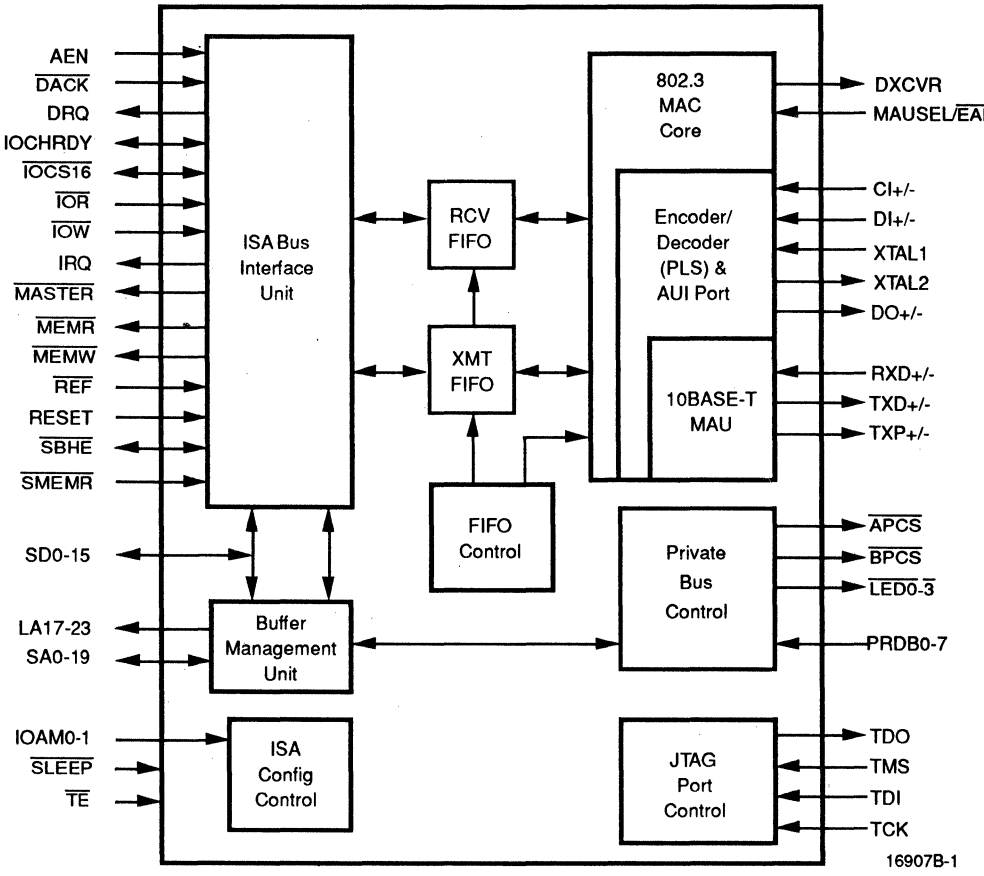
Valid Combinations	
AM79C960	KC, KC\W

Valid Combinations

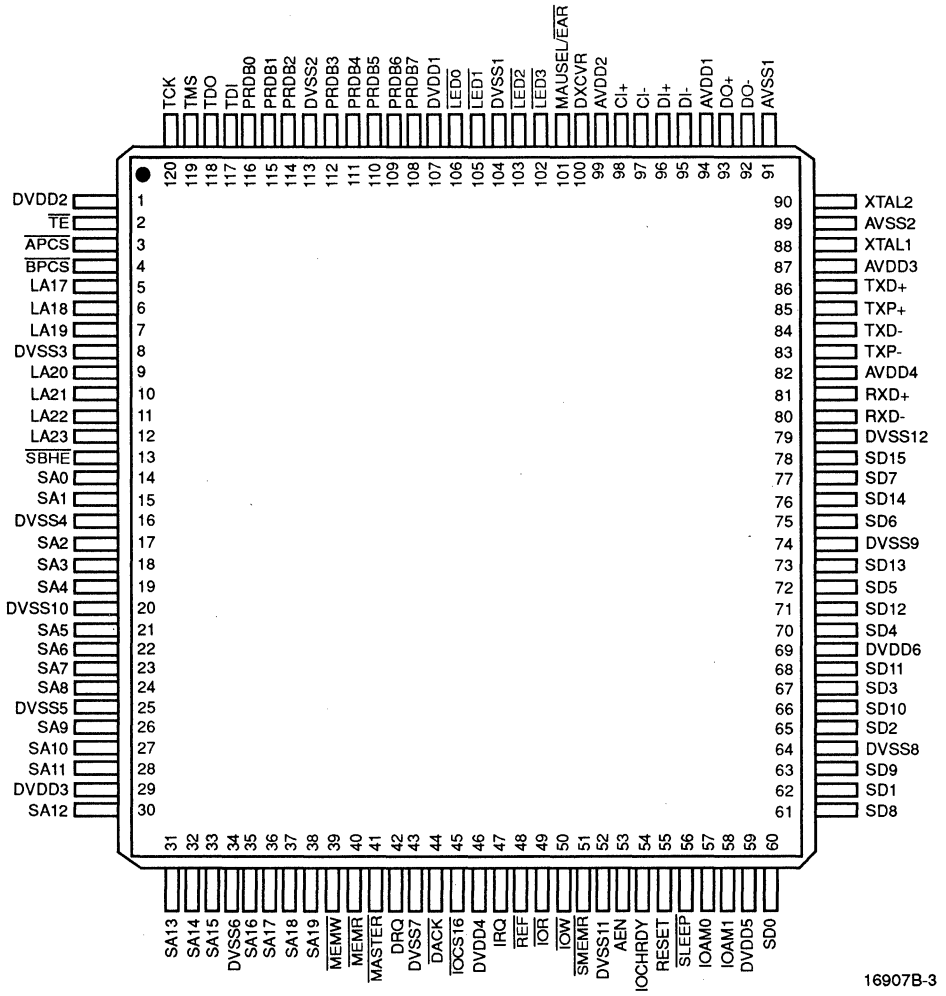
The Valid Combinations table lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

TABLE OF CONTENTS

BLOCK DIAGRAM: BUS MASTER MODE



CONNECTION DIAGRAM: BUS MASTER



16907B-3

PIN DESIGNATIONS: BUS MASTER
Listed by Pin Number

Pin #	Name	Pin #	Name	Pin #	Name	Pin #	Name
1	DVDD2	31	SA13	61	SD8	91	AVSS1
2	$\overline{TE}$	32	SA14	62	SD1	92	DO–
3	$\overline{APCS}$	33	SA15	63	SD9	93	DO+
4	$\overline{BPCS}$	34	DVSS6	64	DVSS8	94	AVDD1
5	LA17	35	SA16	65	SD2	95	DI–
6	LA18	36	SA17	66	SD10	96	DI+
7	LA19	37	SA18	67	SD3	97	CI–
8	DVSS3	38	SA19	68	SD11	98	CI+
9	LA20	39	$\overline{MEMW}$	69	DVDD6	99	AVDD2
10	LA21	40	$\overline{MEMR}$	70	SD4	100	DXCVR
11	LA22	41	$\overline{MASTER}$	71	SD12	101	MAUSEL/$\overline{EAR}$
12	LA23	42	DRQ	72	SD5	102	$\overline{LED3}$
13	$\overline{SBHE}$	43	DVSS7	73	SD13	103	$\overline{LED2}$
14	SA0	44	$\overline{DACK}$	74	DVSS9	104	DVSS1
15	SA1	45	$\overline{IOCS16}$	75	SD6	105	$\overline{LED1}$
16	DVSS4	46	DVDD4	76	SD14	106	$\overline{LED0}$
17	SA2	47	IRQ	77	SD7	107	DVDD1
18	SA3	48	$\overline{REF}$	78	SD15	108	PRDB7
19	SA4	49	$\overline{IOR}$	79	DVSS12	109	PRDB6
20	DVSS10	50	$\overline{IOW}$	80	RXD–	110	PRDB5
21	SA5	51	$\overline{SMEMR}$	81	RXD+	111	PRDB4
22	SA6	52	DVSS11	82	AVDD4	112	PRDB3
23	SA7	53	AEN	83	TXP–	113	DVSS2
24	SA8	54	IOCHRDY	84	TXD–	114	PRDB2
25	DVSS5	55	RESET	85	TXP+	115	PRDB1
26	SA9	56	$\overline{SLEEP}$	86	TXD+	116	PRDB0
27	SA10	57	IOAM0	87	AVDD3	117	TDI
28	SA11	58	IOAM1	88	XTAL1	118	TDO
29	DVDD3	59	DVDD5	89	AVSS2	119	TMS
30	SA12	60	SD0	90	XTAL2	120	TCK

PIN DESIGNATIONS: BUS MASTER
Listed by Pin Name

Name	Pin #	Name	Pin #	Name	Pin #	Name	Pin #
AEN	53	DVSS8	64	PRDB2	114	$\overline{\text{SBHE}}$	13
$\overline{\text{APCS}}$	3	DVSS9	74	PRDB3	112	SD0	60
AVDD1	94	DVSS10	20	PRDB4	111	SD1	62
AVDD2	99	DVSS11	52	PRDB5	110	SD2	65
AVDD3	87	DVSS12	79	PRDB6	109	SD3	67
AVDD4	82	DXCVR	100	PRDB7	108	SD4	70
AVSS1	91	IOAM0	57	$\overline{\text{REF}}$	48	SD5	72
AVSS2	89	IOAM1	58	RESET	55	SD6	75
$\overline{\text{BPCS}}$	4	IOCHRDY	54	RXD–	80	SD7	77
CI–	97	$\overline{\text{IOCS16}}$	45	RXD+	81	SD8	61
CI+	98	$\overline{\text{IOR}}$	49	SA0	14	SD9	63
$\overline{\text{DACK}}$	44	$\overline{\text{IOW}}$	50	SA1	15	SD10	66
DI–	95	IRQ	47	SA2	17	SD11	68
DI+	96	LA17	5	SA3	18	SD12	71
DO–	92	LA18	6	SA4	19	SD13	73
DO+	93	LA19	7	SA5	21	SD14	76
DRQ	42	LA20	9	SA6	22	SD15	78
DVDD1	107	LA21	10	SA7	23	$\overline{\text{SLEEP}}$	56
DVDD2	1	LA22	11	SA8	24	$\overline{\text{SMEMR}}$	51
DVDD3	29	LA23	12	SA9	26	TCK	120
DVDD4	46	$\overline{\text{LED0}}$	106	SA10	27	TDI	117
DVDD5	59	$\overline{\text{LED1}}$	105	SA11	28	TDO	118
DVDD6	69	$\overline{\text{LED2}}$	103	SA12	30	$\overline{\text{TE}}$	2
DVSS1	104	$\overline{\text{LED3}}$	102	SA13	31	TMS	119
DVSS2	113	$\overline{\text{MASTER}}$	41	SA14	32	TXD–	84
DVSS3	8	MAUSEL/$\overline{\text{EAR}}$	101	SA15	33	TXD+	86
DVSS4	16	$\overline{\text{MEMR}}$	40	SA16	35	TXP–	83
DVSS5	25	$\overline{\text{MEMW}}$	39	SA17	36	TXP+	85
DVSS6	34	PRDB0	116	SA18	37	XTAL1	88
DVSS7	43	PRDB1	115	SA19	38	XTAL2	90

PIN DESIGNATIONS: BUS MASTER
Listed by Group

Pin Name	Pin Function	I/O	Driver
ISA Bus Interface			
AEN	Address Enable	I	
$\overline{\text{DACK}}$	DMA Acknowledge	I	
DRQ	DMA Request	O	TS3
IOCHRDY	I/O Channel Ready	I/O	OD3
$\overline{\text{IOCS16}}$	I/O Chip Select 16	I/O	OD3
$\overline{\text{IOR}}$	I/O Read Select	I	
$\overline{\text{IOW}}$	I/O Write Select	I	
IRQ	Interrupt Request	O	OD3
LA17-23	Unlatched Address Bus	O	TS3
$\overline{\text{MASTER}}$	Master Transfer in Progress	O	OC3
$\overline{\text{MEMR}}$	Memory Read Select	O	TS3
$\overline{\text{MEMW}}$	Memory Write Select	O	TS3
$\overline{\text{REF}}$	Memory Refresh Active	I	
RESET	System Reset	I	
SA0-19	System Address Bus	I/O	TS3
$\overline{\text{SBHE}}$	System Byte High Enable	I/O	TS3
SD0-15	System Data Bus	I/O	TS3
$\overline{\text{SMEMR}}$	System Memory Read Select	I	
Board Interfaces			
$\overline{\text{APCS}}$	Address PROM Chip Select	O	TS1
$\overline{\text{BPCS}}$	Boot PROM Chip Select	O	TS1
DXCVR	Disable Transceiver	O	TS1
IOAM0-1	Input/Output Address Map	I	
$\overline{\text{LED0}}$	LED0/LNKST	O	TS2
$\overline{\text{LED1}}$	LED1/SFBD/RCVACT	O	TS2
$\overline{\text{LED2}}$	LED2/SRD/RXPOL	O	TS2
$\overline{\text{LED3}}$	LED3/SRDCLK/XMTACT	O	TS2
MAUSEL/$\overline{\text{EAR}}$	MAU SELect/External Address Reject	I	
PRDB0-7	PROM Data Bus	I	
$\overline{\text{SLEEP}}$	Sleep Mode	I	
$\overline{\text{TE}}$	Test Enable	I	
XTAL1	Crystal Input	I	
XTAL2	Crystal Output	O	

PIN DESIGNATIONS: BUS MASTER (continued)
Listed by Group

Pin Name	Pin Function	I/O	Driver
Attachment Unit Interface (AUI)			
CI±	Collision Inputs	I	
DI±	Receive Data	I	
DO±	Transmit Data	O	
Twisted Pair Transceiver Interface (10BASE-T)			
RXD±	10BASE-T Receive Data	I	
TXD±	10BASE-T Transmit Data	O	
TXP±	10BASE-T Predistortion Control	O	
IEEE 1149.1 Test Access Port Interface (JTAG)			
TCK	Test Clock	I	
TDI	Test Data Input	I	
TDO	Test Data Output	O	TS2
TMS	Test Mode Select	I	
Power Supplies			
AVDD	Analog Power		
AVSS	Analog Ground		
DVDD	Digital Power		
DVSS	Digital Ground		

Table: Output Driver Types

Name	Type	Iol (mA)	Ioh (mA)	pF
TS1	Tri-State	4	–1	50
TS2	Tri-State	12	–4	50
TS3	Tri-State	24	–3	120
OD3	Open Drain	24	–3	120

PIN DESCRIPTION: BUS MASTER MODE

These pins are part of the bus master mode. In order to understand the pin descriptions, definition of some terms from a draft of IEEE P996 are included.

IEEE P996 Terminology

Alternate Master: Any device that can take control of the bus through assertion of the MASTER signal. It has the ability to generate addresses and bus control signals in order to perform bus operations. All Alternate Masters must be 16 bit devices and drive SBHE.

Bus Ownership: The Current Master possesses bus ownership and can assert any bus control, address and data lines.

Current Master: The Permanent Master, Temporary Master or Alternate Master which currently has ownership of the bus.

Permanent Master: Each P996 bus will have a device known as the Permanent Master that provides certain signals and bus control functions as described in Section 3.5 (of the IEEE P996 spec), "Permanent Master". The Permanent Master function can reside on a Bus Adapter or on the backplane itself.

Temporary Master: A device that is capable of generating a DMA request to obtain control of the bus and directly asserting only the memory and I/O strobes during bus transfer. Addresses are generated by the DMA device on the Permanent Master.

ISA Interface

AEN
Address Enable *Input*

This signal must be driven LOW when the bus performs an I/O access to the device.

DACK
DMA Acknowledge *Input*

Asserted LOW when the Permanent Master acknowledges a DMA request. When DACK is asserted the PCnet-ISA controller becomes the Current Master by asserting the MASTER signal.

DRQ
DMA Request *Output*

When the PCnet-ISA controller needs to perform a DMA transfer, it asserts DRQ. The Permanent Master acknowledges DRQ with assertion of DACK. When the PCnet-ISA controller does not need the bus it deasserts DRQ.

IOCHRDY
I/O Channel Ready *Input/Output*

When the PCnet-ISA controller is being accessed, IOCHRDY HIGH indicates that valid data exists on the data bus for reads and that data has been latched for writes. When the PCnet-ISA controller is the Current Master on the ISA bus, it extends the bus cycle as long as IOCHRDY is LOW.

IOCS16
I/O Chip Select 16 *Input/Output*

When an I/O read or write operation is performed, the PCnet-ISA controller will drive the IOCS16 pin LOW to indicate that the chip supports a 16-bit operation at this address. (If the motherboard does not receive this signal, then the motherboard will convert a 16-bit access to two 8-bit accesses.) The IOCS16 pin is also an input and must go HIGH at least once after reset for the PCnet-ISA controller to perform 16-bit I/O operations. If this pin is grounded then the PCnet-ISA controller only performs 8-bit I/O operations.

The PCnet-ISA controller follows the IEEE P996 specification that recommends this function be implemented as a pure decode of SA0-9 and AEN, with no dependency on SMEMR, MEMR, MEMW, IOR, or IOW; however, some PC/AT clone systems are not compatible with this approach. For this reason, the PCnet-ISA controller is recommended to be configured to run 8-bit I/O on all machines. Since data is moved by memory cycles there is virtually no performance loss incurred by running 8-bit I/O and compatibility problems are virtually eliminated. The PCnet-ISA controller can be configured to run 8-bit-only I/O by disconnecting the IOCS16 pin from the ISA bus and tying the IOCS16 pin to ground instead.

IOR
I/O Read *Input*

IOR is driven LOW by the host to indicate that an Input/Output Read operation is taking place. IOR is only valid if the AEN signal is LOW and the external address matches the PCnet-ISA controller's predefined I/O address location. If valid, IOR indicates that a slave read operation is to be performed.

IOW
I/O Write *Input*

IOW is driven LOW by the host to indicate that an Input/Output Write operation is taking place. IOW is only valid if AEN signal is LOW and the external address matches the PCnet-ISA controller's predefined I/O address location. If valid, IOW indicates that a slave write operation is to be performed.

IRQ
Interrupt Request *Output*

An attention signal which indicates that one or more of the following status flags is set: BABL, MISS, MERR, RINT, IDON, RCVCCO, JAB, MFCO, or TXSTRT. All status flags have a mask bit which allows for suppression of INTR assertion. These flags have the following meaning:

BABL	Babble
RCVCCO	Receive Collision Count Overflow
JAB	Jabber
MISS	Missed Frame
MERR	Memory Error
MFCO	Missed Frame Count Overflow
RINT	Receive Interrupt
IDON	Initialization Done
TXSTRT	Transmit Start

LA17-23
Unlatched Address Bus *Output*

The unlatched address bus is driven by the PCnet-ISA controller during bus master cycle.

The functions of these unlatched address pins will change when GPSI mode is invoked. The table below shows the pin configuration in GPSI mode. Please refer to the section on General Purpose Serial Interface for detailed information on accessing this mode.

Pin Number	Pin Function in Bus Master Mode	Pin Function in GPSI Mode
5	LA17	RXDAT
6	LA18	SRDCLK
7	LA19	RXCRS
9	LA20	CLSN
10	LA21	STDCLK
11	LA22	TXEN
12	LA23	TXDAT

$\overline{\text{MASTER}}$
Master Mode *Output*

This signal indicates that the PCnet-ISA controller has become the Current Master of the ISA bus. After the PCnet-ISA controller has received a DMA Acknowledge ($\overline{\text{DACK}}$) in response to a DMA Request (DRQ), the Ethernet controller asserts the $\overline{\text{MASTER}}$ signal to indicate to the Permanent Master that the PCnet-ISA controller is becoming the Current Master.

$\overline{\text{MEMR}}$
Memory Read *Output*

$\overline{\text{MEMR}}$ goes LOW to perform a memory read operation.

$\overline{\text{MEMW}}$
Memory Write *Output*

$\overline{\text{MEMW}}$ goes LOW to perform a memory write operation.

$\overline{\text{REF}}$
Memory Refresh *Input*

When $\overline{\text{REF}}$ is asserted, a memory refresh is active. The PCnet-ISA controller uses this signal to mask inadvertent DMA Acknowledge assertion during memory refresh periods. If $\overline{\text{DACK}}$ is asserted when $\overline{\text{REF}}$ is active, $\overline{\text{DACK}}$ assertion is ignored. $\overline{\text{REF}}$ is monitored to eliminate a bus arbitration problem observed on some ISA platforms.

RESET
Reset *Input*

When RESET is asserted HIGH the PCnet-ISA controller performs an internal system reset. RESET must be held for a minimum of 10 XTAL1 periods before being deasserted. While in a reset state, the PCnet-ISA controller will tristate or deassert all outputs to predefined reset levels. The PCnet-ISA controller resets itself upon power-up.

SA0-19
System Address Bus *Input/Output*

This bus contains address information, which is stable during a bus operation, regardless of the source. SA17-19 contain the same values as the unlatched address LA17-19. When the PCnet-ISA controller is the Current Master, SA0-19 will be driven actively. When the PCnet-ISA controller is not the Current Master, the SA0-19 lines are continuously monitored to determine if an address match exists for I/O slave transfers or Boot PROM accesses.

$\overline{\text{SBHE}}$
System Byte High Enable *Input/Output*

This signal indicates the high byte of the system data bus is to be used. $\overline{\text{SBHE}}$ is driven by the PCnet-ISA controller when performing bus mastering operations.

SD0-15
System Data Bus *Input/Output*

These pins are used to transfer data to and from the PCnet-ISA controller to system resources via the ISA data bus. SD0-15 is driven by the PCnet-ISA controller when performing bus master writes and slave read operations. Likewise, the data on SD0-15 is latched by the PCnet-ISA controller when performing bus master reads and slave write operations.

$\overline{\text{SMEMR}}$
System Memory Read *Input*

This pin is used during Boot PROM access. The Boot PROM can be disabled by not connecting this pin.

Board Interface

$\overline{\text{APCS}}$
Address PROM Chip Select *Output*

This signal is asserted when the external Address PROM is read. When an I/O read operation is performed on the first 16 bytes in the PCnet-ISA controller's I/O space, $\overline{\text{APCS}}$ is asserted. The outputs of the external Address PROM drive the PROM Data Bus. The PCnet-ISA controller buffers the contents of the PROM data bus and drives them on the lower eight bits of the System Data Bus.

BPCS
Boot PROM Chip Select *Output*

This signal is asserted when the Boot PROM is read. If SA0-19 lines match a predefined address block and SMEMR is active and REF inactive, the BPCS signal will be asserted. The outputs of the external Boot PROM drive the PROM Data Bus. The PCnet-ISA controller buffers the contents of the PROM data bus and drives them on the lower eight bits of the System Data Bus.

DXCVR
Disable Transceiver *Output*

This pin disables the transceiver. The DXCVR output is configured in the initialization sequence. A HIGH level indicates the Twisted Pair port is active and the AUI port is inactive, or SLEEP mode has been entered. A LOW level indicates the AUI port is active and the Twisted Pair port is inactive.

IOAM0-1
Input/Output Address Map *Input*

These inputs configure I/O address space for the PCnet-ISA controller and memory address space for the optional Remote Boot PROM with user selectable jumpers. The pins are pulled HIGH internally. The SA1-9 inputs are used for I/O address comparisons and the SA14-19 inputs are used for Boot PROM matching.

IOAM1,0	I/O Base	Memory Base
0 0	300 Hex	C8000 Hex
0 1	320 Hex	CC000 Hex
1 0	340 Hex	D0000 Hex
1 1	360 Hex	D4000 Hex

LED0-3
LED Drivers *Output*

These pins sink 12 mA each for driving LEDs. Their meaning is software configurable (see section *ISA Bus Configuration Registers*) and they are active LOW.

When EADI mode is selected, the pins named LED1, LED2, and LED3 change in function while LED0 continues to indicate 10BASE-T Link Status. The MAUSEL input becomes the EAR input.

LED	EADI Function
1	SF/BD
2	SRD
3	SRDCLK

MAUSEL/EAR
MAU Select/
External Address Reject *Input*

This pin selects the 10BASE-T MAU when HIGH and the AUI interface when LOW if the XMAUSEL register bit in ISACSR2 (ISA Configuration Register) is set. If the XMAUSEL register bit is cleared, the MAUSEL pin is ignored and the network interface is software selected. This pin has a default value of HIGH if left unconnected.

If EADI mode is selected, this pin becomes the EAR input. The incoming frame will be checked against the internally active address detection mechanisms and the result of this check will be OR'd with the value on the EAR pin. The EAR pin is defind as REJECT. See the EADI section for details regarding the function and timing of this signal.

PRDB0-7
Private Data Bus *Input*

This is the data bus for the Boot PROM and the Address PROM.

SLEEP
Sleep *Input*

When SLEEP pin is asserted (active LOW), the PCnet-ISA controller performs an internal system reset and proceeds into a power savings mode. All outputs will be placed in their normal reset condition. All PCnet-ISA controller inputs will be ignored except for the SLEEP pin itself. Deassertion of SLEEP results in wake-up. The system must delay the starting of the network controller by 0.5 seconds to allow internal analog circuits to stabilize.

TE
Test Enable *Input*

This pin is for factory use only. It has a default value of HIGH if left unconnected. It is recommended that this pin always be connected to V_{DD}.

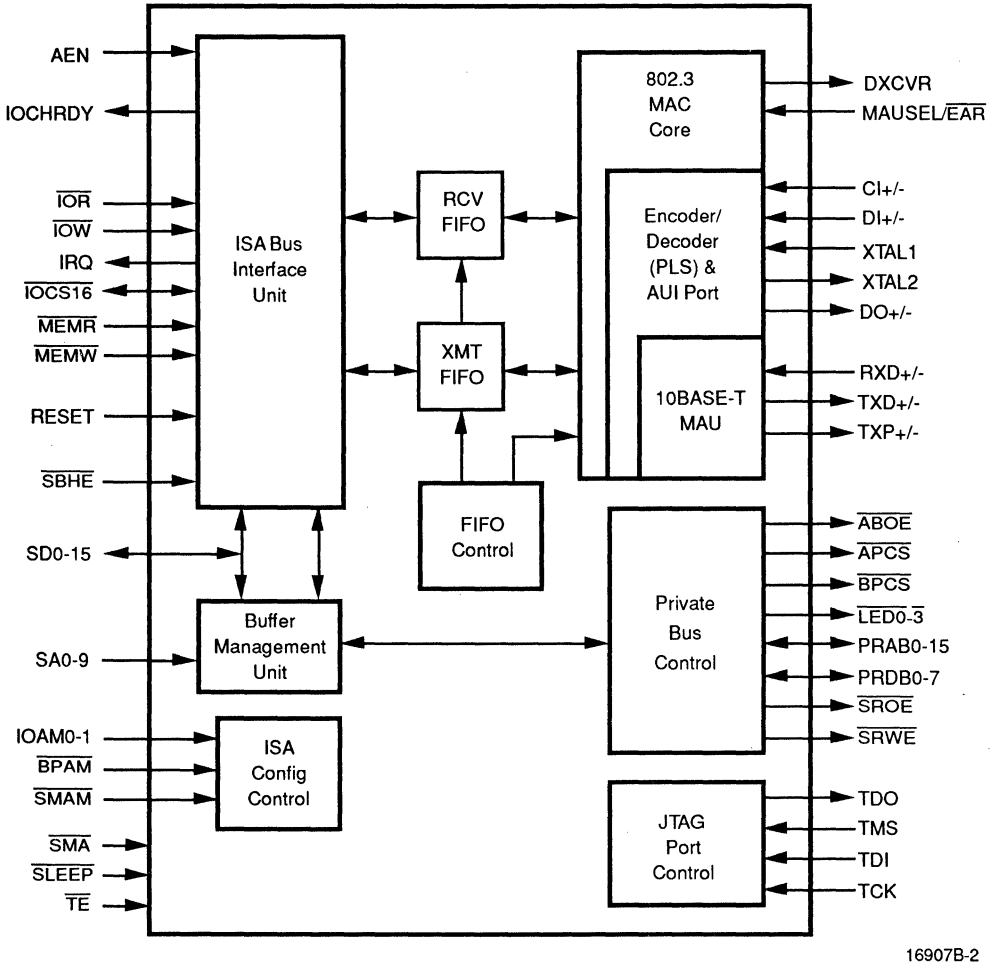
XTAL1
Crystal Connection *Input*

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. Alternatively, an external 20 MHz CMOS-compatible clock signal can be used to drive this pin. Refer to the section on External Crystal Characteristics for more details.

XTAL2
Crystal Connection *Output*

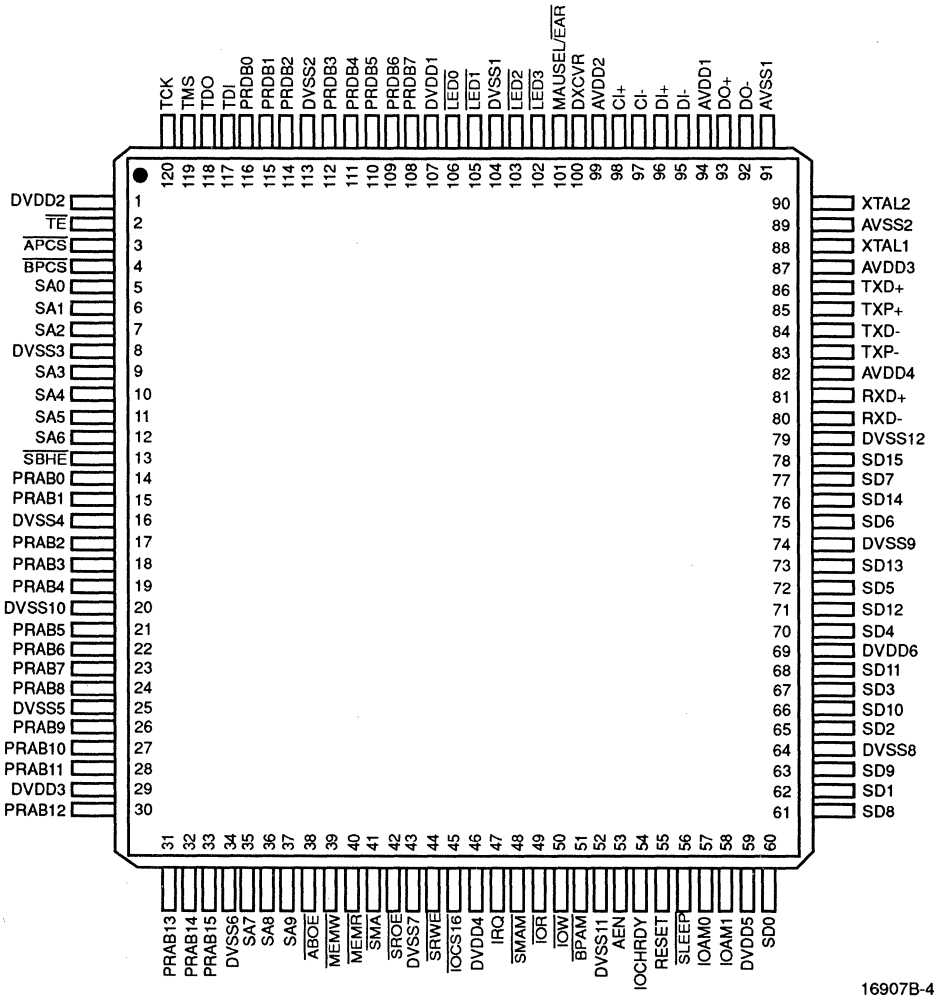
The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. If an external clock is used, this pin should be left unconnected.

BLOCK DIAGRAM: SHARED MEMORY MODE



16907B-2

CONNECTION DIAGRAM: SHARED MEMORY



16907B-4

PIN DESIGNATIONS: SHARED MEMORY
Listed by Pin Number

Pin #	Name	Pin #	Name	Pin #	Name	Pin #	Name
1	DVDD2	31	PRAB13	61	SD8	91	AVSS1
2	$\overline{TE}$	32	PRAB14	62	SD1	92	DO–
3	$\overline{APCS}$	33	PRAB15	63	SD9	93	DO+
4	$\overline{BPCS}$	34	DVSS6	64	DVSS8	94	AVDD1
5	SA0	35	SA7	65	SD2	95	DI–
6	SA1	36	SA8	66	SD10	96	DI+
7	SA2	37	SA9	67	SD3	97	CI–
8	DVSS3	38	$\overline{ABOE}$	68	SD11	98	CI+
9	SA3	39	$\overline{MEMW}$	69	DVDD6	99	AVDD2
10	SA4	40	$\overline{MEMR}$	70	SD4	100	DXCVR
11	SA5	41	$\overline{SMA}$	71	SD12	101	MAUSEL/$\overline{EAR}$
12	SA6	42	$\overline{SROE}$	72	SD5	102	$\overline{LED3}$
13	$\overline{SBHE}$	43	DVSS7	73	SD13	103	$\overline{LED2}$
14	PRAB0	44	$\overline{SRWE}$	74	DVSS9	104	DVSS1
15	PRAB1	45	$\overline{IOCS16}$	75	SD6	105	$\overline{LED1}$
16	DVSS4	46	DVDD4	76	SD14	106	$\overline{LED0}$
17	PRAB2	47	IRQ	77	SD7	107	DVDD1
18	PRAB3	48	$\overline{SMAM}$	78	SD15	108	PRDB7
19	PRAB4	49	$\overline{IOR}$	79	DVSS12	109	PRDB6
20	DVSS10	50	$\overline{IOW}$	80	RXD–	110	PRDB5
21	PRAB5	51	$\overline{BPAM}$	81	RXD+	111	PRDB4
22	PRAB6	52	DVSS11	82	AVDD4	112	PRDB3
23	PRAB7	53	AEN	83	TXP–	113	DVSS2
24	PRAB8	54	IOCHRDY	84	TXD–	114	PRDB2
25	DVSS5	55	RESET	85	TXP+	115	PRDB1
26	PRAB9	56	$\overline{SLEEP}$	86	TXD+	116	PRDB0
27	PRAB10	57	IOAM0	87	AVDD3	117	TDI
28	PRAB11	58	IOAM1	88	XTAL1	118	TDO
29	DVDD3	59	DVDD5	89	AVSS2	119	TMS
30	PRAB12	60	SD0	90	XTAL2	120	TCK

PIN DESIGNATIONS: SHARED MEMORY
Listed by Pin Name

Name	Pin #	Name	Pin #	Name	Pin #	Name	Pin #
$\overline{\text{ABOE}}$	38	DVSS8	64	PRAB10	27	SD2	65
AEN	53	DVSS9	74	PRAB11	28	SD3	67
$\overline{\text{APCS}}$	3	DVSS10	20	PRAB12	30	SD4	70
AVDD1	94	DVSS11	52	PRAB13	31	SD5	72
AVDD2	99	DVSS12	79	PRAB14	32	SD6	75
AVDD3	87	DXCVR	100	PRAB15	33	SD7	77
AVDD4	82	IOAM0	57	PRDB0	116	SD8	61
AVSS1	91	IOAM1	58	PRDB1	115	SD9	63
AVSS2	89	IOCHRDY	54	PRDB2	114	SD10	66
$\overline{\text{BPAM}}$	51	$\overline{\text{IOCS16}}$	45	PRDB3	112	SD11	68
$\overline{\text{BPCS}}$	4	$\overline{\text{IOR}}$	49	PRDB4	111	SD12	71
CI–	97	$\overline{\text{IOW}}$	50	PRDB5	110	SD13	73
CI+	98	IRQ	47	PRDB6	109	SD14	76
DI–	95	$\overline{\text{LED0}}$	106	PRDB7	108	SD15	78
DI+	96	$\overline{\text{LED1}}$	105	RESET	55	$\overline{\text{SLEEP}}$	56
DO–	92	$\overline{\text{LED2}}$	103	RXD–	80	$\overline{\text{SMA}}$	41
DO+	93	$\overline{\text{LED3}}$	102	RXD+	81	$\overline{\text{SMAM}}$	48
DVDD1	107	MAUSEL/$\overline{\text{EAR}}$	101	SA0	5	$\overline{\text{SROE}}$	42
DVDD2	1	$\overline{\text{MEMR}}$	40	SA1	6	$\overline{\text{SRWE}}$	44
DVDD3	29	$\overline{\text{MEMW}}$	39	SA2	7	TCK	120
DVDD4	46	PRAB0	14	SA3	9	TDI	117
DVDD5	59	PRAB1	15	SA4	10	TDO	118
DVDD6	69	PRAB2	17	SA5	11	$\overline{\text{TE}}$	2
DVSS1	104	PRAB3	18	SA6	12	TMS	119
DVSS2	113	PRAB4	19	SA7	35	TXD–	84
DVSS3	8	PRAB5	21	SA8	36	TXD+	86
DVSS4	16	PRAB6	22	SA9	37	TXP–	83
DVSS5	25	PRAB7	23	$\overline{\text{SBHE}}$	13	TXP+	85
DVSS6	34	PRAB8	24	SD0	60	XTAL1	88
DVSS7	43	PRAB9	26	SD1	62	XTAL2	90

PIN DESIGNATIONS: SHARED MEMORY
Listed by Group

Pin Name	Pin Function	I/O	Driver
ISA Bus Interface			
AEN	Address Enable	I	
IOCHRDY	I/O Channel Ready	O	OD3
$\overline{\text{IOCS16}}$	I/O Chip Select 16	I/O	OD3
$\overline{\text{IOR}}$	I/O Read Select	I	
$\overline{\text{IOW}}$	I/O Write Select	I	
IRQ	Interrupt Request	O	OD3
$\overline{\text{MEMR}}$	Memory Read Select	I	
$\overline{\text{MEMW}}$	Memory Write Select	I	
RESET	System Reset	I	
SA0-9	System Address Bus	I	
$\overline{\text{SBHE}}$	System Byte High Enable	I	
SD0-15	System Data Bus	I/O	TS3
Board Interfaces			
$\overline{\text{ABOE}}$	Address Buffer Output Enable	O	TS3
$\overline{\text{APCS}}$	Address PROM Chip Select	O	TS1
$\overline{\text{BPAM}}$	Boot PROM Address Match	I	
$\overline{\text{BPCS}}$	Boot PROM Chip Select	O	TS1
DXCVR	Disable Transceiver	O	TS1
IOAM0-1	Input/Output Address Map	I	
$\overline{\text{LED0}}$	LED0/LNKST	O	TS2
$\overline{\text{LED1}}$	LED1	O	TS2
$\overline{\text{LED2}}$	LED2	O	TS2
$\overline{\text{LED3}}$	LED3	O	TS2
MAUSEL/$\overline{\text{EAR}}$	MAU SELect/External Address Reject	I	
PRAB0-15	PRivate Address Bus	I/O	TS3
PRDB0-7	PRivate Data Bus	I/O	TS1
$\overline{\text{SLEEP}}$	Sleep Mode	I	
$\overline{\text{SMA}}$	Shared Memory Architecture	I	
$\overline{\text{SMAM}}$	Shared Memory Address Match	I	
$\overline{\text{SROE}}$	Static RAM Output Enable	O	TS3
$\overline{\text{SRWE}}$	Static RAM Write Enable	O	TS1
$\overline{\text{TE}}$	Test Enable	I	
XTAL1	Crystal Oscillator Input	I	
XTAL2	Crystal Oscillator OUTPUT	O	

PIN DESIGNATIONS: SHARED MEMORY (continued)
Listed by Group

Pin Name	Pin Function	I/O	Driver
Attachment Unit Interface (AUI)			
CI±	Collision Inputs	I	
DI±	Receive Data	I	
DO±	Transmit Data	O	
Twisted Pair Transceiver Interface (10BASE–T)			
RXD±	10BASE-T Receive Data	I	
TXD±	10BASE-T Transmit Data	O	
TXP±	10BASE-T Predistortion Control	O	
IEEE 1149.1 Test Access Port Interface (JTAG)			
TCK	Test Clock	I	
TDI	Test Data Input	I	
TDO	Test Data Output	O	TS2
TMS	Test Mode Select	I	
Power Supplies			
AVDD	Analog Power		
AVSS	Analog Ground		
DVDD	Digital Power		
DVSS	Digital Ground		

Table: Output Driver Types

Name	Type	Iol (mA)	Ioh (mA)	pF
TS1	Tri-State	4	–1	50
TS2	Tri-State	12	–4	50
TS3	Tri-State	24	–3	120
OD3	Open Drain	24	–3	120

PIN DESCRIPTION: SHARED MEMORY MODE

ISA Interface

AEN
Address Enable *Input*

This signal must be driven LOW when the bus performs an I/O access to the device.

IOCHRDY
I/O Channel Ready *Output*

When the PCnet-ISA controller is being accessed, a HIGH on IOCHRDY indicates that valid data exists on the data bus for reads and that data has been latched for writes.

$\overline{\text{IOCS16}}$
I/O Chip Select 16 *Input/Output*

When an I/O read or write operation is performed, the PCnet-ISA controller will drive this pin LOW to indicate that the chip supports a 16-bit operation at this address. (If the motherboard does not receive this signal, then the motherboard will convert a 16-bit access to two 8-bit accesses.) The $\overline{\text{IOCS16}}$ pin is also an input and must go HIGH at least once after reset for the PCnet-ISA controller to perform 16-bit I/O operations. If this pin is grounded then the PCnet-ISA controller only performs 8-bit I/O operations.

The PCnet-ISA controller follows the IEEE P996 specification that recommends this function be implemented as a pure decode of SA0-9 and AEN, with no dependency on $\overline{\text{SMEMR}}$, $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$, or $\overline{\text{IOW}}$; however, some PC/AT clone systems are not compatible with this approach. For this reason, the PCnet-ISA controller is recommended to be configured to run 8-bit I/O on all machines. Since data is moved by memory cycles there is virtually no performance loss incurred by running 8-bit I/O and compatibility problems are virtually eliminated. The PCnet-ISA controller can be configured to run 8-bit-only I/O by disconnecting the $\overline{\text{IOCS16}}$ pin from the ISA bus and tying the $\overline{\text{IOCS16}}$ pin to ground instead.

$\overline{\text{IOR}}$
I/O Read *Input*

To perform an Input/Output Read operation on the device $\overline{\text{IOR}}$ must be asserted. $\overline{\text{IOR}}$ is only valid if the AEN signal is LOW and the external address matches the PCnet-ISA controller's predefined I/O address location. If valid, $\overline{\text{IOR}}$ indicates that a slave read operation is to be performed.

$\overline{\text{IOW}}$
I/O Write *Input*

To perform an Input/Output write operation on the device $\overline{\text{IOW}}$ must be asserted. $\overline{\text{IOW}}$ is only valid if AEN signal is LOW and the external address matches the PCnet-ISA controller's predefined I/O address location. If valid, $\overline{\text{IOW}}$ indicates that a slave write operation is to be performed.

IRQ
Interrupt Request *Output*

An attention signal which indicates that one or more of the following status flags is set: BABL, MISS, MERR, RINT, IDON or TXSTRT. All status flags have a mask bit which allows for suppression of INTR assertion. These flags have the following meaning:

BABL	Babble
RCVCCO	Receive Collision Count Overflow
JAB	Jabber
MISS	Missed Frame
MERR	Memory Error
MFCO	Missed Frame Count Overflow
RINT	Receive Interrupt
IDON	Initialization Done
TXSTRT	Transmit Start

$\overline{\text{MEMR}}$
Memory Read *Input*

$\overline{\text{MEMR}}$ goes LOW to perform a memory read operation.

$\overline{\text{MEMW}}$
Memory Write *Input*

$\overline{\text{MEMW}}$ goes LOW to perform a memory write operation.

RESET
Reset *Input*

When RESET is asserted HIGH, the PCnet-ISA controller performs an internal system reset. RESET must be held for a minimum of 10 XTAL1 periods before being deasserted. While in a reset state, the PCnet-ISA controller will tristate or deassert all outputs to predefined reset levels. The PCnet-ISA controller resets itself upon power-up.

SA0-9
System Address Bus *Input*

This bus carries the address inputs from the system address bus. Address data is stable during command active cycle.

$\overline{\text{SBHE}}$
System Bus High Enable *Input*

This signal indicates the HIGH byte of the system data bus is to be used. There is a weak pull-up resistor on this pin. If the PCnet-ISA controller is installed in an 8-bit only system like the PC/XT, $\overline{\text{SBHE}}$ will always be HIGH and the PCnet-ISA controller will perform only 8-bit operations. There must be at least one LOW going edge on this signal before the PCnet-ISA controller will perform 16-bit operations.

SD0-15
System Data Bus *Input/Output*

This bus is used to transfer data to and from the PCnet-ISA controller to system resources via the ISA data bus. SD0-15 is driven by the PCnet-ISA controller when performing slave read operations.

Likewise, the data on SD0-15 is latched by the PCnet-ISA controller when performing slave write operations.

Board Interface

$\overline{\text{ABOE}}$
Address Buffer Output Enable *Output*

This pin goes LOW to enable an external octal buffer to drive the contents of SA10-15 onto PRAB10-15. Only six of the eight buffers are needed.

$\overline{\text{APCS}}$
Address PROM Chip Select *Output*

This signal is asserted when the external Address PROM is read. When an I/O read operation is performed on the first 16 bytes in the PCnet-ISA controller's I/O space, $\overline{\text{APCS}}$ is asserted. The outputs of the external Address PROM drive the PROM Data Bus. The PCnet-ISA controller buffers the contents of the PROM data bus and drives them on the lower eight bits of the System Data Bus. $\overline{\text{IOCS16}}$ is not asserted during this cycle.

$\overline{\text{BPAM}}$
Boot PROM Address Match *Input*

This pin indicates a Boot PROM access cycle. If no Boot PROM is installed, this pin has a default value of HIGH and thus may be left connected to V_{DD}.

$\overline{\text{BPCS}}$
Boot PROM Chip Select *Output*

This signal is asserted when the Boot PROM is read. If $\overline{\text{BPAM}}$ is active and $\overline{\text{MEMR}}$ is active, the $\overline{\text{BPCS}}$ signal will be asserted. The outputs of the external Boot PROM drive the PROM Data Bus. The PCnet-ISA controller buffers the contents of the PROM data bus and drives them on the System Data Bus. $\overline{\text{IOCS16}}$ is not asserted during this cycle. If 16-bit cycles are performed, it is the responsibility of external logic to assert $\overline{\text{MEMCS16}}$ signal.

DXCVR
Disable Transceiver *Output*

This pin disables the transceiver. A high level indicates the Twisted Pair Interface is active and the AUI interface is inactive, or SLEEP mode has been entered. A low level indicates the AUI interface is active and the Twisted Pair interface is inactive.

IOAM0-1
Input/Output Address Map *Input*

These inputs configure I/O address space for the PCnet-ISA controller. The pins have an on-chip pullup resistor and are pulled HIGH internally. The SA1-9 inputs are used for I/O address comparisons.

IOAM1,0	I/O Base
0 0	300 Hex
0 1	320 Hex
1 0	340 Hex
1 1	360 Hex

$\overline{\text{LED0-3}}$
LED Drivers *Output*

These pins sink 12 mA each for driving LEDs. Their meaning is software configurable (see section *ISA Bus Configuration Registers*) and they are active LOW.

When EADI mode is selected, the pins named $\overline{\text{LED1}}$, $\overline{\text{LED2}}$, and $\overline{\text{LED3}}$ change in function while $\overline{\text{LED0}}$ continues to indicate 10BASE-T Link Status. The MAUSEL input becomes the $\overline{\text{EAR}}$ input.

LED	EADI Function
1	SF/BD
2	SRD
3	SRDCLK

MAUSEL/$\overline{\text{EAR}}$
MAU Select/
External Address Reject *Input*

This pin selects the 10BASE-T MAU when HIGH and the AUI interface when LOW if the XMAUSEL register bit in ISACSR2 (ISA Configuration Register) is set. If the XMAUSEL register bit is cleared, the MAUSEL pin is ignored and the network interface is software selected. This pin has a default value of HIGH if left unconnected.

If EADI mode is selected, this pin becomes the $\overline{\text{EAR}}$ input. The incoming frame will be checked against the internally active address detection mechanisms and the result of this check will be OR'd with the value on the $\overline{\text{EAR}}$ pin. The $\overline{\text{EAR}}$ pin is defined as $\overline{\text{REJECT}}$. See the EADI section for details regarding the function and timing of this signal.

PRAB0-15
Private Address Bus *Input/Output*

The Private Address Bus is the address bus used to drive the Address PROM, Remote Boot PROM, and SRAM. PRAB10-15 are required to be buffered by a Bus Buffer with $\overline{\text{ABOE}}$ as its control and SA10-15 as its inputs.

PRDB0-7
Private Data Bus *Input/Output*

This is the data bus for the static RAM, the Boot PROM, and the Address PROM.

$\overline{\text{SLEEP}}$
Sleep *Input*

When $\overline{\text{SLEEP}}$ input is asserted (active LOW), the PCnet-ISA controller performs an internal system reset and proceeds into a power savings mode. All outputs will be placed in their normal reset condition. All PCnet-ISA controller inputs will be ignored except for the $\overline{\text{SLEEP}}$ pin itself. Deassertion of $\overline{\text{SLEEP}}$ results in wake-up. The system must delay the starting of the network controller by 0.5 seconds to allow internal analog circuits to stabilize.

$\overline{\text{SMA}}$
Shared Memory Architecture *Input*

This pin is sampled after the hardware RESET sequence. The pin must be pulled permanently LOW for operation in the shared memory mode.

$\overline{\text{SMAM}}$
Shared Memory Address Match *Input*

This pin indicates an access to shared memory when active. The type of access is decided by $\overline{\text{MEMR}}$ or $\overline{\text{MEMW}}$.

$\overline{\text{SROE}}$
Static RAM Output Enable *Output*

This pin directly controls the external SRAM's $\overline{\text{OE}}$ pin.

$\overline{\text{SRWE}}$
Static RAM Write Enable *Output*

This pin directly controls the external SRAM's $\overline{\text{WE}}$ pin.

$\overline{\text{TE}}$
Test Enable *Input*

This pin is for factory use only. It has a default value of HIGH if left unconnected. It is strongly recommended that this pin always be connected to V_{DD}.

XTAL1
Crystal Connection *Input*

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. Alternatively, an external 20 MHz CMOS-compatible clock signal can be used to drive this pin. Refer to the section on External Crystal Characteristics for more details.

XTAL2
Crystal Connection *Output*

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. If an external clock is used, this pin should be left unconnected.

PIN DESCRIPTION: NETWORK INTERFACES

AUI Interface

CI+, CI–
Control Input *Input*

This is a differential input pair used to detect Collision (Signal Quality Error Signal).

DI+, DI–
Data In *Input*

This is a differential receive data input pair to the PCnet-ISA controller.

DO+, DO–
Data Out *Output*

This is a differential transmit data output pair from the PCnet-ISA controller.

Twisted Pair Interface

RXD+, RXD–
Receive Data *Input*

This is the 10BASE-T port differential receive input pair.

TXD+, TXD–
Transmit Data *Output*

These are the 10BASE-T port differential transmit drivers.

TXP+, TXP–
Transmit Predistortion Control *Output*

These are 10BASE-T transmit waveform pre-distortion control differential outputs.

PIN DESCRIPTION IEEE 1149.1 (JTAG) TEST ACCESS PORT

TCK
Test Clock *Input*

This is the clock input for the boundary scan test mode operation. TCK can operate up to 10 MHz. If left unconnected, this pin has a default value of HIGH.

TDI
Test Data Input *Input*

This is the test data input path to the PCnet-ISA controller. If left unconnected, this pin has a default value of HIGH.

TDO
Test Data Output *Output*

This is the test data output path from the PCnet-ISA controller. TDO is tri-stated when JTAG port is inactive.

TMS
Test Mode Select *Input*

This is a serial input bit stream used to define the specific boundary scan test to be executed. If left unconnected, this pin has a default value of HIGH.

PIN DESCRIPTION: POWER SUPPLIES

All power pins with a "D" prefix are digital pins connected to the digital circuitry and digital I/O buffers. All power pins with an "A" prefix are analog power pins connected to the analog circuitry. Not all analog pins are quiet and special precaution must be taken when doing board layout. Some analog pins are more noisy than others and must be separated from the other analog pins.

AVDD1–4

Analog Power (4 Pins) ***Power***

Supplies power to analog portions of the PCnet-ISA controller. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. These supply lines should be kept separate from the DVDD supply pins and as far back to the power supply as is practically possible. AVDD3 is an exception and should be connected to DVDD supply and away from remaining AVDD supply pins. See the table below for more details.

AVSS1–2

Analog Ground (2 Pins) ***Power***

Supplies ground reference to analog portions of PCnet-ISA controller. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. These supply lines should be kept separate from the DVSS ground pins and as far back to the power supply as is practically possible. AVSS1 is an exception and should be connected to DVSS supply and away from remaining AVSS supply pins. See the table below for more details.

DVDD1–6

Digital Power (6 Pins) ***Power***

Supplies power to digital portions of PCnet-ISA controller. Four pins are used by Input/Output buffer drivers and two are used by the internal digital circuitry.

DVSS1–12

Digital Ground (12 Pins) ***Power***

Supplies ground reference to digital portions of PCnet-ISA controller. Ten pins are used by Input/Output buffer drivers and two are used by the internal digital circuitry.

Analog Power Pins and the Circuits to Which They are Connected

Analog Power	Analog Ground	Circuit	Comments
AVDD2 and AVDD4	AVSS2	These pins are connected to the analog voltage reference circuit and VCO.	These pins should be kept quiet. They should be kept separated with low- and high-frequency by-pass capacitors.
AVDD1		These pins are connected to analog circuits such as AUI and Twisted Pair receive logic.	These pins are moderately quiet and should be connected to the VDD supply a short distance away from the DVDD pins.
AVDD3	AVSS1	These pins are connected to the AUI and Twisted Pair drivers.	These pins are more noisy and should be connected to the DVDD/DVSS supplies.

FUNCTIONAL DESCRIPTION

The PCnet-ISA controller is a highly integrated system solution for the PC-AT ISA architecture. It provides an Ethernet controller, AUI port, and 10BASE-T transceiver. The PCnet-ISA controller can be directly interfaced to an ISA system bus. The PCnet-ISA controller contains an ISA bus interface unit, DMA Buffer Management Unit, 802.3 Media Access Control function, separate 136-byte transmit and 128-byte receive FIFOs, IEEE defined Attachment Unit Interface (AUI), and Twisted-Pair Transceiver Media Attachment Unit. In addition, a Sleep function has been incorporated which provides low standby current for power sensitive applications.

The PCnet-ISA controller is register compatible with the LANCE (Am7990) Ethernet controller and PCnet-ISA+ controller (Am79C961). The DMA Buffer Management Unit supports the LANCE descriptor software model and the PCnet-ISA controller is software compatible with the Novell NE2100 and NE1500T add-in cards.

External remote boot and Ethernet physical address PROMs are supported. The location of the I/O registers and PROMs are configured by selected pins and internal address comparators (in bus master mode) or external logic (in shared memory mode).

The PCnet-ISA controller's bus master architecture brings to system manufacturers (adapter card and motherboard makers alike) something they have not been able to enjoy with other architectures—a low-cost system solution that provides the lowest parts count and highest performance. As a bus-mastering device, costly and power-hungry external SRAMs are not needed for packet buffering. This results in lower system cost due to fewer components, less real-estate and less power. The PCnet-ISA controller's advanced bus mastering architecture also provides high data throughput and low CPU utilization for even better performance.

To offer greater flexibility, the PCnet-ISA controller has a shared memory mode to meet varying application needs. The shared memory architecture is compatible with very low-end machines, such as PC/XTs that do not support bus mastering, and very high end machines which require local packet buffering for increased system latency.

The network interface provides an Attachment Unit Interface and Twisted-Pair Transceiver functions. Only one interface is active at any particular time. The AUI allows for connection via isolation transformer to 10BASE5 and 10BASE2, thick and thin based coaxial cables. The Twisted-Pair Transceiver interface allows for connection of unshielded twisted-pair cables as specified by the Section 14 supplement to IEEE 802.3 Standard (Type 10BASE-T).

Bus Master Mode

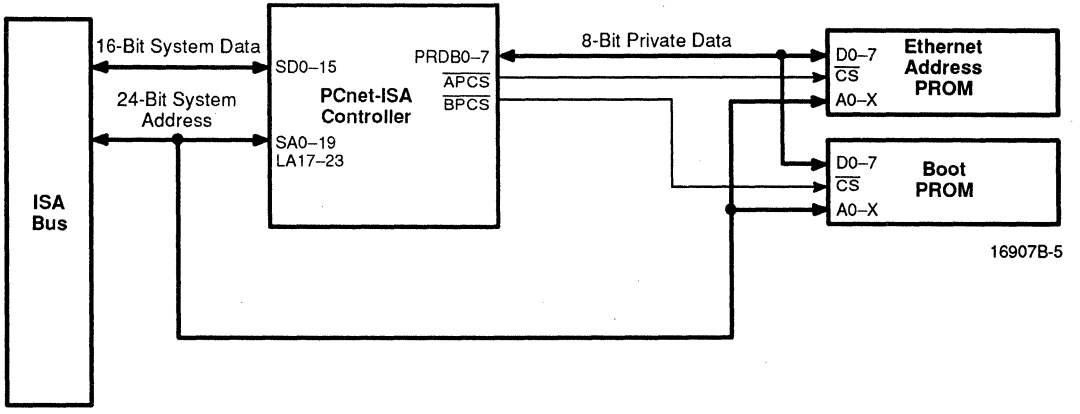
System Interface

The PCnet-ISA controller has two fundamental operating modes, Bus Master and Shared Memory. The selection of either the Bus Master mode or the Shared Memory mode must be done through hard wiring; it is not software configurable. The Bus Master mode provides an Am7990 (LANCE) compatible Ethernet controller, an Ethernet Address PROM, a Boot PROM, and a set of device configuration registers.

The optional Boot PROM is in memory address space and is expected to be 16 kilobytes or less in size. The memory address is always related to the I/O address. For example, 0x300 is always associated with 0xC8000. On-chip address comparators control device selection based on the value of the input pins IOAM0 and IOAM1. The $\overline{\text{SMEMR}}$ input pin can be left unconnected for applications where a Remote Boot PROM is not needed.

The address PROM, board configuration registers, and the Ethernet controller occupy 24 bytes of I/O space and can be located on four different starting addresses.

Data buffers are located in motherboard memory and can be accessed by the PCnet-ISA controller when the device becomes the Current Master.



Bus Master Block Diagram

Shared Memory Mode

System Interface

The Shared Memory mode is the other fundamental operating mode available on the PCnet-ISA controller. The PCnet-ISA controller uses the same descriptor and buffer architecture as the LANCE, but these data structures are stored in static RAM controlled by the PCnet-ISA controller. The static RAM is visible as a memory resource to the PC. The other resources look the same as in the Bus Master mode.

The Boot PROM is selected by an external device which drives the Boot PROM Address Match ($\overline{BPAM}$) input to the PCnet-ISA controller. The PCnet-ISA controller can perform two 8-bit accesses from the 8-bit Boot PROM and present16-bits of data. The shared memory works the same way, with an external device generating Shared Memory Address Match and the PCnet-ISA controller performing the read or write and the 8 to 16-bit data conversion.

Converting shared memory accesses from 8-bit cycles to 16-bit cycles allows use of the much faster 16-bit cycle timing while cutting the number of bus cycles in half. This raises performance to more than 400% of what could be achieved with 8-bit cycles. Converting boot PROM accesses to 16-bit cycles allows the two memory resources to be in the same 128 Kbyte block of memory without a clash between two devices with different data widths.

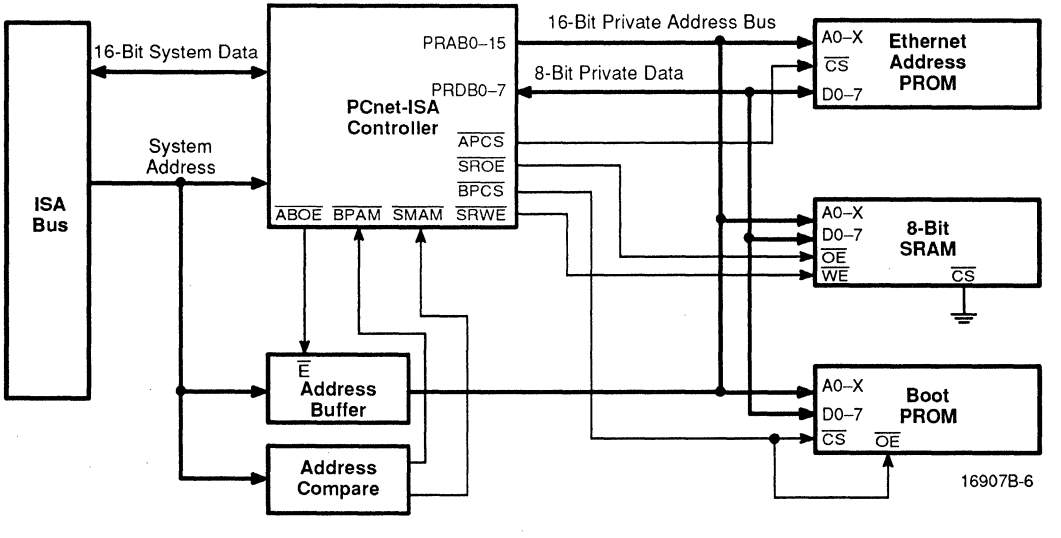
Note that the external address buffer must drive all the bits of PRAB10-15 even if the static RAM is less than 64 Kbytes. The PCnet-ISA controller uses an internal address comparator to perform SRAM prefetches on the Private Data Bus; the PRAB0-15 signals are used internally to determine whether a SRAM read cycle prefetch is a match or a miss.

Access to the Ethernet controller registers, board configuration registers, and Address PROM is done with on-chip address comparators.

Network Interface

The PCnet-ISA controller can be connected to an IEEE 802.3 network via one of two network interface ports. The Attachment Unit Interface (AUI) provides an IEEE 802.3 compliant differential interface to a remote MAU or a transceiver on the system board. The 10BASE-T interface provides a twisted-pair Ethernet port. The PCnet-ISA controller provides three modes of network interface selection: automatic selection, software selection, and jumper selection of AUI or 10BASE-T interface.

In the automatic selection mode, the PCnet-ISA controller will select the interface that is connected to the network by checking the Link Status state machine. If both AUI and 10BASE-T interfaces are connected, the 10BASE-T interface is selected over AUI. If the PCnet-ISA controller is initialized for software selection of network interface, it will read the PORTSEL [1:0] bits in the Mode register (CSR15.8 and CSR15.7) to determine which interface needs to be activated. For jumper selection of the network interface, the MAUSEL pin is used. When the XMAUSEL bit in ISACSR2 is set, a HIGH on the pin will select the 10BASE-T interface, and a LOW on the pin will select the AUI interface.



Shared Memory Block Diagram

DETAILED FUNCTIONS

Bus Interface Unit (BIU)

The bus interface unit is a mixture of a 20 MHz state machine and asynchronous logic. It handles two types of accesses: accesses where the PCnet-ISA controller is a slave and accesses where the PCnet-ISA controller is the Current Master.

In slave mode, signals like $\overline{\text{IOCS16}}$ are asserted and deasserted as soon as the appropriate inputs are received. IOCHRDY is asynchronously driven LOW if the PCnet-ISA controller needs a wait state. It is released synchronously when the PCnet-ISA controller is ready.

When the PCnet-ISA controller is the Current Master, all the signals it generates are synchronous to the on-chip 20 MHz clock.

DMA Transfers

The BIU will initiate DMA transfers according to the type of operation being performed. There are three primary types of DMA transfers:

1. Initialization Block DMA Transfers

Once the BIU has been granted bus mastership, it will perform four data transfer cycles (eight bytes) before relinquishing the bus. The four transfers within the mastership period will always be read cycles to contiguous addresses. There are 12 words to transfer so there will be three bus mastership periods.

2. Descriptor DMA Transfers

Once the BIU has been granted bus mastership, it will perform the appropriate number of data transfer cycles before relinquishing the bus. The transfers within the mastership period will always be of the same type (either all read or all write), but may be to non-contiguous addresses. Only the bytes which need to be read or written are accessed.

3. Burst-Cycle DMA Transfers

Once the BIU has been granted bus mastership, it will perform a series of consecutive data transfer cycles before relinquishing the bus. Each data transfer will be performed sequentially, with the issue of the address, and the transfer of the data with appropriate output signals to indicate selection of the active data bytes during the transfer. All transfers within the mastership cycle will be either read or write cycles, and will be to contiguous addresses. The number of data transfer cycles within the burst is dependent on the programming of the DMAPLUS option (CSR4, bit 14).

If DMAPLUS = 0, a maximum of 16 transfers will be performed. This may be changed by writing to the burst register (CSR80), but the default takes the same amount of time as the Am2100 family of LANCE-based boards, a little over 5 microseconds.

If DMAPLUS = 1, the burst will continue until the FIFO is filled to its high threshold (32 bytes in transmit operation) or emptied to its low threshold (16 bytes in receive operation). The exact number of transfer cycles in this case will be dependent on the latency of the system bus to the BIU's mastership request and the speed of bus operation.

Buffer Management Unit (BMU)

The buffer management unit is a micro-coded 20 MHz state machine which implements the initialization block and the descriptor architecture.

Initialization

PCnet-ISA controller initialization includes the reading of the initialization block in memory to obtain the operating parameters. The initialization block is read when the INIT bit in CSR0 is set. The INIT bit should be set before or concurrent with the STRT bit to insure correct operation. Four words at a time are read and the bus is released at the end of each block of reads, for a total of three arbitration cycles. Once the initialization block has been read in and processed, the BMU knows where the receive and transmit descriptor rings are. On completion of the read operation and after internal registers have been updated, IDON will be set in CSR0, and an interrupt generated if IENA is set.

The Initialization Block is vectored by the contents of CSR1 (least significant 16 bits of address) and CSR2 (most significant 8 bits of address). The block contains the user defined conditions for PCnet-ISA controller operation, together with the address and length information to allow linkage of the transmit and receive descriptor rings.

There is an alternative method to initialize the PCnet-ISA controller. Instead of initialization via the initialization block in memory, data can be written directly into the appropriate registers. Either method may be used at the discretion of the programmer. If the registers are written to directly, the INIT bit must not be set, or the initialization block will be read in, thus overwriting the previously written information. Please refer to Appendix C for details on this alternative method.

Reinitialization

The transmitter and receiver section of the PCnet-ISA controller can be turned on via the initialization block (MODE Register DTX, DRX bits CSR15[1:0]). The state of the transmitter and receiver are monitored through CSR0 (RXON, TXON bits). The PCnet-ISA controller should be reinitialized if the transmitter and/or the receiver were not turned on during the original initialization and it was subsequently required to activate them, or if either section shut off due to the detection of an error condition (MERR, UFLO, TX BUFF error).

Reinitialization may be done via the initialization block or by setting the STOP bit in CSR0, followed by writing to CSR15, and then setting the START bit in CSR0. Note that this form of restart will not perform the same in the PCnet-ISA controller as in the LANCE. In particular, the PCnet-ISA controller reloads the transmit and receive descriptor pointers with their respective base ad-

dresses. This means that the software must clear the descriptor own bits and reset its descriptor ring pointers before the restart of the PCnet-ISA controller. The reload of descriptor base addresses is performed in the LANCE only after initialization, so a restart of the LANCE without initialization leaves the LANCE pointing at the same descriptor locations as before the restart.

Buffer Management

Buffer management is accomplished through message descriptor entries organized as ring structures in memory. There are two rings, a receive ring and a transmit ring. The size of a message descriptor entry is 4 words (8 bytes).

Descriptor Rings

Each descriptor ring must be organized in a contiguous area of memory. At initialization time (setting the INIT bit in CSR0), the PCnet-ISA controller reads the user-defined base address for the transmit and receive descriptor rings, which must be on an 8-byte boundary, as well as the number of entries contained in the descriptor rings. By default, a maximum of 128 ring entries is permitted when utilizing the initialization block, which uses values of TLEN and RLEN to specify the transmit and receive descriptor ring lengths. However, the ring lengths can be manually defined (up to 65535) by writing the transmit and receive ring length registers (CSR76,78) directly.

Each ring entry contains the following information:

- The address of the actual message data buffer in user or host memory
- The length of the message buffer
- Status information indicating the condition of the buffer

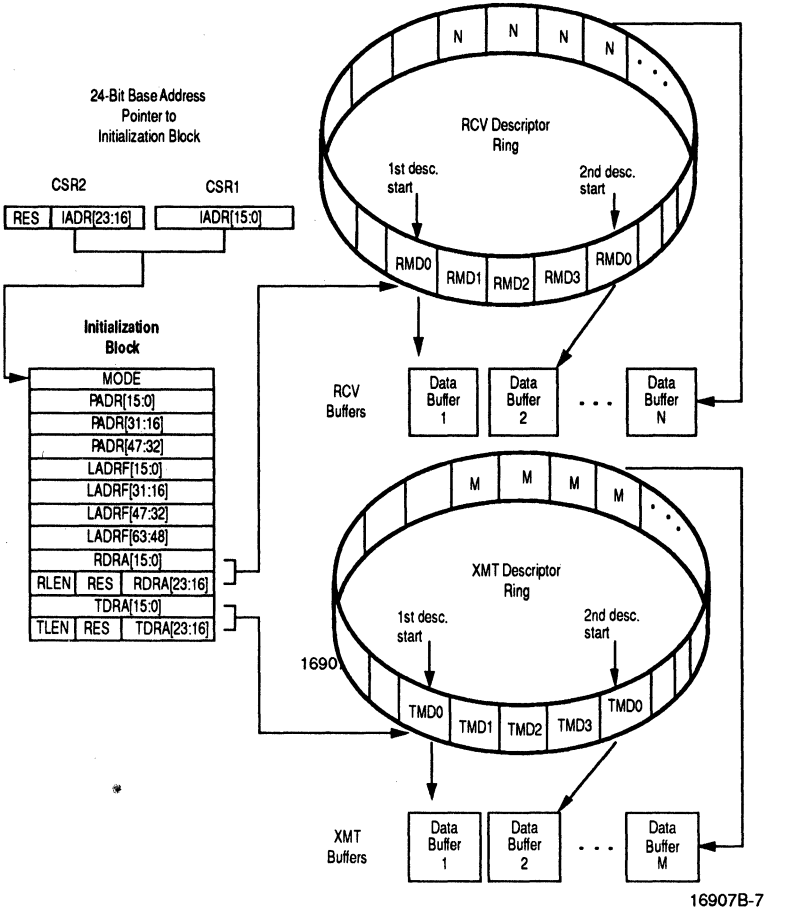
Receive descriptor entries are similar (but not identical) to transmit descriptor entries. Both are composed of four registers, each 16 bits wide for a total of 8 bytes.

To permit the queuing and de-queuing of message buffers, ownership of each buffer is allocated to either the PCnet-ISA controller or the host. The OWN bit within the descriptor status information, either TMD or RMD (see section on TMD or RMD), is used for this purpose. "Deadly Embrace" conditions are avoided by the ownership mechanism. Only the owner is permitted to relinquish ownership or to write to any field in the descriptor entry. A device that is not the current owner of a descriptor entry cannot assume ownership or change any field in the entry.

Descriptor Ring Access Mechanism

At initialization, the PCnet-ISA controller reads the base address of both the transmit and receive descriptor rings into CSRs for use by the PCnet-ISA controller during subsequent operation.

When transmit and receive functions begin, the base address of each ring is loaded into the current descriptor address registers and the address of the next descriptor entry in the transmit and receive rings is computed and loaded into the next descriptor address registers.



Initialization Block and Descriptor Rings

Polling

When there is no channel activity and there is no pre- or post-receive or transmit activity being performed by the PCnet-ISA controller, then the PCnet-ISA controller will periodically poll the current receive and transmit descriptor entries in order to ascertain their ownership. If the DPOLL bit in CSR4 is set, then the transmit polling function is disabled.

A typical polling operation consists of the following: The PCnet-ISA controller will use the current receive descriptor address stored internally to vector to the appropriate Receive Descriptor Table Entry (RDTE). It will then use the current transmit descriptor address (stored internally) to vector to the appropriate Transmit Descriptor Table Entry (TDTE). These accesses will be made to RMD1 and RMD0 of the current RDTE and TMD1 and TMD0 of the current TDTE at periodic polling intervals. All information collected during polling activity will be stored internally in the appropriate CSRs. (i.e. CSR18–19, CSR20–21, CSR40, CSR42, CSR50, CSR52). UnOWNed descriptor status will be internally ignored.

A typical receive poll occurs under the following conditions:

1) PCnet-ISA controller does not possess ownership of the current RDTE and
the poll time has elapsed and
RXON = 1,

or

2) PCnet-ISA controller does not possess ownership of the next RDTE and
the poll time has elapsed and
RXON = 1.

If RXON = 0, the PCnet-ISA controller will never poll RDTE locations.

If RXON=1, the system should always have at least one RDTE available for the possibility of a receive event. When there is only one RDTE, there is no polling for next RDTE.

A typical transmit poll occurs under the following conditions:

1) PCnet-ISA controller does not possess ownership of the current TDTE and
DPOLL = 0 and
TXON = 1 and
the poll time has elapsed,

or

2) PCnet-ISA controller does not possess ownership of the current TDTE and
DPOLL = 0 and
TXON = 1 and
a packet has just been received,

or

3) PCnet-ISA controller does not possess ownership of the current TDTE and
DPOLL = 0 and
TXON = 1 and
a packet has just been transmitted.

The poll time interval is nominally defined as 32,768 crystal clock periods, or 1.6 ms. However, the poll time register is controlled internally by microcode, so any other microcode controlled operation will interrupt the incrementing of the poll count register. For example, when a receive packet is accepted by the PCnet-ISA controller, the device suspends execution of the poll-time-incrementing microcode so that a receive microcode routine may instead be executed. Poll-time-incrementing code is resumed when the receive operation has completely finished. Note, however, that following the completion of any receive or transmit operation, a poll operation will always be performed. The poll time count register is never reset. Note that if a non-default value is desired, then a strict sequence of setting the INIT bit in CSR0, waiting for INITDONE, then writing to CSR47, and then setting STRT in CSR0 must be observed, otherwise the default value will not be overwritten. See the CSR47 section for details.

Setting the TDMD bit of CSR0 will cause the microcode controller to exit the poll counting code and immediately perform a polling operation. If RDTE ownership has not been previously established, then an RDTE poll will be performed ahead of the TDTE poll.

Transmit Descriptor Table Entry (TDTE)

If, after a TDTE access, the PCnet-ISA controller finds that the OWN bit of that TDTE is not set, then the PCnet-ISA controller resumes the poll time count and reexamines the same TDTE at the next expiration of the poll time count.

If the OWN bit of the TDTE is set, but STP = 0, the PCnet-ISA controller will immediately request the bus in order to reset the OWN bit of this descriptor; this condition would normally be found following a LCOL or RETRY error that occurred in the middle of a transmit packet chain of buffers. After resetting the OWN bit of this descriptor, the PCnet-ISA controller will again immediately request the bus in order to access the next TDTE location in the ring.

If the OWN bit is set and the buffer length is 0, the OWN bit will be reset. In the LANCE the buffer length of 0 is interpreted as a 4096-byte buffer. It is acceptable to have a 0 length buffer on transmit with STP = 1 or STP = 1 *and* ENP = 1. It is not acceptable to have 0 length buffer with STP = 0 *and* ENP = 1.

If the OWN bit is set and the start of packet (STP) bit is set, then microcode control proceeds to a routine that will enable transmit data transfers to the FIFO.

If the transmit buffers are data chained (ENP=0 in the first buffer), then the PCnet-ISA controller will look ahead to the next transmit descriptor after it has performed at least one transmit data transfer from the first buffer. More than one transmit data transfer may possibly take place, depending upon the state of the transmitter. The transmit descriptor lookahead reads TMD0 first and TMD1 second. The contents of TMD0 and TMD1 will be stored in Next TX Descriptor Address (CSR32), Next TX Byte Count (CSR66) and Next TX Status (CSR67) regardless of the state of the OWN bit. This transmit descriptor lookahead operation is performed only once.

If the PCnet-ISA controller does not own the next TDTE (i.e. the second TDTE for this packet), then it will complete transmission of the current buffer and then update the status of the current (first) TDTE with the BUFF and UFLO bits being set. This will cause the transmitter to be disabled (CSR0, TXON = 0). The PCnet-ISA controller will have to be restarted to restore the transmit function. The situation that matches this description implies that the system has not been able to stay ahead of the PCnet-ISA controller in the transmit descriptor ring and, therefore, the condition is treated as a fatal error. To avoid this situation, the system should always set the transmit chain descriptor own bits in reverse order.

If the PCnet-ISA controller does own the second TDTE in a chain, it will gradually empty the contents of the first buffer (as the bytes are needed by the transmit operation), perform a single-cycle DMA transfer to update the status (reset the OWN bit in TMD1) of the first descriptor, and then it may perform one data DMA access on the second buffer in the chain before executing another lookahead operation. (i.e. a lookahead to the third descriptor.)

The PCnet-ISA controller can queue up to two packets in the transmit FIFO. Call them packet "X" and packet "Y", where "Y" is after "X". Assume that packet "X" is currently being transmitted. Because the PCnet-ISA controller can perform lookahead data transfer over an ENP, it is possible for the PCnet-ISA controller to update a TDTE in a buffer belonging to packet "Y" while packet "X" is being transmitted if packet "Y" uses data chaining. This operation will result in non-sequential TDTE accesses as packet "X" completes transmission and the PCnet-ISA controller writes out its status, since packet "X"'s TDTE is before the TDTE accessed as part of the lookahead data transfer from packet "Y".

This should not cause any problem for properly written software which processes buffers in sequence, waiting for ownership before proceeding.

If an error occurs in the transmission before all of the bytes of the current buffer have been transferred, then TMD2 and TMD1 of the current buffer will be written; in that case, data transfers from the next buffer will not commence. Instead, following the TMD2/TMD1 update, the PCnet-ISA controller will go to the next transmit packet, if any, skipping over the rest of the packet which experienced an error, including chained buffers.

This is done by returning to the polling microcode where it will immediately access the next descriptor and find the condition OWN = 1 and STP = 0 as described earlier. In that case, the PCnet-ISA controller will reset the own bit for this descriptor and continue in like manner until a descriptor with OWN = 0 (no more transmit packets in the ring) or OWN = 1 and STP = 1 (the first buffer of a new packet) is reached.

At the end of any transmit operation, whether successful or with errors, and the completion of the descriptor updates, the PCnet-ISA controller will always perform another poll operation. As described earlier, this poll operation will begin with a check of the current RDTE, unless the PCnet-ISA controller already owns that descriptor. Then the PCnet-ISA controller will proceed to polling the next TDTE. If the transmit descriptor OWN bit has a zero value, then the PCnet-ISA controller will resume poll time count incrementation. If the transmit descriptor OWN bit has a value of ONE, then the PCnet-ISA controller will begin filling the FIFO with transmit data and initiate a transmission. This end-of-operation poll avoids inserting poll time counts between successive transmit packets.

Whenever the PCnet-ISA controller completes a transmit packet (either with or without error) and writes the status information to the current descriptor, then the TINT bit of CSR0 is set to indicate the completion of a transmission. This causes an interrupt signal if the IENA bit of CSR0 has been set and the TINTM bit of CSR3 is reset.

Receive Descriptor Table Entry (RDTE)

If the PCnet-ISA controller does not own both the current and the next Receive Descriptor Table Entry, then the PCnet-ISA controller will continue to poll according to the polling sequence described above. If the receive descriptor ring length is 1, there is no next descriptor, and no look ahead poll will take place.

If a poll operation has revealed that the current and the next RDTE belongs to the PCnet-ISA controller, then additional poll accesses are not necessary. Future poll operations will not include RDTE accesses as long as the PCnet-ISA controller retains ownership to the current and the next RDTE.

When receive activity is present on the channel, the PCnet-ISA controller waits for the complete address of the message to arrive. It then decides whether to accept or reject the packet based on all active addressing schemes. If the packet is accepted the PCnet-ISA controller checks the current receive buffer status register CRST (CSR40) to determine the ownership of the current buffer.

If ownership is lacking, then the PCnet-ISA controller will immediately perform a (last ditch) poll of the current RDTE. If ownership is still denied, then the PCnet-ISA controller has no buffer in which to store the incoming message. The MISS bit will be set in CSR0 and an interrupt will be generated if IENA = 1 (CSR0) and MISSM = 0 (CSR3). Another poll of the current RDTE will not occur until the packet has finished.

If the PCnet-ISA controller sees that the last poll (either a normal poll or the last-ditch effort described in the above paragraph) of the current RDTE shows valid ownership, then it proceeds to a poll of the next RDTE. Following this poll, and regardless of the outcome of this poll, transfers of receive data from the FIFO may begin.

Regardless of ownership of the second receive descriptor, the PCnet-ISA controller will continue to perform receive data DMA transfers to the first buffer, using burst-cycle DMA transfers. If the packet length exceeds the length of the first buffer, and the PCnet-ISA controller does not own the second buffer, ownership of the current descriptor will be passed back to the system by writing a zero to the OWN bit of RMD1 and status will be written indicating buffer (BUFF = 1) and possibly overflow (OFLO = 1) errors.

If the packet length exceeds the length of the first (current) buffer, and the PCnet-ISA controller does own the second (next) buffer, ownership will be passed back to the system by writing a zero to the OWN bit of RMD1 when the first buffer is full. Receive data transfers to the second buffer may occur before the PCnet-ISA controller proceeds to look ahead to the ownership of the third buffer. Such action will depend upon the state of the FIFO when the status has been updated on the first de-

scriptor. In any case, lookahead will be performed to the third buffer and the information gathered will be stored in the chip, regardless of the state of the ownership bit. As in the transmit flow, lookahead operations are performed only once.

This activity continues until the PCnet-ISA controller recognizes the completion of the packet (the last byte of this receive message has been removed from the FIFO). The PCnet-ISA controller will subsequently update the current RDTE status with the end of packet (ENP) indication set, write the message byte count (MCNT) of the complete packet into RMD2 and overwrite the "current" entries in the CSRs with the "next" entries.

Media Access Control

The Media Access Control engine incorporates the essential protocol requirements for operation of a compliant Ethernet/802.3 node, and provides the interface between the FIFO sub-system and the Manchester Encoder/Decoder (MENDEC).

The MAC engine is fully compliant to Section 4 of ISO/IEC 8802-3 (ANSI/IEEE Standard 1990 Second Edition) and ANSI/IEEE 802.3 (1985).

The MAC engine provides programmable enhanced features designed to minimize host supervision and pre or post-message processing. These features include the ability to disable retries after a collision, dynamic FCS generation on a packet-by-packet basis, and automatic pad field insertion and deletion to enforce minimum frame size attributes.

The two primary attributes of the MAC engine are:

- Transmit and receive message data encapsulation
  - — Framing (frame boundary delimitation, frame synchronization)
  - — Addressing (source and destination address handling)
  - — Error detection (physical medium transmission errors)
- Media access management
  - — Medium allocation (collision avoidance)
  - — Contention resolution (collision handling)

Transmit And Receive Message Data Encapsulation

The MAC engine provides minimum frame size enforcement for transmit and receive packets. When APAD_XMT = 1 (bit 11 in CSR4), transmit messages will be padded with sufficient bytes (containing 00h) to ensure that the receiving station will observe an information field (destination address, source address, length/type, data and FCS) of 64 bytes. When ASTRP_RCV = 1 (bit 10 in CSR4), the receiver will automatically strip pad bytes from the received message by observing the value in the length field, and stripping excess bytes if this value is below the minimum data size (46 bytes). Both features can be independently overridden to allow illegally short (less than 64 bytes of packet data) messages to be transmitted and/or received.

Framing (Frame Boundary Delimitation, Frame Synchronization)

The MAC engine will autonomously handle the construction of the transmit frame. Once the Transmit FIFO has been filled to the predetermined threshold (set by XMTSP in CSR80), and providing access to the channel is currently permitted, the MAC engine will commence the 7-byte preamble sequence (10101010b, where first bit transmitted is a 1). The MAC engine will subsequently append the Start Frame Delimiter (SFD) byte (10101011b) followed by the serialized data from the Transmit FIFO. Once the data has been completed, the MAC engine will append the FCS (most significant bit first) which was computed on the entire data portion of the message.

Note that the user is responsible for the correct ordering and content in each of the fields in the frame, including the destination address, source address, length/type and packet data.

The receive section of the MAC engine will detect an incoming preamble sequence and lock to the encoded clock. The internal MENDEC will decode the serial bit stream and present this to the MAC engine. The MAC will discard the first 8 bits of information before searching for the SFD sequence. Once the SFD is detected, all subsequent bits are treated as part of the frame. The MAC engine will inspect the length field to ensure minimum frame size, strip unnecessary pad characters (if enabled), and pass the remaining bytes through the Receive FIFO to the host. If pad stripping is performed, the MAC engine will also strip the received FCS bytes, although the normal FCS computation and checking will occur. Note that apart from pad stripping, the frame will be passed unmodified to the host. If the length field has a value of 46 or greater, the MAC engine will not attempt to validate the length against the number of bytes contained in the message.

If the frame terminates or suffers a collision before 64 bytes of information (after SFD) have been received, the MAC engine will automatically delete the frame from the Receive FIFO, without host intervention.

Addressing (Source and Destination Address Handling)

The first 6 bytes of information after SFD will be interpreted as the destination address field. The MAC engine provides facilities for physical, logical, and broadcast address reception. In addition, multiple physical addresses can be constructed (perfect address filtering) using external logic in conjunction with the EADI™ interface.

Error Detection (Physical Medium Transmission Errors)

The MAC engine provides several facilities which report and recover from errors on the medium. In addition, the

network is protected from gross errors due to inability of the host to keep pace with the MAC engine activity.

On completion of transmission, the following transmit status is available in the appropriate TMD and CSR areas:

- The exact number of transmission retry attempts (ONE, MORE, or RTRY)
- Whether the MAC engine had to Defer (DEF) due to channel activity
- Loss of Carrier, indicating that there was an interruption in the ability of the MAC engine to monitor its own transmission. Repeated LCAR errors indicate a potentially faulty transceiver or network connection.
- Late Collision (LCOL) indicates that the transmission suffered a collision after the slot time. This is indicative of a badly configured network. Late collisions should not occur in a normal operating network.
- Collision Error (CERR) indicates that the transceiver did not respond with an SQE Test message within the predetermined time after a transmission completed. This may be due to a failed transceiver, disconnected or faulty transceiver drop cable, or the fact the transceiver does not support this feature (or the feature is disabled).

In addition to the reporting of network errors, the MAC engine will also attempt to prevent the creation of any network error due to the inability of the host to service the MAC engine. During transmission, if the host fails to keep the Transmit FIFO filled sufficiently, causing an underflow, the MAC engine will guarantee the message is either sent as a runt packet (which will be deleted by the receiving station) or has an invalid FCS (which will also cause the receiver to reject the message).

The status of each receive message is available in the appropriate RMD and CSR areas. FCS and Framing errors (FRAM) are reported, although the received frame is still passed to the host. The FRAM error will only be reported if an FCS error is detected and there are a non-integral number of bits in the message. The MAC engine will ignore up to seven additional bits at the end of a message (dribbling bits), which can occur under normal network operating conditions. The reception of eight additional bits will cause the MAC engine to de-serialize the entire byte, and will result in the received message and FCS being modified.

The PCnet-ISA controller can handle up to 7 dribbling bits when a received packet terminates. During the reception, the CRC is generated on every serial bit (including the dribbling bits) coming from the cable, although the internally saved CRC value is only updated on the eighth bit (on each byte boundary). The framing error is reported to the user as follows:

1. If the number of the dribbling bits are 1 to 7 and there is no CRC error, then there is no Framing error (FRAM = 0).
2. If the number of the dribbling bits are less than 8 and there is a CRC error, then there is also a Framing error (FRAM = 1).
3. If the number of dribbling bits = 0, then there is no Framing error. There may or may not be a CRC (FCS) error.

Counters are provided to report the Receive Collision Count and Runt Packet Count and used for network statistics and utilization calculations.

Note that if the MAC engine detects a received packet which has a 00b pattern in the preamble (after the first 8 bits, which are ignored), the entire packet will be ignored. The MAC engine will wait for the network to go inactive before attempting to receive the next packet.

Media Access Management

The basic requirement for all stations on the network is to provide fairness of channel allocation. The 802.3/Ethernet protocol defines a media access mechanism which permits all stations to access the channel with equality. Any node can attempt to contend for the channel by waiting for a predetermined time (Inter Packet Gap interval) after the last activity, before transmitting on the medium. The channel is a multidrop communications medium (with various topological configurations permitted) which allows a single station to transmit and all other stations to receive. If two nodes simultaneously contend for the channel, their signals will interact, causing loss of data (defined as a collision). It is the responsibility of the MAC to attempt to avoid and recover from a collision, to guarantee data integrity for the end-to-end transmission to the receiving station.

Medium Allocation (Collision Avoidance)

The IEEE 802.3 Standard (ISO/IEC 8802-3 1990) requires that the CSMA/CD MAC monitor the medium traffic by looking for carrier activity. When carrier is detected the medium is considered busy, and the MAC should defer to the existing message.

The IEEE 802.3 Standard also allows optional two part deferral after a receive message.

See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.1:

> ***"Note**: It is possible for the PLS carrier sense indication to fail to be asserted during a collision on the media. If the deference process simply times the interpacket gap based on this indication it is possible for a short interFrame gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness the following optional measures, as specified in 4.2.8, are recom-*

mended when InterFrameSpacingPart1 is other than zero:

(1) Upon completing a transmission, start timing the interpacket gap, as soon as transmitting and carrierSense are both false.

(2) When timing an interpacket gap following reception, reset the interpacket gap timing if carrier Sense becomes true during the first 2/3 of the interpacket gap timing interval. During the final 1/3 of the interval the timer shall not be reset to ensure fair access to the medium. An initial period shorter than 2/3 of the interval is permissible including zero."

The MAC engine implements the optional receive two part deferral algorithm, with a first part inter-frame-spacing time of 6.0 µs. The second part of the inter-frame-spacing interval is therefore 3.6 µs.

The PCnet-ISA controller will perform the two-part deferral algorithm as specified in Section 4.2.8 (Process Deference). The Inter Packet Gap (IPG) timer will start timing the 9.6 µs InterFrameSpacing after the receive carrier is de-asserted. During the first part deferral (InterFrameSpacingPart1 - IFS1) the PCnet-ISA controller will defer any pending transmit frame and respond to the receive message. The IPG counter will be reset to zero continuously until the carrier de-asserts, at which point the IPG counter will resume the 9.6 µs count once again. Once the IFS1 period of 6.0 µs has elapsed, the PCnet-ISA controller will begin timing the second part deferral (InterFrameSpacingPart2 - IFS2) of 3.6 µs. Once IFS1 has completed, and IFS2 has commenced, the PCnet-ISA controller will not defer to a receive packet if a transmit packet is pending. This means that the PCnet-ISA controller will not attempt to receive the receive packet, since it will start to transmit, and generate a collision at 9.6 µs. The PCnet-ISA controller will guarantee to complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

In addition, transmit two part deferral is implemented as an option which can be disabled using the DXMT2PD bit (CSR3). Two-part deferral after transmission is useful for ensuring that severe IPG shrinkage cannot occur in specific circumstances, causing a transmit message to follow a receive message so closely as to make them indistinguishable.

During the time period immediately after a transmission has been completed, the external transceiver (in the case of a standard AUI connected device), should generate the SQE Test message (a nominal 10 MHz burst of 5-15 Bit Times duration) on the CI± pair (within 0.6 µs – 1.6 µs after the transmission ceases). During the time period in which the SQE Test message is expected the PCnet-ISA controller will not respond to receive carrier sense.

See ANSI/IEEE Std 802.3-1990 Edition, 7.2.4.6 (1)):

"At the conclusion of the output function, the DTE opens a time window during which it expects to see the signal_quality_error *signal asserted on the Control In circuit. The time window begins when the CARRIER_STATUS becomes CARRIER_OFF. If execution of the output function does not cause CARRIER_ON to occur, no SQE test occurs in the DTE. The duration of the window shall be at least 4.0 µs but no more than 8.0 µs. During the time window the Carrier Sense Function is inhibited."*

The PCnet-ISA controller implements a carrier sense "blinding" period within 0 – 4.0 µs from deassertion of carrier sense after transmission. This effectively means that when transmit two part deferral is enabled (DXMT2PD is cleared) the IFS1 time is from 4 µs to 6 µs after a transmission. However, since IPG shrinkage below 4 µs will rarely be encountered on a correctly configured network, and since the fragment size will be larger than the 4 µs blinding window, then the IPG counter will be reset by a worst case IPG shrinkage/fragment scenario and the PCnet-ISA controller will defer its transmission. In addition, the PCnet-ISA controller will not restart the "blinding" period if carrier is detected within the 4.0 µs – 6.0 µs IFS1 period, but will commence timing of the entire IFS1 period.

Contention Resolution (Collision Handling)

Collision detection is performed and reported to the MAC engine by the integrated Manchester Encoder/Decoder (MENDEC).

If a collision is detected before the complete preamble/SFD sequence has been transmitted, the MAC Engine will complete the preamble/SFD before appending the jam sequence. If a collision is detected after the preamble/SFD has been completed, but prior to 512 bits being transmitted, the MAC Engine will abort the transmission, and append the jam sequence immediately. The jam sequence is a 32-bit all zeroes pattern.

The MAC Engine will attempt to transmit a frame a total of 16 times (initial attempt plus 15 retries) due to normal collisions (those within the slot time). Detection of collision will cause the transmission to be re-scheduled, dependent on the backoff time that the MAC Engine computes. If a single retry was required, the ONE bit will be set in the Transmit Frame Status (TMD1 in the Transmit Descriptor Ring). If more than one retry was required, the MORE bit will be set. If all 16 attempts experienced collisions, the RTRY bit (in TMD2) will be set (ONE and MORE will be clear), and the transmit message will be flushed from the FIFO. If retries have been disabled by setting the DRTY bit in the MODE register (CSR15), the MAC Engine will abandon transmission of the frame on detection of the first collision. In this case, only the RTRY bit will be set and the transmit message will be flushed from the FIFO.

If a collision is detected after 512 bit times have been transmitted, the collision is termed a late collision. The

MAC Engine will abort the transmission, append the jam sequence, and set the LCOL bit. No retry attempt will be scheduled on detection of a late collision, and the FIFO will be flushed.

The IEEE 802.3 Standard requires use of a "truncated binary exponential backoff" algorithm which provides a controlled pseudo-random mechanism to enforce the collision backoff interval, before re-transmission is attempted.

See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.5:

"At the end of enforcing a collision (jamming), the CSMA/CD sublayer delays before attempting to re-transmit the frame. The delay is an integer multiple of slotTime. The number of slot times to delay before the nth re-transmission attempt is chosen as a uniformly distributed random integer r in the range:

$0 \leq r < 2^k$, *where* $k = min(n,10)$."

The PCnet-ISA controller provides an alternative algorithm, which suspends the counting of the slot time/IPG during the time that receive carrier sense is detected. This algorithm aids in networks where large numbers of nodes are present, and numerous nodes can be in collision. The algorithm effectively accelerates the increase in the backoff time in busy networks, and allows nodes not involved in the collision to access the channel while the colliding nodes await a reduction in channel activity. Once channel activity is reduced, the nodes resolving the collision time out their slot time counters as normal.

Manchester Encoder/Decoder (MENDEC)

The integrated Manchester Encoder/Decoder provides the PLS (Physical Layer Signaling) functions required for a fully compliant IEEE 802.3 station. The MENDEC provides the encoding function for data to be transmitted on the network using the high accuracy on-board oscillator, driven by either the crystal oscillator or an external CMOS-level compatible clock. The MENDEC also provides the decoding function from data received from the network. The MENDEC contains a Power On Reset (POR) circuit, which ensures that all analog portions of the PCnet-ISA controller are forced into their correct state during power-up, and prevents erroneous data transmission and/or reception during this time.

External Crystal Characteristics

When using a crystal to drive the oscillator, the crystal specification shown in the table may be used to ensure less than ±0.5 ns jitter at DO±.

Table : External Crystal Characteristics

Parameter	Min	Nom	Max	Units
1. Parallel Resonant Frequency		20		MHz
2. Resonant Frequency Error (CL = 20 pF)	–50		+50	PPM
3. Change in Resonant Frequency With Respect To Temperature (0° – 70° C; CL = 20 pF)*	–40		+40	PPM
4. Crystal Capacitance			20	pF
5. Motional Crystal Capacitance (C1)		0.022		pF
6. Series Resistance			25	Ω
7. Shunt Capacitance			7	pF
8. Drive Level			TBD	mW

** Requires trimming crystal spec; no trim is 50 ppm total*

External Clock Drive Characteristics

When driving the oscillator from an external clock source, XTAL2 must be left floating (unconnected). An external clock having the following characteristics must be used to ensure less than ±0.5 ns jitter at DO±:

Clock Frequency:	20 MHz ±0.01%
Rise/Fall Time (tR/tF):	< 6 ns from 0.5 V to V_{DD}–0.5
XTAL1 HIGH/LOW Time (tHIGH/tLOW):	20 ns min
XTAL1 Falling Edge to Falling Edge Jitter:	< ±0.2 ns at 2.5 V input (V_{DD}/2)

MENDEC Transmit Path

The transmit section encodes separate clock and NRZ data input signals into a standard Manchester encoded serial bit stream. The transmit outputs (DO±) are designed to operate into terminated transmission lines. When operating into a 78 Ω terminated transmission line, the transmit signaling meets the required output levels and skew for Cheapernet, Ethernet, and IEEE-802.3.

Transmitter Timing and Operation

A 20 MHz fundamental-mode crystal oscillator provides the basic timing reference for the MENDEC portion of the PCnet-ISA controller. The crystal input is divided by two to create the internal transmit clock reference. Both clocks are fed into the Manchester Encoder to generate the transitions in the encoded data stream. The internal transmit clock is used by the MENDEC to internally synchronize the Internal Transmit Data (ITXDAT) from the controller and Internal Transmit Enable (ITXEN). The internal transmit clock is also used as a stable bit-rate clock by the receive section of the MENDEC and controller.

The oscillator requires an external 0.005% crystal, or an external 0.01% CMOS-level input as a reference. The accuracy requirements, if an external crystal is used, are tighter because allowance for the on-chip oscillator must be made to deliver a final accuracy of 0.01%.

Transmission is enabled by the controller. As long as the ITXEN request remains active, the serial output of the controller will be Manchester encoded and appear at DO±. When the internal request is dropped by the controller, the differential transmit outputs go to one of two idle states, dependent on TSEL in the Mode Register (CSR15, bit 9):

TSEL LOW:	The idle state of DO± yields "zero" differential to operate transformer-coupled loads.
TSEL HIGH:	In this idle state, DO+ is positive with respect to DO– (logical HIGH).

Receive Path

The principal functions of the receiver are to signal the PCnet-ISA controller that there is information on the receive pair, and to separate the incoming Manchester encoded data stream into clock and NRZ data.

The receiver section (see Receiver Block Diagram) consists of two parallel paths. The receive data path is a zero threshold, wide bandwidth line receiver. The carrier path is an offset threshold bandpass detecting line receiver. Both receivers share common bias networks to allow operation over a wide input common mode range.

Input Signal Conditioning

Transient noise pulses at the input data stream are rejected by the Noise Rejection Filter. Pulse width rejection is proportional to transmit data rate. DC inputs more negative than minus 100 mV are also suppressed.

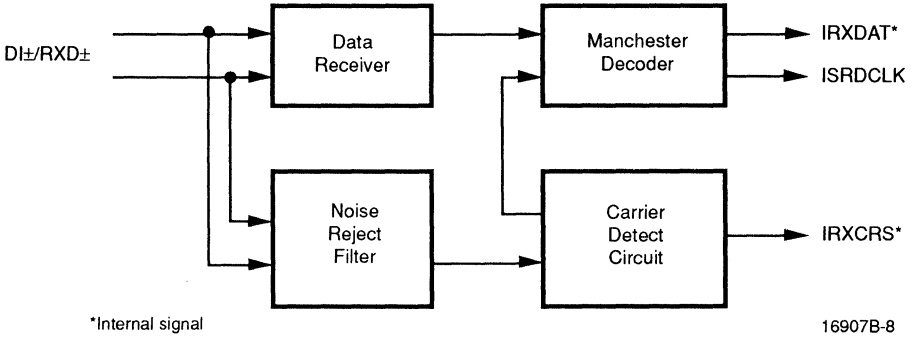
The Carrier Detection circuitry detects the presence of an incoming data packet by discerning and rejecting noise from expected Manchester data, and controls the stop and start of the phase-lock loop during clock acquisition. Clock acquisition requires a valid Manchester bit pattern of 1010b to lock onto the incoming message.

When input amplitude and pulse width conditions are met at DI±, a clock acquisition cycle is initiated.

Clock Acquisition

When there is no activity at DI± (receiver is idle), the receive oscillator is phase-locked to the internal transmit clock. The first negative clock transition (bit cell center of first valid Manchester "0") after IRXCRS is asserted interrupts the receive oscillator. The oscillator is then restarted at the second Manchester "0" (bit time 4) and is phase-locked to it. As a result, the MENDEC acquires the clock from the incoming Manchester bit pattern in 4 bit times with a "1010" Manchester bit pattern.

ISRDCLK and IRXDAT are enabled 1/4 bit time after clock acquisition in bit cell 5. IRXDAT is at a HIGH state when the receiver is idle (no ISRDCLK). IRXDAT however, is undefined when clock is acquired and may remain HIGH or change to LOW state whenever ISRDCLK is enabled. At 1/4 bit time through bit cell 5, the controller portion of the PCnet-ISA controller sees the first ISRDCLK transition. This also strobes in the incoming fifth bit to the MENDEC as Manchester "1". IRXDAT may make a transition after the ISRDCLK rising edge in bit cell 5, but its state is still undefined. The Manchester "1" at bit 5 is clocked to IRXDAT output at 1/4 bit time in bit cell 6.



Receiver Block Diagram

PLL Tracking

After clock acquisition, the phase-locked clock is compared to the incoming transition at the bit cell center (BCC) and the resulting phase error is applied to a correction circuit. This circuit ensures that the phase-locked clock remains locked on the received signal. Individual bit cell phase corrections of the Voltage Controlled Oscillator (VCO) are limited to 10% of the phase difference between BCC and phase-locked clock. Hence, input data jitter is reduced in ISRDCLK by 10 to 1.

Carrier Tracking and End of Message

The carrier detection circuit monitors the DI± inputs after IRXCRS is asserted for an end of message. IRXCRS de-asserts 1 to 2 bit times after the last positive transition on the incoming message. This initiates the end of reception cycle. The time delay from the last rising edge of the message to IRXCRS deassert allows the last bit to be strobed by ISRDCLK and transferred to the controller section, but prevents any extra bit(s) at the end of message.

Data Decoding

The data receiver is a comparator with clocked output to minimize noise sensitivity to the DI±/RXD± inputs. Input error is less than ± 35 mV to minimize sensitivity to input rise and fall time. ISRDCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit, and clocks the data out on IRXDAT on the following ISRDCLK. The data receiver also generates the signal used for phase detector comparison to the internal MENDEC voltage controlled oscillator (VCO).

Jitter Tolerance Definition

The MENDEC utilizes a clock capture circuit to align its internal data strobe with an incoming bit stream. The clock acquisition circuitry requires four valid bits with the values 1010b. Clock is phase-locked to the negative transition at the bit cell center of the second "0" in the pattern.

Since data is strobed at 1/4 bit time, Manchester transitions which shift from their nominal placement through 1/4 bit time will result in improperly decoded data. With this as the criteria for an error, a definition of "Jitter Handling" is:

> The peak deviation approaching or crossing 1/4 bit cell position from nominal input transition, for which the MENDEC section will properly decode data.

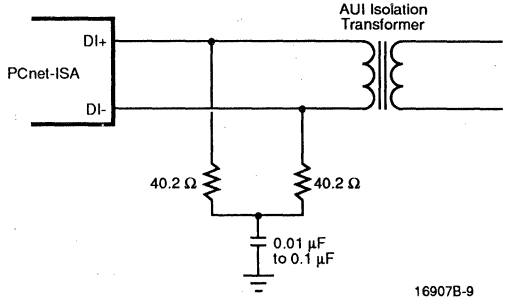
Attachment Unit Interface (AUI)

The AUI is the PLS (Physical Layer Signaling) to PMA (Physical Medium Attachment) interface which connects the DTE to a MAU. The differential interface provided by the PCnet-ISA controller is fully compliant with Section 7 of ISO 8802-3 (ANSI/IEEE 802.3).

After the PCnet-ISA controller initiates a transmission, it will expect to see data "looped-back" on the DI± pair (when the AUI port is selected). This will internally generate a "carrier sense", indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted within sometime before end of transmission. If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in the Transmit Descriptor Ring (TMD3, bit 11) after the packet has been transmitted.

Differential Input Terminations

The differential input for the Manchester data (DI±) is externally terminated by two 40.2 Ω ±1% resistors and one optional common-mode bypass capacitor, as shown in the Differential Input Termination diagram below. The differential input impedance, ZIDF, and the common-mode input impedance, ZICM, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. If SIP devices are used, 39 Ω is the nearest usable equivalent value. The CI± differential inputs are terminated in exactly the same way as the DI± pair.



Differential Input Termination

Collision Detection

A MAU detects the collision condition on the network and generates a differential signal at the CI± inputs. This collision signal passes through an input stage which detects signal levels and pulse duration. When the signal is detected by the MENDEC it sets the ICLSN line HIGH. The condition continues for approximately 1.5 bit times after the last LOW-to-HIGH transition on CI±.

Twisted Pair Transceiver (T-MAU)

The T-MAU implements the Medium Attachment Unit (MAU) functions for the Twisted Pair Medium, as specified by the supplement to IEEE 802.3 standard (Type 10BASE-T). The T-MAU provides twisted pair driver and receiver circuits, including on-board transmit digital predistortion and receiver squelch, and a number of additional features including Link Status indication, Automatic Twisted Pair Receive Polarity Detection/ Correction and indication, Receive Carrier Sense, Transmit Active and Collision Present indication.

Twisted Pair Transmit Function

The differential driver circuitry in the TXD± and TXP± pins provides the necessary electrical driving capability and the pre-distortion control for transmitting signals over maximum length Twisted Pair cable, as specified by the 10BASE-T supplement to the IEEE 802.3 Standard. The transmit function for data output meets the propagation delays and jitter specified by the standard.

Twisted Pair Receive Function

The receiver complies with the receiver specifications of the IEEE 802.3 10BASE-T Standard, including noise immunity and received signal rejection criteria ('Smart Squelch'). Signals meeting this criteria appearing at the RXD± differential input pair are routed to the MENDEC. The receiver function meets the propagation delays and jitter requirements specified by the standard. The receiver squelch level drops to half its threshold value after unsquelch to allow reception of minimum amplitude signals and to offset carrier fade in the event of worst case signal attenuation conditions.

Note that the 10BASE-T Standard defines the receive input amplitude at the external Media Dependent Interface (MDI). Filter and transformer loss are not specified. The T-MAU receiver squelch levels are designed to account for a 1 dB insertion loss at 10 MHz for the type of receive filters and transformers usually used.

Normal 10BASE-T compatible receive thresholds are invoked when the LRT bit (CSR15, bit 9) is LOW. When the LRT bit is set, the Low Receive Threshold option is invoked, and the sensitivity of the T-MAU receiver is increased. Increasing T-MAU sensitivity allows the use of lines longer than the 100 m target distance of standard 10BASE-T (assuming typical 24 AWG cable). Increased receiver sensitivity compensates for the increased signal attenuation caused by the additional cable distance.

However, making the receiver more sensitive means that it is also more susceptible to extraneous noise, primarily caused by coupling from co-resident services (crosstalk). For this reason, end users may wish to invoke the Low Receive Threshold option on 4-pair cable only. Multi-pair cables within the same outer sheath have lower crosstalk attenuation, and may allow noise emitted from adjacent pairs to couple into the receive pair, and be of sufficient amplitude to falsely unsquelch the T-MAU.

Link Test Function

The link test function is implemented as specified by 10BASE-T standard. During periods of transmit pair inactivity, 'Link beat pulses' will be periodically sent over the twisted pair medium to constantly monitor medium integrity.

When the link test function is enabled (DLNKTST bit in CSR15 is cleared), the absence of link beat pulses and receive data on the RXD± pair will cause the TMAU to go into the Link Fail state. In the Link Fail state, data transmission, data reception, data loopback and the collision detection functions are disabled and remain disabled until valid data or greater than 5 consecutive link pulses appear on the RXD± pair. During Link Fail, the Link Status (LNKST indicated by $\overline{\text{LED0}}$) signal is inactive. When the link is identified as functional, the LNKST signal is asserted, and $\overline{\text{LED0}}$ output will be activated.

In order to inter-operate with systems which do not implement Link Test, this function can be disabled by setting the DLNKTST bit. With Link Test disabled, the Data Driver, Receiver and Loopback functions as well as Collision Detection remain enabled irrespective of the presence or absence of data or link pulses on the RXD± pair. Link Test pulses continue to be sent regardless of the state of the DLNKTST bit.

Polarity Detection and Reversal

The T-MAU receive function includes the ability to invert the polarity of the signals appearing at the RXD± pair if the polarity of the received signal is reversed (such as in the case of a wiring error). This feature allows data packets received from a reverse wired RXD± input pair to be corrected in the T-MAU prior to transfer to the MENDEC. The polarity detection function is activated following reset or Link Fail, and will reverse the receive polarity based on both the polarity of any previous link beat pulses and the polarity of subsequent packets with a valid End Transmit Delimiter (ETD).

When in the Link Fail state, the T-MAU will recognize link beat pulses of either positive or negative polarity. Exit from the Link Fail state occurs at the reception of 5–6 consecutive link beat pulses of identical polarity. On entry to the Link Pass state, the polarity of the last 5 link beat pulses is used to determine the initial receive polarity configuration and the receiver is reconfigured to subsequently recognize only link beat pulses of the previously recognized polarity.

Positive link beat pulses are defined as transmitted signal with a positive amplitude greater than 585 mV (LRT = HIGH) with a pulse width of 60 ns–200 ns. This positive excursion may be followed by a negative excursion. This definition is consistent with the expected received signal at a correctly wired receiver, when a link beat pulse, which fits the template of Figure 14-12 of the 10BASE-T Standard, is generated at a transmitter and passed through 100 m of twisted pair cable.

Negative link beat pulses are defined as transmitted signals with a negative amplitude greater than 585 mV with a pulse width of 60 ns–200 ns. This negative excursion may be followed by a positive excursion. This definition is consistent with the expected received signal at a reverse wired receiver, when a link beat pulse which fits the template of Figure 14–12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

The polarity detection/correction algorithm will remain "armed" until two consecutive packets with valid ETD of identical polarity are detected. When "armed," the receiver is capable of changing the initial or previous polarity configuration according to the detected ETD polarity.

On receipt of the first packet with valid ETD following reset or Link Fail, the T-MAU will use the inferred polarity information to configure its RXD± input, regardless of its previous state. On receipt of a second packet with a valid ETD with correct polarity, the detection/correction algorithm will "lock-in" the received polarity. If the second (or subsequent) packet is not detected as confirming the previous polarity decision, the most recently detected ETD polarity will be used as the default. Note that packets with invalid ETD have no effect on updating the previous polarity decision. Once two consecutive packets with valid ETD have been received, the T-MAU will lock the correction algorithm until either a Link Fail condition occurs or RESET is asserted.

During polarity reversal, an internal POL signal will be active. During normal polarity conditions, this internal POL signal is inactive. The state of this signal can be read by software and/or displayed by LED when enabled by the LED control bits in the ISA Bus Configuration Registers (ISACSR5, 6, 7).

Twisted Pair Interface Status

Three signals (XMT, RCV and COL) indicate whether the T-MAU is transmitting, receiving, or in a collision state. These signals are internal signals and the behavior of the LED outputs depends on how the LED output circuitry is programmed.

The T-MAU will power up in the Link Fail state and the normal algorithm will apply to allow it to enter the Link Pass state. In the Link Pass state, transmit or receive activity will be indicated by assertion of RCV signal going active. If T-MAU is selected using the PORTSEL bits in CSR15 or MAUSEL pin, then when moving from AUI to T-MAU selection the T-MAU will be forced into the Link Fail state.

In the Link Fail state, XMT, RCV and COL are inactive.

Collision Detect Function

Activity on both twisted pair signals RXD± and TXD± constitutes a collision, thereby causing the COL signal to be asserted. (COL is used by the LED control circuits.) The COL will remain asserted until one of the two colliding signals changes from active to idle. During collision condition, data presented on the DI± pair will be sourced from the RXD± input. COL stays active for two bit times at the end of a collision.

Signal Quality Error (SQE) Test (Heartbeat) Function

The SQE function is disabled when the 10BASE-T port is selected and in Link Fail state.

Jabber Function

The Jabber function inhibits the twisted pair transmit function of the T-MAU if theTXD± circuit is active for an excessive period (20–150 ms). This prevents any one node from disrupting the network due to a 'stuck-on' or faulty transmitter. If this maximum transmit time is exceeded, the T-MAU transmitter circuitry is disabled, the JAB bit is set (CSR4, bit 1), and COL signal asserted. Once the transmit data stream to the T-MAU is removed, an "unjab" time of 250–750 ms will elapse before the T-MAU deasserts COL and re-enables the transmit circuitry.

Power Down

The T-MAU circuitry can be made to go into low power mode. This feature is useful in battery powered or low duty cycle systems. The T-MAU will go into power down mode when RESET is active, **coma mode** is active, or the T-MAU is not selected. Refer to the Power Down Mode section for a description of the various power down modes.

Any of the three conditions listed above resets the internal logic of the T-MAU and places the device into power down mode. In this mode, the Twisted Pair driver pins (TXD±,TXP±) are asserted LOW, and the internal T-MAU status signals (LNKST, RCVPOL, XMT, RCV and COLLISION) are inactive.

Once the $\overline{\text{SLEEP}}$ pin is deasserted, the T-MAU will be forced into the Link Fail state. The T-MAU will move to the Link Pass state only after 5 - 6 link beat pulses and/or a single received message is detected on the RXD± pair.

In **snooze** mode, the T-MAU receive circuitry will remain enabled even while the $\overline{\text{SLEEP}}$ pin is driven LOW.

The T-MAU circuitry will always go into power down mode if RESET is asserted, **coma mode** is enabled, or the T-MAU is not selected.

EADI (External Address Detection Interface)

This interface is provided to allow external address filtering. It is selected by setting the EADISEL bit in ISACSR2. This feature is typically utilized for terminal servers, bridges and/or router type products. The use of external logic is required to capture the serial bit stream from the PCnet-ISA controller, compare it with a table of stored addresses or identifiers, and perform the desired function.

The EADI interface operates directly from the NRZ decoded data and clock recovered by the Manchester decoder or input to the GPSI, allowing the external address detection to be performed in parallel with frame reception and address comparison in the MAC Station Address Detection (SAD) block.

SRDCLK is provided to allow clocking of the receive bit stream into the external address detection logic. SRDCLK runs only during frame reception activity. Once a received frame commences and data and clock are available, the EADI logic will monitor the alternating ("1,0") preamble pattern until the two ones of the Start Frame Delimiter ("1,0,1,0,1,0,1,1") are detected, at which point the SF/BD output will be driven HIGH.

After SF/BD is asserted the serial data from SRD should be de-serialized and sent to a content addressable memory (CAM) or other address detection device.

To allow simple serial to parallel conversion, SF/BD is provided as a strobe and/or marker to indicate the delineation of bytes, subsequent to the SFD. This provides a mechanism to allow not only capture and/or decoding of the physical or logical (group) address, it also facilitates the capture of header information to determine protocol and or inter-networking information. The $\overline{EAR}$ pin is driven LOW by the external address comparison logic to reject the frame.

If an internal address match is detected by comparison with either the Physical or Logical Address field, the frame will be accepted regardless of the condition of $\overline{EAR}$. Incoming frames which do not pass the internal address comparison will continue to be received. This allows approximately 58 byte times after the last destination address bit is available to generate the $\overline{EAR}$ signal, assuming the device is not configured to accept runt packets. $\overline{EAR}$ will be ignored after 64 byte times after the SFD, and the frame will be accepted if $\overline{EAR}$ has not been asserted before this time. If Runt Packet Accept is configured, the $\overline{EAR}$ signal must be generated prior to the receive message completion, which could be as short as 12 byte times (assuming 6 bytes for source address, 2 bytes for length, no data, 4 bytes for FCS) after the last bit of the destination address is available. $\overline{EAR}$ must have a pulse width of at least 200 ns.

Note that setting the PROM bit (CSR15, bit 15) will cause all receive frames to be received, regardless of the state of the $\overline{EAR}$ input.

If the DRCVPA bit (CSR15.13) is set and the logical address (LADRF) is set to zero, only frames which are not rejected by $\overline{EAR}$ will be received.

The EADI interface will operate as long as the STRT bit in CSR0 is set, even if the receiver and/or transmitter are disabled by software (DTX and DRX bits in CSR15 set). This situation is useful as a power down mode in that the PCnet-ISA controller will not perform any DMA operations; this saves power by not utilizing the ISA bus driver circuits. However, external circuitry could still respond to specific frames on the network to facilitate remote node control.

The table below summarizes the operation of the EADI features.

Table: Internal/External Address Recognition Capabilities

PROM	$\overline{EAR}$	Required Timing	Received Messages
1	X	No timing requirements	All Received Frames
0	1	No timing requirements	All Received Frames
0	0	Low for 200 ns within 512 bits after SFD	Physical/Logical Matches

General Purpose Serial Interface (GPSI)

The PCnet-ISA controller contains a General Purpose Serial Interface (GPSI) designed for testing the digital portions of the chip. The MENDEC, AUI, and twisted pair interface are by-passed once the device is set up in the special "test mode" for accessing the GPSI functions. Although this access is intended only for testing the device, some users may find the non-encoded data functions useful in some special applications. Note, however, that the GPSI functions can be accessed only when the PCnet-ISA devices operate as a bus master.

The PCnet-ISA GPSI signals are consistent with the LANCE digital serial interface. Since the GPSI functions can be accessed only through a special test mode, expect some loss of functionality to the device when the GPSI is invoked. The AUI and 10BASE-T analog interfaces are disabled along with the internal MENDEC logic. The LA (unlatched address) pins are removed and become the GPSI signals, therefore, only 20 bits of address space is available. The table below shows the GPSI pin configuration:

To invoke the GPSI signals, follow the procedure below:

1. After reset or I/O read of Reset Address, write 10b to PORTSEL bits in CSR15.
2. Set the ENTST bit in CSR4
3. Set the GPSIEN bit in CSR124 (see note below)

   (The pins LA17–LA23 will change function after the completion of the above three steps.)
4. Clear the ENTST bit in CSR4
5. Clear both media select bits in ISACSR2
6. Define the PORTSEL bits in the MODE register (CSR15) to be 10b to define GPSI port. The MODE register image is in the initialization block.

Note: *LA pins will be tristated before writing to CORETST bit. After writing to GPSIEN, LA[17–21] will be inputs, LA[22–23] will be outputs.*

Table: GPSI Pin Configurations

GPSI Function	GPSI I/O Type	LANCE/ C-LANCE GPSI Pin	PCnet-ISA GPSI Pin Function	PCnet-ISA Pin Number	PCnet-ISA Normal Pin Function
Receive Data	I	RX	RXDAT	5	LA17
Receive Clock	I	RCLK	SRDCLK	6	LA18
Receive Carrier Sense	I	RENA	RXCRS	7	LA19
Collision	I	CLSN	CLSN	9	LA20
Transmit Clock	I	TCLK	STDCLK	10	LA21
Transmit Enable	O	TENA	TXEN	11	LA22
Transmit Data	O	TX	TXDAT	12	LA23

Note: *The GPSI function is only available in the Bus Master mode of operation.*

IEEE 1149.1 Test Access Port Interface

An IEEE 1149.1 compatible boundary scan Test Access Port is provided for board-level continuity test and diagnostics. All digital input, output, and input/output pins are tested. Analog pins, including the AUI differential driver (DO±) and receivers (DI±, CI±), and the crystal input (XTAL1/XTAL2) pins, are tested. The T-MAU drivers TXD±, TXP±, and receiver RXD± are also tested.

The following is a brief summary of the IEEE 1149.1 compatible test functions implemented in the PCnet-ISA controller.

Boundary Scan Circuit

The boundary scan test circuit requires four extra pins (TCK, TMS, TDI and TDO), defined as the Test Access Port (TAP). It includes a finite state machine (FSM), an instruction register, a data register array, and a power-on reset circuit. Internal pull-up resistors are provided for the TDI, TCK, and TMS pins. The TCK pin must not be left unconnected. The boundary scan circuit remains active during sleep.

TAP FSM

The TAP engine is a 16-state FSM, driven by the Test Clock (TCK) and the Test Mode Select (TMS) pins. This FSM is in its reset state at power-up or RESET. An independent power-on reset circuit is provided to ensure the FSM is in the TEST_LOGIC_RESET state at power-up.

Supported Instructions

In addition to the minimum IEEE 1149.1 requirements (BYPASS, EXTEST and SAMPLE instructions), three additional instructions (IDCODE, TRIBYP and SETBYP) are provided to further ease board-level testing.

All unused instruction codes are reserved. See the table below for a summary of supported instructions.

Instruction Register and Decoding Logic

After hardware or software RESET, the IDCODE instruction is always invoked. The decoding logic gives signals to control the data flow in the DATA registers according to the current instruction.

Boundary Scan Register (BSR)

Each BSR cell has two stages. A flip-flop and a latch are used in the SERIAL SHIFT STAGE and the PARALLEL OUTPUT STAGE, respectively.

There are four possible operational modes in the BSR cell:

1	Capture
2	Shift
3	Update
4	System Function

Other Data Registers

(1) BYPASS REG (1 BIT)

(2) DEV ID REG (32 bits)

Bits 31–28:	Version
Bits 27–12:	Part number (0003H)
Bits 11–1:	Manufacturer ID. The 11 bit manufacturer ID code for AMD is 00000000001 according to JEDEC Publication 106-A.
Bit 0:	Always a logic 1

Table: IEEE 1149.1 Supported Instruction Summary

Instruction Name	Description	Selected Data Reg	Mode	Instruction Code
EXTEST	External Test	BSR	Test	0000
IDCODE	ID Code Inspection	ID REG	Normal	0001
SAMPLE	Sample Boundary	BSR	Normal	0010
TRIBYP	Force Tristate	Bypass	Normal	0011
SETBYP	Control Boundary To 1/0	Bypass	Test	0100
BYPASS	Bypass Scan	Bypass	Normal	1111

Power Savings Modes

The PCnet-ISA controller supports two hardware power-savings modes. Both are entered by asserting the SLEEP pin LOW.

In **coma** mode, the PCnet-ISA controller will go into deep sleep with no support to automatically wake itself up. Coma mode is enabled when the AWAKE bit in ISACSR2 is reset. This mode is the default power down mode.

In **snooze** mode, enabled by setting the AWAKE bit in ISACSR2 and driving the SLEEP pin LOW, the T-MAU receive circuitry will remain enabled even while the SLEEP pin is driven LOW. The LED0 output will also continue to function, indicating a good 10BASE-T link if there are link beat pulses or valid frames present. This LED0 pin can be used to drive a LED and/or external hardware that directly controls the SLEEP pin of the PCnet-ISA controller. This configuration effectively wakes the system when there is any activity on the 10BASE-T link.

Access Operations (Software)

We begin by describing how byte and word data are addressed on the ISA bus, including conversion cycles where 16-bit accesses are turned into 8-bit accesses because the resource accessed did not support 16-bit operations. Then we describe how registers and other resources are accessed. This section is for the device programmer, while the next section (bus cycles) is for the hardware designer.

I/O Resources

The PCnet-ISA controller has both I/O and memory resources. In the I/O space the resources are organized as indicated in the following table:

Offset	#Bytes	Register
0h	16	IEEE Address PROM
10h	2	RDP
12h	2	RAP (shared by RDP and IDP)
14h	2	Reset
16h	2	IDP

The PCnet-ISA controller does not respond to any addresses outside of the offset range 0-17h. I/O offsets 18h and up are not used by the PCnet-ISA controller.

I/O Register Access

The register address port (RAP) is shared by the register data port (RDP) and the ISACSR data port (IDP) to save registers. To access the Ethernet controller's RDP or IDP, the RAP should be written first, followed by the read or write access to the RDP or IDP. I/O register accesses should be coded as 16-bit accesses, even if the PCnet-ISA controller is hardware configured for 8-bit I/O bus cycles. It is acceptable (and transparent) for the motherboard to turn a 16-bit software access into two separate 8-bit hardware bus cycles. The motherboard accesses the low byte before the high byte and the PCnet-ISA controller has circuitry to specifically support this type of access.

The reset register causes a reset when read. Any value will be accepted and the cycle may be 8 or 16 bits wide. Writes are ignored.

All PCnet-ISA controller register accesses should be coded as 16-bit operations.

**Note that the RAP is cleared on Reset.*

Address PROM Access

The address PROM is an external memory device that contains the node's unique physical Ethernet address and any other data stored by the board manufacturer. The software accesses may be 8- or 16-bit.

Boot PROM Access

The boot PROM is an external memory resource located at the address selected by the IOAM0 and IOAM1 pins in bus master mode, or the BPAM input in shared memory mode. It may be software accessed as an 8- or 16-bit resource but the latter is recommended for best performance.

Static RAM Access

The static RAM is only present in the shared memory mode. It is located at the address selected by the SMAM input. It may be accessed as an 8- or 16-bit resource but the latter is recommended for best performance.

Bus Cycles (Hardware)

The PCnet-ISA controller supports both 8- and 16-bit hardware bus cycles. The following sections outline where any limitations apply based upon the architecture mode and/or the resource that is being accessed (PCnet-ISA controller registers, address PROM, boot PROM, or shared memory SRAM). For completeness, the following sections are arranged by architecture (Bus Master Mode or Shared Memory Mode). SRAM resources apply only to Shared Memory Mode.

All resources (registers, PROMs, SRAM) are presented to the ISA bus by the PCnet-ISA controller. With few exceptions, these resources can be configured for either 8-bit or 16-bit bus cycles. The I/O resources (registers, address PROM) are width configured using the IOCS16 pin on the PCnet-ISA controller. The memory resources (boot PROM, SRAM) are width configured by external hardware.

For 16-bit memory accesses, hardware external to the PCnet-ISA controller asserts MEMCS16 when either of the two memory resources is selected. The ISA bus requires that all memory resources within a block of 128 Kbytes be the same width, either 8- or 16-bits. The reason for this is that the MEMCS16 signal is generally a decode of the LA17-23 address lines. 16-bit memory

capability is desirable since two 8-bit accesses take the same amount of time as four 16-bit accesses.

All accesses to 8-bit resources (which do not return MEMCS16 or IOCS16) use SD0-7. If an odd byte is accessed, the Current Master swap buffer turns on. During an odd byte read the swap buffer copies the data from SD0-7 to the high byte. During an odd byte write the Current Master swap buffer copies the data from the high byte to SD0-7. The PCnet-ISA controller can be configured to be an 8-bit I/O resource even in a 16-bit system; this is accomplished by disconnecting IOCS16 from the ISA bus and tying IOCS16 to ground. It is recommended that the PCnet-ISA controller be hardware configured for 8-bit only I/O bus cycles for maximum compatibility with PC/AT clone motherboards.

When the PCnet-ISA controller is in an 8-bit system such as a PC/XT, SBHE and IOCS16 must be left unconnected (these signals do not exist in the PC/XT). This will force ALL resources (I/O and memory) to support only 8-bit bus cycles. The PCnet-ISA controller will function in an 8-bit system only if configured for Shared Memory Mode.

Accesses to 16-bit resources (which do return MEMCS16 or IOCS16) use either or both SD0-7 and SD8-15. A word access is indicated by A0=0 and SBHE=0 and data is transferred on all 16 data lines. An even byte access is indicated by A0=0 and SBHE=1 and data is transferred on SD0-7. An odd-byte access is indicated by A0=1 and SBHE=0 and data is transferred on SD8-15. It is illegal to have A0=1 and SBHE=1 in any bus cycle. The PCnet-ISA controller returns only IOCS16; MEMCS16 must be generated by external hardware if desired. The use of MEMCS16 applies only to Shared Memory Mode.

The following table describes all possible types of ISA bus accesses, including Permanent Master as Current Master and PCnet-ISA controller as Current Master. The PCnet-ISA controller will not work with 8-bit memory while it is Current Master. Any descriptions of 8-bit memory accesses are for when the Permanent Master is Current Master.

The two byte columns (D0-7 and D8-15) indicate whether the bus master or slave is driving the byte. CS16 is a shorthand for MEMCS16 and IOCS16.

Bus Master Mode

The PCnet-ISA controller can be configured as a Bus Master only in systems that support bus mastering. In addition, the system is assumed to support 16-bit memory (DMA) cycles (the PCnet-ISA controler does not use the MEMCS16 signal on the ISA bus). This does not preclude the PCnet-ISA controller from doing 8-bit I/O transfers. The PCnet-ISA controller will not function as a bus master in 8-bit platforms such as the PC/XT.

Table: ISA Bus Accesses

R/W	A0	SBHE	CS16	D0-7	D8-15	Comments
RD	0	1	x	Slave	Float	Low byte RD
RD	1	0	1	Slave	Float*	High byte RD with swap
RD	0	0	1	Slave	Float	16-Bit RD converted to low byte RD
RD	1	0	0	Float	Slave	High byte RD
RD	0	0	0	Slave	Slave	16-Bit RD
WR	0	1	x	Master	Float	Low byte WR
WR	1	0	1	Float*	Master	High byte WR with swap
WR	0	0	1	Master	Master	16-Bit WR converted to low byte WR
WR	1	0	0	Float	Master	High byte WR
WR	0	0	0	Master	Master	16-Bit WR

**Motherboard SWAP logic drives*

Refresh Cycles

Although the PCnet-ISA controller is neither an originator or a receiver of refresh cycles, it does need to avoid unintentional activity during a refresh cycle in bus master mode. A refresh cycle is performed as follows: First, the REF signal goes active. Then a valid refresh address is placed on the address bus. MEMR goes active, the refresh is performed, and MEMR goes inactive. The refresh address is held for a short time and then goes invalid. Finally, REF goes inactive. During a refresh cycle, as indicated by REF being active, the PCnet-ISA controller inhibits its SMEMR inputs and ignores DACK if it goes active until it goes inactive. It is necessary to ignore DACK during a refresh because some motherboards generate a false DACK at that time.

Address PROM Cycles

The Address PROM is a small (16 bytes) 8-bit PROM connected to the PCnet-ISA controller Private Data Bus. The PCnet-ISA controller will support only 8-bit ISA I/O bus cycles for the address PROM; this limitation is transparent to software and does not preclude 16-bit software I/O accesses. An access cycle begins with the Permanent Master driving AEN LOW, driving the addresses valid, and driving IOR active. The PCnet-ISA controller detects this combination of signals and arbitrates for the Private Data Bus (PRDB) if necessary. IOCHRDY is driven LOW during accesses to the address PROM.

When the Private Data Bus becomes available, the PCnet-ISA controller drives APCS active, releases IOCHRDY, turns on the data path from PRD0-7, and enables the SD0-7 drivers (but not SD8-15). During this bus cycle, IOCS16 is not driven active. This condition is maintained until IOR goes inactive, at which time the bus cycle ends. Data is removed from SD0-7 within 30 ns.

Ethernet Controller Register Cycles

Ethernet controller registers (RAP, RDP, IDP) are naturally 16-bit resources but can be configured to operate with 8-bit bus cycles provided the proper protocol is followed. If IOCS16 has never gone HIGH since RESET, then all controller register bus cycles will be 8-bit only. This situation would occur if the IOCS16 pin is left unconnected to the ISA bus and tied to ground. This means on a read, the PCnet-ISA controller will only drive the low byte of the system data bus; if an odd byte is accessed, it will be swapped down. The high byte of the system data bus is never driven by the PCnet-ISA controller under these conditions. On a write cycle, the even byte is placed in a holding register. An odd byte write is internally swapped up and augmented with the even byte in the holding register to provide an internal 16-bit write. This allows the use of 8-bit I/O bus cycles which are more likely to be compatible with all ISA-compatible clones, but requires that both bytes be written in immediate succession. This is accomplished simply by treating the PCnet-ISA controller controller registers as 16-bit software resources. The motherboard will convert the 16-bit accesses done by software into two sequential 8-bit accesses, an even byte access followed immediately by an odd byte access.

An access cycle begins with the Permanent Master driving AEN LOW, driving the address valid, and driving IOR or IOW active. The PCnet-ISA controller detects this combination of signals and drives IOCHRDY LOW. IOCS16 will also be driven LOW if 16-bit I/O bus cycles are enabled. When the register data is ready, IOCHRDY will be released HIGH. This condition is maintained until IOR or IOW goes inactive, at which time the bus cycle ends.

RESET Cycles

A read to the reset address causes an PCnet-ISA controller reset. This has the same effect as asserting the RESET pin on the PCnet-ISA controller, such as happens during a system power-up or hard boot. The subsequent write cycle needed in the NE2100 LANCE based family of Ethernet cards is not required but does not have any harmful effects. IOCS16 is not asserted in this cycle.

ISA Configuration Register Cycles

The ISA configuration registers are accessed by placing the address of the desired register into the RAP and reading the IDP. The ISACSR bus cycles are identical to all other PCnet-ISA controller register bus cycles.

Boot PROM Cycles

The Boot PROM is an 8-bit PROM connected to the PCnet-ISA controller Private Data Bus (PRDB) and can occupy up to 16 Kbytes of address space. Since the PCnet-ISA controller does not generate MEMCS16, only 8-bit ISA memory bus cycles to the boot PROM are supported in Bus Master Mode; this limitation is transparent to software and does not preclude 16-bit software memory accesses. A boot PROM access cycle begins with the Permanent Master driving the addresses valid, REF inactive, and SMEMR active. (AEN is not involved in memory cycles). The PCnet-ISA controller detects this combination of signals, drives IOCHRDY LOW, and reads a byte out of the Boot PROM. The data byte read is driven onto the lower system data bus lines and IOCHRDY is released. This condition is maintained until SMEMR goes inactive, at which time the access cycle ends.

The BPCS signal generated by the PCnet-ISA controller is three 20 MHz clock cycles wide (150 ns). Including delays, the Boot PROM has 120 ns to respond to the BPCS signal from the PCnet-ISA controller. This signal is intended to be connected to the CS pin on the boot

PROM, with the PROM OE pin tied to ground. When using a PROM with an access time slower than 120 ns, BPCS may be connected to the OE pin of the boot PROM while tying the PROM CS pin to ground.

Current Master Operation

Current Master operation only occurs in the bus master mode. It does not occur in shared memory mode.

There are three phases to the use of the bus by the PCnet-ISA controller as Current Master, the Obtain Phase, the Access Phase, and the Release Phase.

Obtain Phase

A Master Mode Transfer Cycle begins by asserting DRQ. When the Permanent Master asserts $\overline{DACK}$, the PCnet-ISA controller asserts $\overline{MASTER}$, signifying it has taken control of the ISA bus. The Permanent Master tri-states the address, command, and data lines within 60 ns of $\overline{DACK}$ going active. The Permanent Master drives AEN inactive within 71 ns of $\overline{MASTER}$ going active.

Access Phase

The ISA bus requires a wait of at least 125 ns after $\overline{MASTER}$ is asserted before the new master is allowed to drive the address, command, and data lines. The PCnet-ISA controller will actually wait 3 clock cycles or 150 ns.

The following signals are not driven by the Permanent Master and are simply pulled HIGH: BALE, IOCHRDY, $\overline{IOCS16}$, $\overline{MEMCS16}$, $\overline{SRDY}$. Therefore, the PCnet-ISA controller assumes the memory which it is accessing is 16 bits wide and can complete an access in the time programmed for the PCnet-ISA controller $\overline{MEMR}$ and $\overline{MEMW}$ signals. Refer to the ISA Bus Configuration Register description section.

Release Phase

When the PCnet-ISA controller is finished with the bus, it drives the command lines inactive. 50 ns later, the controller tri-states the command, address, and data lines and drives DRQ inactive. 50 ns later, the controller drives $\overline{MASTER}$ inactive.

At least 375 ns after DRQ goes inactive, the Permanent Master drives $\overline{DACK}$ inactive.

The Permanent Master drives AEN active within 71 ns of $\overline{MASTER}$ going inactive. The Permanent Master is allowed to drive the command lines no sooner than 60 ns after $\overline{DACK}$ goes inactive.

Master Mode Memory Read Cycle

After the PCnet-ISA controller has acquired the ISA bus, it can perform a memory read cycle. All timing is generated relative to the 20 MHz clock (network clock). Since there is no way to tell if memory is 8- or 16-bit or when it is ready, the PCnet-ISA controller by default assumes 16-bit, 1 wait state memory. The wait state assumption is based on the default value in the MSRDA register in ISACSR0.

The cycle begins with SA0-19, $\overline{SBHE}$, and LA17-23 being presented. The ISA bus requires them to be valid for at least 28 ns before a read command and the PCnet-ISA controller provides one clock or 50 ns of setup time before asserting $\overline{MEMR}$.

The ISA bus requires $\overline{MEMR}$ to be active for at least 219 ns, and the PCnet-ISA controller provides a default of 5 clocks, or 250 ns, but this can be tuned for faster systems with the Master Mode Read Active (MSRDA) register (see section 2.5.2). Also, if IOCHRDY is driven LOW, the PCnet-ISA controller will wait. The wait state counter must expire and IOCHRDY must be HIGH for the PCnet-ISA controller to continue.

The PCnet-ISA controller then accepts the memory read data. The ISA bus requires all command lines to remain inactive for at least 97 ns before starting another bus cycle and the PCnet-ISA controller provides at least two clocks or 100 ns of inactive time.

The ISA bus requires read data to be valid no more than 173 ns after receiving $\overline{MEMR}$ active and the PCnet-ISA controller requires 10 ns of data setup time. The ISA bus requires read data to provide at least 0 ns of hold time and to be removed from the bus within 30 ns after $\overline{MEMR}$ goes inactive. The PCnet-ISA controller requires 0 ns of data hold time.

Master Mode Memory Write Cycle

After the PCnet-ISA controller has acquired the ISA bus, it can perform a memory write cycle. All timing is generated relative to a 20 MHz clock which happens to be the same as the network clock. Since there is no way to tell if memory is 8- or 16-bit or when it is ready, the PCnet-ISA controller by default assumes 16-bit, 1 wait state memory. The wait state assumption is based on the default value in the MSWRA register in ISACSR1.

The cycle begins with SA0-19, $\overline{SBHE}$, and LA17-23 being presented. The ISA bus requires them to be valid at least 28 ns before $\overline{MEMW}$ goes active and data to be valid at least 22 ns before $\overline{MEMW}$ goes active. The PCnet-ISA controller provides one clock or 50 ns of setup time for all these signals.

The ISA bus requires $\overline{MEMW}$ to be active for at least 219 ns, and the PCnet-ISA controller provides a default of 5 clocks, or 250 ns, but this can be tuned for faster systems with the Master Mode Write Active (MSWRA) register (ISACSR1). Also, if IOCHRDY is driven LOW, the PCnet-ISA controller will wait. IOCHRDY must be HIGH for the PCnet-ISA controller to continue.

The ISA bus requires data to be valid for at least 25 ns after $\overline{MEMW}$ goes inactive, and the PCnet-ISA controller provides one clock or 50 ns.

The ISA bus requires all command lines to remain inactive for at least 97 ns before starting another bus cycle. The PCnet-ISA controller provides at least two clocks or 100 ns of inactive time when bit 4 in ISACSR2 is set. The EISA bus requires all command lines to remain inactive for at least 170 ns before starting another bus cycle. When bit 4 in ISACSR4 is cleared, the PCnet-ISA controller provides 200 ns of inactive time.

Shared Memory Mode

Address PROM Cycles

The Address PROM is a small (16 bytes) 8-bit PROM connected to the PCnet-ISA controller Private Data Bus (PRDB). The PCnet-ISA controller will support only 8-bit ISA I/O bus cycles for the address PROM; this limitation is transparent to software and does not preclude 16-bit software I/O accesses. An access cycle begins with the Permanent Master driving AEN LOW, driving the addresses valid, and driving IOR active. The PCnet-ISA controller detects this combination of signals and arbitrates for the Private Data Bus if necessary. IOCHRDY is always driven LOW during address PROM accesses.

When the Private Data Bus becomes available, the PCnet-ISA controller drives APCS active, releases IOCHRDY, turns on the data path from PRD0-7, and enables the SD0-7 drivers (but not SD8-15). During this bus cycle, IOCS16 is not driven active. This condition is maintained until IOR goes inactive, at which time the access cycle ends. Data is removed from SD0-7 within 30 ns.

The PCnet-ISA controller will perform 8-bit ISA bus cycle operation for all resources (registers, PROMs, SRAM) if SBHE has been left unconnected, such as in the case of an 8-bit system like the PC/XT.

Ethernet Controller Register Cycles

Ethernet controller registers (RAP, RDP, ISACSR) are naturally 16-bit resources but can be configured to operate with 8-bit bus cycles provided the proper protocol is followed. If IOCS16 has never gone HIGH since RESET, then all controller register bus cycles will be 8-bit only. This situation would occur if the IOCS16 pin is disconnected from the ISA bus and tied to ground. This means on a read, the PCnet-ISA controller will only drive the low byte of the system data bus; if an odd byte is accessed, it will be swapped down. The high byte of the system data bus is never driven by the PCnet-ISA controller under these conditions. On a write, the even byte is placed in a holding register. An odd-byte write is internally swapped up and augmented with the even byte in the holding register to provide an internal 16-bit write. This allows the use of 8-bit I/O bus cycles which are more likely to be compatible with all clones, but requires that both bytes be written in immediate succession. This is accomplished simply by treating the PCnet-ISA controller controller registers as 16-bit software resources. The motherboard will convert the 16-bit accesses done by software into two sequential 8-bit accesses, an even-byte access followed immediately by an odd-byte access.

An access cycle begins with the Permanent Master driving AEN LOW, driving the address valid, and driving IOR or IOW active. The PCnet-ISA controller detects this combination of signals and drives IOCHRDY LOW. IOCS16 will also be driven LOW if 16-bit I/O bus cycles are enabled. When the register data is ready, IOCHRDY will be released HIGH. This condition is maintained until IOR or IOW goes inactive, at which time the bus cycle ends.

The PCnet-ISA controller will perform 8-bit ISA bus cycle operation for all resources (registers, PROMs, SRAM) if SBHE has been left unconnected, such as in the case of an 8-bit system like the PC/XT.

RESET Cycles

A read to the reset address causes an PCnet-ISA controller reset. This has the same effect as asserting the RESET pin on the PCnet-ISA controller, such as happens during a system power-up or hard boot. The subsequent write cycle needed in the NE2100 LANCE-based family of Ethernet cards is not required but does not have any harmful effects. IOCS16 is not asserted in this cycle.

ISA Configuration Register Cycles

The ISA configuration register is accessed by placing the address of the desired register into the RAP and reading the IDP. The ISACSR bus cycles are identical to all other PCnet-ISA controller register bus cycles.

Boot PROM Cycles

The Boot PROM is an 8-bit PROM connected to the PCnet-ISA controller Private Data Bus (PRDB), and can occupy up to 64 Kbytes of address space. In Shared Memory Mode, an external address comparator is responsible for asserting BPAM to the PCnet-ISA controller. BPAM is intended to be a perfect decode of the boot PROM address space, i.e. REF, LA17-23, SA14-16 for a 16 Kbyte PROM. The LA bus must be latched with BALE in order to provide stable signal for BPAM. REF inactive must be used by the external logic to gate boot PROM address decoding. This same logic must assert MEMCS16 to the ISA bus if 16-bit Boot PROM bus cycles are desired.

The PCnet-ISA controller assumes 16-bit ISA memory bus cycles for the boot PROM. A 16-bit boot PROM bus cycle begins with the Permanent Master driving the addresses valid, REF inactive, and SMEMR active. (AEN is not involved in memory cycles). External hardware would assert BPAM and MEMCS16. The PCnet-ISA controller detects this combination of signals, drives IOCHRDY LOW, and reads two bytes out of the boot PROM. The data bytes read from the PROM are driven by the PCnet-ISA controller onto SD0-15 and IOCHRDY is released. This condition is maintained until MEMR goes inactive, at which time the access cycle ends.

The PCnet-ISA controller can be made to support only 8-bit ISA memory bus cycles for the boot PROM. This can be accomplished by asserting BPAM and SMAM simultaneously; the PCnet-ISA controller would respond using 8-bit ISA memory bus cycles only. Since this is an illegal situation for simple address decoders, the external address decoder must artificially drive SMAM LOW when the (8-bit) boot PROM address space is being accessed. In this case, MEMCS16 must not be asserted.

The PCnet-ISA controller will perform 8-bit ISA bus cycle operation for all resource (registers, PROMs, SRAM) if SBHE has been left unconnected, such as in the case of an 8-bit system like the PC/XT.

The BPCS signal generated by the PCnet-ISA controller is three 20 MHz clock cycles wide (150 ns). Including delays, the Boot PROM has 120 ns to respond to the BPCS signal from the PCnet-ISA controller. This signal is intended to be connected to the CS pin on the boot PROM, with the PROM OE pin tied to ground. The access time of the boot PROM must be 120 ns or faster when 16-bit ISA memory cycles are to be supported.

Static RAM Cycles

The shared memory SRAM is an 8-bit device connected to the PCnet-ISA controller Private Bus, and can occupy up to 64 Kbytes of address space. In Shared Memory Mode, an external address comparator is responsible for asserting SMAM to the PCnet-ISA controller. SMAM is intended to be a perfect decode of the SRAM address space, i.e. REF, LA17-23, SA16 for 64 Kbytes of SRAM. The LA signals must be latched by BALE in order to provide a stable decode for SMAM. The PCnet-ISA controller assumes 16-bit ISA memory bus cycles for the SRAM, so this same logic must assert MEMCS16 to the ISA bus if 16-bit bus cycles are to be supported.

A 16-bit SRAM bus cycle begins with the Permanent Master driving the addresses valid, REF inactive, and either MEMR or MEMW active. (AEN is not involved in memory cycles). External hardware would assert SMAM and MEMCS16. The PCnet-ISA controller detects this combination of signals and initiates the SRAM access.

In a write cycle, the PCnet-ISA controller stores the data into an internal holding register, allowing the ISA bus cycle to finish normally. The data in the holding register will then be written to the SRAM without the need for ISA bus control. In the event the holding register is already filled with unwritten SRAM data, the PCnet-ISA controller will extend the ISA write cycle by driving IOCHRDY LOW until the unwritten data is stored in the SRAM. The current ISA bus cycle will then complete normally.

In a read cycle, the PCnet-ISA controller arbitrates for the Private Bus. If it is unavailable, the PCnet-ISA controller drives IOCHRDY LOW. When the Private Data Bus is available, the PCnet-ISA controller asserts the Address Buffer Output Enable (ABOE) signal to drive the upper 6 bits of the Private Address Bus from the System Address Bus. The PCnet-ISA controller itself drives the lower 10 bits of the Private Address Bus from the System Address Bus and compares the 16 bits of address on the Private Address Bus with that of a SRAM data word held in an internal pre-fetch buffer.

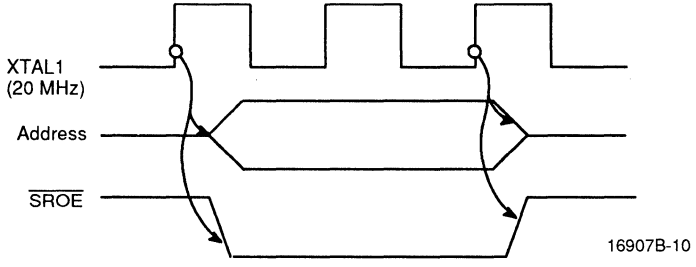
If the address does not match that of the prefetched SRAM data, then the PCnet-ISA controller drives IOCHRDY LOW and reads two bytes from the SRAM. The PCnet-ISA controller then proceeds as though the addressed data location had been prefetched.

If the internal prefetch buffer contains the correct data, then the pre-fetch buffer data is driven on the System Data bus. If IOCHRDY was previously driven LOW due to either Private Data Bus arbitration or SRAM access, then it is released HIGH. The PCnet-ISA controller remains in this state until MEMR is de-asserted, at which time the PCnet-ISA controller performs a new prefetch of the SRAM. In this way memory read wait states can be minimized.

The PCnet-ISA controller performs prefetches of the SRAM between ISA bus cycles. The SRAM is prefetched in an incrementing word address fashion. Prefetched data are invalidated by any other activity on the Private Bus, including Shared Memory Writes by either the ISA bus or the network interface, and also address and boot PROM reads.

The only way to configure the PCnet-ISA controller for 8-bit ISA bus cycles for SRAM accesses is to configure the entire PCnet-ISA controller to support only 8-bit ISA bus cycles. This is accomplished by leaving the SBHE pin disconnected. The PCnet-ISA controller will perform 8-bit ISA bus cycle operation for all resources (registers, PROMs, SRAM) if SBHE has never been driven active since the last RESET, such as in the case of an 8-bit system like the PC/XT. In this case, the external address decode logic must not assert MEMCS16 to the ISA bus, which will be the case if MEMCS16 is left unconnected. It is possible to manufacture a dual 8/16 bit PCnet-ISA controller adapter card, as the MEMCS16 and SBHE signals do not exist in the PC/XT environment.

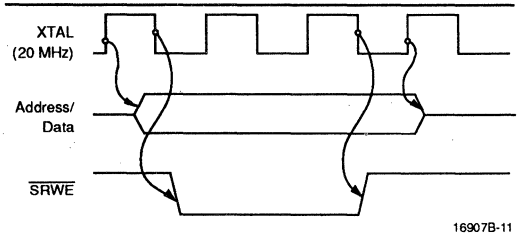
At the memory device level, each SRAM Private Bus read cycle takes two 50 ns clock periods for a maximum read access time of 75 ns. The timing looks like this:



Static RAM Read Cycle

The address and $\overline{SROE}$ go active within 20 ns of the clock going HIGH. Data is required to be valid 5 ns before the end of the second clock cycle. Address and $\overline{SROE}$ have a 0 ns hold time after the end of the second clock cycle. Note that the PCnet-ISA controller does not provide a separate SRAM $\overline{CS}$ signal; SRAM $\overline{CS}$ must always be asserted.

SRAM Private Bus write cycles require three 50 ns clock periods to guarantee non-negative address setup and hold times with regard to $\overline{SRWE}$. The timing is illustrated as follows:



Static RAM Write Cycle

Address and data are valid 20 ns after the rising edge of the first clock period. $\overline{SRWE}$ goes active 20 ns after the falling edge of the first clock period. $\overline{SRWE}$ goes inactive 20 ns after the falling edge of the third clock period. Address and data remain valid until the end of the third clock period. Rise and fall times are nominally 5 ns. Non-negative setup and hold times for address and data with respect to $\overline{SRWE}$ are guaranteed. $\overline{SRWE}$ has a pulse width of typically 100 ns, minimum 75 ns.

Transmit Operation

The transmit operation and features of the PCnet-ISA controller are controlled by programmable options.

Transmit Function Programming

Automatic transmit features, such as retry on collision, FCS generation/transmission, and pad field insertion, can all be programmed to provide flexibility in the (re-)transmission of messages.

Disable retry on collision (DRTY) is controlled by the DRTY bit of the Mode register (CSR15) in the initialization block.

Automatic pad field insertion is controlled by the APAD_XMT bit in CSR4. If APAD_XMT is set, automatic pad field insertion is enabled, the DXMTFCS feature is over-ridden, and the 4-byte FCS will be added to the transmitted frame unconditionally. If APAD_XMT is cleared, no pad field insertion will take place and runt packet transmission is possible.

The disable FCS generation/transmission feature can be programmed dynamically on a frame by frame basis. See the ADD_FCS description of TMD1.

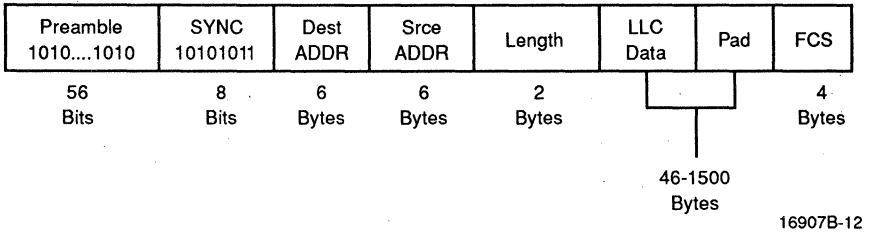
Transmit FIFO Watermark (XMTFW in CSR80) sets the point at which the BMU (Buffer Management Unit) requests more data from the transmit buffers for the FIFO. This point is based upon how many 16-bit bus transfers (2 bytes) could be performed to the existing empty space in the transmit FIFO.

Transmit Start Point (XMTSP in CSR80) sets the point when the transmitter actually tries to go out on the media. This point is based upon the number of bytes written to the transmit FIFO for the current frame.

When the entire frame is in the FIFO, attempts at transmission of preamble will commence regardless of the value in XMTSP. The default value of XMTSP is 10b, meaning 64 bytes full.

Automatic Pad Generation

Transmit frames can be automatically padded to extend them to 64 data bytes (excluding preamble). This allows the minimum frame size of 64 bytes (512 bits) for 802.3/Ethernet to be guaranteed with no software intervention from the host/controlling process. Setting the APAD_XMT bit in CSR4 enables the automatic padding feature. The pad is placed between the LLC data field and FCS field in the 802.3 frame. FCS is always added if the frame is padded, regardless of the state of DXMTFCS. The transmit frame will be padded by bytes with the value of 00h. The default value of APAD_XMT is 0, and this will disable auto pad generation after RESET.



ISO 8802–3 (IEEE/ANSI 802.3) Data Frame

It is the responsibility of upper layer software to correctly define the actual length field contained in the message to correspond to the total number of LLC Data bytes encapsulated in the packet (length field as defined in the IEEE 802.3 standard). The length value contained in the message is not used by the PCnet-ISA controller to compute the actual number of pad bytes to be inserted. The PCnet-ISA controller will append pad bytes dependent on the actual number of bits transmitted onto the network. Once the last data byte of the frame has completed prior to appending the FCS, the PCnet-ISA controller will check to ensure that 544 bits have been transmitted. If not, pad bytes are added to extend the frame size to this value, and the FCS is then added.

The 544 bit count is derived from the following:

Minimum frame size (excluding preamble, including FCS)	64 bytes	512 bits
Preamble/SFD size	8 bytes	64 bits
FCS size	4 bytes	32 bits

To be classed as a minimum-size frame at the receiver, the transmitted frame must contain:

Preamble + (Min Frame Size + FCS) bits

At the point that FCS is to be appended, the transmitted frame should contain:

Preamble + (Min Frame Size - FCS) bits
64 + (512 - 32) bits

A minimum-length transmit frame from the PCnet-ISA controller will, therefore, be 576 bits after the FCS is appended.

Transmit FCS Generation

Automatic generation and transmission of FCS for a transmit frame depends on the value of DXMTFCS bit in CSR15. When DXMTFCS = 0 the transmitter will generate and append the FCS to the transmitted frame. If the automatic padding feature is invoked (APAD_XMT is SET in CSR4), the FCS will be appended by the PCnet-ISA controller regardless of the state of DXMTFCS. Note that the calculated FCS is transmitted most-significant bit first. The default value of DXMTFCS is 0 after RESET.

Transmit Exception Conditions

Exception conditions for frame transmission fall into two distinct categories; those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-ISA controller are basically collisions within the slot time with automatic retry. The PCnet-ISA controller will ensure that collisions which occur within 512 bit times from the start of transmission (including preamble) will be automatically retried with no host intervention. The transmit FIFO ensures this by guaranteeing that data contained within the FIFO will not be overwritten until at least 64 bytes (512 bits) of data have been successfully transmitted onto the network.

If 16 total attempts (initial attempt plus 15 retries) fail, the PCnet-ISA controller sets the RTRY bit in the current transmit TDTE in host memory (TMD2), gives up ownership (sets the OWN bit to zero) for this packet, and processes the next packet in the transmit ring for transmission.

Abnormal network conditions include:

- Loss of carrier
- Late collision
- SQE Test Error (does not apply to 10BASE-T port)

These should not occur on a correctly configured 802.3 network, and will be reported if they do.

When an error occurs in the middle of a multi-buffer frame transmission, the error status will be written in the current descriptor. The OWN bit(s) in the subsequent descriptor(s) will be reset until the STP (the next frame) is found.

Loss of Carrier

A loss of carrier condition will be reported if the PCnet-ISA controller cannot observe receive activity while it is transmitting on the AUI port. After the PCnet-ISA controller initiates a transmission, it will expect to see data "looped back" on the DI± pair. This will internally generate a "carrier sense," indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted before the end of the transmission. If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in TMD2 after the frame has been transmitted. The frame will not be re-tried on the basis of an LCAR error. In 10BASE-T mode LCAR will indicate that Jabber or Link Fail state has occurred.

Late Collision

A late collision will be reported if a collision condition occurs after one slot time (512 bit times) after the transmit process was initiated (first bit of preamble commenced). The PCnet-ISA controller will abandon the transmit process for the particular frame, set Late Collision (LCOL) in the associated TMD3, and process the next transmit frame in the ring. Frames experiencing a late collision will not be re-tried. Recovery from this condition must be performed by upper-layer software.

SQE Test Error

During the inter packet gap time following the completion of a transmitted message, the AUI CI± pair is asserted by some transceivers as a self-test. The integral Manchester Encoder/Decoder will expect the SQE Test Message (nominal 10 MHz sequence) to be returned via the CI± pair within a 40 network bit time period after DI± pair goes inactive. If the CI± inputs are not asserted within the 40 network bit time period following

the completion of transmission, then the PCnet-ISA controller will set the CERR bit in CSR0. CERR will be asserted in 10BASE-T mode after transmit if T-MAU is in Link Fail state. CERR will never cause INTR to be activated. It will, however, set the ERR bit in CSR0.

Host related transmit exception conditions include BUFF and UFLO as described in the Transmit Descriptor section.

Receive Operation

The receive operation and features of the PCnet-ISA controller are controlled by programmable options.

Receive Function Programming

Automatic pad field stripping is enabled by setting the ASTRP_RCV bit in CSR4; this can provide flexibility in the reception of messages using the 802.3 frame format.

All receive frames can be accepted by setting the PROM bit in CSR15. When PROM is set, the PCnet-ISA controller will attempt to receive all messages, subject to minimum frame enforcement. Promiscuous mode overrides the effect of the Disable Receive Broadcast bit on receiving broadcast frames.

The point at which the BMU will start to transfer data from the receive FIFO to buffer memory is controlled by the RCVFW bits in CSR80. The default established during reset is 10b, which sets the threshold flag at 64 bytes empty.

Automatic Pad Stripping

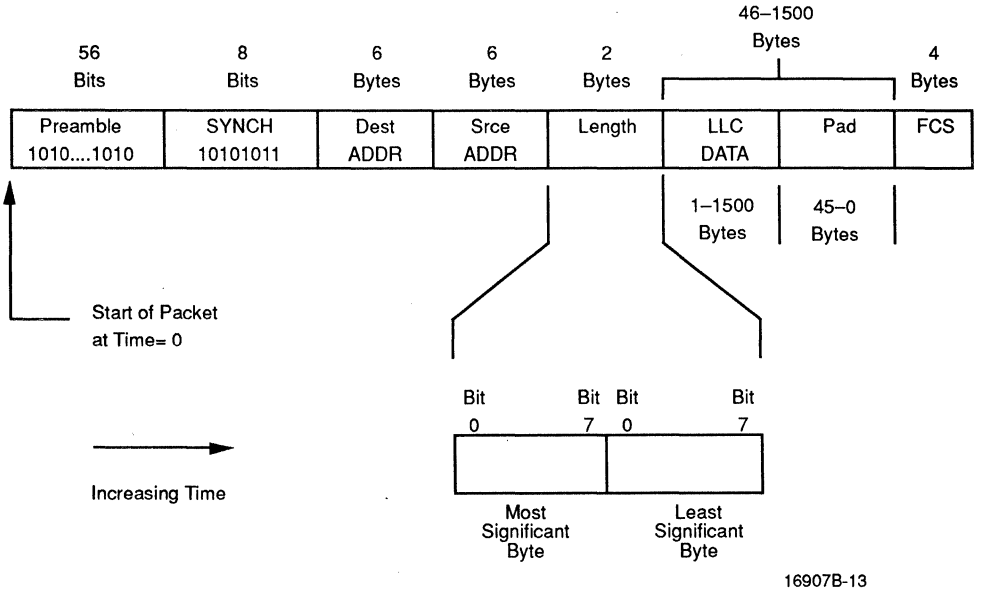
During reception of an 802.3 frame the pad field can be stripped automatically. ASTRP_RCV (bit 10 in CSR4) = 1 enables the automatic pad stripping feature. The pad field will be stripped before the frame is passed to the FIFO, thus preserving FIFO space for additional frames. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has had the pad characters stripped.

The number of bytes to be stripped is calculated from the embedded length field (as defined in the IEEE 802.3 definition) contained in the frame. The length indicates the actual number of LLC data bytes contained in the message. Any received frame which contains a length field less than 46 bytes will have the pad field stripped (if ASTRP_RCV is set). Receive frames which have a length field of 46 bytes or greater will be passed to the host unmodified.

Since any valid Ethernet Type field value will always be greater than a normal 802.3 Length field (≥46), the PCnet-ISA controller will not attempt to strip valid Ethernet frames.

Note that for some network protocols the value passed in the Ethernet Type and/or 802.3 Length field is not compliant with either standard and may cause problems.

The diagram below shows the byte/bit ordering of the received length field for an 802.3 compatible frame format.



802.3 Frame and Length Field Transmission Order

Receive FCS Checking

Reception and checking of the received FCS is performed automatically by the PCnet-ISA controller. Note that if the Automatic Pad Stripping feature is enabled, the received FCS will be verified against the value computed for the incoming bit stream including pad characters, but it will not be passed to the host. If a FCS error is detected, this will be reported by the CRC bit in RMD1.

Receive Exception Conditions

Exception conditions for frame reception fall into two distinct categories; those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-ISA controller are basically collisions within the slot time and automatic runt packet rejection. The PCnet-ISA controller will ensure that collisions which occur within 512 bit times from the start of reception (excluding preamble) will be automatically deleted from the receive FIFO with no host intervention. The receive FIFO will delete any frame which is composed of fewer than 64 bytes provided that the Runt Packet Accept (RPA bit in CSR124) feature has not been enabled. This criteria will be met regardless of whether the receive frame was the first (or only) frame in the FIFO or if the receive frame was queued behind a previously received message.

Abnormal network conditions include:

- FCS errors
- Late collision

These should not occur on a correctly configured 802.3 network and will be reported if they do.

Host related receive exception conditions include MISS, BUFF, and OFLO. These are described in the Receive Descriptor section.

Loopback Operation

During loopback, the FCS logic can be allocated to the receiver by setting the DXMTFCS bit in CSR15.

If DXMTFCS=0, the MAC Engine will calculate and append the FCS to the transmitted message. In this loopback configuration, the receive circuitry cannot detect FCS errors if they occur.

If DXMTFCS=1, the last four bytes of the transmit message must contain the (software generated) FCS computed for the transmit data preceding it. The MAC Engine will transmit the data without addition of an FCS field, and the FCS will be calculated and verified at the receiver.

The loopback facilities of the MAC Engine allow full operation to be verified without disturbance to the network. Loopback operation is also affected by the state of the Loopback Control bits (LOOP, MENDECL, and INTL) in CSR15. This affects whether the internal MENDEC is considered part of the internal or external loopback path.

When in the loopback mode(s), the multicast address detection feature of the MAC Engine, programmed by the contents of the Logical Address Filter (LADRF [63:0] in CSR 8-11) can only be tested when DXMTFCS= 1, allocating the FCS generator to the receiver. All other features operate identically in loopback as in normal operation, such as automatic transmit padding and receive pad stripping.

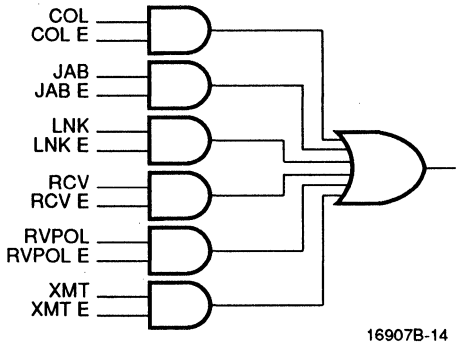
When performing an internal loopback, no frame will be transmitted to the network. However, when the PCnet-ISA controller is configured for internal loopback the receiver will not be able to detect network traffic. AUI external loopback tests will transmit frames onto the network, and the PCnet-ISA controller will receive network traffic while configured for external loopback. 10BASE-T external loopback should never be used in a live network. 10BASE-T external loopback provides a means of looping Transmit data to the receive input without asserting a collision. This mode allows a board test to verify both the transmit and receive paths to the 10BASE-T connector. Unless the Runt Packet Accept feature is enabled, all loopback frames must contain at least 64 bytes of data.

LEDs

The PCnet-ISA controller's LED control logic allows programming of the status signals, which are displayed on 3 LED outputs. One LED ($\overline{\text{LED0}}$) is dedicated to displaying 10BASE-T Link Status. The status signals available are Collision, Jabber, Receive, Receive Polarity (active when receive polarity is okay), and Transmit. If more than one status signal is enabled, they are ORed together. An optional pulse stretcher is available for each programmable output. This allows emulation of the TPEX (Am79C98) and TPEX+ (Am79C100) LED outputs.

Signal	Behavior
LNKST	Active during Link OK Not active during Link Down
RCV	Active while receiving data
RVPOL	Active during receive polarity is OK Not active during reverse receive polarity
XMT	Active while transmitting data

Each status signal is ANDed with its corresponding enable signal. The enabled status signals run to a common OR gate:



LED Control Logic

The output from the OR gate is run through a pulse stretcher, which consists of a 3-bit shift register clocked at 38 Hz. The data input of the shift register is at logic 0. The OR gate output asynchronously sets all three bits of the shift register when its output goes active. The output of the shift register controls the associated $\overline{\text{LEDx}}$ pin. Thus, the pulse stretcher provides an LED output of 52 ms to 78 ms.

PCnet-ISA CONTROLLER REGISTERS

The PCnet-ISA controller implements all LANCE (Am7990) registers, plus a number of additional registers. The PCnet-ISA controller registers are compatible with the original LANCE, but there are some places where previously reserved LANCE bits are now used by the PCnet-ISA controller. If the reserved LANCE bits were used as recommended, there should be no compatibility problems.

Register Access

Internal registers are accessed in a two-step operation. First, the address of the register to be accessed is written into the register address port (RAP). Subsequent read or write operations will access the register pointed to by the contents of the RAP. The data will be read from (or written to) the selected register through the data port, either the register data port (RDP) for control and status status registers (CSR) or the ISACSR register data port (IDP) for ISA control and status registers (ISACSR)

RAP: Register Address Port

Bit	Name	Description
15-7	RES	Reserved locations. Read and written as zeroes.
6-0	RAP	Register Address Port select. Selects the CSR or ISACSR location to be accessed. RAP is cleared by RESET.

Control and Status Registers

CSR0: PCnet-ISA Controller Status

Bit	Name	Description
15	ERR	Error is set by the ORing of BABL, CERR, MISS, and MERR. ERR remains set as long as any of the error flags are true. ERR is read only; write operations are ignored.
14	BABL	Babble is a transmitter time-out error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum length frame. BABL will be set if 1519 bytes or greater are transmitted. When BABL is set, IRQ is asserted if IENA = 1 and the mask bit BABLM (CSR3.14) is clear. BABL assertion will set the ERR bit. BABL is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. BABL is cleared by RESET or by setting the STOP bit.
13	CERR	Collision Error indicates that the collision inputs to the AUI port failed to activate within 20 network bit times after chip terminated transmission (SQE Test). This feature is a transceiver test feature. CERR will be set in 10BASE-T mode during trasmit if in Link Fail state. CERR assertion will not result in an interrupt being generated. CERR assertion will set the ERR bit. CERR is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. CERR is cleared by RESET or by setting the STOP bit.
12	MISS	Missed Frame is set when PCnet-ISA controller has lost an incoming receive frame because a Receive Descriptor was not available. This bit is the only indication that receive data has been lost since there is no receive descriptor available for status information. When MISS is set, IRQ is asserted if IENA = 1 and the mask bit MISSM (CSR3.12) is clear. MISS assertion will set the ERR bit. MISS is set by the Buffer Management Unit and cleared by writing a "1". Writing a "0" has no effect. MISS is cleared by RESET or by setting the STOP bit.
11	MERR	Memory Error is set when PCnet-ISA controller is a bus master and has not received DACK assertion after 50 µs after DRQ assertion. Memory Error indicates that PCnet-ISA controller is not receiving bus mastership in time to prevent overflow/underflow conditions in the receive and transmit FIFOs. (MERR indicates a slightly different condition for the LANCE; for the LANCE MERR occurs when READY has not been asserted 25.6 µs after the address has been asserted.) When MERR is set, IRQ is asserted if IENA = 1 and the mask bit MERRM (CSR3.11) is clear.

MERR assertion will set the ERR bit.

MERR is set by the Bus Interface Unit and cleared by writing a "1". Writing a "0" has no effect. MERR is cleared by RESET or by setting the STOP bit.

10 RINT — Receive Interrupt is set after reception of a receive frame and toggling of the OWN bit in the last buffer in the Receive Descriptor Ring.

When RINT is set, IRQ is asserted if IENA = 1 and the mask bit RINTM (CSR3.10) is clear.

RINT is set by the Buffer Management Unit after the last receive buffer has been updated and cleared by writing a "1". Writing a "0" has no effect. RINT is cleared by RESET or by setting the STOP bit.

9 TINT — Transmit Interrupt is set after transmission of a transmit frame and toggling of the OWN bit in the last buffer in the Transmit Descriptor Ring.

When TINT is set, IRQ is asserted if IENA = 1 and the mask bit TINTM (CSR3.9) is clear.

TINT is set by the Buffer Management Unit after the last transmit buffer has been updated and cleared by writing a "1". Writing a "0" has no effect. TINT is cleared by RESET or by setting the STOP bit.

8 IDON — Initialization Done indicates that the initialization sequence has completed. When IDON is set, PCnet-ISA controller has read the Initialization block from memory.

When IDON is set, IRQ is asserted if IENA = 1 and the mask bit IDONM (CSR3.8) is clear.

IDON is set by the Buffer Management Unit after the initialization block has been read from memory and cleared by writing a "1". Writing a "0" has no effect. IDON is cleared by RESET or by setting the STOP bit.

7 INTR — Interrupt Flag indicates that one or more of the following interrupt causing conditions has occurred: BABL, MISS, MERR, MPCO, RCVCCO, RINT, TINT, IDON, JAB or TXSTRT; and its associated mask bit is clear. If IENA = 1 and INTR is set, IRQ will be active.

INTR is cleared automatically when the condition that caused interrupt is cleared.

INTR is read only. INTR is cleared by RESET or by setting the STOP bit.

6 IENA — Interrupt Enable allows IRQ to be active if the Interrupt Flag is set. If IENA = "0" then IRQ will be disabled regardless of the state of INTR.

IENA is set by writing a "1" and cleared by writing a "0". IENA is cleared by RESET or by setting the STOP bit.

5 RXON — Receive On indicates that the Receive function is enabled. RXON is set if DRX (CSR15.0) = "0" after the START bit is set. If INIT and START are set together, RXON will not be set until after the initialization block has been read in.

RXON is read only. RXON is cleared by RESET or by setting the STOP bit.

4 TXON — Transmit On indicates that the Transmit function is enabled. TXON is set if DTX (CSR15.1) = "0" after the START bit is set. If INIT and START are set together, TXON will not be set until after the initialization block has been read in.

TXON is read only. TXON is cleared by RESET or by setting the STOP bit.

3 TDMD — Transmit Demand, when set, causes the Buffer Management Unit to access the Transmit Descriptor Ring without waiting for the poll-time counter to elapse. If TXON is not enabled, TDMD bit will be reset and no Transmit Descriptor Ring access will occur. TDMD is required to be set if the DPOLL bit in CSR4 is set; setting TDMD while DPOLL = 0 merely hastens the PCnet-ISA controller's response to a Transmit Descriptor Ring Entry.

TDMD is set by writing a "1". Writing a "0" has no effect. TDMD will be cleared by the Buffer Management Unit when it fetches a Transmit Descriptor. TDMD is cleared by RESET or by setting the STOP bit.

2	STOP	STOP assertion disables the chip from all external activity. The chip remains inactive until either STRT or INIT are set. If STOP, STRT and INIT are all set together, STOP will override STRT and INIT. STOP is set by writing a "1" or by RESET. Writing a "0" has no effect. STOP is cleared by setting either STRT or INIT.
1	STRT	STRT assertion enables PCnet-ISA controller to send and receive frames, and perform buffer management operations. Setting STRT clears the STOP bit. If STRT and INIT are set together, PCnet-ISA controller initialization will be performed first. STRT is set by writing a "1". Writing a "0" has no effect. STRT is cleared by RESET or by setting the STOP bit.
0	INIT	INIT assertion enables PCnet-ISA controller to begin the initialization procedure which reads in the initialization block from memory. Setting INIT clears the STOP bit. If STRT and INIT are set together, PCnet-ISA controller initialization will be performed first. INIT is not cleared when the initialization sequence has completed. INIT is set by writing a "1". Writing a "0" has no effect. INIT is cleared by RESET or by setting the STOP bit.

CSR1: IADR[15:0]

Bit	Name	Description
15-0	IADR [15:0]	Lower address of the Initialization address register. Bit location 0 must be zero. Whenever this register is written, CSR16 is updated with CSR1's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR2: IADR[23:16]

Bit	Name	Description
15-8	RES	Reserved locations. Read and written as zero.
7-0	IADR [23:16]	Upper 8 bits of the address of the Initialization Block. Bit locations 15-8 must be written with zeros. Whenever this register is written, CSR17 is updated with CSR2's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR3: Interrupt Masks and Deferral Control

Bit	Name	Description
15	RES	Reserved location. Written as zero and read as undefined.
14	BABLM	Babble Mask. If BABLM is set, the BABL bit in CSR0 will be masked and will not set INTR flag in CSR0. BABLM is cleared by RESET and is not affected by STOP.
13	RES	Reserved location. Written as zero and read as undefined.
12	MISSM	Missed Frame Mask. If MISSM is set, the MISS bit in CSR0 will be masked and will not set INTR flag in CSR0. MISSM is cleared by RESET and is not affected by STOP.
11	MERRM	Memory Error Mask. If MERRM is set, the MERR bit in CSR0 will be masked and will not set INTR flag in CSR0. MERRM is cleared by RESET and is not affected by STOP.
10	RINTM	Receive Interrupt Mask. If RINTM is set, the RINT bit in CSR0 will be masked and will not set INTR flag in CSR0. RINTM is cleared by RESET and is not affected by STOP.
9	TINTM	Transmit Interrupt Mask. If TINTM is set, the TINT bit in CSR0 will be masked and will not set INTR flag in CSR0. TINTM is cleared by RESET and is not affected by STOP.
8	IDONM	Initialization Done Mask. If IDONM is set, the IDON bit in CSR0 will be masked and will not set INTR flag in CSR0. IDONM is cleared by RESET and is not affected by STOP.
7-5	RES	Reserved locations. Written as zero and read as undefined.

4	DXMT2PD	Disable Transmit Two Part Deferral. If DXMT2PD is set, Transmit Two Part Deferral will be disabled. DXMT2PD is cleared by RESET and is not affected by STOP.
3	EMBA	Enable Modified Back-off Algorithm. If EMBA is set, a modified back-off algorithm is implemented. Read/Write accessible. EMBA is cleared by RESET and is not affected by STOP.
2-0	RES	Reserved locations. Written as zero and read as undefined.

CSR4: Test and Features Control

Bit	Name	Description
15	ENTST	Enable Test Mode operation. When ENTST is set, writing to test mode registers CSR124 and CSR126 is allowed, and other register test functions are enabled. In order to set ENTST, it must be written with a "1" during the first write access to CSR4 after RESET. Once a "0" is written to this bit location, ENTST cannot be set until after the PCnet-ISA controller is reset. ENTST is cleared by RESET.
14	DMAPLUS	When DMAPLUS = "1", the burst transaction counter in CSR80 is disabled. If DMAPLUS = "0", the burst transaction counter is enabled. DMA-PLUS is cleared by RESET.
13	TIMER	Timer Enable Register. If TIMER is set, the Bus Timer Register, CSR82, is enabled. If TIMER is set, CSR82 must be written with a value. If TIMER is cleared, the Bus Timer Register is disabled. TIMER is cleared by RESET.
12	DPOLL	Disable Transmit Polling. If DPOLL is set, the Buffer Management Unit will disable transmit polling. Likewise, if DPOLL is cleared, automatic transmit polling is enabled. If DPOLL is set, TDMD bit in CSR0 must be periodically set in order to initiate a manual poll of a transmit descriptor. Transmit descriptor polling will not take place if TXON is reset. DPOLL is cleared by RESET.
11	APAD_XMT	Auto Pad Transmit. When set, APAD_XMT enables the automatic padding feature. Transmit frames will be padded to extend them to 64 bytes, including FCS. The FCS is calculated for the entire frame (including pad) and appended after the pad field. APAD_XMT will override the programming of the DXMTFCS bit (CSR15.3). APAD_ XMT is reset by activation of the RESET pin.
10	ASTRP_RCV	ASTRP_RCV enables the automatic pad stripping feature. The pad and FCS fields will be stripped from receive frames and not placed in the FIFO. ASTRP_ RCV is reset by activation of the RESET pin.
9	MFCO	Missed Frame Counter Overflow Interrupt. This bit indicates the MFC (CSR112) has overflowed. Can be cleared by writing a "1" to this bit. Also cleared by RESET or setting the STOP bit. Writing a "0" has no effect.
8	MFCOM	Missed Frame Counter Overflow Mask. If MFCOM is set, MFCO will not set INTR in CSR0. MFCOM is set by Reset and is not affected by STOP.
7-6	RES	Reserved locations. Read and written as zero.
5	RCVCCO	Receive Collision Counter Overflow. This bit indicates the Receive Collision Counter (CSR114) has overflowed. It can be cleared by writing a 1 to this bit. Also cleared by RESET or setting the STOP bit. Writing a 0 has no effect.
4	RCVCCOM	Receive Collision Counter Overflow Mask. If RCVCCOM is set, RCVCCO will not set INTR in CSR0. RCVCCOM is set by RESET and is not affected by STOP.
3	TXSTRT	Transmit Start status is set whenever PCnet-ISA controller begins trans- mission of a frame. When TXSTRT is set, IRQ is asserted if IENA = 1 and the mask bit TXSTRTM (CSR4.2) is clear.

		TXSTRT is set by the MAC Unit and cleared by writing a "1", setting RESET or setting the STOP bit. Writing a "0" has no effect.
2	TXSTRTM	Transmit Start Mask. If TXSTRTM is set, the TXSTRT bit in CSR4 will be masked and will not set INTR flag in CSR0. TXS-TRTM is set by RESET and is not affected by STOP.
1	JAB	Jabber Error is set when the PCnet-ISA controller Twisted-pair MAU function exceeds an allowed transmission limit. Jabber is set by the TMAU cell and can only be asserted in 10BASE-T mode. When JAB is set, IRQ is asserted if IENA = 1 and the mask bit JABM (CSR4.0) is clear. The JAB bit can be reset even if the jabber condition is still present. JAB is set by the TMAU circuit and cleared by writing a "1". Writing a "0" has no effect. JAB is also cleared by RESET or setting the STOP bit.
0	JABM	Jabber Error Mask. If JABM is set, the JAB bit in CSR4 will be masked and will not set INTR flag in CSR0. JABM is set by RESET and is not affected by STOP.

CSR6: RCV/XMT Descriptor Table Length

Bit	Name	Description
15-12	TLEN	Contains a copy of the transmit encoded ring length (TLEN) field read from the initialization block during PCnet-ISA controller initialization. This field is written during the PCnet-ISA controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. TLEN is only defined after initialization.
11-8	RLEN	Contains a copy of the receive encoded ring length (RLEN) read from the initialization block during PCnet-ISA controller initialization. This field is written during the PCnet-ISA controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. RLEN is only defined after initialization.
7-0	RES	Reserved locations. Read as zero. Write operations should not be performed.

CSR8: Logical Address Filter, LADRF[15:0]

Bit	Name	Description
15-0	LADRF[15:0]	Logical Address Filter, LADRF [15:0]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set.

CSR9: Logical Address Filter, LADRF[31:16]

Bit	Name	Description
15-0	LADRF[31:16]	Logical Address Filter, LADRF[31:16]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set.

CSR10: Logical Address Filter, LADRF[47:32]

Bit	Name	Description
15-0	LADRF[47:32]	Logical Address Filter, LADRF[47:32]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set.

CSR11: Logical Address Filter, LADRF[63:48]

Bit	Name	Description
15-0	LADRF[63:48]	Logical Address Filter, LADRF[63:48]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register.

Read/write accessible only when STOP bit is set.

CSR12: Physical Address Register, PADR[15:0]

Bit	Name	Description
15-0	PADR[15:0]	Physical Address Register, PADR[15:0]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set.

CSR13: Physical Address Register, PADR[31:16]

Bit	Name	Description
15-0	PADR[31:16]	Physical Address Register, PADR[31:16]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set.

CSR14: Physical Address Register, PADR[47:32]

Bit	Name	Description
15-0	PADR[47:32]	Physical Address Register, PADR[47:32]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set.

CSR15: Mode Register

Bit	Name	Description
		This register's fields are loaded during the PCnet-ISA controller initialization routine with the corresponding Initialization Block values. The register can also be loaded directly by an I/O write. Activating the RESET pin clears all bits of CSR15 to zero.
15	PROM	Promiscuous Mode. When PROM = "1", all incoming receive frames are accepted. Read/write accessible only when STOP bit is set.
14	DRCVBC	Disable Receive Broadcast. When set, disables the PCnet-ISA controller from responding to broadcast messages. Used for protocols that do not support broadcast addressing, except as a function of multicast. DRCVBC is cleared by activation of the RESET pin (broadcast messages will be received). Read/write accessible only when STOP bit is set.
13	DRCVPA	Disable Receive Physical Address. When set, the physical address detection (Station or node ID) of the PCnet-ISA controller will be disabled. Frames addressed to the nodes individual physical address will not be recognized (although the frame may be accepted by the EADI mechanism). Read/write accessible only when STOP bit is set.
12	DLNKTST	Disable Link Status. When DLNKTST = "1", monitoring of Link Pulses is disabled. When DLNKTST = "0", monitoring of Link Pulses is enabled. This bit only has meaning when the 10BASE-T network interface is selected. Read/write accessible only when STOP bit is set.
11	DAPC	Disable Automatic Polarity Correction. When DAPC = "1", the 10BASE-T receive polarity reversal algorithm is disabled. Likewise, when DAPC = "0", the polarity reversal algorithm is enabled. This bit only has meaning when the 10BASE-T network interface is selected. Read/write accessible only when STOP bit is set.
10	MENDECL	MENDEC Loopback Mode. See the description of the LOOP bit in CSR15. Read/write accessible only when STOP bit is set.
9	LRT/TSEL	Low Receive Threshold (T-MAU Mode only)

Transmit Mode Select (AUI Mode only)

LRT — Low Receive Threshold. When LRT = "1", the internal twisted pair receive thresholds are reduced by 4.5 dB below the standard 10BASE-T value (approximately 3/5) and the unsquelch threshold for the RXD circuit will be 180–312 mV peak.

When LRT = "0", the unsquelch threshold for the RXD circuit will be the standard 10BASE-T value, 300–520 mV peak.

In either case, the RXD circuit post squelch threshold will be one half of the unsquelch threshold.

This bit only has meaning when the 10BASE-T network interface is selected.

Read/write accessible only when STOP bit is set. Cleared by RESET.

TSEL — Transmit Mode Select. TSEL controls the levels at which the AUI drivers rest when the AUI transmit port is idle. When TSEL = 0, DO+ and DO- yield "zero" differential to operate transformer coupled loads (Ethernet 2 and 802.3). When TSEL = 1, the DO+ idles at a higher value with respect to DO- , yielding a logical HIGH state (Ethernet 1).

This bit only has meaning when the AUI network interface is selected.

Read/write accessible only when STOP bit is set. Cleared by RESET.

8-7 PORTSEL [1:0] — Port Select bits allow for software controlled selection of the network medium. Medium selection can be over ridden by the MAUSEL pin if the XMAUSEL bit in the ISA Configuration Register is set.

Read/write accessible only when STOP bit is set. Cleared by RESET.

The network port configuration are as follows:

PORTSEL[1:0]	Network Port
0 0	AUI
0 1	10BASE-T
1 0	GPSI*
1 1	Reserved

**Refer to the section on General Purpose Serial Interface for detailed information on accessing GPSI.*

6 INTL — Internal Loopback. See the description of LOOP, CSR15.2.

Read/write accessible only when STOP bit is set.

5 DRTY — Disable Retry. When DRTY = "1", PCnet-ISA controller will attempt only one transmission. If DRTY = "0", PCnet-ISA controller will attempt 16 retry attempts before signaling a retry error.

Read/write accessible only when STOP bit is set.

4 FCOLL — Force Collision. This bit allows the collision logic to be tested. PCnet-ISA controller must be in internal loopback for FCOLL to be valid. If FCOLL = "1", a collision will be forced during loopback transmission attempts; a Retry Error will ultimately result. If FCOLL = "0", the Force Collision logic will be disabled.

Read/write accessible only when STOP bit is set.

3 DXMTFCS — Disable Transmit CRC (FCS). When DXMTFCS = 0, the transmitter will generate and append a FCS to the transmitted frame. When DXMTFCS = 1, the FCS logic is allocated to the receiver and no FCS is generated or sent with the transmitted frame.

See also the ADD_FCS bit in TMD1. If DXMTFCS is set, no FCS will be generated. If both DXMTFCS is set and ADD_FCS is clear for a particular frame, no FCS will be generated. If ADD_FCS is set for a particular frame, the state of DXMTFCS is ignored and a FCS will be appended on that frame by the transmit circuitry.

In loopback mode, this bit determines if the transmitter appends FCS or if the receiver checks the FCS.

This bit was called DTCR in the LANCE (Am7990).

Read/write accessible only when STOP bit is set.

2 LOOP — Loopback Enable allows PCnet-ISA controller to operate in full duplex mode for test purposes. When LOOP = "1", loopback is enabled. In combination with INTL and MENDECL, various loopback modes are defined as follows:

LOOP	INTL	MENDECL	Loopback Mode
0	X	X	Non-loopback
1	0	X	External Loopback
1	1	0	Internal Loopback Include MENDEC
1	1	1	Internal Loopback Exclude MENDEC

Read/write accessible only when STOP bit is set. LOOP is cleared by RESET.

1 DTX — Disable Transmit. If this bit is set, the PCnet-ISA controller will not access the Transmit Descriptor Ring and, therefore, no transmissions will occur. DTX = "0" will set TXON bit (CSR0.4) after STRT (CSR0.1) is asserted. DTX is defined after the initialization block is read.

Read/write accessible only when STOP bit is set.

0 DRX — Disable Receiver. If this bit is set, the PCnet-ISA controller will not access the Receive Descriptor Ring and, therefore, all receive frame data are ignored. DRX = "0" will set RXON bit (CSR0.5) after STRT (CSR0.1) is asserted. DRX is defined after the initialization block is read.

Read/write accessible only when STOP bit is set.

CSR16: Initialization Block Address Lower

Bit	Name	Description
15-0	IADR	Lower 16 bits of the address of the Initialization Block. Bit location 0 must be zero. This register is an alias of CSR1. Whenever this register is written, CSR1 is updated with CSR16's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR17: Initialization Block Address Upper

Bit	Name	Description
15-8	RES	Reserved locations. Written as zero and read as undefined.
7-0	IADR	Upper 8 bits of the address of the Initialization Block. Bit locations 15-8 must be written with zeros. This register is an alias of CSR2. Whenever this register is written, CSR2 is updated with CSR17's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR18-19: Current Receive Buffer Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	CRBA	Contains the current receive buffer address to which the PCnet-ISA controller will store incoming frame data. Read/write accessible only when STOP bit is set.

CSR20-21: Current Transmit Buffer Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	CXBA	Contains the current transmit buffer address from which the PCnet-ISA controller is transmitting. Read/write accessible only when STOP bit is set.

CSR22-23: Next Receive Buffer Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	NRBA	Contains the next receive buffer address to which the PCnet-ISA

		controller will store incoming frame data. Read/write accessible only when STOP bit is set.

CSR24-25: Base Address of Receive Ring

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	BADR	Contains the base address of the Receive Ring. Read/write accessible only when STOP bit is set.

CSR26-27: Next Receive Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	NRDA	Contains the next RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR28-29: Current Receive Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	CRDA	Contains the current RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR30-31: Base Address of Transmit Ring

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	BADX	Contains the base address of the Transmit Ring. Read/write accessible only when STOP bit is set.

CSR32-33: Next Transmit Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	NXDA	Contains the next TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR34-35: Current Transmit Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	CXDA	Contains the current TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR36-37: Next Next Receive Descriptor Address

Bit	Name	Description
31-0	NNRDA	Contains the next next RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR38-39: Next Next Transmit Descriptor Address

Bit	Name	Description
31-0	NNXDA	Contains the next next TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR40-41: Current Receive Status and Byte Count

Bit	Name	Description
31-24	CRST	Current Receive Status. This field is a copy of bits 15:8 of RMD1 of the current receive descriptor.

		Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined.
11-0	CRBC	Current Receive Byte Count. This field is a copy of the BCNT field of RMD2 of the current receive descriptor. Read/write accessible only when STOP bit is set.

CSR42-43: Current Transmit Status and Byte Count

Bit	Name	Description
31-24	CXST	Current Transmit Status. This field is a copy of bits 15:8 of TMD1 of the current transmit descriptor. Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined.
11-0	CXBC	Current Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the current transmit descriptor. Read/write accessible only when STOP bit is set.

CSR44-45: Next Receive Status and Byte Count

Bit	Name	Description
31-24	NRST	Next Receive Status. This field is a copy of bits 15:8 of RMD1 of the next receive descriptor. Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined.
11-0	NRBC	Next Receive Byte Count. This field is a copy of the BCNT field of RMD2 of the next receive descriptor. Read/write accessible only when STOP bit is set.

CSR46: Poll Time Counter

Bit	Name	Description
15-0	POLL	Poll Time Counter. This counter is incremented by the PCnet-ISA controller microcode and is used to trigger the descriptor ring polling operation of the PCnet-ISA controller. Read/write accessible only when STOP bit is set.

CSR47: Polling Interval

Bit	Name	Description
31-16	RES	Reserved locations. Written as zero and read as undefined.
15-0	POLLINT	Polling Interval. This register contains the time that the PCnet-ISA controller will wait between successive polling operations. The POLLINT value is expressed as the two's complement of the desired interval, where each bit of POLLINT represents one-half of an XTAL1 period of time. POLLINT[3:0] are ignored. (POLINT[16] is implied to be a one, so POLLINT[15] is significant, and does not represent the sign of the two's complement POLLINT value.) The default value of this register is 0000. This corresponds to a polling interval of 32,768 XTAL1 periods. The POLINT value of 0000 is created during the microcode initialization routine, and therefore might not be seen when reading CSR47 after RESET. If the user desires to program a value for POLLINT other than the default, then the correct procedure is to first set INIT only in CSR0. Then, when the initialization sequence is complete, the user must set STOP in CSR0. Then the user may write to CSR47 and then set STRT in CSR0. In this way, the default value of 0000 in CSR47 will be overwritten with the desired user value. Read/write accessible only when STOP bit is set.

CSR48-49: Temporary Storage

Bit	Name	Description
31-0	TMP0	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR50-51: Temporary Storage

Bit	Name	Description
31-0	TMP1	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR52-53: Temporary Storage

Bit	Name	Description
31-0	TMP2	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR54-55: Temporary Storage

Bit	Name	Description
31-0	TMP3	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR56-57: Temporary Storage

Bit	Name	Description
31-0	TMP4	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR58-59: Temporary Storage

Bit	Name	Description
31-0	TMP5	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR60-61: Previous Transmit Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	PXDA	Contains the previous TDRE address pointer. The PCnet-ISA controller has the capability to stack multiple transmit frames. Read/write accessible only when STOP bit is set.

CSR62-63: Previous Transmit Status and Byte Count

Bit	Name	Description
31-24	PXST	Previous Transmit Status. This field is a copy of bits 15:8 of TMD1 of the previous transmit descriptor. Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined. Accessible only when STOP bit is set.
11-0	PXBC	Previous Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the previous transmit descriptor. Read/write accessible only when STOP bit is set.

CSR64-65: Next Transmit Buffer Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	NXBA	Contains the next transmit buffer address from which the PCnet-ISA controller will transmit an outgoing frame. Read/write accessible only when STOP bit is set.

CSR66-67: Next Transmit Status and Byte Count

Bit	Name	Description
31-24	NXST	Next Transmit Status. This field is a copy of bits 15:8 of TMD1 of the next transmit descriptor. Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined. Accessible only when STOP bit is set.
11-0	NXBC	Next Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the next transmit descriptor. Read/write accessible only when STOP bit is set.

CSR68-69: Transmit Status Temporary Storage

Bit	Name	Description
31-0	XSTMP	Transmit Status Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR70-71: Temporary Storage

Bit	Name	Description
31-0	TMP8	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR72: Receive Ring Counter

Bit	Name	Description
15-0	RCVRC	Receive Ring Counter location. Contains a Two's complement binary number used to number the current receive descriptor. This counter interprets the value in CSR76 as pointing to the first descriptor; a two's complement value of -1 (FFFFh) corresponds to the last descriptor in the ring. Read/write accessible only when STOP bit is set.

CSR74: Transmit Ring Counter

Bit	Name	Description
15-0	XMTRC	Transmit Ring Counter location. Contains a Two's complement binary number used to number the current transmit descriptor. This counter interprets the value in CSR78 as pointing to the first descriptor; a two's complement value of -1 (FFFFh) corresponds to the last descriptor in the ring. Read/write accessible only when STOP bit is set.

CSR76: Receive Ring Length

Bit	Name	Description
15-0	RCVRL	Receive Ring Length. Contains a Two's complement binary number of the receive descriptor ring length. This register is initialized during the PCnet-ISA controller initialization routine based on the value in the RLEN field of the initialization block. This register can be manually altered; the actual receive ring length is defined by the current value in this register. Read/write accessible only when STOP bit is set.

CSR78: Transmit Ring Length

Bit	Name	Description
15-0	XMTRL	Transmit Ring Length. Contains the two's complement of the transmit descriptor ring length. This register is initialized during the PCnet-ISA controller initialization routine based on the value in the TLEN field of the initialization block. This register can be manually altered; the actual transmit ring length is defined by the current value in this register. Read/write accessible only when STOP bit is set.

CSR80: Burst and FIFO Threshold Control

Bit	Name	Description
15-14	RES	Reserved locations. Read as ones. Written as zero.
13-12	RCVFW[1:0]	Receive FIFO Watermark. RCVFW controls the point at which ISA bus receive DMA is requested in relation to the number of received bytes in the receive FIFO. RCVFW specifies the number of bytes which must be present (once the frame has been verified as a non-runt) before receive DMA is requested. Note however that in order for receive DMA to be performed for a new frame, at least 64 bytes must have been received. This effectively avoids having to react to receive frames which are runts or suffer a collision during the slot time (512 bit times). If the Runt Packet Accept feature is enabled, receive DMA will be requested as soon as either the RCVFW threshold is reached, or a complete valid receive frame is detected (regardless of length). RCVFW is set to a value of 10b (64 bytes) after RESET. Read/write accessible only when STOP bit is set.

RCVFW[1:0]	Bytes Received
00	16
01	32
10	64
11	Reserved

11-10XMTSP[1:0] Transmit Start Point. XMTSP controls the point at which preamble transmission attempts commence in relation to the number of bytes written to the transmit FIFO for the current transmit frame. When the entire frame is in the FIFO, transmission will start regardless of the value in XMTSP. XMTSP is given a value of 10b (64 bytes) after RESET. Regardless of XMTSP, the FIFO will not internally over write its data until at least 64 bytes (or the entire frame if <64 bytes) have been transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be re-written to the transmit FIFO, and re-tries will be handled autonomously by the MAC. This bit is read/write accessible only when the STOP bit is set.

XMTSP[1:0]	Bytes Written
00	4
01	16
10	64
11	112

9-8 XMTFW[1:0] Transmit FIFO Watermark. XMTFW specifies the point at which transmit DMA stops, based upon the number of write cycles that could be performed to the transmit FIFO without FIFO overflow. Transmit DMA is allowed at any time when the number of write cycles specified by XMTFW could be executed without causing transmit FIFO overflow. XMTFW is set to a value of 00b (8 cycles) after hardware RESET. Read/write accessible only when STOP bit is set.

XMTFW[1:0]	Write Cycles
00	8
01	16
10	32
11	Reserved

7-0 DMABR DMA Burst Register. This register contains the maximum allowable number of transfers to system memory that the Bus Interface will perform during a single DMA cycle. The Burst Register is not used to limit the number of transfers during Descriptor transfers. A value of zero will be interpreted as one transfer. During RESET a value of 16 is loaded in the BURST register. If DMAPLUS (CSR4.14) is set, the DMA Burst Register is disabled.

When the Bus Activity Timer register (CSR82: DMABAT) is enabled, the PCnet-ISA controller will relinquish the bus when either the time specified in DMABAT has elapsed or the number of transfers specified in DMABR have occured. When ENTST (CSR4.15) is asserted, all writes to this register will automatically perform a decrement cycle.

Read/write accessible only when STOP bit is set.

CSR82: Bus Activity Timer

Bit	Name	Description
15-0	DMABAT	Bus Activity Timer. If the TIMER bit in CSR4 is set, this register contains the maximum allowable time that the PCnet-ISA controller will take up on the system bus during FIFO data transfers in each bus mastership period. The DMABAT starts counting upon receipt of DACK from the host system. The DMABAT Register does not limit the number of transfers during Descriptor transfers. A value of zero will limit the PCnet-ISA controller to one bus cycle per mastership period. A non-zero value is interpreted as an unsigned number with a resolution of 100 ns. For instance, a value of 51 micro seconds would be programmed with a value of 510. When the TIMER bit in CSR4 is set, DMABAT is enabled and must be initialized by the user. The DMABAT register is undefined until written. When the

ENTST bit in CSR4 is set, all writes to this register will automatically perform a decrement cycle.

When the Bus Activity Timer register (CSR82: DMABAT) is enabled, the PCnet-ISA controller will relinquish the bus when either the time specified in DMABAT has elapsed or the number of transfers specified in DMABR have occured. When ENTST (CSR4.15) is asserted, all writes to this register will automatically perform a decrement cycle.

Read/write accessible only when STOP bit is set.

CSR84-85: DMA Address

Bit	Name	Description
31-0	DMABA	DMA Address Register. This register contains the address of system memory for the current DMA cycle. The Bus Interface Unit controls the Address Register by issuing increment commands to increment the memory address for sequential operations. The DMABA register is undefined until the first PCnet-ISA controller DMA operation. When the ENTST bit in CSR4 is set, all writes to this register will automatically perform an increment cycle. This register has meaning only if the PCnet-ISA controller is in Bus Master Mode. Read/write accessible only when STOP bit is set.

CSR86: Buffer Byte Counter

Bit	Name	Description
15-12	RES	Reserved, Read and written with ones.
11-0	DMABC	DMA Byte Count Register. Contains a Two's complement of the current size of the remaining transmit or receive buffer in bytes. This register is incremented by the Bus Interface Unit. The DMABC register is undefined until written. When ENTST (CSR4.15) is asserted, all writes to this register will automatically perform an increment cycle. Read/write accessible only when STOP bit is set.

CSR88-89: Chip ID

Bit	Name	Description
31-28		Version. This 4-bit pattern is silicon revision dependent.
27-12		Part number. The 16-bit code for the PCnet-ISA controller is 0000000000000011b.
11-1		Manufacturer ID. The 11-bit manufacturer code for AMD is 00000000001b. This code is per the JEDEC Publication 106-A.
0		Always a logic 1. This register is exactly the same as the Chip ID register in the JTAG description.

CSR92: Ring Length Conversion

Bit	Name	Description
15-0	RCON	Ring Length Conversion Register. This register performs a ring length conversion from an encoded value as found in the initialization block to a Two's complement value used for internal counting. By writing bits 15-12 with an encoded ring length, a Two's complemented value is read. The RCON register is undefined until written. Read/write accessible only when STOP bit is set.

CSR94:Transmit Time Domain Reflectometry Count

Bit	Name	Description
15-10	RES	Reserved locations. Read and written as zero.
9-0	XMTTDR	Time Domain Reflectometry reflects the state of an internal counter that counts from the start of transmission to the occurrence of loss of carrier. TDR is incremented at a rate of 10 MHz. Read accessible only when STOP bit is set. Write operations are ignored. XMTTDR is cleared by RESET.

CSR96-97: Bus Interface Scratch Register 0

Bit	Name	Description
31-0	SCR0	This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. The SCR0 register is undefined until written. Read/write accessible only when STOP bit is set.

CSR98-99: Bus Interface Scratch Register 1

Bit	Name	Description
31-0	SCR1	This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. Read/write accessible only when STOP bit is set.

CSR104-105: SWAP

Bit	Name	Description
31-0	SWAP	This register performs word and byte swapping depending upon if 32-bit or 16-bit internal write operations are performed. This register is used internally by the BIU/BMU as a word or byte swapper. The swap register can perform 32-bit operations that the PC can not; the register is externally accessible for test reasons only. CSR104 holds the lower 16 bits and CSR105 holds the upper 16 bits. The swap function is defined as follows:

Internal Write Operation	SWAP Register Result
32-Bit word	SRC[31:16] → SWAP[15:0] SRC[15:0] → SWAP[31:16]
Lower 16-Bit (CSR104)	SRC[15:8] → SWAP[7: 0] SRC[7:0] → SWAP[15:8]

Read/write accessible only when STOP bit is set.

CSR108-109: Buffer Management Scratch

Bit	Name	Description
31-0	BMSCR	The Buffer Management Scratch register is used for assembling Receive and Transmit Status. This register is also used as the primary scan register for Buffer Management Test Modes. BMSCR register is undefined until written. Read/write accessible only when STOP bit is set.

CSR112: Missed Frame Count

Bit	Name	Description
15-0	MFC	Counts the number of missed frames. This register is always readable and is cleared by STOP. A write to this register performs an increment when the ENTST bit in CSR4 is set. When MFC is all 1's (65535) and a missed frame occurs, MFC increments to 0 and sets MFC0 bit (CSR4.9).

CSR114: Receive Collision Count

Bit	Name	Description
15-0	RCVCC	Counts the number of Receive collisions seen, regular and late. This register is always readable and is cleared by STOP. A write to this register performs an increment when the ENTST bit in CSR4 is set. When RCVCC is all 1's (65535) and a receive collision occurs, RCVCC increments to 0 and sets RCVCC0 bit (CSR4.5)

CSR124: Buffer Management Unit Test

Bit	Name	Description
		This register is used to place the BMU/BIU into various test modes to support Test/Debug. This register is writeable when the ENTST bit in CSR4 is set.

15-5	RES	Reserved locations. Written as zero and read as undefined.
4	GPSIEN	This mode places the PCnet-ISA controller in the GPSI Mode. This mode will reconfigure the External Address Pins so that the GPSI port is exposed. This allows bypassing the MENDEC-TMAU logic. This bit should only be set if the external logic supports GPSI operation. Damage to the device may occur in a non-GPSI configuration. Refer to the GPSI section.
3	RPA	Runt Packet Accept. This bit forces the CORE receive logic to accept Runt Packets. This bit allows for faster testing.
2-0	RES	For test purposes only. Reserved locations. Written as zero and read as undefined.

ISA Bus Configuration Registers

The ISA Bus Data Port (IDP) allows access to registers which are associated with the ISA bus. These registers are called ISA Bus Configuration Registers (ISACSRs), and are indexed by the value in the Register Address Port (RAP). The table below defines the ISACSRs which can be accessed. All registers are 16 bits. The "Default" value is the value in the register after reset and is hexadecimal.

ISACSR	MNEMONIC	Default	Name
0	MSRDA	0005H	Master Mode Read Active
1	MSWRA	0005H	Master Mode Write Active
2	MC	0002H	Miscellaneous Configuration
3	Reserved	N/A	Reserved for future AMD use
4	LED0	0000H	Link Integrity
5	LED1	0084H	Default: RCV
6	LED2	0008H	Default: RCVPOL
7	LED3	0090H	Default: XMT

ISACSR0: Master Mode Read Active

Bit	Name	Description
15-4	RES	Reserved locations. Written as zero and read as undefined.
3-0	MSRDA	This register is used to tune the MEMR command signal active time. The value stored in MSRDA defines the number of 50 ns periods that the command signal is active. The default value of 5h indicates 250 ns pulse widths. A value of 0 or 1 will generate 50 ns wide commands.

ISACSR1: Master Mode Write Active

Bit	Name	Description
15-4	RES	Reserved locations. Written as zero and read as undefined.
3-0	MSWRA	This register is used to tune the MEMW command signal active time. The value stored in MSWRA defines the number of 50 ns periods that the command signal is active. The default value of 5h indicates 250 ns pulse widths. A value of 0 or 1 will generate 50 ns wide commands.

ISACSR2: Miscellaneous Configuration

Bit	Name	Description
15	MODE_STATUS	Mode Status. This is a read-only register which indicates whether the PCnet-ISA is configured in shared memory mode. A set condition indicates shared-memory while a clear condition indicates bus-master condition.
14-8	RES	Reserved locations. Written and read as zero.
7	EISA_LVL	EISA_LVL allows for EISA level-sensitive interrupt support. EISA_LVL is cleared when RESET is asserted. When EISA_LVL is a zero, the IRQ pin is configured for ISA edge sensitive full CMOS driver. When EISA_LVL is set by writing a one, the IRQ pin is configured as an EISA level-sensitive interrupt open drain output. When EISA_LVL is set to one, the IRQ pin assertion level is active low.
6-5	RES	Reserved locations. Written and read as zero.
4	ISAINACT	ISAINACT allows for reduced inactive timing appropriate for modern ISA machines. ISAINACT is cleared when RESET is asserted. When ISAINACT is a zero, tMMR3 and tMMW3 parameters are nominally 200 ns, which is compatible with EISA system. When ISAINACT is set by writing a one,

		tMMR3 and tMMW3 are nominally set to 100 ns.
3	EADISEL	EADI Select. Enables EADI match mode. XMAUSEL must be 0.
2	AWAKE	Auto-Wake. If LNKST is set and AWAKE = "1", the 10BASE-T receive circuitry is active during sleep and listens for Link Pulses. LED0 indicates Link Status and goes active if the 10BASE-T port comes of out of "link fail" state. This LED0 pin can be used by external circuitry to re-enable the PCnet-ISA controller and/or other devices. When AWAKE = "0", the Auto-Wake circuity is disabled. This bit only has meaning when the 10BASE-T network interface is selected.
1	ASEL	Auto Select. When set, the PCnet-ISA controller will automatically select the operating media interface port. Set by Reset.
0	XMAUSEL	External MAU Select allows the hardware selection of AUI or 10BASE-T interfaces when set. When cleared, the interface is selected by software. Cleared by RESET.

ASEL (Bit 1)	XMAUSEL (Bit 0)	Selection Mode
0	0	Software; interface selection is done through the PORTSEL[1:0] bits in CSR15.
0	1	Jumper; interface selection is done through the MAUSEL pin.
1	0	Automatic (default)
1	1	Reserved

ISACSR4: $\overline{\text{LED0}}$ Status (Link Integrity)

Bit	Name	Description
		ISACSR4 is a non-programmable register that uses one bit to reflect the status of the $\overline{\text{LED0}}$ pin. This pin defaults to twisted pair MAU Link Status (LNKST) and is not programmable.
15	LNKST	LNKST is a read-only register bit that indicates whether the Link Status LED is asserted. When LNKST is read as zero, the Link Status LED is not asserted. When LNKST is read as one, the Link Status LED is asserted, indicating good 10BASE-T integrity.
14-0	RES	Reserved locations. Written as 0, read as undefined.

ISACSR5: $\overline{\text{LED1}}$ Status

Bit	Name	Description
		ISACSR5 controls the function(s) that the $\overline{\text{LED1}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. ISACSR5 defaults to Receive Status (RCV) with pulse stretcher enabled (PSE = 1) and is fully programmable.
15	LEDOUT	Indicates the current (non-stretched) state of the function(s) generated. Read only.
14-8	RES	Reserved locations. Read and written as zero.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination for each enabled function occurrence. 0 is disabled, 1 is enabled.
6-5	RES	Reserved locations. Read and written as zero.
4	XMT E	Enable Transmit Status Signal. Indicates PCnet-ISA controller transmit activity . 0 disables the signal, 1 enables the signal.
3	RVPOL E	Enable Receive Polarity Signal. Enables LED pin assertion when receive polarity is correct on the 10BASE-T port. Clearing the bit indicates this function is to be ignored.
2	RCV E	Enable Receive Status Signal. Indicates receive activity on the network. 0 disables the signal, 1 enables the signal.
1	JAB E	Enable Jabber Signal. Indicates the PCnet-ISA controller is jabbering on the network. 0 disables the signal, 1 enables the signal.
0	COL E	Enable Collision Signal. Indicates collision activity on the network. 0 disables the signal, 1 enables the signal.

ISACSR6: $\overline{\text{LED2}}$ Status

Bit	Name	Description
		ISACSR6 controls the function(s) that the $\overline{\text{LED2}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. ISACSR6 defaults to twisted pair MAU Receive Polarity (RCVPOL) with pulse stretcher enabled (PSE = 1) and is fully programmable.
15	LEDOUT	Indicates the current (non-stretched) state of the function(s) generated. Read only.
14-8	RES	Reserved locations. Read and written as zero.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination for each enabled function occurrence.
		0 is disabled, 1 is enabled.
6-5	RES	Reserved locations. Read and written as zero.
4	XMT E	Enable Transmit Status Signal. Indicates PCnet-ISA controller transmit activity .
		0 disables the signal, 1 enables the signal.
3	RVPOL E	Enable Receive Polarity Signal.
		Enables LED pin assertion when receive polarity is correct on the 10BASE-T port. Clearing the bit indicates this function is to be ignored.
2	RCV E	Enable Receive Status Signal. Indicates receive activity on the network.
		0 disables the signal, 1 enables the signal.
1	JAB E	Enable Jabber Signal. Indicates the PCnet-ISA controller is jabbering on the network.
		0 disables the signal, 1 enables the signal.
0	COL E	Enable Collision Signal. Indicates collision activity on the network.
		0 disables the signal, 1 enables the signal.

ISACSR7: $\overline{\text{LED3}}$ Status

Bit	Name	Description
		ISACSR7 controls the function(s) that the $\overline{\text{LED3}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. ISACSR7 defaults to Transmit Status (XMT) with pulse stretcher enabled (PSE = 1) and is fully programmable.
15	LEDOUT	Indicates the current (non-stretched) state of the function(s) generated. Read only.
14-8	RES	Reserved locations. Read and written as zero.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination for each enabled function occurrence.
		0 is disabled, 1 is enabled.
6-5	RES	Reserved locations. Read and written as zero.
4	XMT E	Enable Transmit Status Signal. Indicates PCnet-ISA controller transmit activity .
		0 disables the signal, 1 enables the signal.
3	RVPOL E	Enable Receive Polarity Signal.
		Enables LED pin assertion when receive polarity is correct on the 10BASE-T port. Clearing the bit indicates this function is to be ignored.
2	RCV E	Enable Receive Status Signal. Indicates receive activity on the network.
		0 disables the signal, 1 enables the signal.
1	JAB E	Enable Jabber Signal. Indicates the PCnet-ISA controller is jabbering on the network.
		0 disables the signal, 1 enables the signal.
0	COL E	Enable Collision Signal. Indicates collision activity on the network.
		0 disables the signal, 1 enables the signal.

Initialization Block

The figure below shows the Initialization Block memory configuration. Note that the Initialization Block must be based on a word (16-bit) boundary.

Address	Bits 15–12	Bits 11–8	Bits 7–4	Bits 3–0
IADR+22	TLEN	RES	TDRA 23–16	
IADR+20	TDRA 15–00			
IADR+18	RLEN	RES	RDRA 23–16	
IADR+16	RDRA 15–00			
IADR+14	LADRF 63–48			
IADR+12	LADRF 47–32			
IADR+10	LADRF 31–16			
IADR+08	LADRF 15–00			
IADR+06	PADR 47–32			
IADR+04	PADR 31–16			
IADR+02	PADR 15–00			
IADR+00	MODE 15–00			

RLEN and TLEN

The TLEN and RLEN fields in the initialization block are 3 bits wide, occupying bits 15,14, and 13, and the value in these fields determines the number of Transmit and Receive Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is as follows:

R/TLEN	# of DREs
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

If a value other than those listed in the above table is desired, CSR76 and CSR78 can be written after initialization is complete. See the description of the appropriate CSRs.

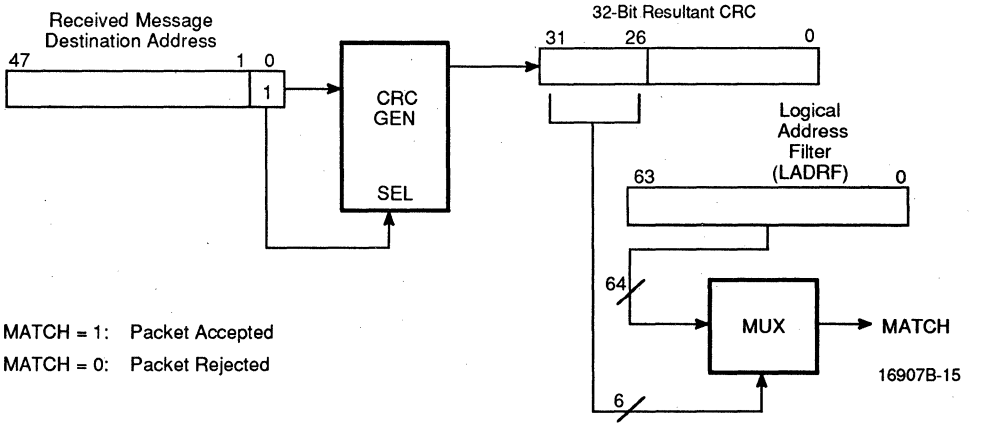
RDRA and TDRA

TDRA and RDRA indicate where the transmit and receive descriptor rings, respectively, begin. Each DRE must be located on an 8-byte boundary.

LADRF

The Logical Address Filter (LADRF) is a 64-bit mask that is used to accept incoming Logical Addresses. If the first bit in the incoming address (as transmitted on the wire) is a "1", the address is deemed logical. If the first bit is a "0", it is a physical address and is compared against the physical address that was loaded through the initialization block.

A logical address is passed through the CRC generator, producing a 32-bit result. The high order 6 bits of the CRC are used to select one of the 64 bit positions in the Logical Address Filter. If the selected filter bit is set, the address is accepted and the frame is placed into memory.



Address Match Logic

The Logical Address Filter is used in multicast addressing schemes. The acceptance of the incoming frame based on the filter value indicates that the message may be intended for the node. It is the node's responsibility to determine if the message is actually intended for the node by comparing the destination address of the stored message with a list of acceptable logical addresses.

If the Logical Address Filter is loaded with all zeroes and promiscuous mode is disabled, all incoming logical addresses except broadcast will be rejected.

The Broadcast address, which is all ones, does not go through the Logical Address Filter and is handled as follows:

1) If the Disable Broadcast Bit is cleared, the broadcast address is accepted.
2) If the Disable Broadcast Bit is set and promiscuous mode is enabled, the broadcast address is accepted.
3) If the Disable Broadcast Bit is set and promiscous mode is disabled, the broadcast address is rejected.

If external loopback is used, the FCS logic must be allocated to the receiver (by setting the DXMTFCS bit in CSR15, and clearing the ADD_FCS bit in TMD1) when using multicast addressing.

PADR

This 48-bit value represents the unique node address assigned by the IEEE and used for internal address comparison. PADR[0] is the first address bit transmitted on the wire, and must be zero. The six-byte nomenclature used by the IEEE maps to the PCnet-ISA controller PADR register as follows: the first byte comprises PADR[7:0], with PADR[0] being the least significant bit of the byte. The second IEEE byte maps to PADR[15:8], again from LSbit to MSbit, and so on. The sixth byte maps to PADR[47:40], the LSbit being PADR[40].

MODE

The mode register in the initialization block is copied into CSR15 and interpreted according to the description of CSR15.

Receive Descriptors

The Receive Descriptor Ring Entries (RDREs) are composed of 4 receive message fields (RMD0-3). Together they contain the following information:

- The address of the actual message data buffer in user (host) memory.
- The length of that message buffer.
- Status information indicating the condition of the buffer. The eight most significant bits of RMD1 (RMD1[15:0]) are collectively termed the STATUS of the receive descriptor.

RMD0

Holds LADR [15:0]. This is combined with HADR [7:0] in RMD1 to form the 24-bit address of the buffer pointed to by this descriptor table entry. There are no restrictions on buffer byte alignment or length.

RMD1

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-ISA controller (OWN=1). The PCnet-ISA controller clears the OWN bit after filling the buffer pointed to by the descriptor entry. The host sets the OWN bit after emptying the buffer. Once the PCnet-ISA controller or host has relinquished ownership of a

		buffer, it must not change any field in the descriptor entry.
14	ERR	ERR is the OR of FRAM, OFLO, CRC, or BUFF. ERR is written by the PCnet-ISA controller.
13	FRAM	FRAMING ERROR indicates that the incoming frame contained a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is written by the PCnet-ISA controller.
12	OFLO	OVERFLOW error indicates that the receiver has lost all or part of the incoming frame, due to an inability to store the frame in a memory buffer before the internal FIFO overflowed. OFLO is valid only when ENP is not set. OFLO is written by the PCnet-ISA controller.
11	CRC	CRC indicates that the receiver has detected a CRC (FCS) error on the incoming frame. CRC is valid only when ENP is set and OFLO is not. CRC is written by the PCnet-ISA controller.
10	BUFF	BUFFER ERROR is set any time the PCnet-ISA controller does not own the next buffer while data chaining a received frame. This can occur in either of two ways: 1) The OWN bit of the next buffer is zero. 2) FIFO overflow occurred before the PCnet-ISA controller polled the next descriptor. If a Buffer Error occurs, an Overflow Error may also occur internally in the FIFO, but will not be reported in the descriptor status entry unless both BUFF and OFLO errors occur at the same time. BUFF is written by the PCnet-ISA controller.
9	STP	START OF PACKET indicates that this is the first buffer used by the PCnet-ISA controller for this frame. It is used for data chaining buffers. STP is written by the PCnet-ISA controller.
8	ENP	END OF PACKET indicates that this is the last buffer used by the PCnet-ISA controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is written by the PCnet-ISA controller.
7-0	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and is not changed by the PCnet-ISA controller.

RMD2

Bit	Name	Description
15-12	ONES	MUST BE ONES. This field is written by the host and unchanged by the PCnet-ISA controller.
11-0	BCNT	BUFFER BYTE COUNT is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and is not changed by the PCnet-ISA controller.

RMD3

Bit	Name	Description
15-12	RES	RESERVED and read as zeros.
11-0	MCNT	MESSAGE BYTE COUNT is the length in bytes of the received message, expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the PCnet-ISA controller and cleared by the host.

Transmit Descriptors

The Transmit Descriptor Ring Entries (TDREs) are composed of 4 transmit message fields (TMD0-3). Together they contain the following information:

- The address of the actual message data buffer in user or host memory.
- The length of the message buffer.
- Status information indicating the condition of the buffer. The eight most significant bits of TMD1 (TMD1[15:8]) are collectively termed the STATUS of the transmit descriptor.

Note that bit 13 of TMD1, which was formerly a reserved bit in the LANCE (Am7990), is assigned a new meaning, ADD_FCS.

TMD0

Holds LADRF [15:0]. This is combined with HADR [7:0] in TMD1 to form a 24-bit address of the buffer pointed to by this descriptor table entry. There are no restrictions on buffer byte alignment or length.

TMD1

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-ISA controller (OWN=1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The PCnet-ISA controller clears the OWN bit after transmitting the contents of the buffer. Both the PCnet-ISA controller and the host must not alter a descriptor entry after it has relinquished ownership.
14	ERR	ERR is the OR of UFLO, LCOL, LCAR, or RTRY. ERR is written by the PCnet-ISA controller. This bit is set in the current descriptor when the error occurs, and therefore may be set in any descriptor of a chained buffer transmission.
13	ADD_FCS	ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. It is valid only if the STP bit is set. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS = 0, FCS generation is controlled by DXMTFCS. ADD_FCS is written by the host, and unchanged by the PCnet-ISA controller. This was a reserved bit in the LANCE (Am7990).
12	MORE	MORE indicates that more than one re-try was needed to transmit a frame. MORE is written by the PCnet-ISA controller. This bit has meaning only if the ENP or the ERR bit is set.
11	ONE	ONE indicates that exactly one re-try was needed to transmit a frame. ONE flag is not valid when LCOL is set. ONE is written by the PCnet-ISA controller. This bit has meaning only if the ENP or the ERR bit is set.
10	DEF	DEFERRED indicates that the PCnet-ISA controller had to defer while trying to transmit a frame. This condition occurs if the channel is busy when the PCnet-ISA controller is ready to transmit. DEF is written by the PCnet-ISA controller. This bit has meaning only if the ENP or ERR bits are set.
9	STP	START OF PACKET indicates that this is the first buffer to be used by the PCnet-ISA controller for this frame. It is used for data chaining buffers. The STP bit must be set in the first buffer of the frame, or the PCnet-ISA controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is written by the host and is not changed by the PCnet-ISA controller.
8	ENP	END OF PACKET indicates that this is the last buffer to be used by the PCnet-ISA controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is written by the host and is not changed by the PCnet-ISA controller.
7-0	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and is not changed by the PCnet-ISA controller.

TMD2

Bit	Name	Description
15-12	ONES	MUST BE ONES. This field is written by the host and unchanged by the PCnet-ISA controller.
11-0	BCNT	BUFFER BYTE COUNT is the length of the buffer pointed to by this descriptor, expressed as the two's com- plement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the PCnet-ISA controller. This field is written by the host and is not changed by

the PCnet-ISA controller. There are no minimum buffer size restrictions. Zero length buffers are allowed for protocols which require it.

TMD3

Bit	Name	Description
15	BUFF	BUFFER ERROR is set by the PCnet-ISA controller during transmission when the PCnet-ISA controller does not find the ENP flag in the current buffer and does not own the next buffer. This can occur in either of two ways: 1) The OWN bit of the next buffer is zero. 2) FIFO underflow occurred before the PCnet-ISA controller obtained the next STATUS byte (TMD1[15:8]). BUFF error will turn off the transmitter (CSR0, TXON = 0). If a Buffer Error occurs, an Underflow Error will also occur. BUFF is not valid when LCOL or RTRY error is set during transmit data chaining. BUFF is written by the PCnet-ISA controller.
14	UFLO	UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that the FIFO has emptied before the end of the frame was reached. Upon UFLO error, the transmitter is turned off (CSR0, TXON = 0). UFLO is written by the PCnet-ISA controller.
13	RES	RESERVED bit. The PCnet-ISA controller will write this bit with a "0".
12	LCOL	LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The PCnet-ISA controller does not re-try on late collisions. LCOL is written by the PCnet-ISA controller.
11	LCAR	LOSS OF CARRIER is set when the carrier is lost during an PCnet-ISA controller-initiated transmission. The PCnet-ISA controller does not stop transmission upon loss of carrier. It will continue to transmit the whole frame until done. LCAR is written by the PCnet-ISA controller.
10	RTRY	RETRY ERROR indicates that the transmitter has failed after 16 attempts to successfully transmit a message, due to repeated collisions on the medium. If DRTY = 1 in the MODE register, RTRY will set after one failed transmission attempt. RTRY is written by the PCnet-ISA controller.
09-00	TDR	TIME DOMAIN REFLECTOMETRY reflects the state of an internal PCnet-ISA controller counter that counts at a 10 MHz rate from the start of a transmission to the occurrence of a collision or loss of carrier. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the PCnet-ISA controller and is valid only if RTRY is set. Note that 10 MHz gives very low resolution and in general has not been found to be particularly useful. This feature is here primarily to maintain full compatibility with the LANCE.

Register Summary

Ethernet Controller Registers (accessed via RDP port)

RAP Addr	Symbol	Width	User Register	Comments
00	CSR0	16-bit	Y	PCnet-ISA Controller Status
01	CSR1	16-bit	Y	Lower IADR: maps to location 16
02	CSR2	16-bit	Y	Upper IADR: maps to location 17
03	CSR3	16-bit	Y	Mask Register
04	CSR4	16-bit	Y	Miscellaneous Register
05	CSR5	16-bit		Reserved
06	CSR6	16-bit		RCV/XMT Descriptor Table Length
07	CSR7	16-bit		Reserved
08	CSR8	16-bit	Y	LADR0: LADRF[15:0]
09	CSR9	16-bit	Y	LADR1: LADRF[31:16]
10	CSR10	16-bit	Y	LADR2: LADRF[47:32]
11	CSR11	16-bit	Y	LADR3: LADRF[63:48]
12	CSR12	16-bit	Y	PADR0: PADR[15:0]
13	CSR13	16-bit	Y	PADR1: PADR[31:16]
14	CSR14	16-bit	Y	PADR2: PADR[47:32]
15	CSR15	16-bit	Y	MODE: Mode Register
16–17	CSR16	32-bit		IADR: Base Address of INIT Block
18–19	CSR18	32-bit		CRBA: Current RCV Buffer Address
20–21	CSR20	32-bit		CXBA: Current XMT Buffer Address
22–23	CSR22	32-bit		NRBA: Next RCV Buffer Address
24–25	CSR24	32-bit	Y	BADR: Base Address of RCV Ring
26–27	CSR26	32-bit		NRDA: Next RCV Descriptor Address
28–29	CSR28	32-bit		CRDA: Current RCV Descriptor Address
30–31	CSR30	32-bit	Y	BADX: Base Address of XMT Ring
32–33	CSR32	32-bit		NXDA: Next XMT Descriptor Address
34–35	CSR34	32-bit		CXDA: Current XMT Descriptor Address
36–37	CSR36	32-bit		Next Next Receive Descriptor Address
38–39	CSR38	32-bit		Next Next Transmit Descriptor Address
40–41	CSR40	32-bit		CRBC: Current RCV Stat and Byte Count
42–43	CSR42	32-bit		CXBC: Current XMT Status and Byte Count
44–45	CSR44	32-bit		NRBC: Next RCV Stat and Byte Count
46	CSR46	16-bit		POLL: Poll Time Counter
47	CSR47	32-bit		Polling Interval
48–49	CSR48	32-bit		TMP0: Temporary Storage
50–51	CSR50	32-bit		TMP1: Temporary Storage
52–53	CSR52	32-bit		TMP2: Temporary Storage
54–55	CSR54	32-bit		TMP3: Temporary Storage
56–57	CSR56	32-bit		TMP4: Temporary Storage
58–59	CSR58	32-bit		TMP5: Temporary Storage
60–61	CSR60	32-bit		PXDA: Previous XMT Descriptor Address
62–63	CSR62	32-bit		PXBC: Previous XMT Status and Byte Count

Register Summary

Ethernet Controller Registers (accessed via RDP port) (continued)

RAP Addr	Symbol	Width	User Registers	Comments
64–65	CSR64	32-bit		NXBA: Next XMT Buffer Address
66–67	CSR66	32-bit		NXBC: Next XMT Status and Byte Count
68–69	CSR68	32-bit		XSTMP: XMT Status Temporary
70–71	CSR70	32-bit		RSTMP: RCV Status Temporary
72	CSR72	16-bit		RCVRC: RCV Ring Counter
74	CSR74	16-bit		XMTRC: XMT Ring Counter
76	CSR76	16-bit	Y	RCVRL: RCV Ring Length
78	CSR78	16-bit	Y	XMTRL: XMT Ring Length
80	CSR80	16-bit	Y	DMABR: Burst Register
82	CSR82	16-bit	Y	DMABAT: Bus Activity Timer
84–85	CSR84	32-bit		DMABA: Address Register
86	CSR86	16-bit		DMABC: Byte Counter/Register
88–89	CSR88	32-bit	Y	Chip ID Register
92	CSR92	16-bit		RCON: Ring Length Conversion Register
94	CSR94	16-bit		XMTTDR: Transmit Time Domain Reflectometry
96–97	CSR96	32-bit		SCR0: BIU Scratch Register 0
98–99	CSR98	32-bit		SCR1: BIU Scratch Register 1
104–105	CSR104	32-bit		SWAP:16-bit word/byte Swap Register
108–109	CSR108	32-bit		BMSCR: BMU Scratch Register
112	CSR112	16-bit	Y	Missed Frame Count
114	CSR114	16-bit	Y	Receive Collision Count
124	CSR124	16-bit	Y	BMU Test Register
126	CSR126	16-bit		Reserved

Note:
Although the PCnet-ISA controller has many registers that can be accessed by software, most of these registers are intended for debugging and production testing purposes only. The registers with a "Y" are the only registers that should be accessed by network software.

Register Summary

ISACSR—ISA Bus Configuration Registers (accessed via IDP port)

RAP Addr	Mnemonic	Default	Name
0	MSRDA	0005H	Master Mode Read Active
1	MSWRA	0005H	Master Mode Write Active
2	MC	0002H	Miscellaneous Configuration
3	Reserved	N/A	Reserved for future AMD use
4	LED0	0000H	$\overline{\text{LED0}}$ Status (Link Integrity)
5	LED1	0084H	$\overline{\text{LED1}}$ Status (Default: RCV)
6	LED2	0008H	$\overline{\text{LED2}}$ Status (Default: RCVPOL)
7	LED3	0090H	$\overline{\text{LED3}}$ Status (Default: XMT)

I/O Address Offset

Offset	#Bytes	Register
0h	16	Address PROM
10h	2	RDP
12h	2	RAP (shared by RDP and IDP)
14h	2	Reset
16h	2	IDP

SYSTEM APPLICATION

ISA Bus Interface

Compatibility Considerations

Although 8 MHz is now widely accepted as the standard speed at which to run the ISA bus, many machines have been built which operate at higher speeds with non-standard timing. Some machines do not correctly support 16-bit I/O operations with wait states. Although the PCnet-ISA controller is quite fast, some operations still require an occasional wait state. The PCnet-ISA controller moves data through memory accesses, therefore, I/O operations do not affect performance. By configuring the PCnet-ISA controller as an 8-bit I/O device, compatibility with PC/AT-class machines is obtained at virtually no cost in performance. To treat the PCnet-ISA controller as an 8-bit software resource (for non-ISA applications), the even-byte must be accessed first, followed by an odd-byte access.

Memory cycle timing is an area where some tradeoffs may be necessary. Any slow down in a memory cycle translates directly into lower bandwidth. The PCnet-ISA controller starts out with much higher bandwidth than most slave type controllers and should continue to be superior even if an extra 50 or 100 ns are added to memory cycles.

The memory cycle active time is tunable in 50 ns increments with a default of 250 ns. The memory cycle idle time defaults to 200 ns and can be reprogrammed to 100 ns. See register description for ISACS42. Most machines should not need tuning.

The PCnet-ISA controller is compatible with NE2100 and NE1500T software drivers. All the resources such as address PROM, boot PROM, RAP, and RDP are in the same location with the same semantics. An additional set of registers (ISA CSR) is available to configure on board resources such as ISA bus timing and LED operation. However, loopback frames for the PCnet-ISA controller must contain more than 64 bytes of data if the Runt Packet Accept feature is not enabled; this size limitation does not apply to LANCE (Am7990) based boards such as the NE2100 and NE1500T.

Bus Master

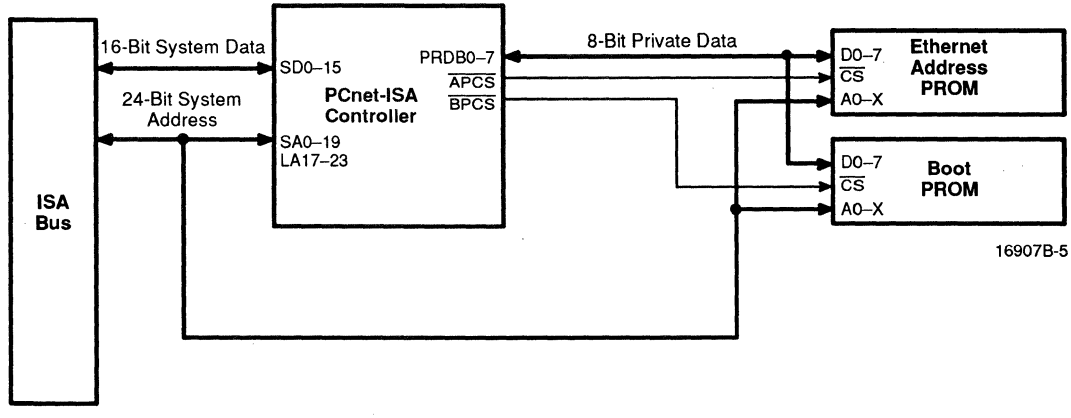
Bus Master mode is the preferred mode for client applications on PC/AT or similar machines supporting 16-bit DMA with its unsurpassed combination of high performance and low cost.

Shared Memory

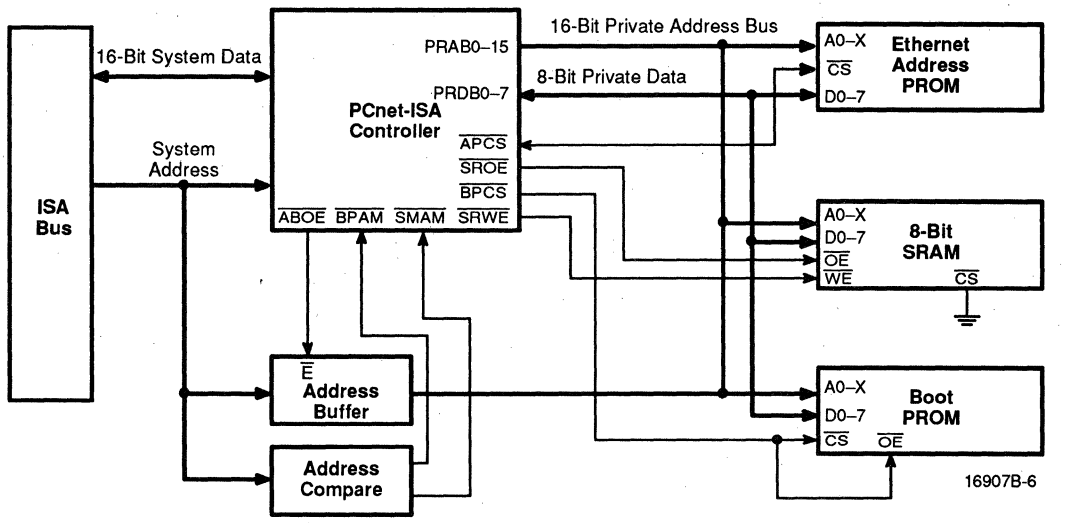
The shared memory mode is recommended for file servers or other applications where there is very high, average or peak latency.

The address compare circuit has the following functions. It receives the 7 LA signals, generates $\overline{\text{MEMCS16}}$, and compares them to the desired shared memory and boot PROM addresses. The logic latches the address compare result when BALE goes inactive and uses this result along with $\overline{\text{REF}}$ (must be deasserted) and the appropriate SA signals to generate $\overline{\text{SMAM}}$ and $\overline{\text{BPAM}}$.

All these functions can be performed in one PAL device. Assume both memories are 8 Kbytes and are in the same 128 Kbyte region. SA16,15,14,13 are required to select 8 Kbytes, and there are 7 LA pins. Counting the $\overline{\text{MEMCS16}}$ pin, the latched compare pin, four SA pins, the $\overline{\text{REF}}$ pin, the $\overline{\text{SMAM}}$ pin and the $\overline{\text{BPAM}}$ pin, we find a total of 16 pins which can easily fit into one PAL device. To operate in an 8–bit PC/XT environment, the LA signals should have weak pull-down resistors connected to them to present a logic 0 level when not driven.



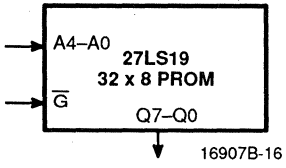
Bus Master Block Diagram



Shared Memory Block Diagram

Address PROM Interface

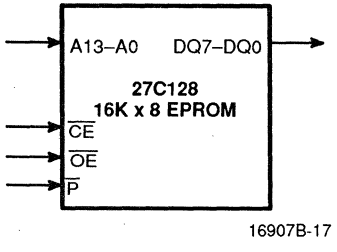
The suggested address PROM is the Am27LS19, a 32x8 device. $\overline{APCS}$ should be connected directly to the device's $\overline{G}$ input.



Address PROM Example

Boot PROM Interface

The boot PROM is a 16Kx8 EPROM. Its program pin $\overline{P}$ should be tied to V_{CC}, output enable $\overline{OE}$ tied to ground, and chip enable $\overline{CE}$ to $\overline{BPCS}$ to minimize power consumption at the expense of speed. If speed is more important, then ground $\overline{CE}$ and connect $\overline{OE}$ to $\overline{BPCS}$.



Boot PROM Example

Static RAM Interface (for Shared Memory only)

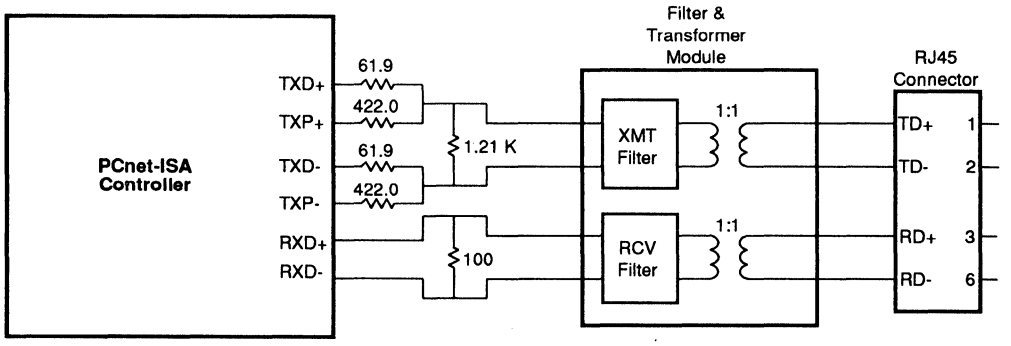
The SRAM is an 8Kx8 or 32Kx8 device. The PCnet-ISA controller can support 64 Kbytes of SRAM address space. The PCnet-ISA controller provides $\overline{SROE}$ and $\overline{SRWE}$ outputs which can go directly to the $\overline{OE}$ and $\overline{WE}$ pins of the SRAM, respectively. The address lines are connected as described in the shared memory section and the data lines go to the Private Data Bus.

AUI

The PCnet-ISA controller drives the AUI interface through a set of transformers. The DI and CI inputs should each be terminated with a pair of matched 39 Ω or 40.2 Ω resistors connected in series with the middle node bypassed to ground with a .01 µF to 0.1 µF capacitor. Refer to the PCnet-ISA Technical Manual (PID #16850B) for network interface design and refer to Appendix A for a list of compatible AUI isolation transformers.

10BASE-T Interface

The diagram below shows the proper 10BASE-T network interface design. Refer to the PCnet-Family Technical Manual (PID #18216A) for more design details, and refer to Appendix A for a list of compatible 10BASE-T filter/transformer modules.



Note:
All resistors are ±1%

10BASE-T External Components and Hookup

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C

Ambient Temperature
Under Bias ..... 0°C to +70°C

Supply Voltage to AVss
or DVss (AV_{DD}, DV_{DD}) ..... −0.3 V to +6.0 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A) ..... 0°C to +70°C

Supply Voltages
(AV_{DD}, DV_{DD}) ..... 5 V ±5%

All inputs within the range: ..... $AV_{SS} - 0.5\ V \le V_{in} \le AV_{DD} + 0.5\ V$, or $DV_{SS} - 0.5\ V \le V_{in} \le DV_{DD} + 0.5\ V$

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (refer to page 19 for driver types)

Parameter Symbol	Parameter Description	Test Conditions		Min	Max	Unit
Digital Input Voltage						
V_{IL}	Input LOW Voltage				0.8	V
V_{IH}	Input HIGH Voltage			2.0	DV_{DD} + 0.5	V
Digital Output Voltage						
V_{OL}	Output LOW Voltage				0.5	V
V_{OH}	Output HIGH Voltage	(Note 1)		2.4		V
Digital Input Leakage Current						
I_{IX}	Input Leakage Current	V_{DD} = 5 V, V_{IN} = 0 V (Note 2)		−10	10	µA
Digital Output Leakage Current						
I_{OZL}	Output Low Leakage Current (Note 3)	V_{OUT} = 0 V		−10		µA
I_{OZH}	Output High Leakage Current (Note 3)	V_{OUT} = V_{DD}			10	µA
Crystal Input Current						
V_{ILX}	XTAL1 Input LOW Threshold Voltage	V_{IN} = External Clock		−0.5	0.8	V
V_{IHX}	XTAL1 Input HIGH Threshold Voltage	V_{IN} = External Clock		3.5	V_{DD} + 0.5	V
I_{ILX}	XTAL1 Input LOW Current	VIN = DVSS	Active	−120	0	µA
			Sleep	−10	+10	µA
I_{IHX}	XTAL1 Input HIGH Current	VIN = VDD	Active	0	120	µA
			Sleep		400	µA
Attachment Unit Interface						
I_{IAXD}	Input Current at DI+ and DI−	AV_{SS} < VIN < AV_{DD}		−500	+500	µA
I_{IAXC}	Input current at CI+ and CI−	AV_{SS} < VIN < AV_{DD}		−500	+500	µA
V_{AOD}	Differential Output Voltage |(DO+)−(DO−)|	R_L = 78 Ω		630	1200	mV
V_{AODOFF}	Transmit Differential Output Idle Voltage	R_L = 78 Ω (Note 5)		−40	+40	mV

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Attachment Unit Interface (Continued)					
I_{AODOFF}	Transmit Differential Output Idle Current	R_L = 78 Ω (Note 4)	–1	+1	mA
V_{CMT}	Transmit Output Common Mode Voltage	R_L = 78 Ω	2.5	AV_{DD}	V
V_{ODI}	DO± Transmit Differential Output Voltage Imbalance	R_L = 78 Ω (Note 5)		25	mV
V_{ATH}	Receive Data Differential Input Threshold	(Note 5)	–35	35	mV
V_{ASQ}	DI± and CI± Differential Input Threshold (Squelch)		–275	–160	mV
V_{IRDVD}	DI± and CI± Differential Mode Input Voltage Range		–1.5	+1.5	V
V_{ICM}	DI± and CI± Input Bias Voltage	I_{IN} = 0 mA	AV_{DD}–3.0	AV_{DD}–1.0	V
V_{OPD}	DO± Undershoot Voltage at Zero Differential on Transmit Return to Zero (ETD)	(Note 5)		–100	mV
Twisted Pair Interface					
I_{IRXD}	Input Current at RXD±	$AV_{SS} < V_{IN} < AV_{DD}$	–500	500	µA
R_{RXD}	RXD± Differential Input Resistance	(Note 5)	10		KΩ
V_{TIVB}	RXD+, RXD– Open Circuit Input Voltage (Bias)	I_{IN} = 0 mA	AV_{DD} – 3.0	AV_{DD} – 1.5	V
V_{TIDV}	Differential Mode Input Voltage Range (RXD±)	AV_{DD} = +5 V	–3.1	+3.1	V
V_{TSQ+}	RXD Positive Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	300	520	mV
V_{TSQ-}	RXD Negative Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	–520	–300	mV
V_{THS+}	RXD Post-Squelch Positive Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	150	293	mV
V_{THS-}	RXD Post-Squelch Negative Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	–293	–150	mV
V_{LTSQ+}	RXD Positive Squelch Threshold (Peak)	LRT = 1 (Note 6)	180	312	mV
V_{LTSQ-}	RXD Negative Squelch Threshold (Peak)	LRT = 1 (Note 6)	–312	–180	mV
V_{LTHS+}	RXD Post-Squelch Positive Threshold (Peak)	LRT = 1 (Note 6)	90	156	mV
V_{LTHS-}	RXD Post-Squelch Negative Threshold (Peak)	LRT = 1 (Note 6)	–156	–90	mV

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Twisted Pair Interface (continued)					
V_{RXDTH}	RXD Switching Threshold	(Note 5)	–35	35	mV
V_{TXH}	TXD± and TXP± Output HIGH Voltage	DV_{SS} = 0 V	DV_{DD} – 0.6	DV_{DD}	V
V_{TXL}	TXD± and TXP± Output LOW Voltage	DV_{DD} = +5 V	DV_{SS}	DV_{SS} + 0.6	V
V_{TXI}	TXD± and TXP± Differential Output Voltage Imbalance		–40	+40	mV
V_{TXOFF}	TXD± and TXP± Idle Output Voltage	DV_{DD} = +5 V	–40	+40	mV
R_{TX}	TXD± Differential Driver Output Impedance	(Note 5)		40	Ω
	TXP± Differential Driver Output Impedance	(Note 5)		80	Ω
IEEE 1149.1 (JTAG) Test Port					
V_{IL}	TCK, TMS, TDI			0.8	V
V_{IH}	TCK, TMS, TDI		2.0		V
V_{OL}	TDO	I_{OL} = 2.0 mA		0.4	V
V_{OH}	TDO	I_{OH} = –0.4 mA	2.4		V
I_{IL}	TCK, TMS, TDI	V_{DD} = 5.5 V, V_I = 0.5 V		–200	µA
I_{IH}	TCK, TMS, TDI	V_{DD} =5.5V, V_I = 2.7V		–100	µA
I_{OZ}	TDO	0.4V < V_{OUT} < V_{DD}	–10	+10	µA
Power Supply Current					
I_{DD}	Active Power Supply Current	XTAL1 = 20 MHz		75	mA
I_{DDCOMA}	Coma Mode Power Supply Current	$\overline{SLEEP}$ active		200	µA
$I_{DDSNOOZE}$	Snooze Mode Power Supply Current	Awake bit set active		10	mA

Notes:

1. *V_{OH} does not apply to open-drain output pins.*
2. *I_{IX} applies to all input only pins except D±, C±, and XTAL1.*
3. *I_{OZL} applies to all three-state output pins and bi-directional pins, except PRDB[7:0]. I_{OZH} applies to pins PRDB[7:0].*
4. *Correlated to other tested parameters—not tested directly.*
5. *Parameter not tested.*
6. *LRT is bit 9 of Mode register (CSR15)*

SWITCHING CHARACTERISTICS: BUS MASTER MODE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Input/Output Write Timing					
t_{IOW1}	AEN, $\overline{SBHE}$, SA0–9 Setup to ↓ $\overline{IOW}$		10		ns
t_{IOW2}	AEN, $\overline{SBHE}$,SA0–9 Hold After ↑ $\overline{IOW}$		5		ns
t_{IOW3}	$\overline{IOW}$ Assertion		100		ns
t_{IOW4}	$\overline{IOW}$ Inactive		55		ns
t_{IOW5}	SD Setup to ↑ $\overline{IOW}$		10		ns
t_{IOW6}	SD Hold After ↑ $\overline{IOW}$		10		ns
t_{IOW7}	↓ IOCHRDY Delay From ↓ $\overline{IOW}$		0	35	ns
t_{IOW8}	IOCHRDY Inactive		125		ns
t_{IOW9}	↑IOCHRDY to ↑ $\overline{IOW}$		0		ns
Input/Output Read Timing					
t_{IOR1}	AEN, $\overline{SBHE}$, SA0–9 Setup to ↓ $\overline{IOR}$		15		ns
t_{IOR2}	AEN, $\overline{SBHE}$,SA0–9 Hold After ↑ $\overline{IOR}$		5		ns
t_{IOR3}	$\overline{IOR}$ Inactive		55		ns
t_{IOR4}	SD Hold After ↑ $\overline{IOR}$		0	20	ns
t_{IOR5}	SD Valid From ↓ $\overline{IOR}$		0	110	ns
t_{IOR6}	↓ IOCHRDY Delay From ↓ $\overline{IOR}$		0	35	ns
t_{IOR7}	IOCHRDY Inactive		125		ns
t_{IOR8}	SD Valid From ↑ IOCHRDY		–130	10	ns
I/O To Memory Command Inactive					
t_{IOM1}	↑ $\overline{IOW}$/$\overline{MEMW}$ to ↓ ($\overline{S}$)$\overline{MEMR}$/$\overline{IOR}$		55		ns
t_{IOM2}	↑ ($\overline{S}$)$\overline{MEMR}$/$\overline{IOR}$ to ↓ $\overline{IOW}$/$\overline{MEMW}$		55		ns
$\overline{IOCS16}$ Timing					
t_{IOCS1}	AEN, $\overline{SBHE}$, SA0–9 to ↓ $\overline{IOCS16}$		0	35	ns
t_{IOCS2}	AEN, $\overline{SBHE}$, SA0–9 to $\overline{IOCS16}$ Tristated		0	25	ns
Master Mode Bus Acquisition					
t_{MMA1}	$\overline{REF}$ Inactive to ↓ $\overline{DACK}$		5		ns
t_{MMA2}	↑ DRQ to ↓ $\overline{DACK}$		0		ns
t_{MMA3}	$\overline{DACK}$ Inactive		55		ns
t_{MMA4}	↓ $\overline{DACK}$ to ↓ $\overline{MASTER}$			35	ns
t_{MMA5}	↓ $\overline{MASTER}$ to Active Command, $\overline{SBHE}$, SA0–19, LA17–23		125	185	ns

SWITCHING CHARACTERISTICS: BUS MASTER MODE (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Master Mode Bus Release					
t_{MMBR1}	Command Deassert to ↓ DRQ		45	65	ns
t_{MMBR2}	↓ DRQ to ↑ $\overline{DACK}$		0		ns
t_{MMBR3}	↓ DRQ to ↑ $\overline{MASTER}$		40	60	ns
t_{MMBR4}	↓ DRQ to Command, $\overline{SBHE}$, SA0–19, LA17–23 Tristated,		–15	0	ns
Master Write Cycles					
t_{MMW1}	$\overline{SBHE}$, SA0–19, LA17–23, Active to ↓ $\overline{MEMW}$	(Note 1)	EXTIME + 45	EXTIME + 65	ns
t_{MMW2}	$\overline{MEMW}$ Active	(Note 2)	MSWRA – 10	MSWRA + 5	ns
t_{MMW3}	$\overline{MEMW}$ Inactive	(Note 1)	EXTIME + 97	EXTIME + 105	ns
t_{MMW4}	↑ $\overline{MEMW}$ to $\overline{SBHE}$, SA0–19, LA17–23,SD Inactive		45	55	ns
t_{MMW5}	$\overline{SBHE}$, SA0–19, LA17–23, SD Hold After ↑ $\overline{MEMW}$		45	60	ns
t_{MMW6}	$\overline{SBHE}$, SA0–19, LA17–23, SD Setup to ↓ $\overline{MEMW}$	(Note 1)	EXTIME + 45	EXTIME + 55	ns
t_{MMW7}	↓ IOCHRDY Delay From ↓ $\overline{MEMW}$		t_{MMW2} – 175		ns
t_{MMW8}	IOCHRDY Inactive		55		ns
t_{MMW9}	↑ IOCHRDY to ↑ $\overline{MEMW}$		130		ns
t_{MMW10}	SD Active to ↓ $\overline{MEMW}$	(Note 1)	EXTIME + 20	EXTIME + 60	ns
t_{MMW11}	SD Setup to ↓ $\overline{MEMW}$	(Note 1)	EXTIME + 20	EXTIME + 60	ns
Master Read Cycles					
t_{MMR1}	$\overline{SBHE}$, SA0–19, LA17–23, Active to ↓ $\overline{MEMR}$	(Note 1)	EXTIME + 45	EXTIME + 60	ns
t_{MMR2}	$\overline{MEMR}$ Active	(Note 2)	MSRDA – 10	MSRDA + 5	ns
t_{MMR3}	$\overline{MEMR}$ Inactive	(Note 1)	EXTIME + 97	EXTIME + 105	ns
t_{MMR4}	↑ $\overline{MEMR}$ to $\overline{SBHE}$, SA0–19, LA17–23 Inactive		45	55	ns
t_{MMR5}	$\overline{SBHE}$, SA0–19, LA17–23 Hold After ↑ $\overline{MEMW}$		45	55	ns
t_{MMR6}	$\overline{SBHE}$, SA0–19, LA17–23 Setup to ↓ $\overline{MEMR}$	(Note 1)	EXTIME + 45	EXTIME + 55	ns
t_{MMR7}	↓ IOCHRDY Delay From ↓ $\overline{MEMR}$		t_{MMR2} – 175		ns
t_{MMR8}	IOCHRDY Inactive		55		ns
t_{MMR9}	↑ IOCHRDY to ↑ $\overline{MEMR}$		130		ns
t_{MMR10}	SD Setup to ↑ $\overline{MEMR}$		30		ns
t_{MMR11}	SD Hold After ↑ $\overline{MEMR}$		0		ns

SWITCHING CHARACTERISTICS: BUS MASTER MODE (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Master Mode Address PROM Read					
t_{MA1}	↓ $\overline{IOR}$ to ↓ $\overline{APCS}$		125	260	ns
t_{MA2}	$\overline{APCS}$ Active		140	155	ns
t_{MA3}	PRDB Setup to ↑ $\overline{APCS}$		20		ns
t_{MA4}	PRDB Hold After ↑ $\overline{APCS}$		0		ns
t_{MA5}	↑ $\overline{APCS}$ to ↑ IOCHRDY		45	65	ns
t_{MA6}	SD Valid From ↑ IOCHRDY		0	10	ns
Master Mode Boot PROM Read					
t_{MB1}	$\overline{REF}$, $\overline{SBHE}$,SA0–19 Setup to ↓ $\overline{SMEMR}$		10		ns
t_{MB2}	$\overline{REF}$, $\overline{SBHE}$,SA0–19 Hold ↑ $\overline{SMEMR}$		5		ns
t_{MB3}	↓ IOCHRDY Delay From ↓ $\overline{SMEMR}$		0	35	ns
t_{MB4}	$\overline{SMEMR}$ Inactive		55		ns
t_{MB5}	↓ $\overline{SMEMR}$ to ↓ $\overline{BPCS}$		125	260	ns
t_{MB6}	$\overline{BPCS}$ Active		140	155	ns
t_{MB7}	↑ $\overline{BPCS}$ to ↑ IOCHRDY		45	65	ns
t_{MB8}	PRDB Setup to ↑ $\overline{BPCS}$		20		ns
t_{MB9}	PRDB Hold After ↑ $\overline{BPCS}$		0		ns
t_{MB10}	SD Valid From ↑ IOCHRDY		0	10	ns
t_{MB11}	SD Hold After ↑ $\overline{SMEMR}$		0	20	ns

Notes:

1. *EXTIME is 100 ns when ISACSR2, bit 4, is cleared (default). EXTIME is 0 ns when ISACSR2, bit 4, is set.*
2. *MSRDA and MSWDA are parameters which are defined in registers ISACSR0 and ISACSR1, respectively.*

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Input/Output Write Timing					
t_{IOW1}	AEN, $\overline{SBHE}$, SA0–9 Setup to ↓ $\overline{IOW}$		10		ns
t_{IOW2}	AEN, $\overline{SBHE}$,SA0–9 Hold After ↑ $\overline{IOW}$		5		ns
t_{IOW3}	$\overline{IOW}$ Assertion		150		ns
t_{IOW4}	$\overline{IOW}$ Inactive		55		ns
t_{IOW5}	SD Setup to ↑ $\overline{IOW}$		10		ns
t_{IOW6}	SD Hold After ↑ $\overline{IOW}$		10		ns
t_{IOW7}	↓ IOCHRDY Delay From ↓ $\overline{IOW}$		0	35	ns
t_{IOW8}	IOCHRDY Inactive		125		ns
t_{IOW9}	↑IOCHRDY to ↑ $\overline{IOW}$		0		ns
Input/Output Read Timing					
t_{IOR1}	AEN, $\overline{SBHE}$, SA0–9 Setup to ↓ $\overline{IOR}$		15		ns
t_{IOR2}	AEN, $\overline{SBHE}$,SA0–9 Hold After ↑ $\overline{IOR}$		5		ns
t_{IOR3}	$\overline{IOR}$ Inactive		55		ns
t_{IOR4}	SD Hold After ↑ $\overline{IOR}$		0	20	ns
t_{IOR5}	SD Valid From ↓ $\overline{IOR}$		0	110	ns
t_{IOR6}	↓ IOCHRDY Delay From ↓ $\overline{IOR}$		0	35	ns
t_{IOR7}	IOCHRDY Inactive		125		ns
t_{IOR8}	SD Valid From ↑ IOCHRDY		–130	10	ns
Memory Write Timing					
t_{MW1}	SA0–9, PRAB10–15, $\overline{SBHE}$, ↓ $\overline{SMAM}$ Setup to ↓ $\overline{MEMW}$		10		ns
t_{MW2}	SA0–9, PRAB10–15, SBHE, ↑ $\overline{SMAM}$ Hold After ↑ $\overline{MEMW}$		5		ns
t_{MW3}	$\overline{MEMW}$ Assertion		150		ns
t_{MW4}	$\overline{MEMW}$ Inactive		55		ns
t_{MW5}	SD Setup to ↑ $\overline{MEMW}$		10		ns
t_{MW6}	SD Hold After ↑ $\overline{MEMW}$		10		ns
t_{MW7}	↓ IOCHRDY Delay From ↓ $\overline{MEMW}$		0	35	ns
t_{MW8}	IOCHRDY Inactive		125		ns
t_{MW9}	↑ $\overline{MEMW}$ to ↑ IOCHRDY		0		ns

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Memory Read Timing					
t_{MR1}	SA0–9, PRAB10–15, $\overline{SBHE}$, ↓ SMAM/$\overline{BPAM}$ Setup to ↓ $\overline{MEMR}$		10		ns
t_{MR2}	SA0–9, PRAB10–15, $\overline{SBHE}$, ↑ SMAM/$\overline{BPAM}$ Hold After ↑ $\overline{MEMR}$		5		ns
t_{MR3}	$\overline{MEMR}$ Inactive		55		ns
t_{MR4}	SD Hold After ↑ $\overline{MEMR}$		0	20	ns
t_{MR5}	SD Valid From ↓ $\overline{MEMR}$		0	110	ns
t_{MR6}	↓ IOCHRDY Delay From ↓ $\overline{MEMR}$		0	35	ns
t_{MR7}	IOCHRDY Inactive		125		ns
t_{MR8}	SD Valid From ↑ IOCHRDY		–130	10	ns
I/O To Memory Command Inactive					
t_{IOM1}	↓ $\overline{IOW}$/$\overline{MEMW}$ to ↓ ($\overline{S}$)$\overline{MEMR}$/$\overline{IOR}$		55		ns
t_{IOM2}	↓ ($\overline{S}$)$\overline{MEMR}$/$\overline{IOR}$ to ↓ $\overline{IOW}$/$\overline{MEMW}$		55		ns
IOCS16 Timing					
t_{IOCS1}	AEN, $\overline{SBHE}$, SA0–9 to ↓ $\overline{IOCS16}$		0	35	ns
t_{IOCS2}	AEN, $\overline{SBHE}$, SA0–9 to $\overline{IOCS16}$ tristated		0	25	ns
SRAM Read/Write, Boot PROM Read, Address PROM Read on Private Bus					
t_{PR1}	↑ $\overline{ABOE}$ to PRAB10–15 Tristated		0	20	ns
t_{PR2}	↑ $\overline{ABOE}$ to PRAB10–15 Active (Driven by Am79C960)		25	55	ns
t_{PR3}	PRAB10–15 Inactive to ↓ $\overline{ABOE}$		5	25	ns
t_{PR4}	PRAB Change to PRAB Change, SRAM Access		95	105	ns
t_{PR5}	PRDB Setup to PRAB Change, SRAM Access		20		ns
t_{PR6}	PRDB Hold After PRAB Change, SRAM Access		0		ns
t_{PR7}	PRAB Change to PRAB Change, APROM Access		145	155	ns

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
SRAM Read/Write, Boot PROM Read, Address PROM Read on Private Bus (continued)					
t_{PR8}	PRDB Setup to PRAB Change, APROM Access		20		ns
t_{PR9}	PRDB Hold After PRAB Change, APROM Access		0		ns
t_{PR10}	PRAB Change to PRAB Change, BPROM Access		145	155	ns
t_{PR11}	PRDB Setup to PRAB Change, BPROM Access		20		ns
t_{PR12}	PRDB Hold After PRAB Change, BPROM Access		0		ns
t_{PR13}	PRAB Change to PRAB Change, SRAM Write		145	155	ns
t_{PR14}	PRAB Change to $\downarrow \overline{SRWE}$		20	30	ns
t_{PR15}	PRAB Change to $\uparrow \overline{SRWE}$		120	130	ns

SWITCHING CHARACTERISTICS: EADI

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
t_{EAD1}	SRD Setup to ↑ SRDCLK		40		ns
t_{EAD2}	SRD Hold to ↑ SRDCLK		40		ns
t_{EAD3}	$\overline{SF/BD}$ Change to ↓ SRDCLK		−15	+15	ns
t_{EAD4}	$\overline{EAR}$ Deassertion to ↑ SRDCLK (First Rising Edge)		50		ns
t_{EAD5}	$\overline{EAR}$ Assertion After SFD Event (Packet Rejection)		0	51,090	ns
t_{EAD6}	$\overline{EAR}$ Assertion		110		ns

Note:

External Address Detection Interface is invoked by setting bit 3 in ISACSR2 and resetting bit 0 in ISACSR2. External MAU select is not available when EADISEL bit is set.

SWITCHING CHARACTERISTICS: JTAG (IEEE 1149.1) INTERFACE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
t_{JTG1}	TCK HIGH Assertion		20		ns
t_{JTG2}	TCK Period		50		ns
t_{JTG3}	TDI Setup Before ↑ TCK		5		ns
t_{JTG4}	TDI, TMS Hold After ↑ TCK		5		ns
t_{JTG5}	TMS Setup Before ↑ TCK		8		ns
t_{JTG6}	TDO Active After ↓ TCK		0	30	ns
t_{JTG7}	TDO Change After ↓ TCK		0	30	ns
t_{JTG8}	TDO Tristate After ↓ TCK		0	25	ns

Note:

JTAG logic is reset with an internal Power-On Reset circuit independent of Sleep Modes.

SWITCHING CHARACTERISTICS: GPSI

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Transmit Timing					
t_{GPT1}	TCLK Period (802.3 Compliant)		99.99	100.01	ns
t_{GPT2}	TCLK HIGH Time		40	60	ns
t_{GPT3}	TX and TENA Delay from ↑ TCLK		0	70	ns
t_{GPT4}	RENA Setup Before ↑ TCLK (Last Bit)		210		ns
t_{GPT5}	RENA Hold After ↓ TENA		0		ns
t_{GPT6}	CLSN Active Time to Trigger Collision	(Note 1)	110		ns
t_{GPT7}	CLSN Active to ↓ RENA to Prevent LCAR Assertion		0		ns
t_{GPT8}	CLSN Active to ↓ RENA for SQE Hearbeat Window		0	4.0	µs
T_{gpt9}	CLSN Active to ↑ Rena for Normal Collision		0	51.2	µs
Receive Timing					
t_{GPR1}	RCLK Period	(Note 2)	80	120	ns
t_{GPR2}	RCLK HIGH Time	(Note 2)	30	80	ns
t_{GPR3}	RCLK LOW Time	(Note 2)	30	80	ns
t_{GPR4}	RX and RENA Setup to ↑ RCLK		15		ns
t_{GPR5}	RX Hold After ↑ RCLK		15		ns
t_{GPR6}	RENA Hold After ↓ RCLK		0		ns
t_{GPR7}	CLSN Active to First ↑ RCLK (Collision Recognition)		0		ns
t_{GPR8}	CLSN Active to ↑ RCLK for Address Type Designation Bit	(Note 3)	51.2		µs
t_{GPR9}	CLSN Setup to Last ↑ RCLK for Collision Recognition		210		ns
t_{GPR10}	CLSN Active		110		ns
t_{GPR11}	CLSN Inactive Setup to First ↑ RCLK		300		ns
t_{GPR12}	CLSN Inactive Hold to Last ↑ RCLK		300		ns

Notes:

1. *CLSN must be asserted for a continuous period of 110 ns or more. Assertion for less than 110 ns period may or may not result in CLSN recognition.*
2. *RCLK should meet jitter requirements of IEEE 802.3 specification.*
3. *CLSN assertion before 51.2 µs will be indicated as a normal collision. CLSN assertion after 51.2 µs will be considered as a Late Receive Collision.*

SWITCHING CHARACTERISTICS: AUI

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
AUI Port					
t_{DOTR}	DO+,DO- Rise Time (10% to 90%)		2.5	5.0	ns
t_{DOTF}	DO+,DO- Fall Time (90% to 10%)		2.5	5.0	ns
t_{DORM}	DO+,DO- Rise and Fall Time Mismatch			1.0	ns
t_{DOETD}	DO+/- End of Transmission		200	375	ns
t_{PWODI}	DI Pulse Width Accept/Reject Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 1)	15	45	ns
t_{PWKDI}	DI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 2)	136	200	ns
t_{PWOCI}	CI Pulse Width Accept/Reject Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 3)	10	26	ns
t_{PWKCI}	CI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 4)	90	160	ns
Internal MENDEC Clock Timing					
t_{X1}	XTAL1 Period	V_{IN} = External Clock	49.995	50.005	ns
t_{X1H}	XTAL1 HIGH Pulse Width	V_{IN} = External Clock	20		ns
t_{X1L}	XTAL1 LOW Pulse Width	V_{IN} = External Clock	20		ns
t_{X1R}	XTAL1 Rise Time	V_{IN} = External Clock		5	ns
t_{X1F}	XTAL1 Fall Time	V_{IN} = External Clock		5	ns

Notes:

1. *DI pulses narrower than t_{PWODI} (min) will be rejected; pulses wider than t_{PWODI} (max) will turn internal DI carrier sense on.*
2. *DI pulses narrower than t_{PWKDI} (min) will maintain internal DI carrier sense on; pulses wider than t_{PWKDI} (max) will turn internal DI carrier sense off.*
3. *CI pulses narrower than t_{PWOCI} (min) will be rejected; pulses wider than t_{PWOCI} (max) will turn internal CI carrier sense on.*
4. *CI pulses narrower than t_{PWKCI} (min) will maintain internal CI carrier sense on; pulses wider than t_{PWKCI} (max) will turn internal CI carrier sense off.*

SWITCHING CHARACTERISTICS: 10BASE-T INTERFACE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Transmit Timing					
t_{TETD}	Transmit Start of Idle		250	350	ns
t_{TR}	Transmitter Rise Time	(10% to 90%)		5.5	ns
t_{TF}	Transmitter Fall Time	(90% to 10%)		5.5	ns
t_{TM}	Transmitter Rise and Fall Time Mismatch			1	ns
t_{PERLP}	Idle Signal Period		8	24	ms
t_{PWLP}	Idle Link Pulse Width	(Note 1)	75	120	ns
t_{PWPLP}	Predistortion Idle Link Pulse Width	(Note 1)	45	55	ns
t_{JA}	Transmit Jabber Activation Time		20	150	ms
t_{JR}	Transmit Jabber Reset Time		250	750	ms
Receive Timing					
t_{PWNRD}	RXD Pulse Width Not to Turn Off Internal Carrier Sense	VIN > VTHS (min)	136	–	ns
t_{PWROFF}	RXD Pulse Width to Turn Off	VIN > VTHS (min)		200	ns

Note:

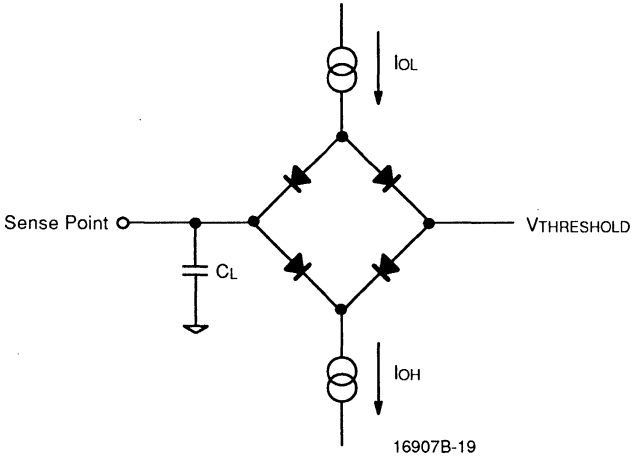
1. Not tested; parameter guaranteed by characterization.

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	Must Be Steady	Will Be Steady
	May Change from H to L	Will Be Changing from H to L
	May Change from L to H	Will Be Changing from L to H
	Don't Care Any Change Permitted	Changing State Unknown
	Does Not Apply	Center Line is High Impedance "Off" State

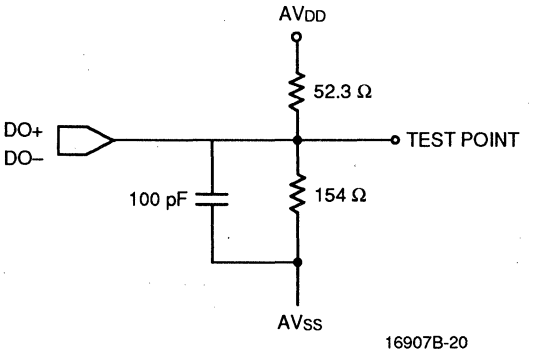
KS000010

SWITCHING TEST CIRCUITS

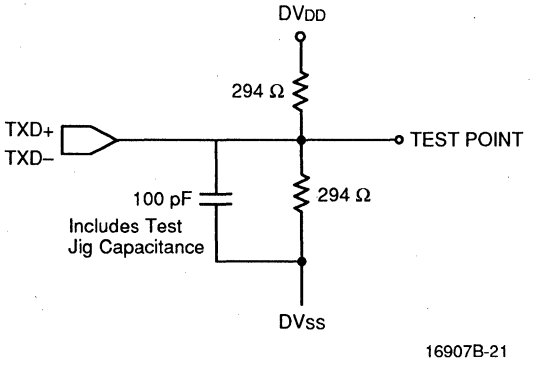


Normal and Three-State Outputs

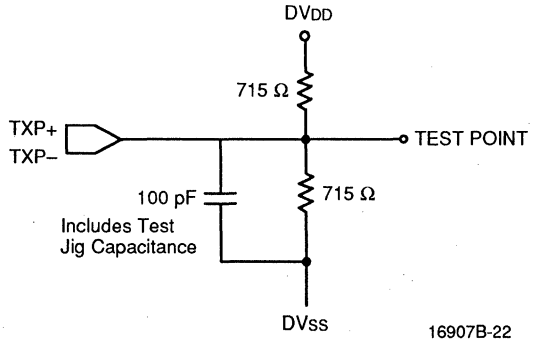
SWITCHING TEST CIRCUITS



AUI DO Switching Test Circuit

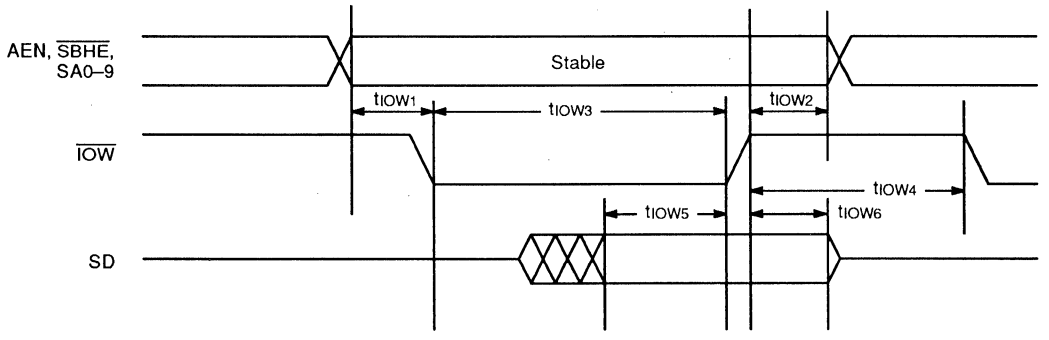


TXD Switching Test Circuit



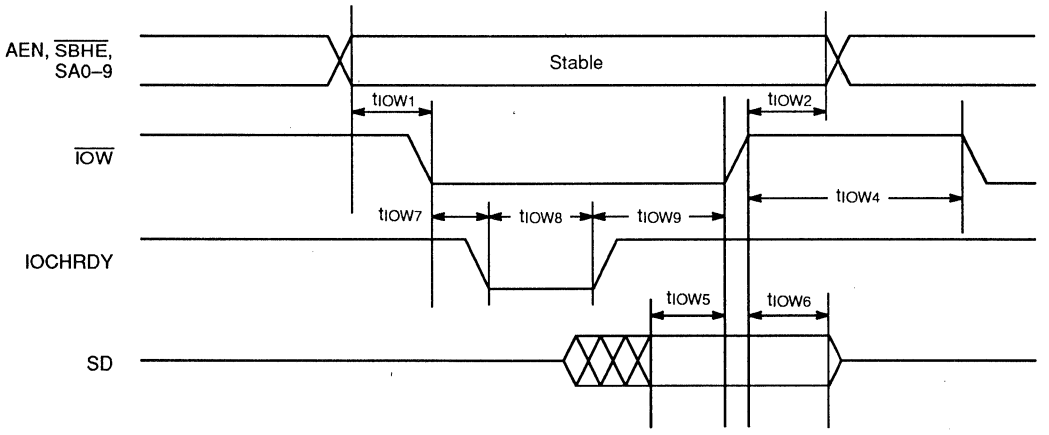
TXP Outputs Test Circuit

SWITCHING WAVEFORMS: BUS MASTER MODE



16907B-23

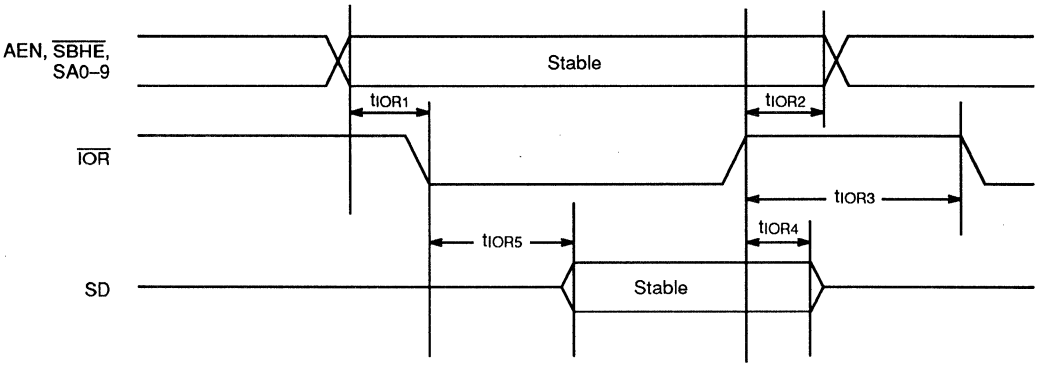
I/O Write without Wait States



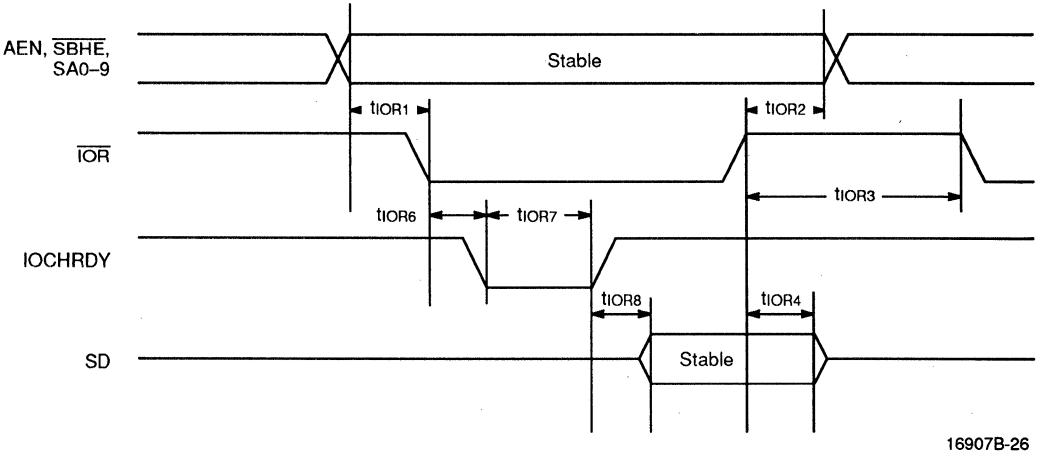
16907B-24

I/O Write with Wait States

SWITCHING WAVEFORMS: BUS MASTER MODE

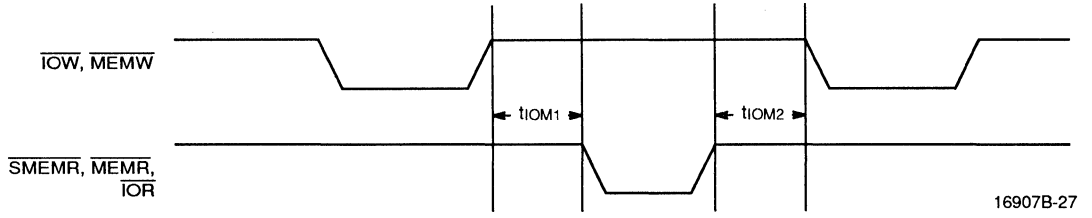


I/O Read without Wait States

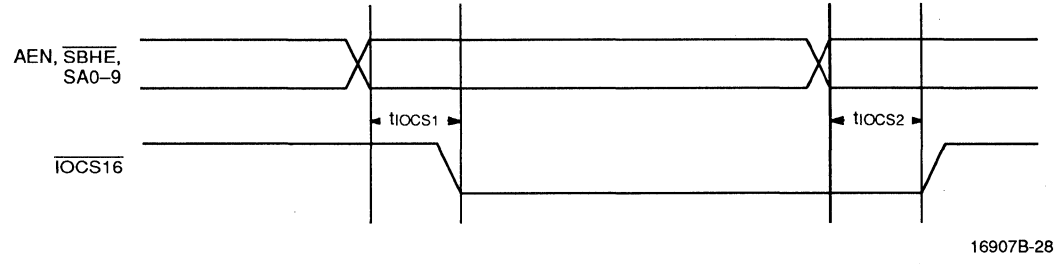


I/O Read with Wait States

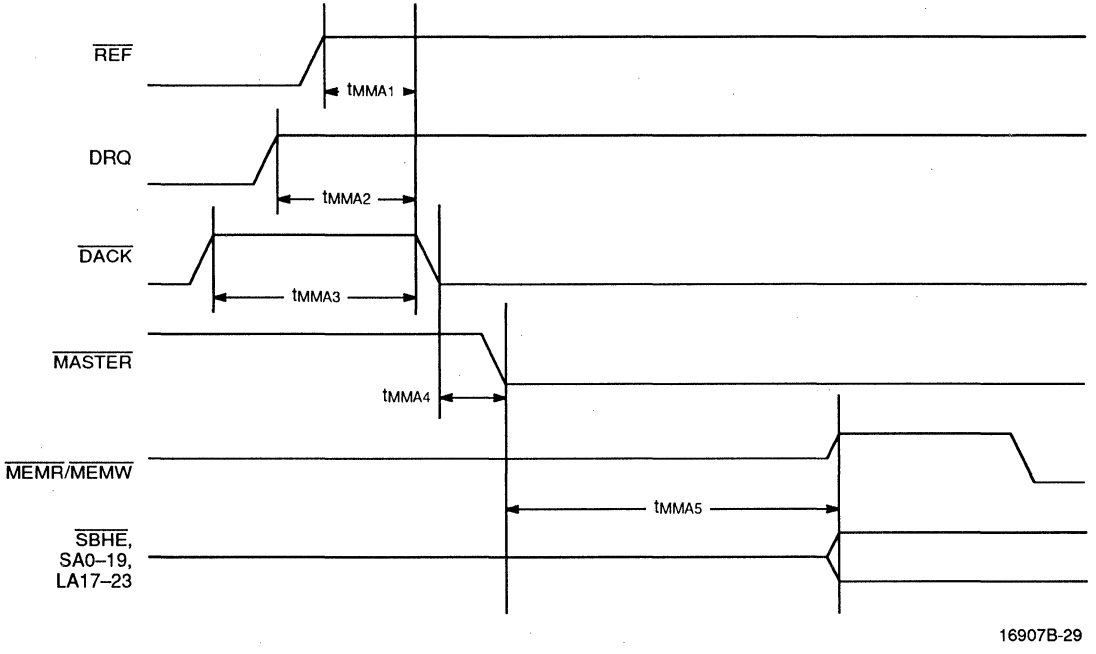
SWITCHING WAVEFORMS: BUS MASTER MODE



I/O to Memory Command Inactive Time

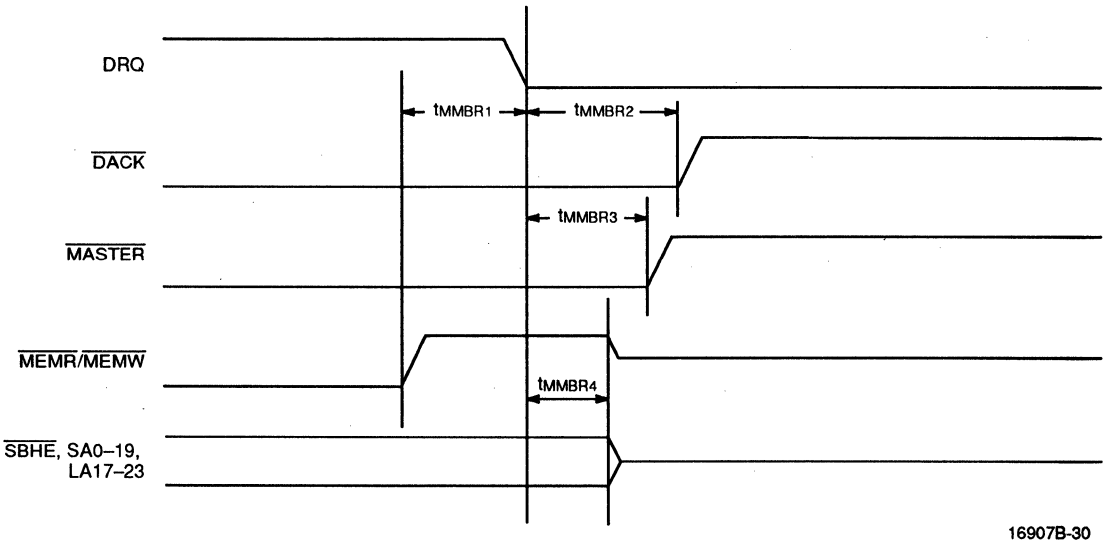


IOCS16 Timings

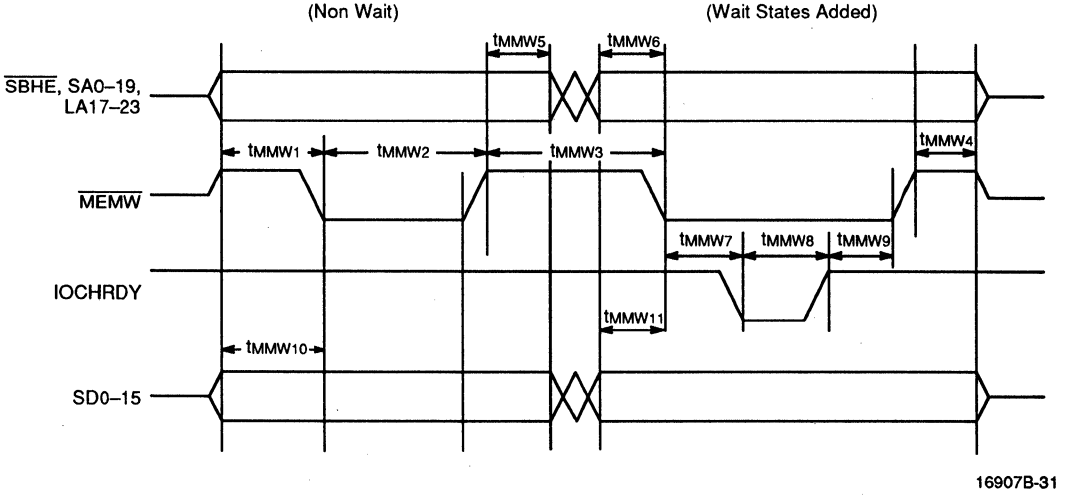


Bus Acquisition

SWITCHING WAVEFORMS: BUS MASTER MODE

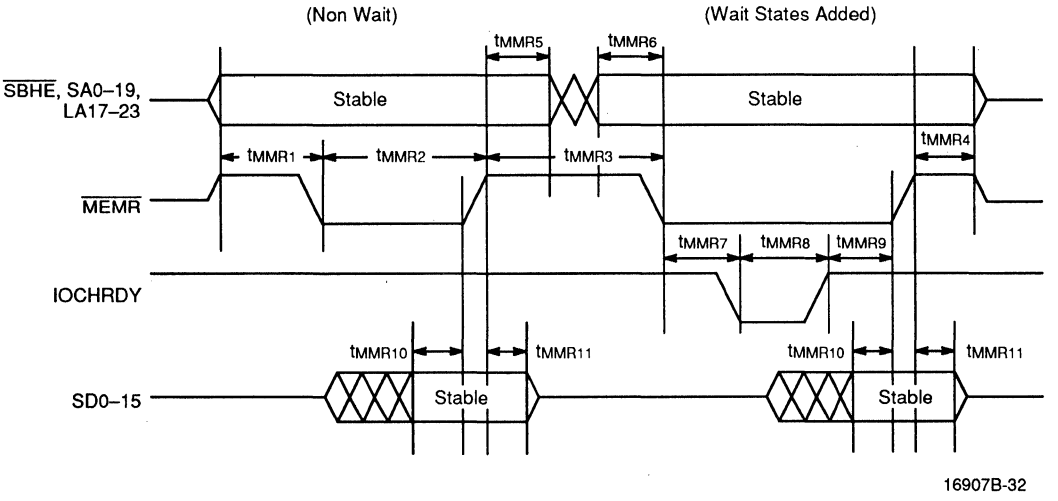


Bus Release

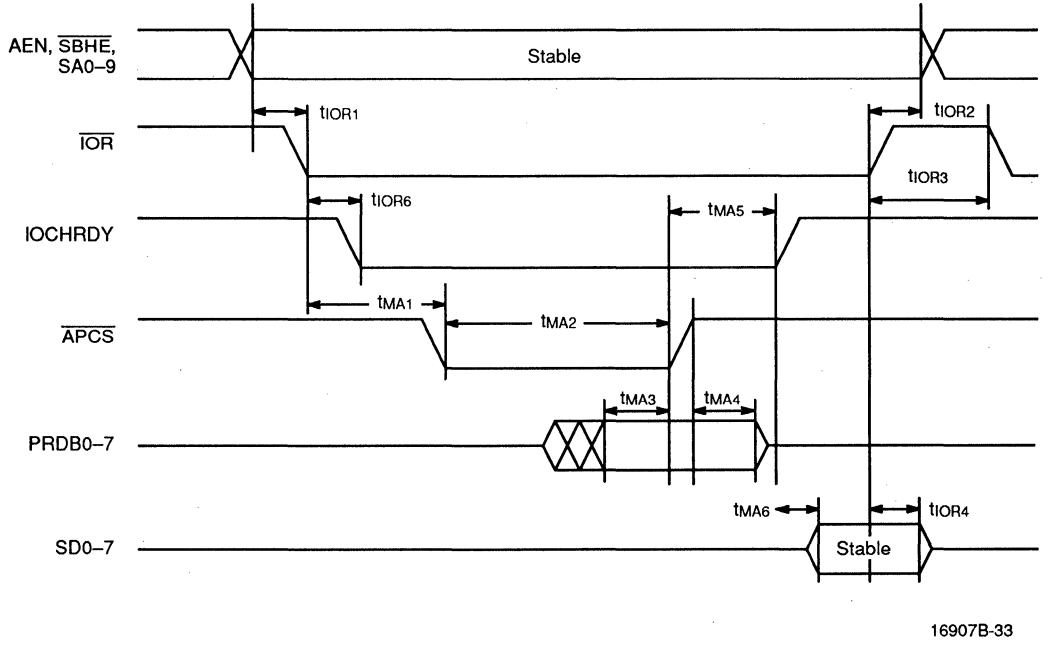


Write Cycles

SWITCHING WAVEFORMS: BUS MASTER MODE

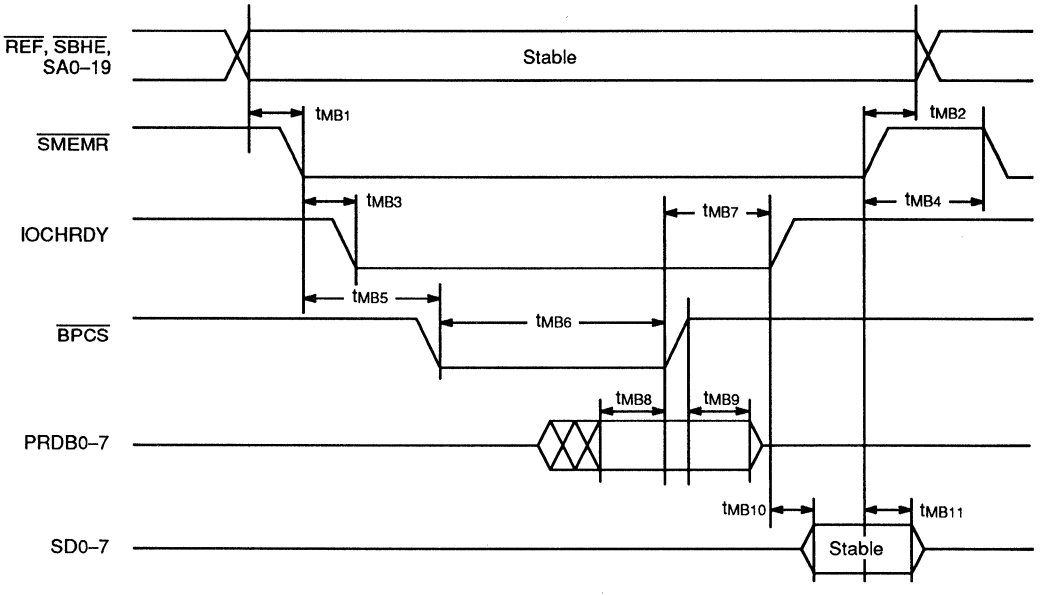


Read Cycles



Address PROM Read Cycle

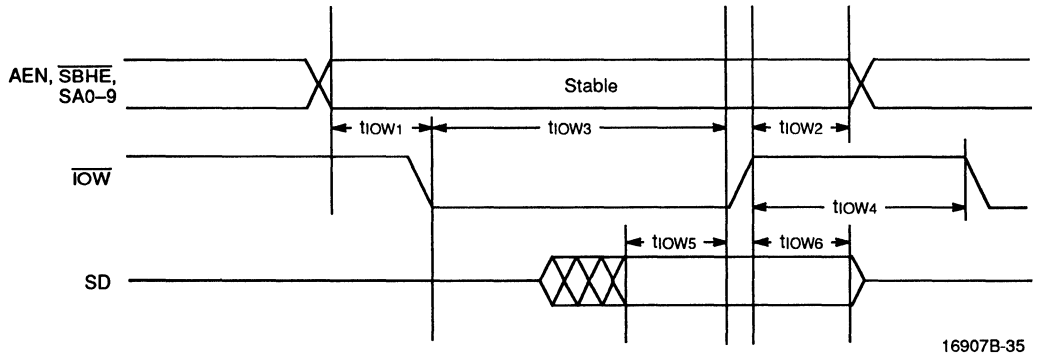
SWITCHING WAVEFORMS: BUS MASTER MODE



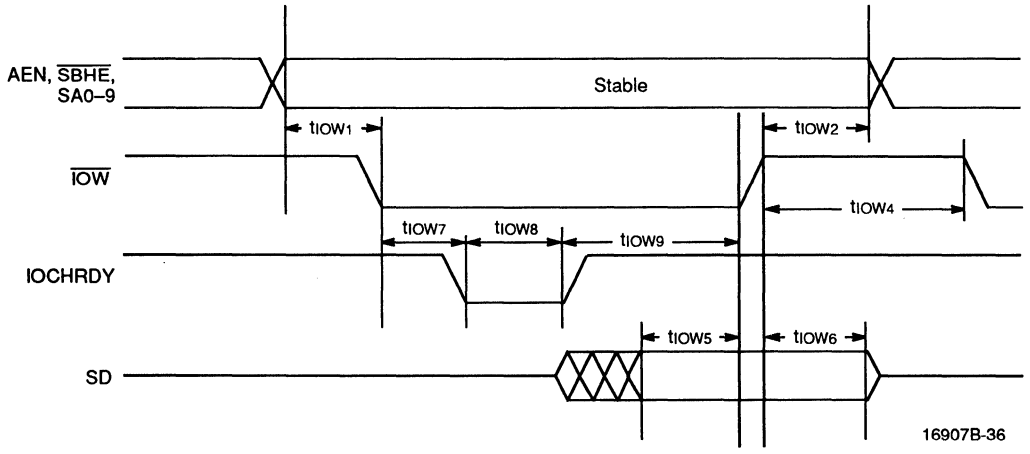
16907B-34

Boot PROM Read Cycle

SWITCHING WAVEFORMS: SHARED MEMORY MODE

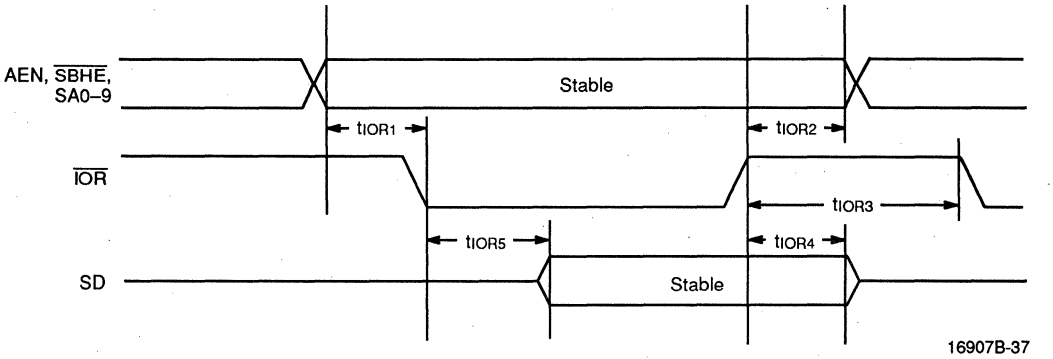


I/O Write without Wait States

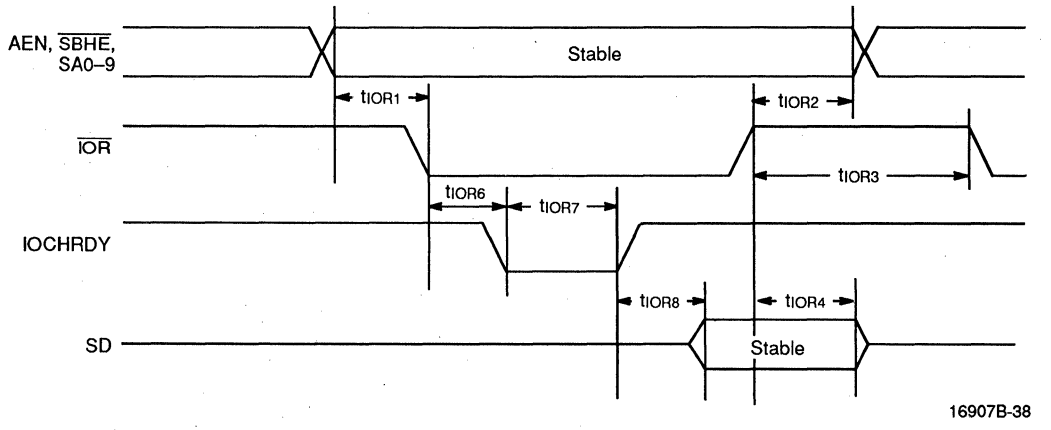


I/O Write with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE

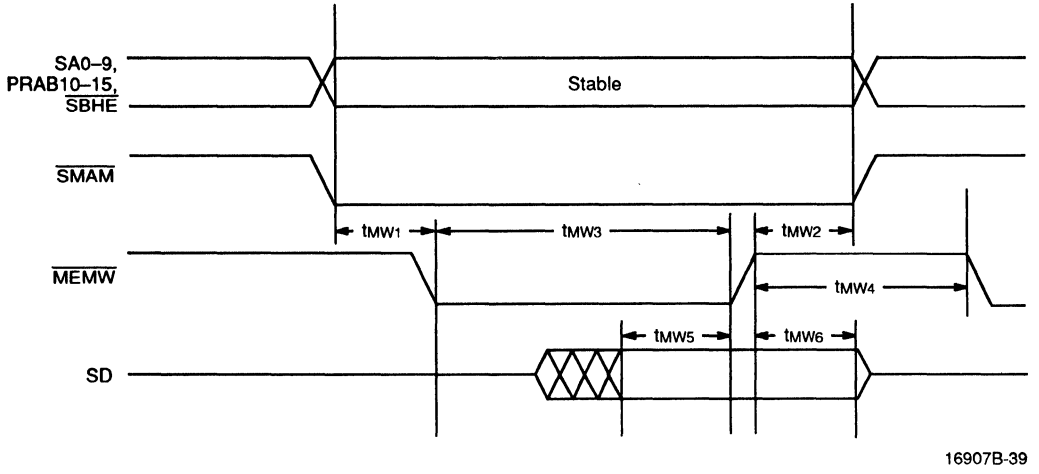


I/O Read without Wait States

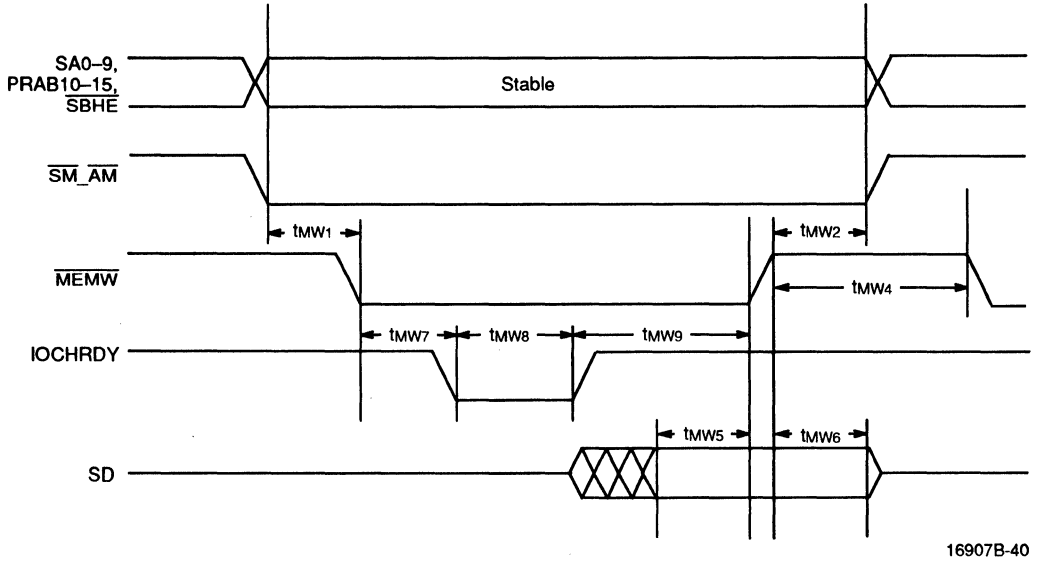


I/O Read with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE

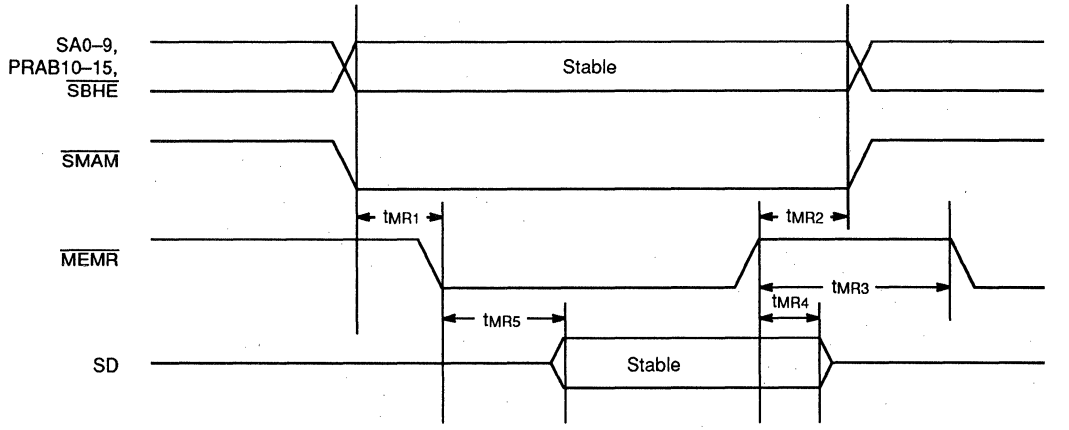


Memory Write without Wait States



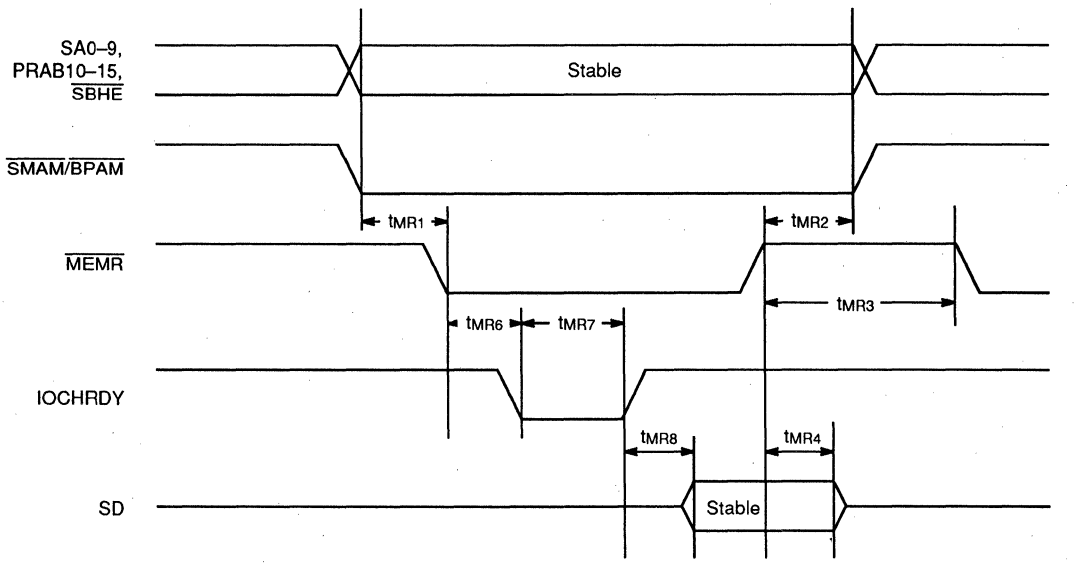
Memory Write with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE



16907B-41

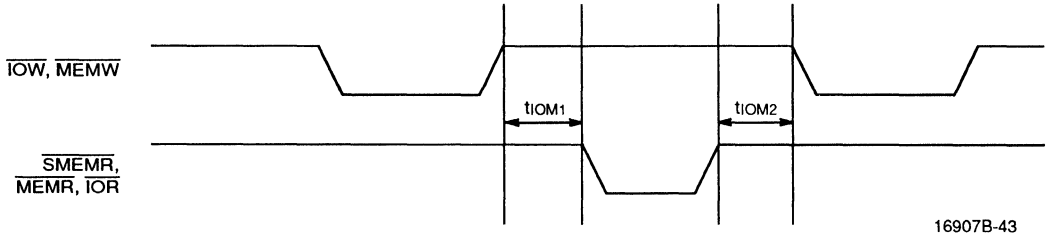
Memory Read without Wait States



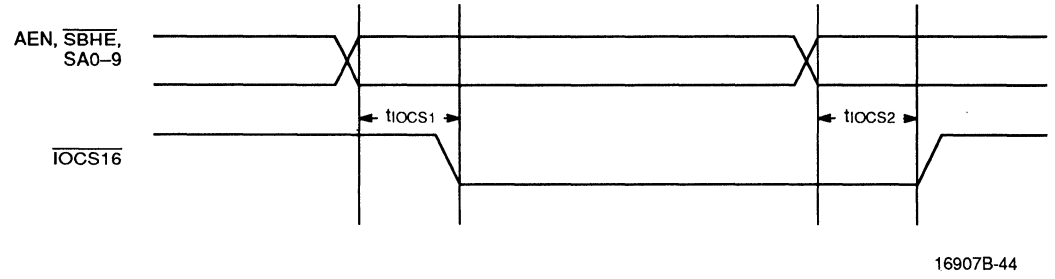
16907B-42

Memory Read with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE

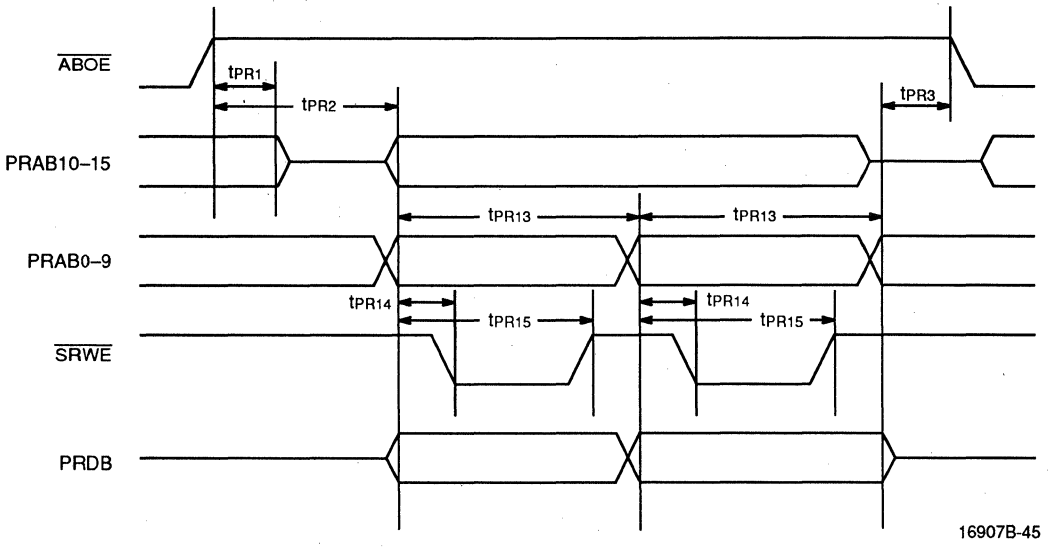


I/O to Memory Command Inactive Time

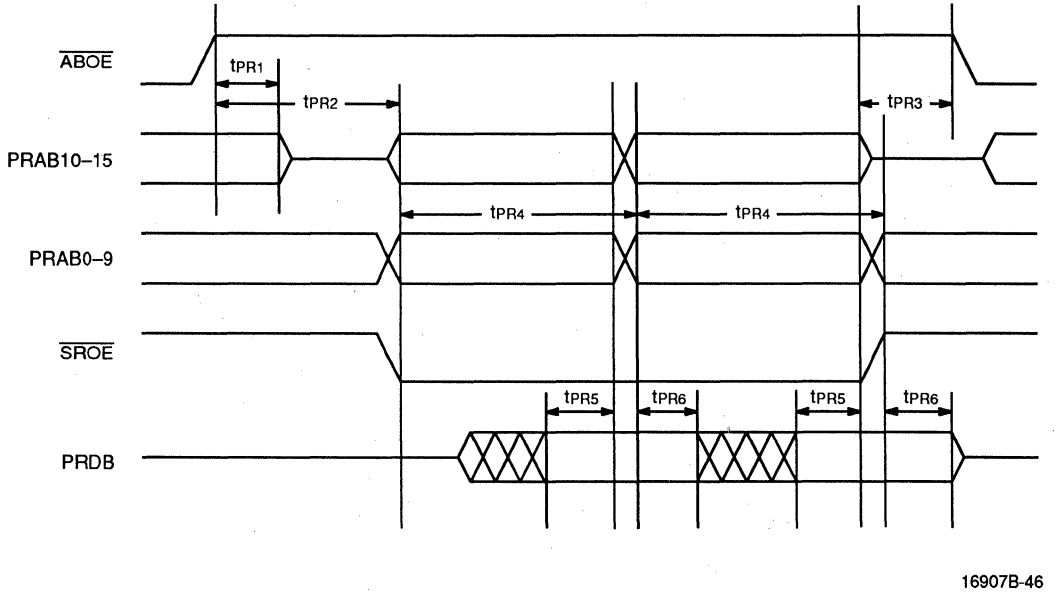


IOCS16 Timings

SWITCHING WAVEFORMS: SHARED MEMORY MODE

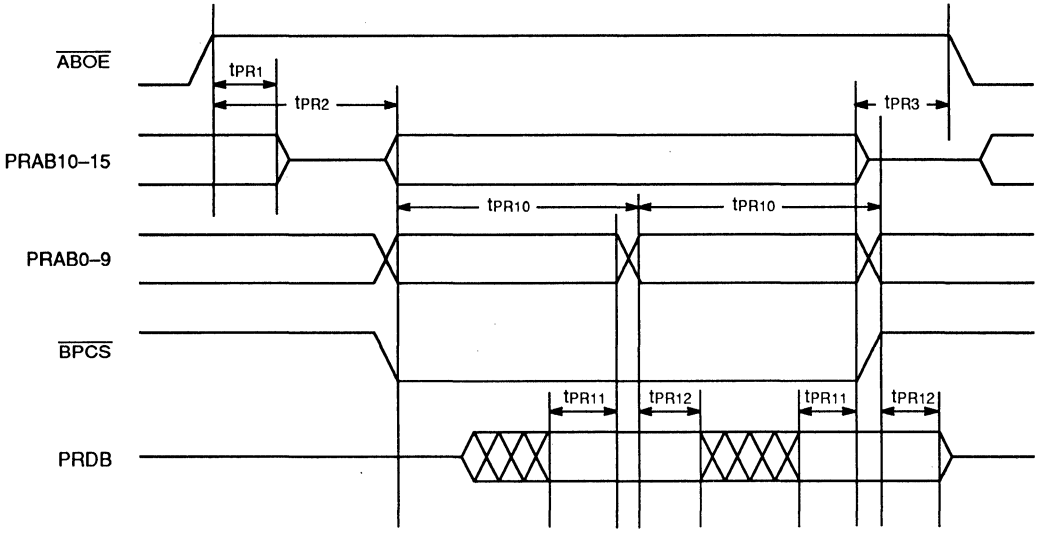


SRAM Write on Private Bus



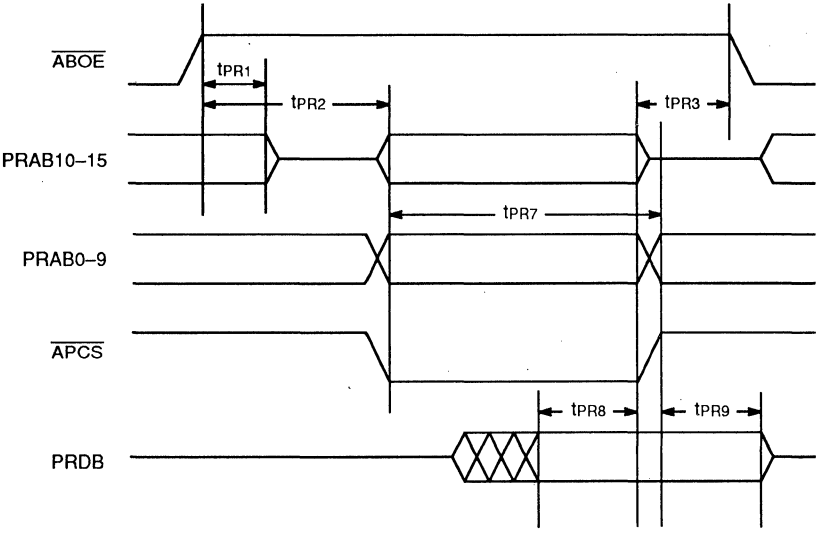
SRAM Read on Private Bus

SWITCHING WAVEFORMS: SHARED MEMORY MODE



16907B-47

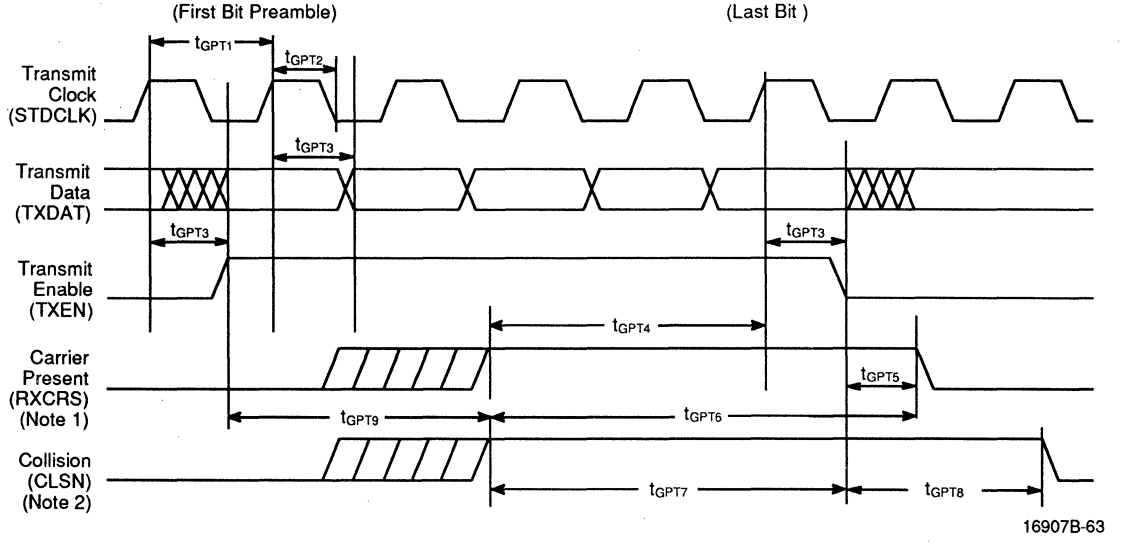
Boot PROM Read on Private Bus



16907B-48

Address PROM Read on Private Bus

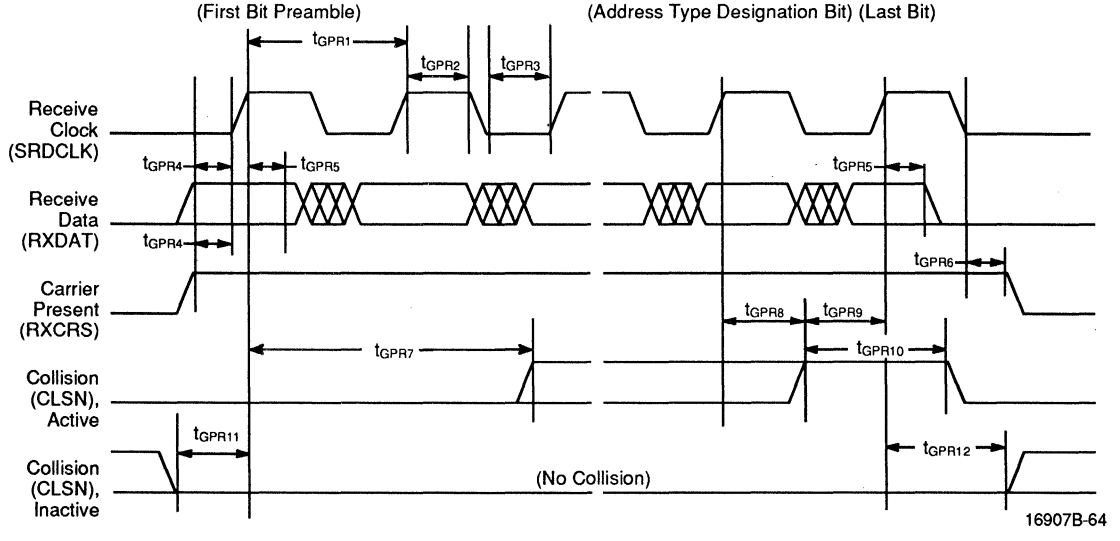
SWITCHING WAVEFORMS: GPSI



Notes:

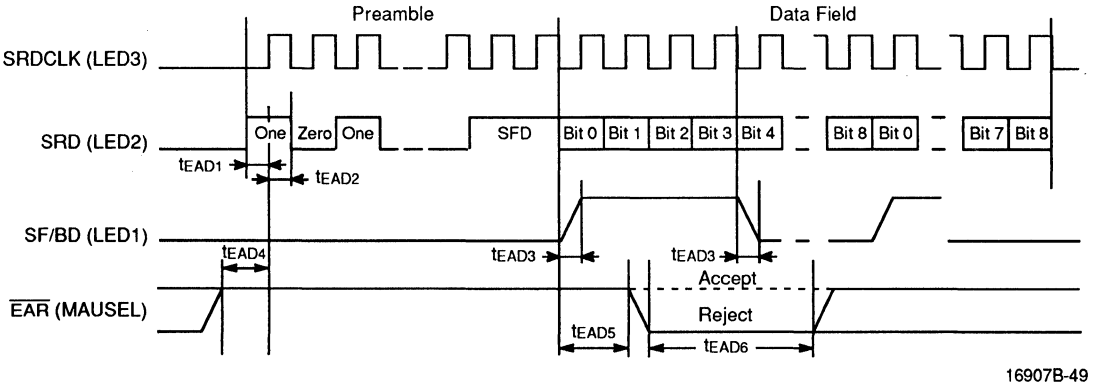
1. *If RXCRS is not present during transmission, LCAR bit in TMD3 will be set.*
2. *If CLSN is not present during or shortly after transmission, CERR in CSR0 will be set.*

Transmit Timing



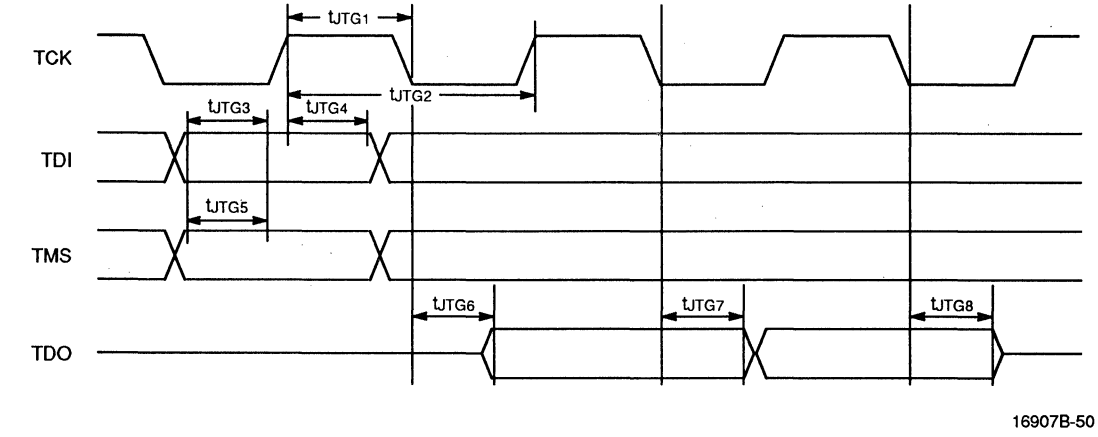
Receive Timing

SWITCHING WAVEFORMS: EADI



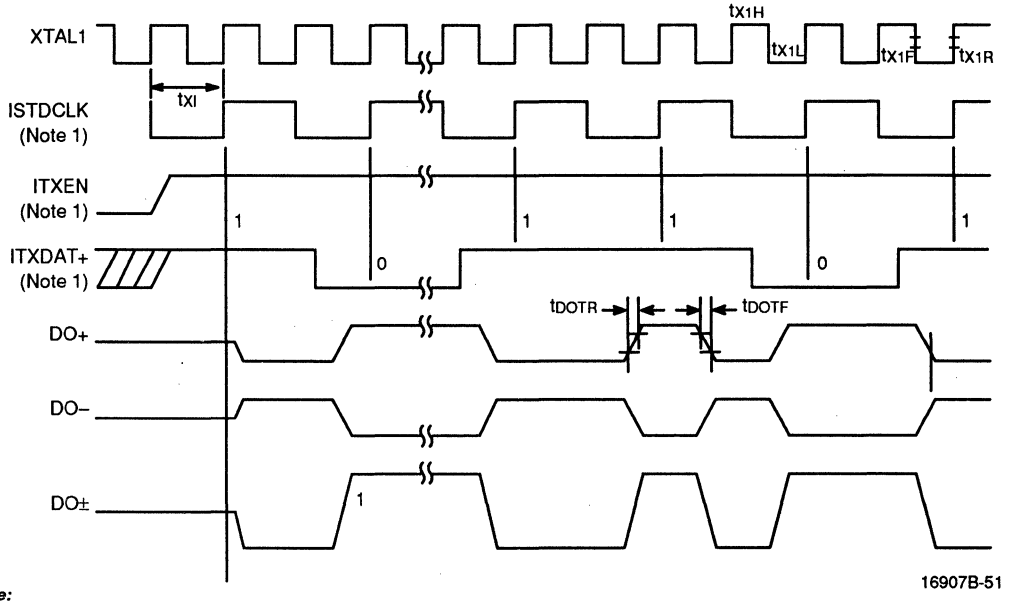
EADI Reject Timing

SWITCHING WAVEFORMS: JTAG (IEEE 1149.1) INTERFACE



Test Access Port Timing

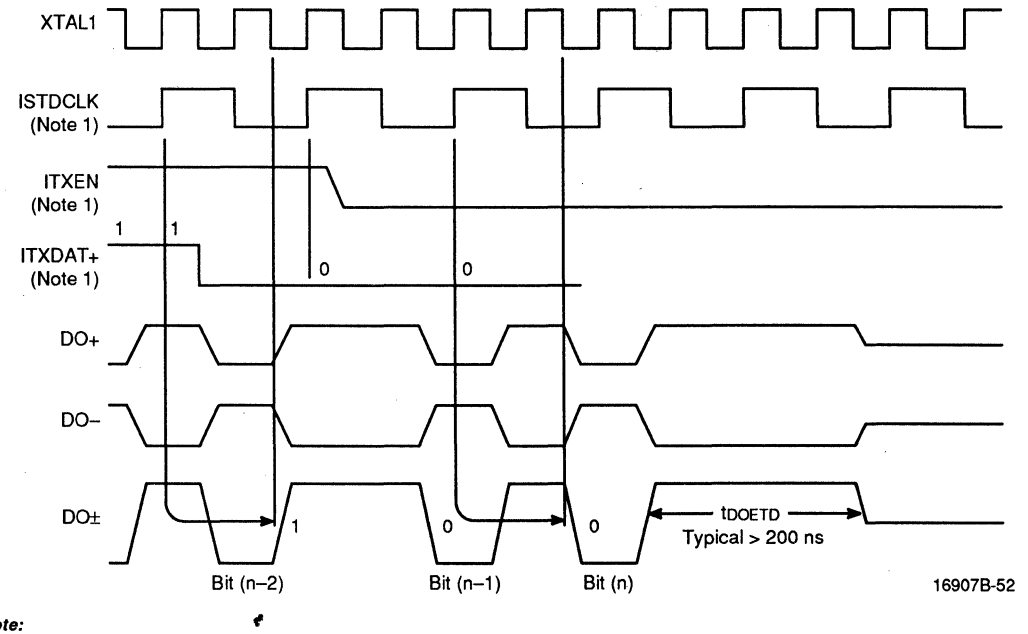
SWITCHING WAVEFORMS: AUI



Note:

1. Internal signal and is shown for clarification only.

Transmit Timing—Start of Packet

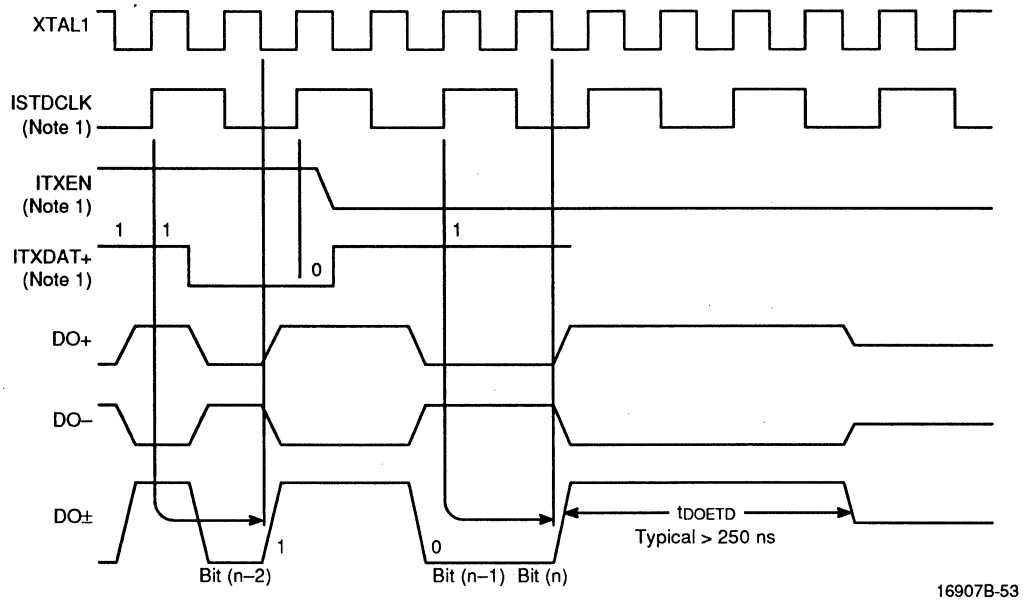


Note:

1. Internal signal and is shown for clarification only.

Transmit Timing—End of Packet (Last Bit = 0)

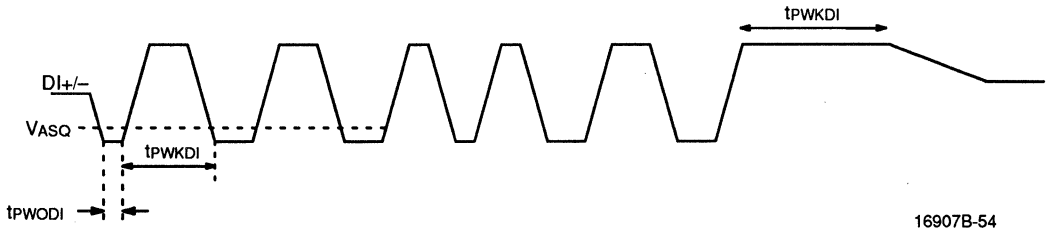
SWITCHING WAVEFORMS: AUI



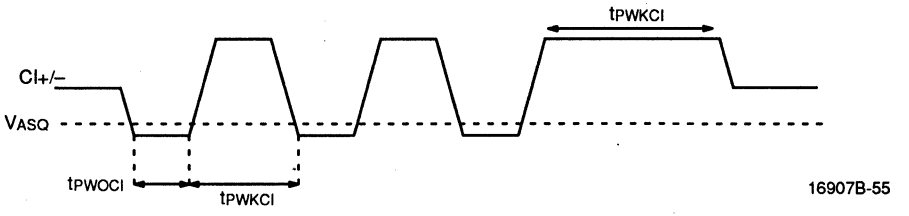
Note:
1. Internal signal and is shown for clarification only.

Transmit Timing—End of Packet (Last Bit = 1)

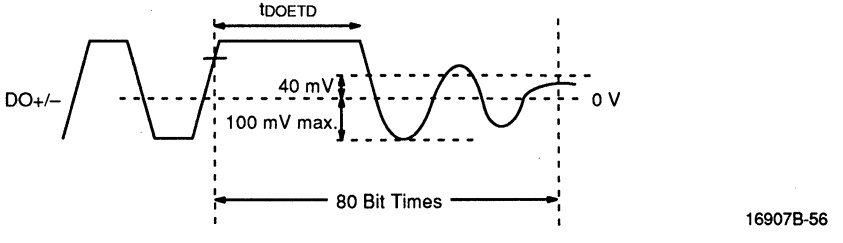
SWITCHING WAVEFORMS: AUI



Receive Timing Diagram

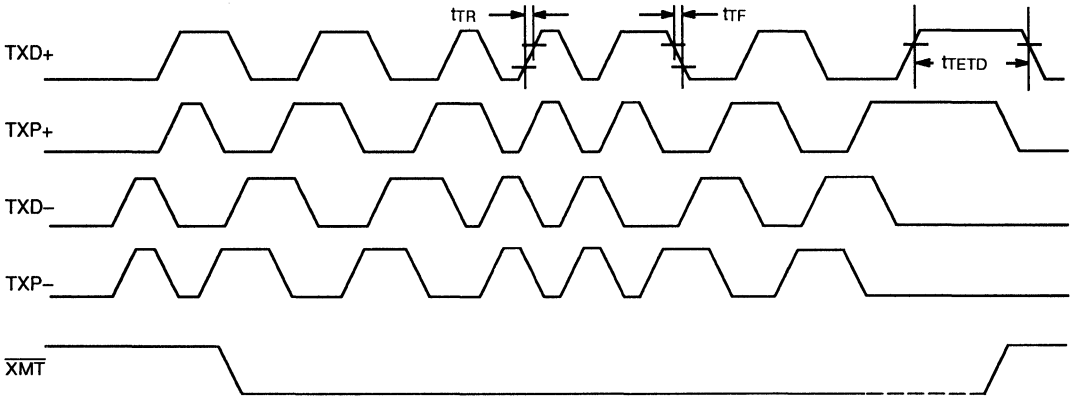


Collision Timing Diagram



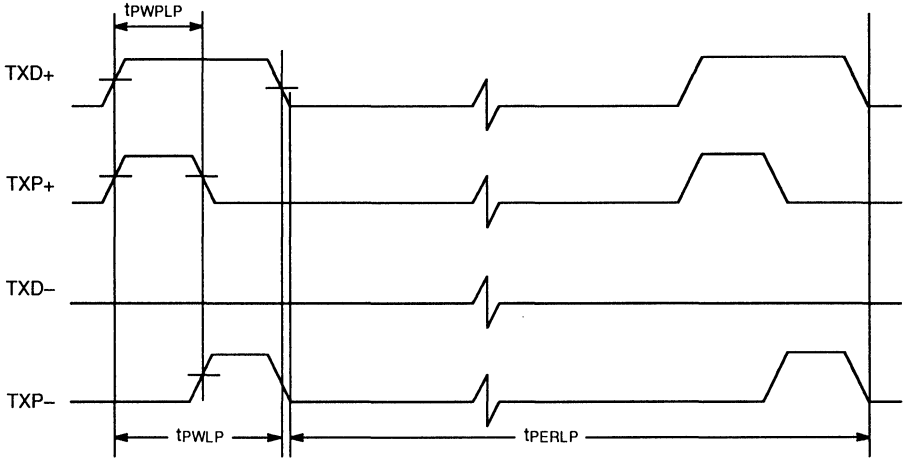
Port DO ETD Waveform

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



16907B-57

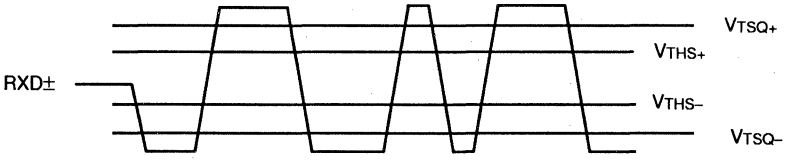
Transmit Timing



16907B-60

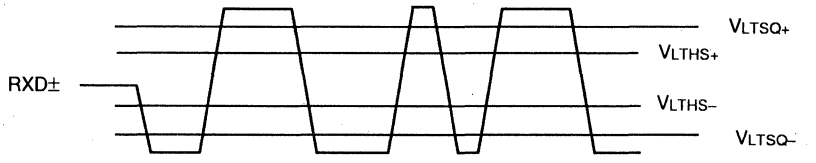
Idle Link Test Pulse

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



16907B-61

Receive Thresholds (LRT = 0; CSR15[9])



16907B-62

Receive Thresholds (LRT = 1; CSR15[9])

APPENDIX A

PCnet-ISA Compatible Media Interface Modules

PCnet-ISA COMPATIBLE 10BASE-T FILTERS AND TRANSFORMERS

The table below provides a sample list of PCnet-ISA compatible 10BASE-T filter and transformer modules available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Filters and Transformers	Filters Transformers and Choke	Filters Transformers Dual Chokes	Filters Transformers Dual Chokes
Bel Fuse	A556-2006-DE	16-pin 0.3" DIL	√			
Bel Fuse	0556-2006-00	14-pin SIP	√			
Bel Fuse	0556-2006-01	14-pin SIP			√	
Bel Fuse	0556-6392-00	16-pin 0.5" DIL			√	
Halo Electronics	FD02-101G	16-pin 0.3" DIL	√			
Halo Electronics	FD12-101G	16-pin 0.3" DIL		√		
Halo Electronics	FD22-101G	16-pin 0.3" DIL			√	
PCA Electronics	EPA1990A	16-pin 0.3" DIL	√			
PCA Electronics	EPA2013D	16-pin 0.3" DIL		√		
PCA Electronics	EPA2162	16-pin 0.3" SIP			√	
Pulse Engineering	PE-65421	16-pin 0.3" DIL	√			
Pulse Engineering	PE-65434	16-pin 0.3" SIL			√	
Pulse Engineering	PE-65445	16-pin 0.3" DIL			√	
Pulse Engineering	PE-65467	12-pin 0.5" SMT				√
Valor Electronics	PT3877	16-pin 0.3" DIL	√			
Valor Electronics	FL1043	16-pin 0.3" DIL			√	

PCnet-ISA Compatible AUI Isolation Transformers

The table below provides a sample list of PCnet-ISA compatible AUI isolation transformers available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3" DIL	50 µH
Bel Fuse	S553-0756-AE	16-pin 0.3" SMD	75 µH
Halo Electronics	TD01-0756K	16-pin 0.3" DIL	75 µH
Halo Electronics	TG01-0756W	16-pin 0.3" SMD	75 µH
PCA Electronics	EP9531-4	16-pin 0.3" DIL	50 µH
Pulse Engineering	PE64106	16-pin 0.3" DIL	50 µH
Pulse Engineering	PE65723	16-pin 0.3" SMT	75 µH
Valor Electronics	LT6032	16-pin 0.3" DIL	75 µH
Valor Electronics	ST7032	16-pin 0.3" SMD	75 µH

PCnet-ISA Compatible DC/DC Converters

The table below provides a sample list of PCnet-ISA compatible DC/DC converters available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Voltage	Remote On/Off
Halo Electronics	DCU0-0509D	24-pin DIP	5/-9	No
Halo Electronics	DCU0-0509E	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1007P	24-pin DIP	5/-9	No
PCA Electronics	EPC1054P	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1078	24-pin DIP	5/-9	Yes
Valor Electronics	PM7202	24-pin DIP	5/-9	No
Valor Electronics	PM7222	24-pin DIP	5/-9	Yes

MANUFACTURER CONTACT INFORMATION

Contact the following companies for further information on their products:

Company		U.S. and Domestic	Asia	Europe
Bel Fuse	Phone: FAX:	(201) 432-0463 (201) 432-9542	852-328-5515 852-352-3706	33-1-69410402 33-1-69413320
Halo Electronics	Phone: FAX:	(415) 969-7313 (415) 367-7158	65-285-1566 65-284-9466	
PCA Electronics (HPC in Hong Kong)	Phone: FAX:	(818) 892-0761 (818) 894-5791	852-553-0165 852-873-1550	33-1-44894800 33-1-42051579
Pulse Engineering	Phone: FAX:	(619) 674-8100 (619) 675-8262	852-425-1651 852-480-5974	353-093-24107 353-093-24459
Valor Electronics	Phone: FAX:	(619) 537-2500 (619) 537-2525	852-513-8210 852-513-8214	49-89-6923122 49-89-6926542

Recommendation for Reducing Noise Injection

DECOUPLING LOW-PASS R/C FILTER DESIGN

The PCnet-ISA controller is an integrated, single-chip Ethernet controller, which contains both digital and analog circuitry. The analog circuitry contains a high speed Phase-Locked Loop (PLL) and Voltage Controlled Oscillator (VCO). Because of the mixed signal characteristics of this chip, some extra precautions must be taken into account when designing with this device.

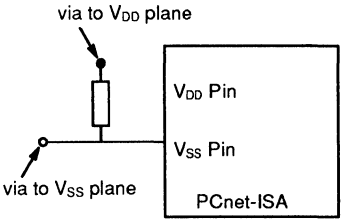
Described in this section is a simple decoupling low-pass R/C filter that can significantly increase noise immunity of the PLL circuit, thus, prevent noise from disrupting the VCO. Bit error rate, a common measurement of network performance, as a result can be drastically reduced. In certain cases the bit error rate can be reduced by orders of magnitude.

Implementation of this filter is not necessary to achieve a functional product that meets the IEEE 802.3 specification and provides adequate performance. However, this filter will help designers meet those specifications with more margin.

Digital Decoupling

The DVSS pins that are sinking the most current are those that provide the ground for the ISA bus output signals since these outputs require 24 mA drivers. The DVSS10 and DVSS12 pins provide the ground for the internal digital logic. In addition, DVSS11 provides ground for the internal digital and for the Input and I/O pins.

The CMOS technology used in fabricating the PCnet-ISA controller employs an n-type substrate. In this technology, all V_{DD} pins are electrically connected to each other internally. Hence, in a 4-layer board, when decoupling between V_{DD} and critical V_{SS} pins, the specific V_{DD} pin that you connect to is not critical. In fact, the V_{DD} connection of the decoupling capacitor can be made directly to the power plane, near the closest V_{DD} pin to the V_{SS} pin of interest. However, we recommend that the V_{SS} connection of the decoupling capacitor be made directly to the V_{SS} pin of interest as shown.



AMD recommends that at least one low-frequency bulk decoupling capacitor be used in the area of the PCnet-ISA controller. 22 µF capacitors have worked well for this. In addition, a total of 4 or 5 0.1 µF capacitors have proven sufficient around the DV_{SS} and DV_{DD} pins that supply the drivers of the ISA bus output pins.

Analog Decoupling

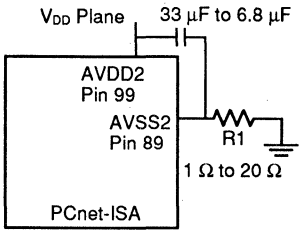
The most critical pins are the analog supply and ground pins. All of the analog supply and ground pins are located in one corner of the device. Specific requirements of the analog supply pins are listed below.

AVSS1 and AVDD3

These pins provide the power and ground for the Twisted Pair and AUI drivers. Hence, they are very noisy. A dedicated 0.1 µF capacitor between these pins is recommended.

AVSS2 and AVDD2

These pins are the most critical pins on the PCnet-ISA controller because they provide the power and ground for the PLL portion of the chip. The VCO portion of the PLL is sensitive to noise in the 60–200 kHz. range. To prevent noise in this frequency range from disrupting the VCO, AMD strongly recommends that the low-pass filter shown below be implemented on these pins. Tests using this filter have shown significantly increased noise immunity and reduced Bit Error Rate (BER) statistics in designs using the PCnet-ISA controller.



To determine the value for the resistor and capacitor, the formula is:

$$R * C \geq 88$$

Where R is in ohms and C is in microfarads. Some possible combinations are given below. To minimize the voltage drop across the resistor, the R value should not be more than 20 Ω.

R	C
2.7 Ω	33 µF
4.3 Ω	22 µF
6.8 Ω	15 µF
10 Ω	10 µF
20 Ω	6.8 µF

AVSS2 and AVDD2/AVDD4

These pins provide power and ground for the AUI and twisted pair receive circuitry. No specific decoupling has been necessary on these pins.

Alternative Method for Initialization

The PCnet-ISA controller may be initialized by performing I/O writes only. That is, data can be written directly to the appropriate control and status registers (CSR) instead of reading from the Initialization Block in memory. The registers that must be written are shown in the table below. These are followed by writing the START bit in CSR0.

Control and Status Register	Comment
CSR8	LADRF[15:0]
CSR9	LADRF[31:16]
CSR10	LADRF[47:32]
CSR11	LADRF[63:48]
CSR12	PADR[15:0]
CSR13	PADR[31:16]
CSR14	PADR[47:32]
CSR15	Mode
CSR24-25	BADR
CSR30-31	BADX
CSR47	POLLINT
CSR76	RCVRL
CSR78	XMTRL

Note: *The INIT bit must not be set or the initialization block will be accessed instead.*

DATA SHEET REVISION SUMMARY

The following list represents the key differences between Revision A (May 1992) and Revision B (May 1994).

Distinctive Characteristics

Added a new bullet on *General Purpose Serial Interface.*

Related Products—Page 1-344:

This table is updated to reflect new members in the PCnet Family.

Table of Contents—Page 1-346:

This section has been moved after *Ordering Information.*

Block Diagram: Bus Master Mode—Page 1-352:

This section has been moved after *Table of Contents.*

The *MAUSEL* pin name has been changed to *MAUSEL/$\overline{EAR}$* to reflect its multiplexed functionality.

Connection Diagram: Bus Master Mode—Page 1-353:

The *MAUSEL* pin name has been changed to *MAUSEL/$\overline{EAR}$* to reflect its multiplexed functionality.

Pin Designations: Bus Master Mode—Page 1-354:

The *MAUSEL* pin name has been changed to *MAUSEL/$\overline{EAR}$* to reflect its multiplexed functionality.

Pin Description: Bus Master Mode—Page 1-358:

This section has been moved after *Pin Designations: Bus Master Mode.*

The descriptions for *LA17–23, $\overline{MASTER}$, $\overline{REF}$, RESET, IOAM0–1, MAUSEL/$\overline{EAR}$, $\overline{SLEEP}$,* and *XTAL1* have been rewritten for clarity.

Block Diagram: Shared Memory Mode—Page 1-361:

This section has been moved after *Pin Description: Bus Master Mode.*

The *MAUSEL* pin name has been changed to *MAUSEL/$\overline{EAR}$* to reflect its multiplexed functionality.

The *TCLK* pin name has been changed to *TCK* for correction.

Connection Diagram: Shared Memory Mode—Page 1-362:

The *MAUSEL* pin name has been changed to *MAUSEL/$\overline{EAR}$* to reflect its multiplexed functionality.

Pin Designations: Shared Memory Mode—Page 1-363:

The *MAUSEL* pin name has been changed to *MAUSEL/$\overline{EAR}$* to reflect its multiplexed functionality.

Pin Description: Shared Memory Mode—Page 1-367:

The descriptions for *IOCHRDY, RESET, $\overline{SBHE}$, IOAM0–1, MAUSEL/$\overline{EAR}$, XTAL1,* and *XTAL2* have been rewritten for clarity.

Pin Description: IEEE 1149.1 (JTAG) Test Access Port

The description for *TCK* has been rewritten for clarity.

Pin Description: Power Supplies

This entire section has been rewritten for clarity.

Functional Description

Page 1-372:

Bus Master Block Diagram
This diagram has been updated for clarity.

Page 1-373:

Shared Memory Block Diagram
This diagram has been updated for clarity.

Network Interface
Clarified the methods of selecting different network interfaces.

Detailed Functions

Page 1-374:

BMU—Initialization
Added description for an alternative initialization method.

BMU—Reinitialization
Clarified text description for alternative reinitialization method.

Page 1-376:

BMU—Polling
Added description for descriptor polling intervals.

Page 1-377:

BMU—TDTE
Added descriptor for zero-length buffers.

Page 1-379:

BMU—Message Data Encapsulation
Added description for framing errors.

Page 1-382:

External Clock Drive Characteristics
Changed the *XTAL1 HIGH/LOW Time* value in the table to *20 ns*.

Page 1-383:

Receiver Block Diagram and Text
Changed *DI±* to *DI±*/RXD±, *IRXD* to *IRXDAT*, *IRCLK* to *ISRDCLK*, and *IRENA* to *IRXCRS*.

Page 1-384:

Differential Input Terminations and Collision Detection
Moved these sections under *Attachment Unit Interface*

Page 1-386:

T-MAU—Power Down
Changed names of power savings modes: *sleep mode* to *coma* mode; and *auto wake mode* to *snooze mode*.

Page 1-387:

EADI
Added more description on the condition for accepting frames.

Page 1-388:

General Purpose Serial Interface
Added this new section to describe how the GPSI mode can be accessed and used.

Page 1-389:

IEEE 1149.1 Test Access Port
Changed *ID CODE* to *IDCODE*, *TRI_ST* to *TRIBYP*, and *SET_1/0* to *SETBYP*.

Page 1-390:

Power Savings Modes
Changed the section title name from Power Down Mode; changed names of power savings modes: *sleep mode* to *coma* mode; and *auto wake mode* to *snooze mode*.

Page 1-399:

Receive Exception Conditions
As a result of text clarification the number of abnormal network conditions has been reduced from two to four.

PCnet-ISA Controller Registers—Page 1-401:

Register Access
Added this new section to clarify how registers are accessed.

Control and Status Registers
Changed the section name from *Ethernet Controller Registers* to *Control and Status Registers*.

CSR4
The name for Bit 9 is changed from *MPCO* to *MFCO*. The name for Bit 10 is changed from *MPCOM* to *MFCOM*.

CSR15
Clarified description for bits 8 and 7.

CSR36–37
The name of the register is changed from *Temporary Storage* to *Next Next Receive Descriptor Address*. The bit name changed from *TMP0* to *NNRDA*. Updated with new bit description.

CSR38–39
The name of the register is changed from *Temporary Storage* to *Next Next Transmit Descriptor Address*. The bit name changed from *TMP1* to *NNTDA*. Updated with new bit description.

CSR47
Added description for *Polling Interval*.

CSR48–49
Changed bit name from *TMP2* to *TMP0*.

CSR50–51
Changed bit name from *TMP3* to *TMP1*.

CSR52–53
Changed bit name from *TMP4* to *TMP2*.

CSR54–55
Changed bit name from *TMP5* to *TMP3*.

CSR56–57
Changed bit name from *TMP6* to *TMP4*.

CSR58–59
Changed bit name from *TMP7* to *TMP5*.

CSR72
Clarified description for *Receive Ring Counter*.

CSR74
Clarified description for *Transmit Ring Counter*.

CSR82
Changed the register name from *Bus Timer Register* to *Bus Activity Timer*. Also added new text description for clarity.

CSR112
Changed bit name from *Count* to *MFC*. Also added new text description for clarity.

CSR114
Changed bit name from *Count* to *RCVCC*. Also added new text description for clarity.

CSR124
Changed the register name from *Buffer Management Scratch Test Register* to *Buffer Management Unit Test*.

Also changed bits 9–5 to Reserved. Changed bit name from Bit 4 from *CORETST* to *GPSIEN*.

ISA Bus Configuration Registers—Page 1-416:

ISACSR0
The description for this register has been rewritten for clarity.

ISACSR1
The description for this register has been rewritten for clarity.

ISACSR2
The bit name for Bit 15 is changed from *MODECONFIG* to *MODE_STATUS* and the description is rewritten for clarity. The bit name for Bit 7 is changed from *IRQLEVEL* to *EISA_LVL*. The description for Bit 0 is rewritten for clarity.

ISACSR4
The description for this register has been added.

ISACSR5–7
The descriptions for these registers have been rewritten for clarity.

Initialization Block

Page 1-419:

Address Match Logic Diagram
The MUX output signal name is changed from $\overline{MATCH}$ to *MATCH*.

Receive Descriptors

Page 1-420:

RMD1
The descriptions for the various bits in this register have been rewritten for clarity.

Transmit Descriptors

Page 1-421:

TMD1
The descriptions for the various bits in this register have been rewritten for clarity.

Register Summary

Page 1-424:

A column indicating whether a register is user accessible has been added.

System Application

Page 1-427:

The text has been rewritten for clarity.

The Bus Master Block Diagram has been edited to provide more detail.

In the Boot PROM Example Diagram the part number for the boot PROM has been corrected. The correct part is 27C128 (16Kx8 EPROM).

10BASE-T Interface

Page 1-429:

The list of compatible filter modules has been moved to Appendix A.

DC Characteristics

Page 1-430:

V_{OL}
This parameter is now guaranteed to have a maximum value of 0.5 V.

I_{OZL}
This parameter is added and has a minimum value of −10 µA.

I_{OZH}
This parameter is added and has a maximum value of +10 µA.

V_{ILX} and V_{IHX}
The two new parameters have been added to the Crystal Input Current list.

I_{ILX}
This parameter is divided into Active mode and Sleep mode. When Active this parameter is now guaranteed to be valid from −120 µA to 0 µA; when in Sleep this parameter is guaranteed to be valid from −10 µA to +10 µA.

I_{IHX}
When Active this parameter is now guaranteed to be valid from 0 µA to 120 µA.

V_{AOD}
This parameter used to be named V_{OD}.

V_{AODOFF}
This parameter used to be named V_{ODOFF}.

I_{AODOFF}
This parameter used to be named I_{ODOFF}.

V_{ATH}
This parameter used to be named V_{IRD}.

V_{ASQ}
This parameter used to be named V_{IDC}.

Page 1-432:

I_{DDCOMA}
This parameter used to be named $I_{DDSLEEP}$. I_{DDCOMA} is guaranteed to have maximum supply current of 200 µA.

$I_{DDSNOOZE}$
This is a new parameter. It is guaranteed to have a maximum current of 10 mA.

Switching Characteristics

Page 1-433:

t_{IOW2}
The parameter description has been rewritten for clarity.

t_{IOW5}
The parameter description has been rewritten for clarity.

t_{IOW6}
The parameter description has been rewritten for clarity. Also, the minimum value is now 10 ns.

t_{IOW7}
The parameter description has been rewritten for clarity.

t_{IOW9}
The parameter description has been rewritten for clarity.

t_{IOR2}
The parameter description has been rewritten for clarity.

t_{IOR4}
The parameter description has been rewritten for clarity.

t_{IOR6}
The parameter description has been rewritten for clarity.

t_{IOR8}
The parameter description has been rewritten for clarity. Also, the minimum value is now –130 ns.

t_{IOM1}
The parameter description has been rewritten for clarity. Also, the maximum value is now 55 ns.

t_{IOM2}
The parameter description has been rewritten for clarity.

t_{MMA4}
The typographical error is now corrected; the value of 35 ns has been moved to the Max column from the Min column.

t_{MMA5}
The parameter description was rewritten for clarity.

Page 1-434:

t_{MMBR1}
This parameter was incorrectly named t_{MMR1}. The parameter description has been rewritten for clarity and the maximum value is now 65 ns.

t_{MMBR2}
This parameter used to be named t_{MMR2}. The parameter description has been rewritten for clarity.

t_{MMBR3}
This parameter used to be named t_{MMR3}. The values are now valid from 40 ns to 60 ns.

t_{MMBR4}
This parameter used to be named t_{MMR4}. The parameter description has been rewritten for clarity.

t_{MMW1}
The parameter description has been rewritten for clarity and the values are now valid from EXTIME+45 to EXTIME+65 ns.

t_{MMW2}
The minimum value is now MSWRA–10 ns.

t_{MMW3}
The values are now valid from EXTIME+97 to EXTIME+105 ns.

t_{MMW5}
The parameter description has been rewritten for clarity and the maximum value is now 60 ns.

t_{MMW6}
The values are now valid from EXTIME+45 to EXTIME+55 ns.

t_{MMW7}
The parameter description has been rewritten for clarity and the minimum value is now t_{MMW2}–175 ns and the maximum value is left open.

t_{MMW9}
The parameter description has been rewritten for clarity and the minimum value is now 130 ns.

t_{MMW10}
This is a new parameter.

t_{MMW11}
This is a new parameter.

t_{MMR1}
The description has been rewritten for clarity and the values are now valid from EXTIME+45 to EXTIME+60 ns.

t_{MMR2}
The minimum value is now MSRDA–10 ns.

t_{MMR3}
The values are now valid from EXTIME+97 to EXTIME+105 ns.

t_{MMR6}
The values are now valid from EXTIME+45 to EXTIME+55 ns.

t_{MMR7}
The description has been rewritten for clarity and the minimum value is now t_{MMR2}–175 ns.

t_{MMR9}
The description has been rewritten for clarity and the minimum value is now 130 ns.

Page 1-435:

t_{MA2}
The minimum value is now 140 ns.

t_{MB6}
The minimum value is now 140 ns.

Page 1-436:

t_{IOW3}
The minimum value is now 150 ns.

t_{IOW6}
The minimum value is now 10 ns.

t_{IOW9}
The description has been rewritten for clarity.

t_{IOR1}
The minimum value is now 15 ns.

t_{IOR8}
The description has been rewritten for clarity and the minimum value is now –130 ns.

t_{MW3}
The minimum value is now 150 ns.

t_{MW4}
The minimum value is now 55 ns.

t_{MW6}
The minimum value is now 10 ns.

Page 1-437:

t_{MR8}
The minimum value is now –130 ns.

t_{IOM1}
The minimum value is now 55 ns.

Page 1-439:

EADI Timing Parameters
These are new additions to the data sheet.

JTAG Timing Parameters
These are new additions to the data sheet.

Page 1-440:

GPSI Timing Parameters
These are new additions to the data sheet.

Page 1-441:

AUI Timing Parameters
There are new additions to the data sheet.

Page 1-442:

10BASE-T Interface Timing
This replaces the Network Interface timing section in Rev. A of the data sheet. t_{JREC}, t_{RETD}, t_{RCVON}, t_{RCVOFF}, t_{COLON}, and t_{COLOFF} have been removed from the table.

Switching Waveforms

Page 1-448:

The LA17–13 reference in the Write Cycles (Bus Master Mode) diagram has been changed to LA17–23.

Page 1-449:

The LA17–13 reference in the Read Cycles (Bus Master Mode) diagram has been changed to LA17–23.

Page 1-458:

Added GPSI Transmit and Receive timing diagrams.

Page 1-459:

Added EADI and JTAG timing diagrams.

Page 1-460, 1-461, 1-462:

Added AUI timing diagrams.

Page 1-463:

The Transmit Timing (10BASE-T Interface) diagram is replaced with a new one.

Appendices

Appendix A
This is a new addition. It lists the compatible filter modules.

Appendix B
This is a new addition. It gives recommendation on reducing noise injected into the Ethernet controller.

Appendix C
This is a new addition. It provides an alternative method on initializing the Ethernet controller.

PRELIMINARY

Am79C961

PCnet™-ISA+ Jumperless Single-Chip Ethernet Controller for ISA

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Single-chip Ethernet controller for the Industry Standard Architecture (ISA) and Extended Industry Standard Architecture (EISA) buses**
- **Supports IEEE 802.3/ANSI 8802-3 and Ethernet standards**
- **Direct interface to the ISA or EISA bus**
- **Software compatible with AMD's Am7990 LANCE register and descriptor architecture**
- **Low power, CMOS design with sleep mode allows reduced power consumption for critical battery powered applications**
- **Individual 136-byte transmit and 128-byte receive FIFOs provide packet buffering for increased system latency, and support the following features:**
  - — Automatic retransmission with no FIFO reload
  - — Automatic receive stripping and transmit padding (individually programmable)
  - — Automatic runt packet rejection
  - — Automatic deletion of received collision frames
- **Dynamic transmit FCS generation programmable on a frame-by-frame basis**
- **Single +5 V power supply**
- **Internal/external loopback capabilities**
- **Supports 8K, 16K, 32K, and 64K Boot PROMs or Flash for diskless node applications**
- **Supports Microsoft's Plug and Play System configuration for jumperless designs**
- **Supports staggered AT bus drive for reduced noise and ground bounce**
- **Supports 8 interrupts on chip**
- **Look Ahead Packet Processing (LAPP) allows protocol analysis to begin before end of receive frame**
- **Supports 4 DMA channels on chip**
- **Supports 16 I/O locations**
- **Supports 16 boot PROM locations**
- **Provides integrated Attachment Unit Interface (AUI) and 10BASE-T transceiver with 2 modes of port selection:**
  - — Automatic selection of AUI or 10BASE-T
  - — Software selection of AUI or 10BASE-T
- **Automatic Twisted Pair receive polarity detection and automatic correction of the receive polarity**
- **Supports bus-master and shared-memory architectures to fit in any PC application**
- **Supports edge and level-sensitive interrupts**
- **DMA Buffer Management Unit for reduced CPU intervention which allows higher throughput by by-passing the platform DMA**
- **JTAG Boundary Scan (IEEE 1149.1) test access port interface for board level production test**
- **Integrated Manchester Encoder/Decoder**
- **Supports the following types of network interfaces:**
  - — AUI to external 10BASE2, 10BASE5, 10BASE-T or 10BASE-F MAU
  - — Internal 10BASE-T transceiver with Smart Squelch to Twisted Pair medium
- **Supports LANCE General Purpose Serial Interface (GPSI)**
- **132-pin PQFP package**

GENERAL DESCRIPTION

The PCnet-ISA+ controller, a single-chip Ethernet controller, is a highly integrated system solution for the PC-AT Industry Standard Architecture (ISA) architecture. It is designed to provide flexibility and compatibility with any existing PC application. This highly integrated 132-pin VLSI device is specifically designed to reduce parts count and cost, and addresses applications where higher system throughput is desired. The PCnet-ISA+ controller is fabricated with AMD's advanced low-power CMOS process to provide low standby current for power sensitive applications.

The PCnet-ISA+ controller is a DMA-based device with a dual architecture that can be configured in two different operating modes to suit a particular PC application. In the Bus Master Mode all transfers are performed using

Publication# 18183 Rev. B Amendment/0
Issue Date: April 1994

the integrated DMA controller. This configuration enhances system performance by allowing the PCnet-ISA+ controller to bypass the platform DMA controller and directly address the full 24-bit memory space. The implementation of Bus Master Mode allows minimum parts count for the majority of PC applications. The PCnet-ISA+ controller can be configured to perform Shared Memory operations for compatibility with low-end machines, such as PC/XTs that do not support Bus Master and high-end machines that require local packet buffering for increased system latency.

The PCnet-ISA+ controller is designed to directly interface with the ISA or EISA system bus. It contains an ISA Plug and Play bus interface unit, DMA Buffer Management Unit, 802.3 Media Access Control function, individual 136-byte transmit and 128-byte receive FIFOs, IEEE 802.3 defined Attachment Unit Interface (AUI), and a Twisted Pair Transceiver Media Attachment Unit. The PCnet-ISA+ controller is also register compatible with the LANCE (Am7990) Ethernet controller and PCnet-ISA. The DMA Buffer Management Unit supports the LANCE descriptor software model. External remote boot and Ethernet physical address PROMs and Electrically Erasable Proms are also supported.

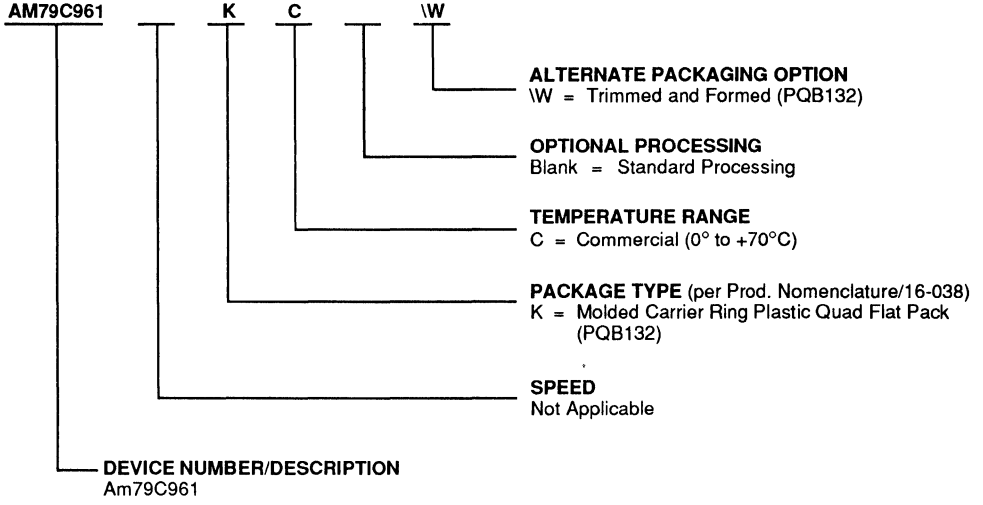
This advanced Ethernet controller has the built-in capability of automatically selecting either the AUI port or the Twisted Pair transceiver. Only one interface is active at any one time. The individual 136-byte transmit and 128-byte receive FIFOs optimize system overhead, providing sufficient latency during packet transmission and reception, and minimizing intervention during normal network error recovery. The integrated Manchester encoder/decoder eliminates the need for an external Serial Interface Adapter (SIA) in the node system. If support for an external encoding/decoding scheme is desired, the embedded General Purpose Serial Interface (GPSI) allows direct access to/from the MAC. In addition, the device provides programmable on-chip LED drivers for transmit, receive, collision, receive polarity, link integrity and activity, or jabber status. The PCnet-ISA+ controller also provides an External Address Detection Interface™ (EADI™) to allow external hardware address filtering in internetworking applications.

RELATED PRODUCTS

Part No.	Description
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted Pair Ethernet Transceiver Plus (TPEX+)
Am7996	IEEE 802.3/Ethernet/Cheapernet Transceiver
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller (for 386, 486, VL local buses)
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



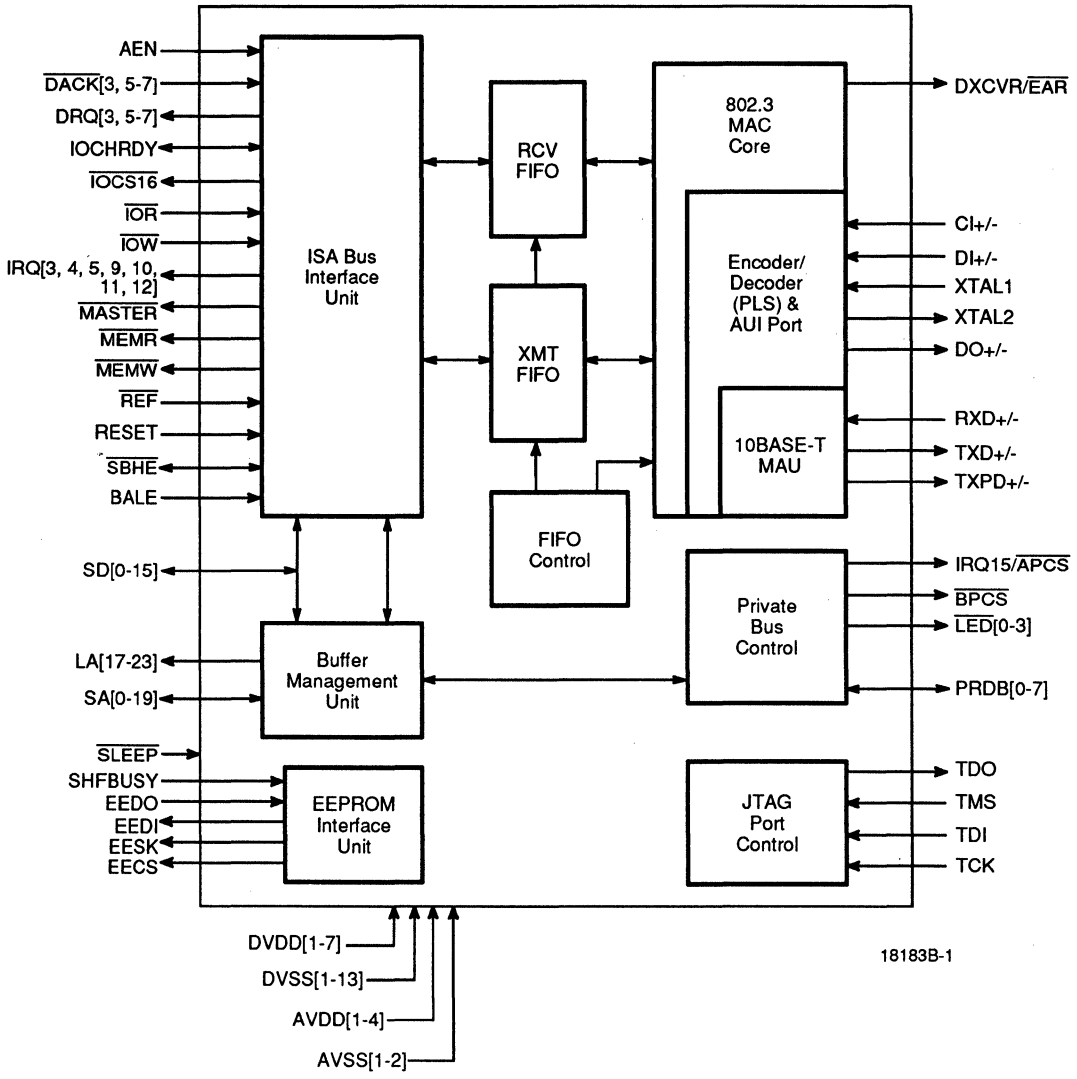
Valid Combinations	
AM79C961	KC, KC\W

Valid Combinations

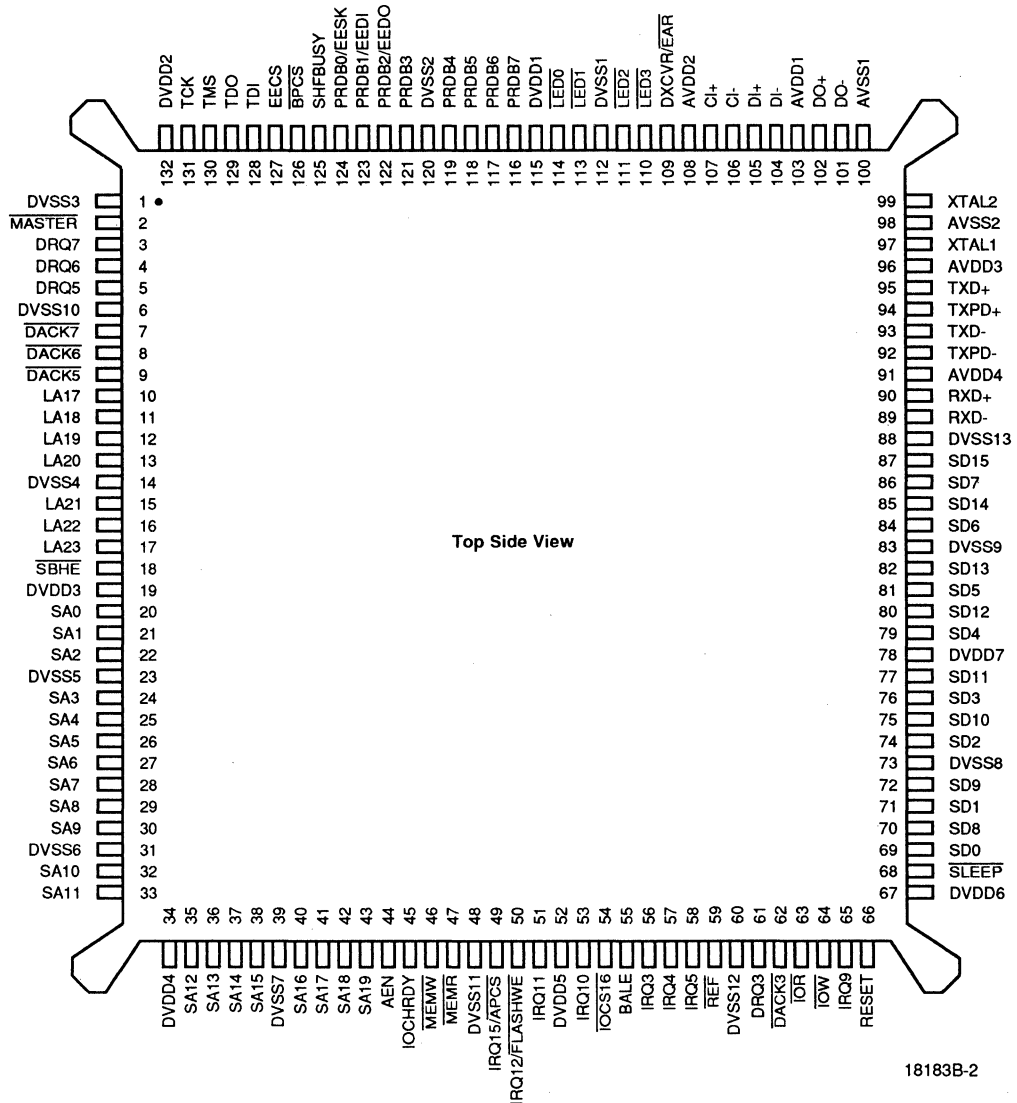
The Valid Combinations table lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

TABLE OF CONTENTS

BLOCK DIAGRAM: BUS MASTER MODE



CONNECTION DIAGRAM: BUS MASTER



PIN DESIGNATIONS: BUS MASTER
Listed by Pin Number

Pin #	Name	Pin #	Name	Pin #	Name
1	DVSS3	45	IOCHRDY	89	RXD-
2	$\overline{\text{MASTER}}$	46	$\overline{\text{MEMW}}$	90	RXD+
3	DRQ7	47	$\overline{\text{MEMR}}$	91	AVDD4
4	DRQ6	48	DVSS11	92	TXPD-
5	DRQ5	49	IRQ15/$\overline{\text{APCS}}$	93	TXD-
6	DVSS10	50	IRQ12/$\overline{\text{FlashWE}}$	94	TXPD+
7	$\overline{\text{DACK7}}$	51	IRQ11	95	TXD+
8	$\overline{\text{DACK6}}$	52	DVDD5	96	AVDD3
9	$\overline{\text{DACK5}}$	53	IRQ10	97	XTAL1
10	LA17	54	$\overline{\text{IOCS16}}$	98	AVSS2
11	LA18	55	BALE	99	XTAL2
12	LA19	56	IRQ3	100	AVSS1
13	LA20	57	IRQ4	101	DO-
14	DVSS4	58	IRQ5	102	DO+
15	LA21	59	$\overline{\text{REF}}$	103	AVDD1
16	LA22	60	DVSS12	104	DI-
17	LA23	61	DRQ3	105	DI+
18	$\overline{\text{SBHE}}$	62	$\overline{\text{DACK3}}$	106	CI-
19	DVDD3	63	$\overline{\text{IOR}}$	107	CI+
20	SA0	64	$\overline{\text{IOW}}$	108	AVDD2
21	SA1	65	IRQ9	109	DXCVR/$\overline{\text{EAR}}$
22	SA2	66	RESET	110	$\overline{\text{LED3}}$
23	DVSS5	67	DVDD6	111	$\overline{\text{LED2}}$
24	SA3	68	$\overline{\text{SLEEP}}$	112	DVSS1
25	SA4	69	SD0	113	$\overline{\text{LED1}}$
26	SA5	70	SD8	114	$\overline{\text{LED0}}$
27	SA6	71	SD1	115	DVDD1
28	SA7	72	SD9	116	PRDB7
29	SA8	73	DVSS8	117	PRDB6
30	SA9	74	SD2	118	PRDB5
31	DVSS6	75	SD10	119	PRDB4
32	SA10	76	SD3	120	DVSS2
33	SA11	77	SD11	121	PRDB3
34	DVDD4	78	DVDD7	122	PRDB2/EEDO
35	SA12	79	SD4	123	PRDB1/EEDI
36	SA13	80	SD12	124	PRDB0/EESK
37	SA14	81	SD5	125	SHFBUSY
38	SA15	82	SD13	126	$\overline{\text{BPCS}}$
39	DVSS7	83	DVSS9	127	EECS
40	SA16	84	SD6	128	TDI
41	SA17	85	SD14	129	TDO
42	SA18	86	SD7	130	TMS
43	SA19	87	SD15	131	TCK
44	AEN	88	DVSS13	132	DVDD2

PIN DESIGNATIONS: BUS MASTER
Listed by Pin Name

Name	Pin #	Name	Pin #	Name	Pin #
AEN	44	EECS	127	SA13	36
AVDD1	103	IOCHRDY	45	SA14	37
AVDD2	108	$\overline{\text{IOCS16}}$	54	SA15	38
AVDD3	96	$\overline{\text{IOR}}$	63	SA16	40
AVDD4	91	$\overline{\text{IOW}}$	64	SA17	41
AVSS1	100	IRQ10	53	SA18	42
AVSS2	98	IRQ11	51	SA19	43
BALE	55	IRQ12/$\overline{\text{FlashWE}}$	50	SA2	22
BPCS	126	IRQ15/$\overline{\text{APCS}}$	49	SA3	24
CI-	106	IRQ3	56	SA4	25
CI+	107	IRQ4	57	SA5	26
$\overline{\text{DACK3}}$	62	IRQ5	58	SA6	27
$\overline{\text{DACK5}}$	9	IRQ9	65	SA7	28
$\overline{\text{DACK6}}$	8	LA17	10	SA8	29
$\overline{\text{DACK7}}$	7	LA18	11	SA9	30
DI-	104	LA19	12	$\overline{\text{SBHE}}$	18
DI+	105	LA20	13	SD0	69
DO-	101	LA21	15	SD1	71
DO+	102	LA22	16	SD10	75
DRQ3	61	LA23	17	SD11	77
DRQ5	5	LED0	114	SD12	80
DRQ6	4	LED1	113	SD13	82
DRQ7	3	LED2	111	SD14	85
DVDD1	115	LED3	110	SD15	87
DVDD2	132	$\overline{\text{MASTER}}$	2	SD2	74
DVDD3	19	$\overline{\text{MEMR}}$	47	SD3	76
DVDD4	34	$\overline{\text{MEMW}}$	46	SD4	79
DVDD5	52	PRDB0/EESK	124	SD5	81
DVDD6	67	PRDB1/EEDI	123	SD6	84
DVDD7	78	PRDB2/EEDO	122	SD7	86
DVSS1	112	PRDB3	121	SD8	70
DVSS10	6	PRDB4	119	SD9	72
DVSS11	48	PRDB5	118	SHFBUSY	125
DVSS12	60	PRDB6	117	$\overline{\text{SLEEP}}$	68
DVSS13	88	PRDB7	116	TCK	131
DVSS2	120	$\overline{\text{REF}}$	59	TDI	128
DVSS3	1	RESET	66	TDO	129
DVSS4	14	RXD-	89	TMS	130
DVSS5	23	RXD+	90	TXD-	93
DVSS6	31	SA0	20	TXD+	95
DVSS7	39	SA1	21	TXPD-	92
DVSS8	73	SA10	32	TXPD+	94
DVSS9	83	SA11	33	XTAL1	97
DXCVR/$\overline{\text{EAR}}$	109	SA12	35	XTAL2	99

PIN DESIGNATIONS: BUS MASTER
Listed by Group

Pin Name	Pin Function	I/O	Driver
ISA Bus Interface			
AEN	Address Enable	I	
BALE	Bus Address Latch Enable	I	
$\overline{\text{DACK}}$[3, 5-7]	DMA Acknowledge	I	
DRQ[3, 5-7]	DMA Request	O	TS3
IOCHRDY	I/O Channel Ready	I/O	OD3
$\overline{\text{IOCS16}}$	I/O Chip Select 16	O	OD3
$\overline{\text{IOR}}$	I/O Read Select	I	
$\overline{\text{IOW}}$	I/O Write Select	I	
IRQ[3, 4, 5, 9, 10, 11, 12, 15]	Interrupt Request	O	TS3/OD3
LA[17-23]	Unlatched Address Bus	I/O	TS3
$\overline{\text{MASTER}}$	Master Transfer in Progress	O	OD3
$\overline{\text{MEMR}}$	Memory Read Select	O	TS3
$\overline{\text{MEMW}}$	Memory Write Select	O	TS3
$\overline{\text{REF}}$	Memory Refresh Active	I	
RESET	System Reset	I	
SA[0-19]	System Address Bus	I/O	TS3
$\overline{\text{SBHE}}$	System Byte High Enable	I/O	TS3
SD[0-15]	System Data Bus	I/O	TS3
Board Interfaces			
IRQ15/$\overline{\text{APCS}}$	IRQ15 or Address PROM Chip Select	O	TS1
$\overline{\text{BPCS}}$	Boot PROM Chip Select	O	TS1
DXCVR/$\overline{\text{EAR}}$	Disable Transceiver	I/O	TS1
$\overline{\text{LED0}}$	$\overline{\text{LED0/LNKST}}$	O	TS2
$\overline{\text{LED1}}$	$\overline{\text{LED1/SFBD/RCVACT}}$	O	TS2
$\overline{\text{LED2}}$	$\overline{\text{LED2/SRD/RXDATPOL}}$	O	TS2
$\overline{\text{LED3}}$	$\overline{\text{LED3/SRDCLK/XMTACT}}$	O	TS2
PRDB[3-7]	PROM Data Bus	I/O	TS1
$\overline{\text{SLEEP}}$	Sleep Mode	I	
XTAL1	Crystal Input	I	
XTAL2	Crystal Output	O	
SHFBUSY	Read access from EEPROM in process	O	
PRDB(0)/EESK	Serial Shift Clock	I/O	
PRDB(1)/EEDI	Serial Shift Data In	I/O	
PRDB(2)/EEDO	Serial Shift Data Out	I/O	
EECS	EEPROM Chip Select	O	

PIN DESIGNATIONS: BUS MASTER (continued)
Listed by Group

Pin Name	Pin Function	I/O	Driver
Attachment Unit Interface (AUI)			
CI±	Collision Inputs	I	
DI±	Receive Data	I	
DO±	Transmit Data	O	
Twisted Pair Transceiver Interface (10BASE-T)			
RXD±	10BASE-T Receive Data	I	
TXD±	10BASE-T Transmit Data	O	
TXPD±	10BASE-T Predistortion Control	O	
IEEE 1149.1 Test Access Port Interface (JTAG)			
TCK	Test Clock	I	
TDI	Test Data Input	I	
TDO	Test Data Output	O	TS2
TMS	Test Mode Select	I	
Power Supplies			
AVDD	Analog Power [1-4]		
AVSS	Analog Ground [1-2]		
DVDD	Digital Power [1-7]		
DVSS	Digital Ground [1-13]		

Output Driver Types

Name	Type	I_{OL} (mA)	I_{OH} (mA)	pF
TS1	Tri-State	4	–1	50
TS2	Tri-State	12	–4	50
TS3	Tri-State	24	–3	120
OD3	Open Drain	24	–3	120

PIN DESCRIPTION: BUS MASTER MODE

These pins are part of the bus master mode. In order to understand the pin descriptions, definition of some terms from a draft of IEEE P996 are included.

IEEE P996 Terminology

Alternate Master: Any device that can take control of the bus through assertion of the MASTER signal. It has the ability to generate addresses and bus control signals in order to perform bus operations. All Alternate Masters must be 16 bit devices and drive SBHE.

Bus Ownership: The Current Master possesses bus ownership and can assert any bus control, address and data lines.

Current Master: The Permanent Master, Temporary Master or Alternate Master which currently has ownership of the bus.

Permanent Master: Each P996 bus will have a device known as the Permanent Master that provides certain signals and bus control functions as described in Section 3.5 (of the IEEE P996 spec), "Permanent Master". The Permanent Master function can reside on a Bus Adapter or on the backplane itself.

Temporary Master: A device that is capable of generating a DMA request to obtain control of the bus and directly asserting only the memory and I/O strobes during bus transfer. Addresses are generated by the DMA device on the Permanent Master.

ISA Interface

AEN
Address Enable *Input*

This signal must be driven LOW when the bus performs an I/O access to the device.

BALE
Used to latch the LA20–23 address lines.

DACK 3, 5-7
DMA Acknowledge *Input*

Asserted LOW when the Permanent Master acknowledges a DMA request. When DACK is asserted the PCnet-ISA+ controller becomes the Current Master by asserting the MASTER signal.

DRQ 3, 5-7
DMA Request *Output*

When the PCnet-ISA+ controller needs to perform a DMA transfer, it asserts DRQ. The Permanent Master acknowledges DRQ with assertion of DACK. When the PCnet-ISA+ controller does not need the bus it deasserts DRQ.

Because of the operation of the Plug and Play registers, the DMA Channels on the PCnet-ISA+ must be attached to specific DRQ and DACK signals on the PC/AT bus.

IOCHRDY
I/O Channel Ready *Input/Output*

When the PCnet-ISA+ controller is being accessed, IOCHRDY HIGH indicates that valid data exists on the data bus for reads and that data has been latched for writes. When the PCnet-ISA+ controller is the Current Master on the ISA bus, it extends the bus cycle as long as IOCHRDY is LOW.

IOCS16
I/O Chip Select 16 *Output*

When an I/O read or write operation is performed, the PCnet-ISA+ controller will drive the IOCS16 pin LOW to indicate that the chip supports a 16-bit operation at this address. (If the motherboard does not receive this signal, then the motherboard will convert a 16-bit access to two 8-bit accesses.)

The PCnet-ISA+ controller follows the IEEE P996 specification that recommends this function be implemented as a pure decode of SA0-9 and AEN, with no dependency on IOR, or IOW; however, some PC/AT clone systems are not compatible with this approach. For this reason, the PCnet-ISA+ controller is recommended to be configured to run 8-bit I/O on all machines. Since data is moved by memory cycles there is virtually no performance loss incurred by running 8-bit I/O and compatibility problems are virtually eliminated. The PCnet-ISA+ controller can be configured to run 8-bit-only I/O by clearing Bit 0 in Plug and Play register F0.

IOR
I/O Read *Input*

IOR is driven LOW by the host to indicate that an Input/Output Read operation is taking place. IOR is only valid if the AEN signal is LOW and the external address matches the PCnet-ISA+ controller's predefined I/O address location. If valid, IOR indicates that a slave read operation is to be performed.

IOW
I/O Write *Input*

IOW is driven LOW by the host to indicate that an Input/Output Write operation is taking place. IOW is only valid if AEN signal is LOW and the external address matches the PCnet-ISA+ controller's predefined I/O address location. If valid, IOW indicates that a slave write operation is to be performed.

IRQ 3, 4, 5, 9, 10, 11, 12, 15
Interrupt Request *Output*

An attention signal which indicates that one or more of the following status flags is set: BABL, MISS, MERR, RINT, IDON, RCVCCO, JAB, MPCO, or TXDATSTRT. All status flags have a mask bit which allows for suppression of IRQ assertion. These flags have the following meaning:

BABL	Babble
RCVCCO	Receive Collision Count Overflow
JAB	Jabber
MISS	Missed Frame
MERR	Memory Error
MPCO	Missed Packet Count Overflow
RINT	Receive Interrupt
IDON	Initialization Done
TXDATSTRT	Transmit Start

Because of the operation of the Plug and Play registers, the interrupts on the PCnet-ISA+ must be attached to specific IRQ signals on the PC/AT bus.

LA17-23
Unlatched Address Bus *Input/Output*

The unlatched address bus is driven by the PCnet-ISA+ controller during bus master cycle.

The functions of these unlatched address pins will change when GPSI mode is invoked. The following table shows the pin configuration in GPSI mode. Please refer to the section on General Purpose Serial Interface for detailed information on accessing this mode.

Pin Number	Pin Function in Bus Master Mode	Pin Function in GPSI Mode
10	LA17	RXDAT
11	LA18	SRDCLK
12	LA19	RXCRS
13	LA20	CLSN
15	LA21	STDCLK
16	LA22	TXEN
17	LA23	TXDAT

$\overline{\text{MASTER}}$
Master Mode *Input/Output*

This signal indicates that the PCnet-ISA+ controller has become the Current Master of the ISA bus. After the PCnet-ISA+ controller has received a DMA Acknowledge ($\overline{\text{DACK}}$) in response to a DMA Request (DRQ), the Ethernet controller asserts the $\overline{\text{MASTER}}$ signal to indicate to the Permanent Master that the PCnet-ISA+ controller is becoming the Current Master.

$\overline{\text{MEMR}}$
Memory Read *Input/Output*

$\overline{\text{MEMR}}$ goes LOW to perform a memory read operation.

$\overline{\text{MEMW}}$
Memory Write *Input/Output*

$\overline{\text{MEMW}}$ goes LOW to perform a memory write operation.

$\overline{\text{REF}}$
Memory Refresh *Input*

When $\overline{\text{REF}}$ is asserted, a memory refresh is active. The PCnet-ISA+ controller uses this signal to mask inadvertent DMA Acknowledge assertion during memory refresh periods. If $\overline{\text{DACK}}$ is asserted when $\overline{\text{REF}}$ is active, $\overline{\text{DACK}}$ assertion is ignored. $\overline{\text{REF}}$ is monitored to eliminate a bus arbitration problem observed on some ISA platforms.

RESET
Reset *Input*

When RESET is asserted HIGH the PCnet-ISA+ controller performs an internal system reset. RESET must be held for a minimum of 10 XTAL1 periods before being deasserted. While in a reset state, the PCnet-ISA+ controller will tristate or deassert all outputs to predefined reset levels. The PCnet-ISA+ controller resets itself upon power-up.

SA0-19
System Address Bus *Input/Output*

This bus contains address information, which is stable during a bus operation, regardless of the source. SA17-19 contain the same values as the unlatched address LA17-19. When the PCnet-ISA+ controller is the Current Master, SA0-19 will be driven actively. When the PCnet-ISA+ controller is not the Current Master, the SA0-19 lines are continuously monitored to determine if an address match exists for I/O slave transfers or Boot PROM accesses.

$\overline{\text{SBHE}}$
System Byte High Enable *Input/Output*

This signal indicates the high byte of the system data bus is to be used. $\overline{\text{SBHE}}$ is driven by the PCnet-ISA+ controller when performing bus mastering operations.

SD0-15
System Data Bus *Input/Output*

These pins are used to transfer data to and from the PCnet-ISA+ controller to system resources via the ISA data bus. SD0-15 is driven by the PCnet-ISA+ controller when performing bus master writes and slave read operations. Likewise, the data on SD0-15 is latched by the PCnet-ISA+ controller when performing bus master reads and slave write operations.

Board Interface

IRQ12/$\overline{\text{FlashWE}}$
Flash Write Enable *Output*

Optional interface to the Flash memory boot PROM Write Enable.

IRQ15/APCS
Address PROM Chip Select *Output*

When programmed as APCS in Plug and Play Register F0, this signal is asserted when the external Address PROM is read. When an I/O read operation is performed on the first 16 bytes in the PCnet-ISA+ controller's I/O space, APCS is asserted. The outputs of the external Address PROM drive the PROM Data Bus. The PCnet-ISA+ controller buffers the contents of the PROM data bus and drives them on the lower eight bits of the System Data Bus.

When programmed to IRQ15 (default), this pin has the same function as IRQ 3, 4, 5, 9, 10, 11, or 12.

BPCS
Boot PROM Chip Select *Output*

This signal is asserted when the Boot PROM is read. If SA0-19 lines match a predefined address block and MEMR is active and REF inactive, the BPCS signal will be asserted. The outputs of the external Boot PROM drive the PROM Data Bus. The PCnet-ISA+ controller buffers the contents of the PROM data bus and drives them on the lower eight bits of the System Data Bus.

DXCVR/EAR
Disable Transceiver/
External Address Reject *Input/Output*

This pin disables the transceiver. The DXCVR output is configured in the initialization sequence. A HIGH level indicates the Twisted Pair port is active and the AUI port is inactive, or SLEEP mode has been entered. A LOW level indicates the AUI port is active and the Twisted Pair port is inactive.

If EADI mode is selected, this pin becomes the EAR input.

The incoming frame will be checked against the internally active address detection mechanisms and the result of this check will be OR'd with the value on the EAR pin. The EAR pin is defined as REJECT. (See the EADI section for details regarding the function and timing of this signal.)

LED0-3
LED Drivers *Output*

These pins sink 12 mA each for driving LEDs. Their meaning is software configurable (see section *The ISA Bus Configuration Registers*) and they are active LOW.

When EADI mode is selected, the pins named LED1, LED2, and LED3 change in function while LED0 continues to indicate 10BASE-T Link Status.

LED	EADI Function
1	SF/BD
2	SRD
3	SRDCLK

PRDB3-7
Private Data Bus *Input/Output*

This is the data bus for the Boot PROM and the Address PROM.

PRDB2/EEDO
Private data bus bit 2/Data Out *Input/Output*

A multifunction pin which serves as PRDB2 of the private data bus and, when ISACSR3 bit 4 is set, changes to become DATA OUT from the EEPROM.

PRDB1/EEDI
Private data bus bit 1/Data In *Input/Output*

A multifunction pin which serves as PRDB1 of the private data bus and, when ISACSR3 bit 4 is set, changes to become DATA In to the EEPROM.

PRDB0/EESK
Private data bus bit 0/
Serial Clock *Input/Output*

A multifunction pin which serves as PRDB0 of the private data bus and, when ISACSR3 bit 4 is set, changes to become Serial Clock to the EEPROM.

SHFBUSY *Input/Output*

An output from PCnet-ISA+ which indicates that a read from the external EEPROM is in progress. It is active only when the hardware reconfigure is running (when data is being shifted out of the EEPROM due to a hardware RESET or the EELOAD command being issued). This pin should have a pull-up resistor (10 KΩ) to VCC.

EECS
EEPROM CHIPSELECT *Output*

This signal is asserted when read or write accesses are being performed to the EEPROM. It is controlled by ISACSR3. It is driven at Reset during EEPROM Read.

SLEEP
Sleep *Input*

When SLEEP pin is asserted (active LOW), the PCnet-ISA+ controller performs an internal system reset and proceeds into a power savings mode. All outputs will be placed in their normal reset condition. All PCnet-ISA+ controller inputs will be ignored except for the SLEEP pin itself. Deassertion of SLEEP results in wake-up. The system must delay the starting of the network controller by 0.5 seconds to allow internal analog circuits to stabilize.

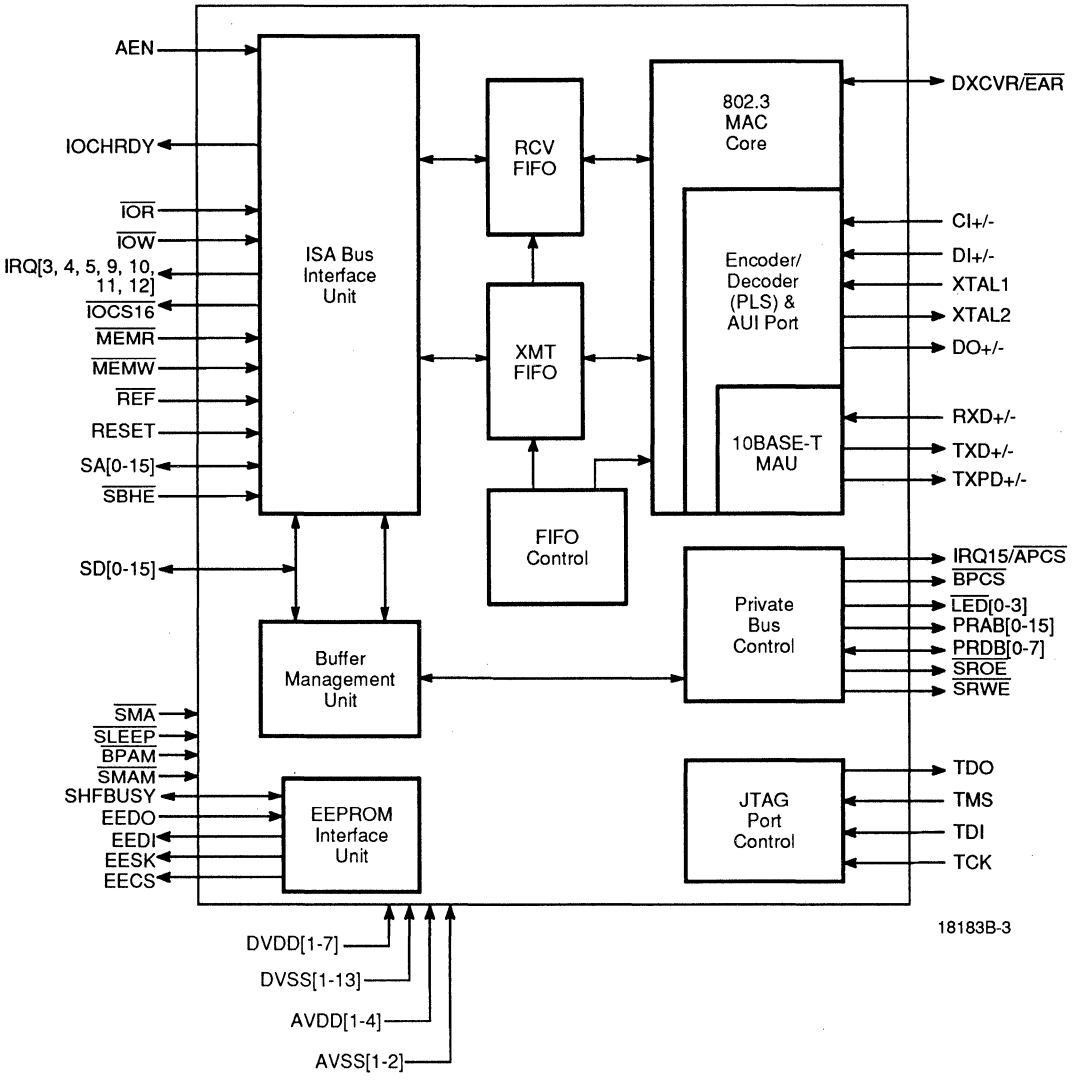
XTAL1
Crystal Connection *Input*

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. Alternatively, an external 20 MHz CMOS-compatible clock signal can be used to drive this pin. Refer to the section on External Crystal Characteristics for more details.

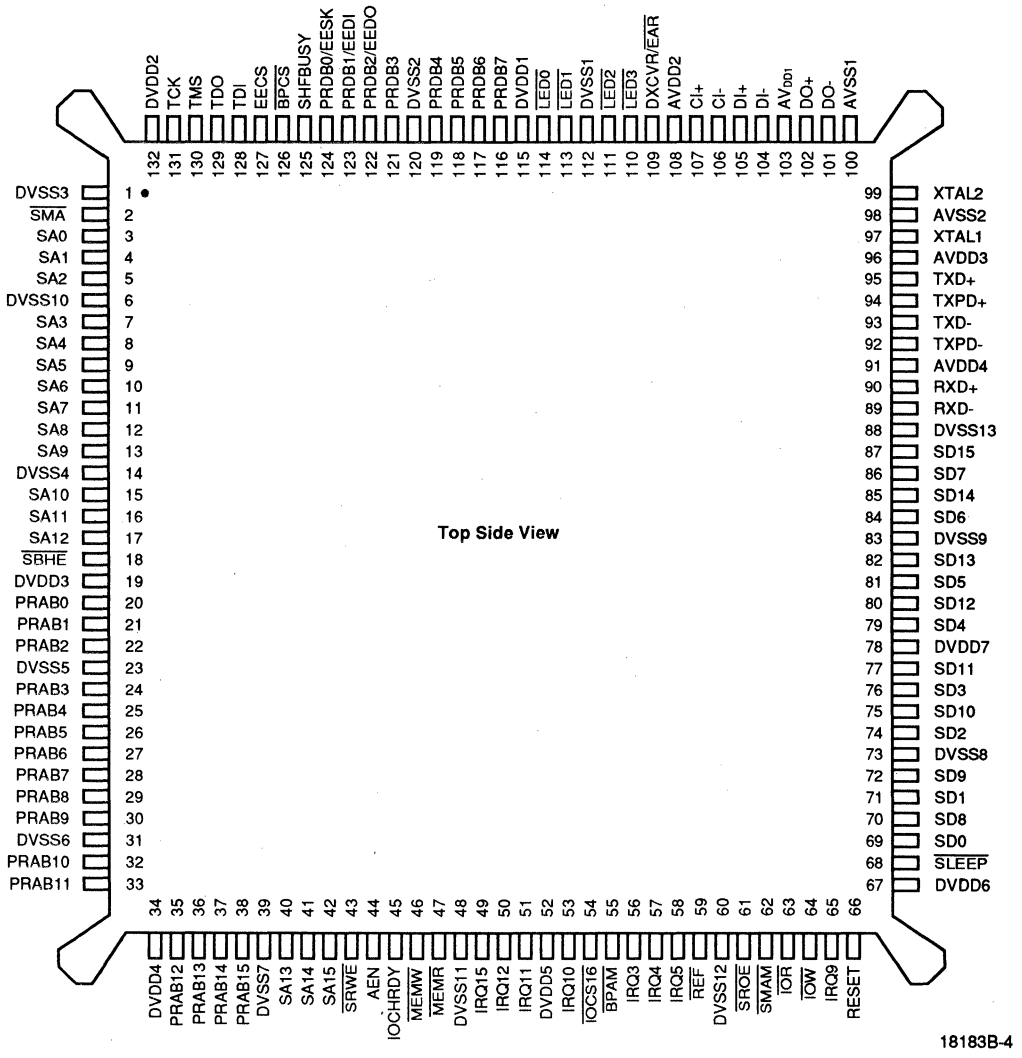
XTAL2
Crystal Connection *Output*

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. If an external clock is used, this pin should be left unconnected.

BLOCK DIAGRAM: SHARED MEMORY MODE



CONNECTION DIAGRAM: SHARED MEMORY



18183B-4

PIN DESIGNATIONS: SHARED MEMORY
Listed by Pin Number

Pin #	Name	Pin #	Name	Pin #	Name
1	DVSS3	45	IOCHRDY	89	RXD-
2	$\overline{\text{SMA}}$	46	$\overline{\text{MEMW}}$	90	RXD+
3	SA0	47	$\overline{\text{MEMR}}$	91	AVDD4
4	SA1	48	DVSS11	92	TXPD-
5	SA2	49	IRQ15	93	TXD-
6	DVSS10	50	IRQ12	94	TXPD+
7	SA3	51	IRQ11	95	TXD+
8	SA4	52	DVDD5	96	AVDD3
9	SA5	53	IRQ10	97	XTAL1
10	SA6	54	$\overline{\text{IOCS16}}$	98	AVSS2
11	SA7	55	$\overline{\text{BPAM}}$	99	XTAL2
12	SA8	56	IRQ3	100	AVSS1
13	SA9	57	IRQ4	101	DO-
14	DVSS4	58	IRQ5	102	DO+
15	SA10	59	$\overline{\text{REF}}$	103	AVDD1
16	SA11	60	DVSS12	104	DI-
17	SA12	61	$\overline{\text{SROE}}$	105	DI+
18	$\overline{\text{SBHE}}$	62	$\overline{\text{SMAM}}$	106	CI-
19	DVDD3	63	$\overline{\text{IOR}}$	107	CI+
20	PRAB0	64	$\overline{\text{IOW}}$	108	AVDD2
21	PRAB1	65	IRQ9	109	DXCVR/$\overline{\text{EAR}}$
22	PRAB2	66	RESET	110	$\overline{\text{LED3}}$
23	DVSS5	67	DVDD6	111	$\overline{\text{LED2}}$
24	PRAB3	68	$\overline{\text{SLEEP}}$	112	DVSS1
25	PRAB4	69	SD0	113	$\overline{\text{LED1}}$
26	PRAB5	70	SD8	114	$\overline{\text{LED0}}$
27	PRAB6	71	SD1	115	DVDD1
28	PRAB7	72	SD9	116	PRDB7
29	PRAB8	73	DVSS8	117	PRDB6
30	PRAB9	74	SD2	118	PRDB5
31	DVSS6	75	SD10	119	PRDB4
32	PRAB10	76	SD3	120	DVSS2
33	PRAB11	77	SD11	121	PRDB3
34	DVDD4	78	DVDD7	122	PRDB2/EEDO
35	PRAB12	79	SD4	123	PRDB1/EEDI
36	PRAB13	80	SD12	124	PRDB0/EESK
37	PRAB14	81	SD5	125	SHFBUSY
38	PRAB15	82	SD13	126	$\overline{\text{BPCS}}$
39	DVSS7	83	DVSS9	127	EECS
40	SA13	84	SD6	128	TDI
41	SA14	85	SD14	129	TDO
42	SA15	86	SD7	130	TMS
43	$\overline{\text{SRWE}}$	87	SD15	131	TCK
44	AEN	88	DVSS13	132	DVDD2

PIN DESIGNATIONS: SHARED MEMORY
Listed by Pin Name

Name	Pin #	Name	Pin #	Name	Pin #
AEN	44	IRQ15	49	SA13	40
AVDD1	103	IRQ3	56	SA14	41
AVDD2	108	IRQ4	57	SA15	42
AVDD3	96	IRQ5	58	SA2	5
AVDD4	91	IRQ9	65	SA3	7
AVSS1	100	$\overline{\text{LED0}}$	114	SA4	8
AVSS2	98	$\overline{\text{LED1}}$	113	SA5	9
$\overline{\text{BPAM}}$	55	$\overline{\text{LED2}}$	111	SA6	10
$\overline{\text{BPCS}}$	126	$\overline{\text{LED3}}$	110	SA7	11
CI-	106	$\overline{\text{MEMR}}$	47	SA8	12
CI+	107	$\overline{\text{MEMW}}$	46	SA9	13
DI-	104	PRAB0	20	$\overline{\text{SBHE}}$	18
DI+	105	PRAB1	21	SD0	69
DO-	101	PRAB10	32	SD1	71
DO+	102	PRAB11	33	SD10	75
DVDD1	115	PRAB12	35	SD11	77
DVDD2	132	PRAB13	36	SD12	80
DVDD3	19	PRAB14	37	SD13	82
DVDD4	34	PRAB15	38	SD14	85
DVDD5	52	PRAB2	22	SD15	87
DVDD6	67	PRAB3	24	SD2	74
DVDD7	78	PRAB4	25	SD3	76
DVSS1	112	PRAB5	26	SD4	79
DVSS10	6	PRAB6	27	SD5	81
DVSS11	48	PRAB7	28	SD6	84
DVSS12	60	PRAB8	29	SD7	86
DVSS13	88	PRAB9	30	SD8	70
DVSS2	120	PRDB0/DO	124	SD9	72
DVSS3	1	PRDB1/DI	123	SHFBUSY	125
DVSS4	14	PRDB2/SCLK	122	$\overline{\text{SLEEP}}$	68
DVSS5	23	PRDB3	121	$\overline{\text{SMA}}$	2
DVSS6	31	PRDB4	119	$\overline{\text{SMAM}}$	62
DVSS7	39	PRDB5	118	$\overline{\text{SROE}}$	61
DVSS8	73	PRDB6	117	$\overline{\text{SRWE}}$	43
DVSS9	83	PRDB7	116	TCK	131
DXCVR/$\overline{\text{EAR}}$	109	$\overline{\text{REF}}$	59	TDI	128
EECS	127	RESET	66	TDO	129
IOCHRDY	45	RXD-	89	TMS	130
$\overline{\text{IOCS16}}$	54	RXD+	90	TXD-	93
$\overline{\text{IOR}}$	63	SA0	3	TXD+	95
$\overline{\text{IOW}}$	64	SA1	4	TXPD-	92
IRQ10	53	SA10	15	TXPD+	94
IRQ11	51	SA11	16	XTAL1	97
IRQ12	50	SA12	17	XTAL2	99

PIN DESIGNATIONS: SHARED MEMORY
Listed by Group

Pin Name	Pin Function	I/O	Driver
ISA Bus Interface			
AEN	Address Enable	I	
IOCHRDY	I/O Channel Ready	O	OD3
$\overline{\text{IOCS16}}$	I/O Chip Select 16	O	OD3
$\overline{\text{IOR}}$	I/O Read Select	I	
$\overline{\text{IOW}}$	I/O Write Select	I	
IRQ[3, 4, 5, 9, 10, 11, 12, 15]	Interrupt Request	O	TS3/OD3
$\overline{\text{MEMR}}$	Memory Read Select	I	
$\overline{\text{MEMW}}$	Memory Write Select	I	
REF	Memory Refresh Active	I	
RESET	System Reset	I	
SA[0-15]	System Address Bus	I	
$\overline{\text{SBHE}}$	System Byte High Enable	I	
SD[0-15]	System Data Bus	I/O	TS3
Board Interfaces			
IRQ15/$\overline{\text{APCS}}$	IRQ15 or Address PROM Chip Select	O	TS1
$\overline{\text{BPCS}}$	Boot PROM Chip Select	O	TS1
$\overline{\text{BPAM}}$	Boot PROM Address Match	I	
DXCVR/$\overline{\text{EAR}}$	Disable Transceiver	I/O	TS1
$\overline{\text{LED0}}$	$\overline{\text{LED0/LNKST}}$	O	TS2
$\overline{\text{LED1}}$	$\overline{\text{LED1/SFBD/RCVACT}}$	O	TS2
$\overline{\text{LED2}}$	$\overline{\text{LED2/SRD/RXDATD01}}$	O	TS2
$\overline{\text{LED3}}$	$\overline{\text{LED3/SRDCLK/XMTACT}}$	O	TS2
PRAB[0-15]	PRivate Address Bus	I/O	TS3
PRDB[3-7]	PRivate Data Bus	I/O	TS1
$\overline{\text{SLEEP}}$	Sleep Mode	I	
$\overline{\text{SMA}}$	Shared Memory Architecture	I	
$\overline{\text{SMAM}}$	Shared Memory Address Match	I	
$\overline{\text{SROE}}$	Static RAM Output Enable	O	TS3
$\overline{\text{SRWE}}$	Static RAM Write Enable	O	TS1
XTAL1	Crystal Oscillator Input	I	
XTAL2	Crystal Oscillator OUTPUT	O	
SHFBUSY	Read access from EEPROM in process	O	
PRDB(0)/EESK	Serial Shift Clock	I/O	
PRDB(1)/EEDI	Serial Shift Data In	I/O	
PRDB(2)/EEDO	Serial Shift Data Out	I/O	
EECS	EEPROM Chip Select	O	

PIN DESIGNATIONS: SHARED MEMORY (continued)
Listed by Group

Pin Name	Pin Function	I/O	Driver
Attachment Unit Interface (AUI)			
CI±	Collision Inputs	I	
DI±	Receive Data	I	
DO±	Transmit Data	O	
Twisted Pair Transceiver Interface (10BASE–T)			
RXD±	10BASE-T Receive Data	I	
TXD±	10BASE-T Transmit Data	O	
TXPD±	10BASE-T Predistortion Control	O	
IEEE 1149.1 Test Access Port Interface (JTAG)			
TCK	Test Clock	I	
TDI	Test Data Input	I	
TDO	Test Data Output	O	TS2
TMS	Test Mode Select	I	
Power Supplies			
AVDD	Analog Power [1-4]		
AVSS	Analog Ground [1-2]		
DVDD	Digital Power [1-7]		
DVSS	Digital Ground [1-13]		

Output Driver Types

Name	Type	I_{OL} (mA)	I_{OH} (mA)	pF
TS1	Tri-State	4	–1	50
TS2	Tri-State	12	–4	50
TS3	Tri-State	24	–3	120
OD3	Open Drain	24	–3	120

PIN DESCRIPTION: SHARED MEMORY MODE

ISA Interface

AEN
Address Enable *Input*

This signal must be driven LOW when the bus performs an I/O access to the device.

IOCHRDY
I/O Channel Ready *Output*

When the PCnet-ISA+ controller is being accessed, a HIGH on IOCHRDY indicates that valid data exists on the data bus for reads and that data has been latched for writes.

$\overline{\text{IOCS16}}$
I/O Chip Select 16 *Input/Output*

When an I/O read or write operation is performed, the PCnet-ISA+ controller will drive this pin LOW to indicate that the chip supports a 16-bit operation at this address. (If the motherboard does not receive this signal, then the motherboard will convert a 16-bit access to two 8-bit accesses.)

The PCnet-ISA+ controller follows the IEEE P996 specification that recommends this function be implemented as a pure decode of SA0-9 and AEN, with no dependency on $\overline{\text{IOR}}$, or $\overline{\text{IOW}}$; however, some PC/AT clone systems are not compatible with this approach. For this reason, the PCnet-ISA+ controller is recommended to be configured to run 8-bit I/O on all machines. Since data is moved by memory cycles there is virtually no performance loss incurred by running 8-bit I/O and compatibility problems are virtually eliminated. The PCnet-ISA+ controller can be configured to run 8-bit-only I/O by clearing Bit 0 in Plug and Play Register F0.

$\overline{\text{IOR}}$
I/O Read *Input*

To perform an Input/Output Read operation on the device $\overline{\text{IOR}}$ must be asserted. $\overline{\text{IOR}}$ is only valid if the AEN signal is LOW and the external address matches the PCnet-ISA+ controller 's predefined I/O address location. If valid, $\overline{\text{IOR}}$ indicates that a slave read operation is to be performed.

$\overline{\text{IOW}}$
I/O Write *Input*

To perform an Input/Output write operation on the device $\overline{\text{IOW}}$ must be asserted. $\overline{\text{IOW}}$ is only valid if AEN signal is LOW and the external address matches the PCnet-ISA+ controller's predefined I/O address location. If valid, $\overline{\text{IOW}}$ indicates that a slave write operation is to be performed.

IRQ3, 4, 5, 9, 10, 11, 15
Interrupt Request *Output*

An attention signal which indicates that one or more of the following status flags is set: BABL, MISS, MERR, RINT, IDON or TXSTRT. All status flags have a mask bit which allows for suppression of IRQ assertion. These flags have the following meaning:

BABL	Babble
RCVCCO	Receive Collision Count Overflow
JAB	Jabber
MISS	Missed Frame
MERR	Memory Error
MPCO	Missed Packet Count Overflow
RINT	Receive Interrupt
IDON	Initialization Done
TXSTRT	Transmit Start

$\overline{\text{MEMR}}$
Memory Read *Input*

$\overline{\text{MEMR}}$ goes LOW to perform a memory read operation.

$\overline{\text{MEMW}}$
Memory Write *Input*

$\overline{\text{MEMW}}$ goes LOW to perform a memory write operation.

RESET
Reset *Input*

When RESET is asserted HIGH, the PCnet-ISA+ controller performs an internal system reset. RESET must be held for a minimum of 10 XTAL1 periods before being deasserted. While in a reset state, the PCnet-ISA+ controller will tristate or deassert all outputs to predefined reset levels. The PCnet-ISA+ controller resets itself upon power-up.

SA0-15
System Address Bus *Input*

This bus carries the address inputs from the system address bus. Address data is stable during command active cycle.

$\overline{\text{SBHE}}$
System Bus High Enable *Input*

This signal indicates the HIGH byte of the system data bus is to be used. There is a weak pull-up resistor on this pin. If the PCnet-ISA+ controller is installed in an 8-bit only system like the PC/XT, $\overline{\text{SBHE}}$ will always be HIGH and the PCnet-ISA+ controller will perform only 8-bit operations. There must be at least one LOW going edge on this signal before the PCnet-ISA+ controller will perform 16-bit operations.

SD0-15
System Data Bus *Input/Output*

This bus is used to transfer data to and from the PCnet-ISA+ controller to system resources via the ISA data bus. SD0-15 is driven by the PCnet-ISA+ controller when performing slave read operations.

Likewise, the data on SD0-15 is latched by the PCnet-ISA+ controller when performing slave write operations.

BOARD INTERFACE

APCS/IRQ15
Address PROM Chip Select *Output*

This signal is asserted when the external Address PROM is read. When an I/O read operation is performed on the first 16 bytes in the PCnet-ISA+ controller's I/O space, APCS is asserted. The outputs of the external Address PROM drive the PROM Data Bus. The PCnet-ISA+ controller buffers the contents of the PROM data bus and drives them on the lower eight bits of the System Data Bus. IOCS16 is not asserted during this cycle.

BPAM
Boot PROM Address Match *Input*

This pin indicates a Boot PROM access cycle. If no Boot PROM is installed, this pin has a default value of HIGH and thus may be left connected to V_{DD}.

BPCS
Boot PROM Chip Select *Output*

This signal is asserted when the Boot PROM is read. If BPAM is active and MEMR is active, the BPCS signal will be asserted. The outputs of the external Boot PROM drive the PROM Data Bus. The PCnet-ISA+ controller buffers the contents of the PROM data bus and drives them on the System Data Bus. IOCS16 is not asserted during this cycle. If 16-bit cycles are performed, it is the responsibility of external logic to assert MEMCS16 signal.

DXCVR/EAR
Disable Transceiver/
External Address Reject *Input/Output*

This pin disables the transceiver. The DXCVR output is configured in the initialization sequence. A high level indicates the Twisted Pair Interface is active and the AUI is inactive, or SLEEP mode has been entered. A low level indicates the AUI is active and the Twisted Pair interface is inactive.

If EADI mode is selected, this pin becomes the EAR input.

The incoming frame will be checked against the internally active address detection mechanisms and the result of this check will be OR'd with the value on the EAR pin. The EAR pin is defined as REJECT. (See the EADI section for details regarding the function and timing of this signal.)

LED0-3
LED Drivers *Output*

These pins sink 12 mA each for driving LEDs. Their meaning is software configurable (see section *The ISA Bus Configuration Registers*) and they are active LOW.

When EADI mode is selected, the pins named LED1, LED2, and LED3 change in function while LED0 continues to indicate 10BASE-T Link Status. The DXCVR input becomes the EAR input.

LED	EADI Function
1	SF/BD
2	SRD
3	SRDCLK

PRAB0-15
Private Address Bus *Input/Output*

The Private Address Bus is the address bus used to drive the Address PROM, Remote Boot PROM, and SRAM. PRAB10-15 are required to be buffered by a Bus Buffer with ABOE as its control and SA10-15 as its inputs.

PRDB3-7
Private Data Bus *Input/Output*

This is the data bus for the static RAM, the Boot PROM, and the Address PROM.

PRDB2/EEDO
Private Data Bus Bit 2/Data Out *Input/Output*

A multifunction pin which serves as PRDB2 of the private data bus and, when ISACSR3 bit 4 is set, changes to become DATA OUT from the EEPROM.

PRDB1/EEDI
Private Data Bus Bit 1/Data In *Input/Output*

A multifunction pin which serves as PRDB1 of the private data bus and, when ISACSR3 bit 4 is set, changes to become DATA In to the EEPROM.

PRDB0/EESK
Private Data Bus Bit 0/
Serial Clock *Input/Output*

A multifunction pin which serves as PRDB0 of the private data bus and, when ISACSR3 bit 4 is set, changes to become Serial Clock to the EEPROM.

SHFBUSY
Shift Busy *Input/Output*

An output from PCnet-ISA+ which indicates that a read from the external EEPROM is in progress. It is active only when the hardware reconfigure is running (when data is being shifted out of the EEPROM due to a hardware RESET or the EELOAD command being issued).

SHFBUSY should be connected to V_{CC} with a 10K Ω resistor.

EECS
EEPROM CHIPSELECT *Output*

This signal is asserted when read or write accesses are being performed to the EEPROM. It is controlled by ISACSR3. It is driven at Reset during EEPROM Read.

SLEEP
Sleep *Input*

When SLEEP input is asserted (active LOW), the PCnet-ISA+ controller performs an internal system reset

and proceeds into a power savings mode. All outputs will be placed in their normal reset condition. All PCnet-ISA+ controller inputs will be ignored except for the SLEEP pin itself. Deassertion of SLEEP results in wake-up. The system must delay the starting of the network controller by 0.5 seconds to allow internal analog circuits to stabilize.

SMA
Shared Memory Architecture ***Input***

This pin is sampled after the hardware RESET sequence. The pin must be pulled permanently LOW for operation in the shared memory mode.

SMAM
Shared Memory Address Match ***Input***

This pin indicates an access to shared memory when active. The type of access is decided by MEMR or MEMW.

SROE
Static RAM Output Enable ***Output***

This pin directly controls the external SRAM's OE pin.

SRCS/IRQ12
Static RAM Chip Select ***Output***

This pin directly controls the external SRAM's chip select (CS) pin when the Flash boot ROM option is selected.

When Flash boot ROM option is not selected, this pin becomes IRQ12.

SRWE/WE
Static RAM Write Enable/ Write Enable ***Output***

This pin (SRWE) directly controls the external SRAM's WE pin when a Flash memory device is not implemented.

When a Flash memory device is implemented, this pin becomes a global write enable (WE) pin.

XTAL1
Crystal Connection ***Input***

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. Alternatively, an external 20 MHz CMOS-compatible clock signal can be used to drive this pin. Refer to the section on External Crystal Characteristics for more details.

XTAL2
Crystal Connection ***Output***

The internal clock generator uses a 20 MHz crystal that is attached to pins XTAL1 and XTAL2. If an external clock is used, this pin should be left unconnected.

PIN DESCRIPTION: NETWORK INTERFACES

AUI

CI+, CI–
Control Input *Input*

This is a differential input pair used to detect Collision (Signal Quality Error Signal).

DI+, DI–
Data In *Input*

This is a differential receive data input pair to the PCnet-ISA+ controller.

DO+, DO–
Data Out *Output*

This is a differential transmit data output pair from the PCnet-ISA+ controller.

Twisted Pair Interface

RXD+, RXD–
Receive Data *Input*

This is the 10BASE-T port differential receive input pair.

TXD+, TXD–
Transmit Data *Output*

These are the 10BASE-T port differential transmit drivers.

TXP+, TXP–
Transmit Predistortion Control *Output*

These are 10BASE-T transmit waveform pre-distortion control differential outputs.

PIN DESCRIPTION: IEEE 1149.1 (JTAG) TEST ACCESS PORT

TCK
Test Clock *Input*

This is the clock input for the boundary scan test mode operation. TCK can operate up to 10 MHz. TCK does not have an internal pullup resistor and must be connected to a valid TTL level of high or low. TCK must not be left unconnected.

TDI
Test Data Input *Input*

This is the test data input path to the PCnet-ISA+ controller. If left unconnected, this pin has a default value of HIGH.

TDO
Test Data Output *Output*

This is the test data output path from the PCnet-ISA+ controller. TDO is tri-stated when JTAG port is inactive.

TMS
Test Mode Select *Input*

This is a serial input bit stream used to define the specific boundary scan test to be executed. If left unconnected, this pin has a default value of HIGH.

PIN DESCRIPTION: POWER SUPPLIES

All power pins with a "D" prefix are digital pins connected to the digital circuitry and digital I/O buffers. All power pins with an "A" prefix are analog power pins connected to the analog circuitry. Not all analog pins are quiet and special precaution must be taken when doing board layout. Some analog pins are more noisy than others and must be separated from the other analog pins.

AVDD1–4
Analog Power (4 Pins) *Power*

Supplies power to analog portions of the PCnet-ISA+ controller. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines.

AVSS1–2
Analog Ground (2 Pins) *Power*

Supplies ground reference to analog portions of PCnet-ISA+ controller. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines.

DVDD1–7
Digital Power (7 Pins) *Power*

Supplies power to digital portions of PCnet-ISA+ controller. Four pins are used by Input/Output buffer drivers and two are used by the internal digital circuitry.

DVSS1–13
Digital Ground (13 Pins) *Power*

Supplies ground reference to digital portions of PCnet-ISA+ controller. Ten pins are used by Input/Output buffer drivers and two are used by the internal digital circuitry.

FUNCTIONAL DESCRIPTION

The PCnet-ISA+ controller is a highly integrated system solution for the PC-AT ISA architecture. It provides an Ethernet controller, AUI port, and 10BASE-T transceiver. The PCnet-ISA+ controller can be directly interfaced to an ISA system bus. The PCnet-ISA+ controller contains an ISA bus interface unit, DMA Buffer Management Unit, 802.3 Media Access Control function, separate 136-byte transmit and 128-byte receive FIFOs, IEEE defined Attachment Unit Interface (AUI), and Twisted-Pair Transceiver Media Attachment Unit. In addition, a Sleep function has been incorporated which provides low standby current for power sensitive applications.

The PCnet-ISA+ controller is register compatible with the LANCE (Am7990) Ethernet controller and PCnet-ISA (Am79C960). The DMA Buffer Management Unit supports the LANCE descriptor software model and the PCnet-ISA+ controller is software compatible with the Novell NE2100 and NE1500T add-in cards.

External remote boot PROMs and Ethernet physical address PROMs are supported. The location of the I/O registers, Ethernet address PROM, and the boot PROM are determined by the programming of the registers internal to PCnet-ISA+. These registers are loaded at RESET from the EEPROM.

Normally, the Ethernet physical address will be stored in the EEPROM with the other configuration data. This reduces the parts count, board space requirements, and power consumption. The option to use a standard parallel 8 bit PROM is provided to manufactures who are concerned about the non-volatile nature of EEPROMs.

The PCnet-ISA+ controller's bus master architecture brings to system manufacturers (adapter card and motherboard makers alike) something they have not been able to enjoy with other architectures—a low-cost system solution that provides the lowest parts count and highest performance. As a bus-mastering device, costly and power-hungry external SRAMs are not needed for packet buffering. This results in lower system cost due to fewer components, less real-estate and less power.

The PCnet-ISA+ controller's advanced bus mastering architecture also provides high data throughput and low CPU utilization for even better performance.

To offer greater flexibility, the PCnet-ISA+ controller has a shared memory mode to meet varying application needs. The shared memory architecture is compatible with very low-end machines, such as PC/XTs that do not support bus mastering, and very high end machines which require local packet buffering for increased system latency.

The network interface provides an Attachment Unit Interface and Twisted-Pair Transceiver functions. Only one interface is active at any particular time. The AUI allows for connection via isolation transformer to 10BASE5 and 10BASE2, thick and thin based coaxial cables. The Twisted-Pair Transceiver interface allows for connection of unshielded twisted-pair cables as specified by the Section 14 supplement to IEEE 802.3 Standard (Type 10BASE-T).

Bus Master Mode

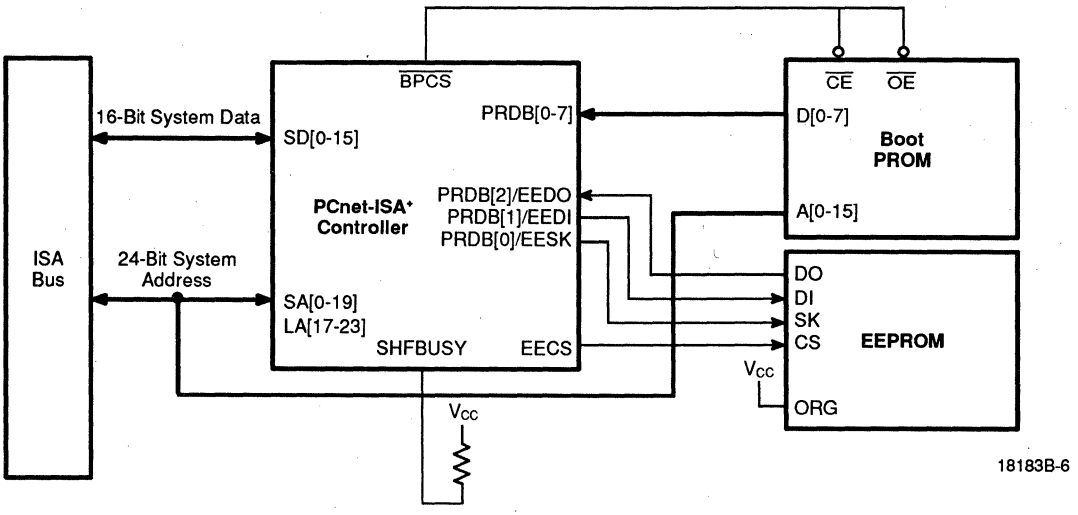
System Interface

The PCnet-ISA+ controller has two fundamental operating modes, Bus Master and Shared Memory. The selection of either the Bus Master mode or the Shared Memory mode must be done through hard wiring; it is not software configurable. The Bus Master mode provides an Am7990 (LANCE) compatible Ethernet controller, an Ethernet Address EEPROM or PROM, a Boot PROM, and a set of device configuration registers.

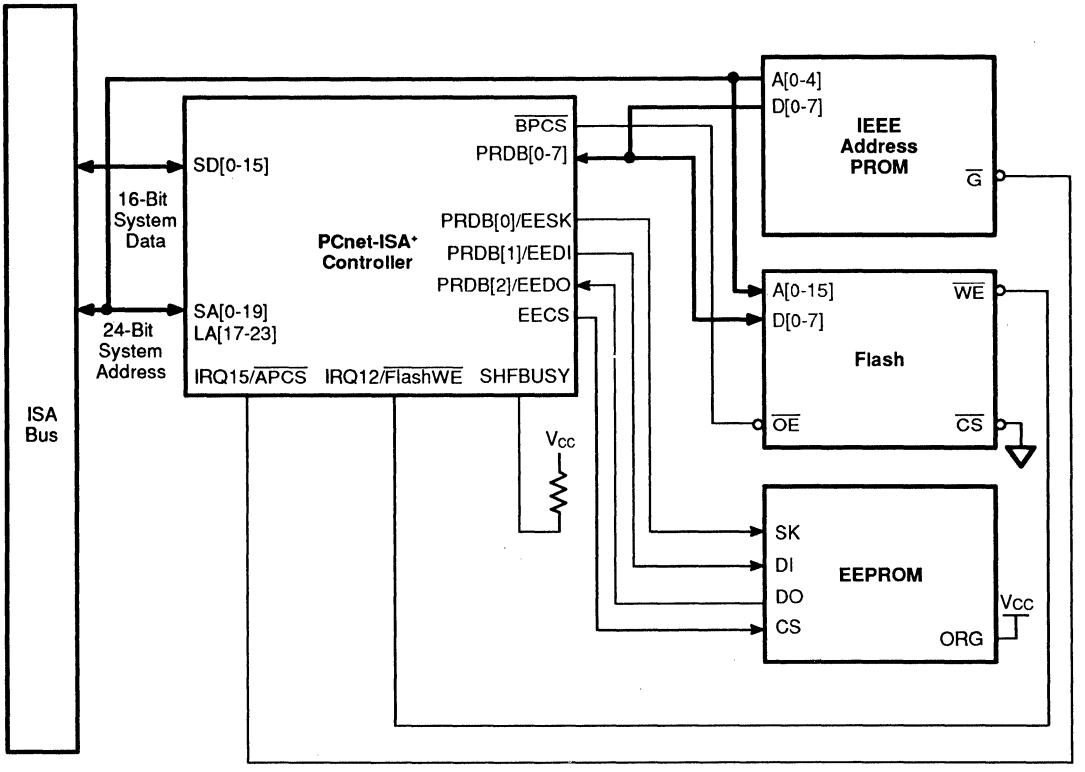
The optional Boot PROM is in memory address space and is expected to be 8–64K. On-chip address comparators control device selection based on the value of the EEPROM.

The address PROM, board configuration registers, and the Ethernet controller occupy 24 bytes of I/O space and can be located on 16 different starting addresses.

Data buffers are located in system memory and can be accessed by the PCnet-ISA+ controller when the device becomes the Current Master.



Bus Master Block Diagram
Plug and Play Compatible



18183B-7

Bus Master Block Diagram
Plug and Play Compatible with Flash Support

Shared Memory Mode

System Interface

The Shared Memory mode is the other fundamental operating mode available on the PCnet-ISA+ controller. The PCnet-ISA+ controller uses the same descriptor and buffer architecture as the LANCE, but these data structures are stored in static RAM controlled by the PCnet-ISA+ controller. The static RAM is visible as a memory resource to the PC. The other resources look the same as in the Bus Master mode.

The Boot PROM is selected by an external device which drives the Boot PROM Address Match ($\overline{BPAM}$) input to the PCnet-ISA+ controller. The PCnet-ISA+ controller can perform two 8-bit accesses from the 8-bit Boot PROM and presents 16-bits of data. The shared memory works the same way, with an external device generating Shared Memory Address Match and the PCnet-ISA+ controller performing the read or write and the 8 to 16-bit data conversion.

Converting shared memory accesses from 8-bit cycles to 16-bit cycles allows use of the much faster 16-bit cycle timing while cutting the number of bus cycles in half. This raises performance to more than 400% of what could be achieved with 8-bit cycles. Converting boot PROM accesses to 16-bit cycles allows the two memory resources to be in the same 128 Kbyte block of memory without a clash between two devices with different data widths.

The PCnet-ISA+ controller uses an internal address comparator to perform SRAM prefetches on the Private Data Bus; the SA0-15 signals are used internally to determine whether a SRAM read cycle prefetch is a match or a miss.

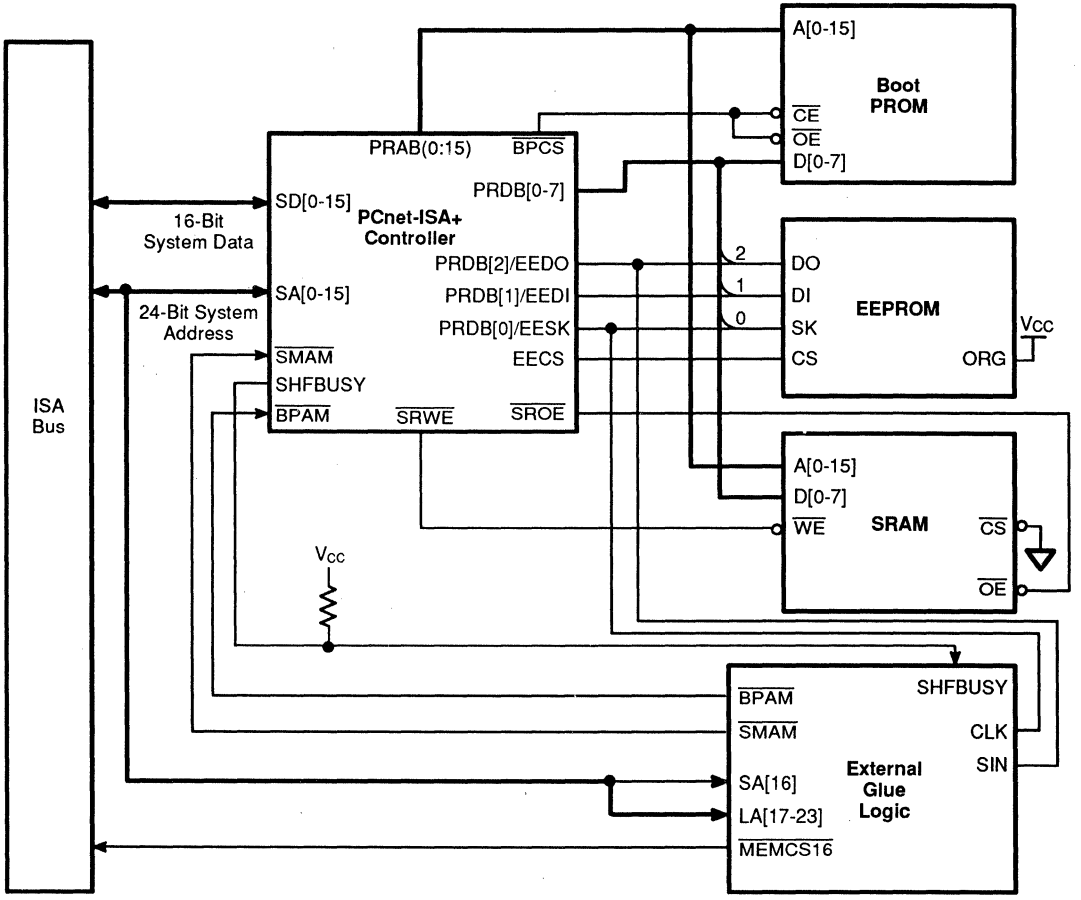
Access to the Ethernet controller registers, board configuration registers, and Address PROM is done with on-chip address comparators.

Network Interface

The PCnet-ISA+ controller can be connected to an IEEE 802.3 network via one of two network interface ports. The Attachment Unit Interface (AUI) provides an IEEE 802.3 compliant differential interface to a remote MAU or an on-board transceiver. The 10BASE-T interface provides a twisted-pair Ethernet port. The PCnet-ISA+ controller provides three modes of network interface

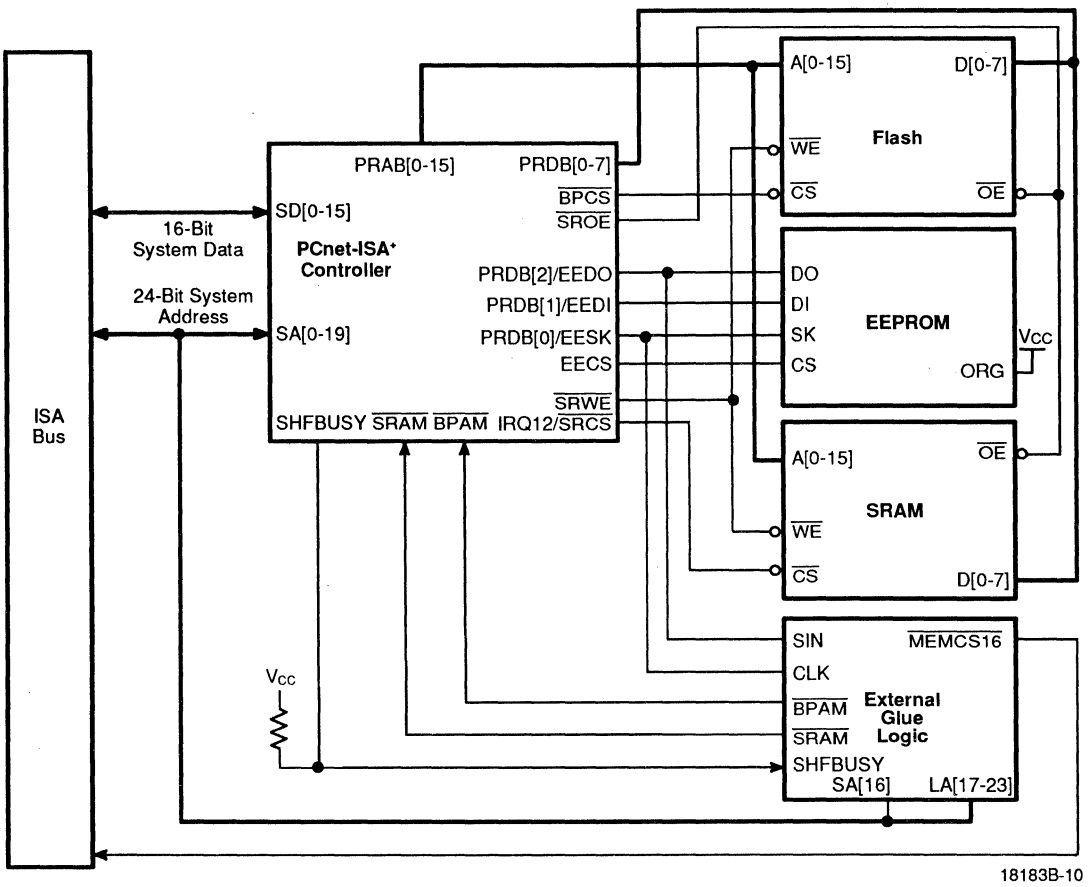
selection: automatic selection, software selection, and jumper selection of AUI or 10BASE-T interface.

In the automatic selection mode, the PCnet-ISA+ controller will select the interface that is connected to the network by checking the Link Status state machine. If both AUI and 10BASE-T interfaces are connected, the 10BASE-T interface is selected over AUI. If the PCnet-ISA+ controller is initialized for software selection of network interface, it will read the PORTSEL [1:0] bits in the Mode register (CSR15.8 and CSR15.7) to determine which interface needs to be activated.



18183B-9

Shared Memory Block Diagram Plug and Play Compatible



Shared Memory Block Diagram
Plug and Play Compatible with Flash Memory Support

PLUG AND PLAY

Plug and Play is a standardized method of configuring jumperless adapter cards in a system. Plug and Play is a Microsoft standard and is based on a central software configuration program, either in the operating system or elsewhere, which is responsible for configuring all Plug and Play cards in a system. Plug and Play is fully supported by the PCnet-ISA+ ethernet controller.

For a copy of the Microsoft Plug and Play specification contact Microsoft Inc. This specification should be referenced in addition to PCnet-ISA+ Technical Reference Manual and this data sheet.

Operation

If the PCnet-ISA+ ethernet controller is used to boot off the network, the device will come up active at RESET, otherwise it will come up inactive. Information stored in the serial EEPROM is used to identify the card and to describe the system resources required by the card, such as I/O space, Memory space, IRQs and DMA channels. This information is stored in a standardized Read Only format. Operation of the Plug and Play system is shown as follows.

- Isolate the Plug and Play card
- Read the cards resource data
- Identify the card
- Configure its resources

The Plug and Play mode of operation allows the following benefits to the end user.

- Eliminates all jumpers or dip switches from the adapter card
- Ease of use is greatly enhanced
- Allows the ability to uniquely address identical cards in a system, without conflict
- Allows the software configuration program or OS to read out the system resource requirements required by the card
- Defines a mechanism to set or modify the current configuration of each card
- Maintain backward compatability with other ISA bus adapters

Auto-Configuration Ports

Three 8 bit I/O ports are used by the Plug and Play configuration software on each Plug and Play device to communicate with the Plug and Play registers. The ports are listed in the table below. The software configuration space is defined as a set of 8 bit registers. These registers are used by the Plug and Play software configuration to issue commands, access the resource information, check status, and configure the PCnet-ISA+ controller hardware.

Port Name	Location	Type
ADDRESS	0X279 (Printer Status Port)	Write-only
WRITE-DATA	0xA79 (Printer status port + 0x0800)	Write-only
READ-DATA	Relocatable in range 0x0203-0x03FF	Read-only

The address and Write_DATA ports are located at fixed, predefined I/O addresses. The Write_Data port is located at an alias of the Address port. All three auto-configuration ports use a 12-bit ISA address decode.

The READ_DATA port is relocatable within the range 0x203–0x3FF by a command written to the WRITE_DATA port.

ADDRESS PORT

The internal Plug and Play registers are accessed by writing the address to the ADDRESS PORT and then either reading the READ_DATA PORT or writing to the WRITE_DATA PORT. Once the ADDRESS PORT has been written, any number of reads or writes can occur without having to rewrite the ADDRESS PORT.

The ADDRESS PORT is also the address to which the initiation key is written to, which is described later.

WRITE_DATA PORT

The WRITE_DATA PORT is the address to which all writes to the internal Plug and Play registers occur. The destination of the data written to the WRITE_DATA PORT is determined by the last value written to the ADDRESS PORT.

READ_DATA PORT

The READ_DATA PORT is used to read information from the internal Plug and Play registers. The register to be read is determined by the last value of the ADDRESS PORT.

The I/O address of the READ_DATA PORT is set by writing the chosen I/O location to Plug and Play Register 0. The isolation protocol can determine that the address chosen is free from conflict with other devices I/O ports.

Initiation Key

The PCnet-ISA+ controller is disabled at reset when operating in Plug and Play mode. It will not respond to any memory or I/O accesses, nor will the PCnet-ISA+ controller drive any interrupts or DMA channels.

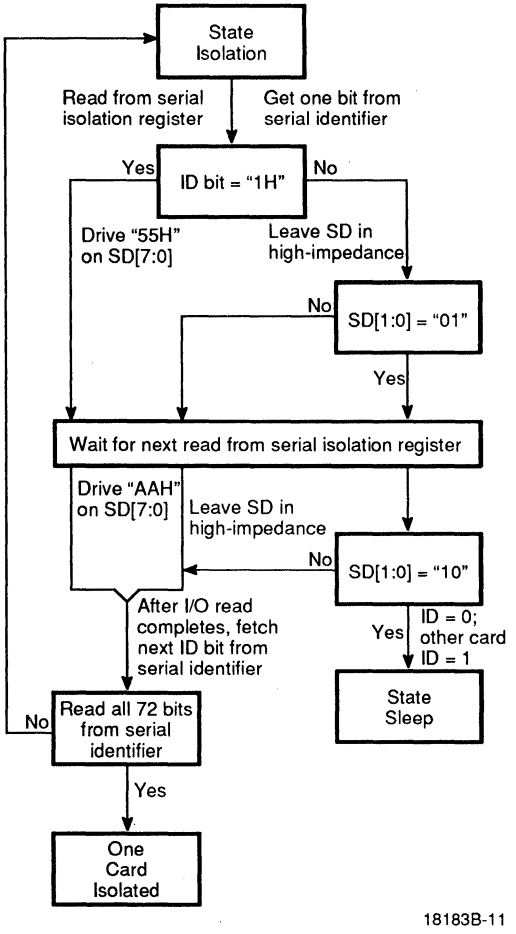
The initiation key places the PCnet-ISA+ device into the configuration mode. This is done by writing a predefined pattern to the ADDRESS PORT. If the proper sequence of I/O writes are detected by the PCnet-ISA+ device, the Plug and Play auto-configuration ports are enabled. This sequence must be sequential, i.e., any other I/O access to this I/O port will reset the state machine which is checking the pattern. Interrupts should be disabled during this time to eliminate any extraneous I/O cycles.

The exact sequence for the initiation key is listed below in hexadecimal.

```
6A, B5, DA, ED, F6, FB, 7D, BE
DF, 6F, 37, 1B, 0D, 86, C3, 61
B0, 58, 2C, 16, 8B, 45, A2, D1
E8, 74, 3A, 9D, CE, E7, 73, 39
```

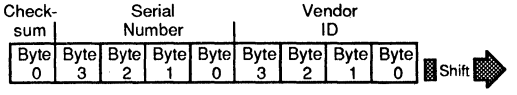
Isolation Protocol

A simple algorithm is used to isolate each Plug and Play card. This algorithm uses the signals on the ISA bus and requires lock-step operation between the Plug and Play hardware and the isolation software.



Plug and Play ISA Card Isolation Algorithm

The key element of this mechanism is that each card contains a unique number, referred to as the serial identifier for the rest of the discussion. The serial identifier is a 72-bit unique, non-zero, number composed of two, 32-bit fields and an 8-bit checksum. The first 32-bit field is a vendor identifier. The other 32 bits can be any value, for example, a serial number, part of a LAN address, or a static number, as long as there will never be two cards in a single system with the same 64 bit number. The serial identifier is accessed bit-serially by the isolation logic and is used to differentiate the cards.



Shifting of Serial Identifier

The shift order for all Plug and Play serial isolation and resource data is defined as bit[0], bit[1], and so on through bit[7].

Hardware Protocol

The isolation protocol can be invoked by the Plug and Play software at any time. The initiation key, described earlier, puts all cards into configuration mode. The hardware on each card expects 72 pairs of I/O read accesses to the READ_DATA port. The card's response to these reads depends on the value of each bit of the serial identifier which is being examined one bit at a time in the sequence shown above.

If the current bit of the serial identifier is a "1", then the card will drive the data bus to 0x55 to complete the first I/O read cycle. If the bit is "0", then the card puts its data bus driver into high impedance. All cards in high impedance will check the data bus during the I/O read cycle to sense if another card is driving D[1:0] to "01". During the second I/O read, the card(s) that drove the 0x55, will now drive a 0xAA. All high impedance cards will check the data bus to sense if another card is driving D[1:0] to "10". Between pairs of Reads, the software should wait at least 30 µs.

If a high impedance card sensed another card driving the data bus with the appropriate data during both cycles, then that card ceases to participate in the current iteration of card isolation. Such cards, which lose out, will participate in future iterations of the isolation protocol.

NOTE: *During each read cycle, the Plug and Play hardware drives the entire 8-bit databus, but only checks the lower 2 bits.*

If a card was driving the bus or if the card was in high impedance and did not sense another card driving the bus, then it should prepare for the next pair of I/O reads. The card shifts the serial identifier by one bit and uses the shifted bit to decide its response. The above sequence is repeated for the entire 72-bit serial identifier.

At the end of this process, one card remains. This card is assigned a handle referred to as the *Card Select Number* (CSN) that will be used later to select the card. Cards which have been assigned a CSN will not participate in subsequent iterations of the isolation protocol.

Cards must be assigned a CSN before they will respond to the other commands defined in the specification.

It should be noted that the protocol permits the 8-bit checksum to be stored in non-volatile memory on the card or generated by the on-card logic in real-time. The same LFSR algorithm described in the initiation key section of the Plug and Play specification is used in the checksum generation.

Software Protocol

The Plug and Play software sends the initiation key to all Plug and Play cards to place them into configuration mode. The software is then ready to perform the isolation protocol.

The Plug and Play software generates 72 pairs of I/O read cycles from the READ_DATA port. The software checks the data returned from each pair of I/O reads for the 0x55 and 0xAA driven by the hardware. If both 0x55 and 0xAA are read back, then the software assumes that the hardware had a "1" bit in that position. All other results are assumed to be a "0."

During the first 64 bits, software generates a checksum using the received data. The checksum is compared with the checksum read back in the last 8 bits of the sequence.

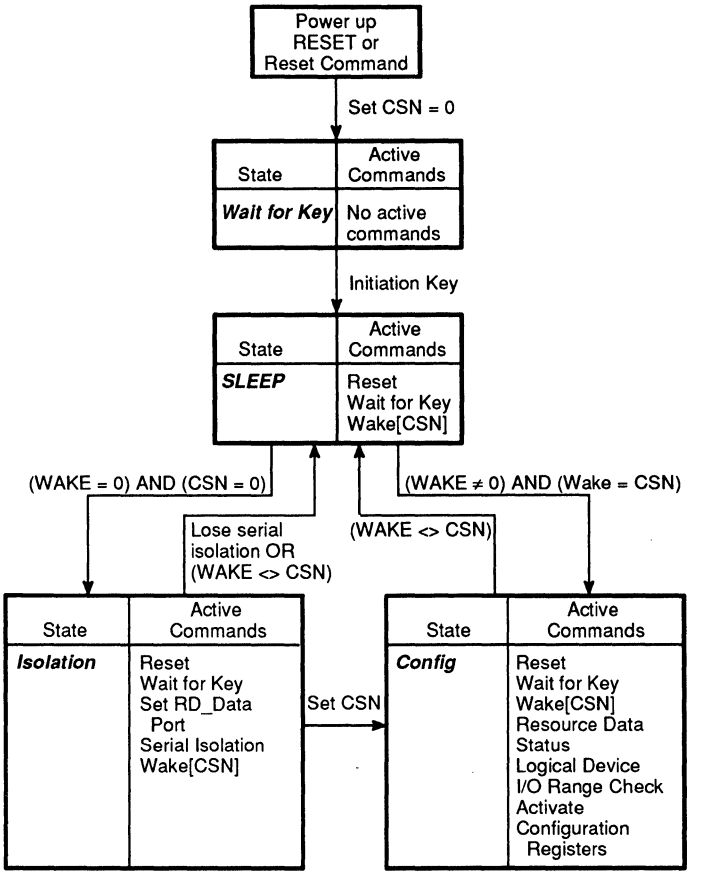
There are two other special considerations for the software protocol. During an iteration, it is possible that the 0x55 and 0xAA combination is never detected. It is also possible that the checksum does not match If either of these cases occur on the first iteration, it must be assumed that the READ_DATA port is in conflict. If a conflict is detected, then the READ_DATA port is relocated. The above process is repeated until a non-conflicting location for the READ_DATA port is found. The entire range between 0x203 and 0x3FF is available, however in practice it is expected that only a few locations will be tried before software determines that no Plug and Play cards are present.

During subsequent iterations, the occurrence of either of these two special cases should be interpreted as the absence of any further Plug and Play cards (i.e. the last card was found in the previous iteration). This terminates the isolation protocol.

NOTE: *The software must delay 1 ms prior to starting the first pair of isolation reads, and must wait 250 µsec between each subsequent pair of isolation reads. This delay gives the ISA card time to access information from possibly very slow storage devices.*

Plug and Play Card Control Registers

The state transitions and card control commands for the PCnet-ISA+ controller are shown in the following figure.



Notes

1. *CSN = Card Select Number*
2. *RESET or the Reset command causes a state transition from the current state to Wait for Key and sets all CSNs to zero.*
3. *The Wait for Key command causes a state transition from the current state to Wait for Key.*

Plug and Play ISA Card State Transitions

Plug and Play Registers

The PCnet-ISA+ controller supports all of the defined Plug and Play card control registers. Refer to the tables on the following pages for detailed information.

Plug and Play Standard Registers

Name	Address Port Value	Definition
Set RD_DATA Port	0x00	Writing to this location modifies the address of the port used for reading from the Plug and Play ISA cards. Bits[7:00] become I/O read port address bits [9:02]. Reads from this register are ignored. I/O Address bits 11:10 should = 00, and 1:0 = 11.
Serial Isolation	0x01	A read to this register causes a Plug and Play card in the ***Isolation*** state to compare one bit of the board's ID. This process is fully described above. This register is read only.
Config Control	0x02	Bit[0] - Reset all logical devices and restore configuration registers to their power-up values. Bit[1] - Return to the ***Wait for Key*** state Bit[2] - Reset CSN to 0 A write to bit[0] of this register performs a reset function on all logical devices. This resets the contents of configuration registers to their default state. All card's logical devices enter their default state and the CSN is preserved. A write to bit[1] of this register causes all cards to enter the ***Wait for Key*** state but all CSNs are preserved and logical devices are not affected. A write to bit[2] of this register causes all cards to reset their CSN to zero. This register is write-only. The values are not sticky, that is, hardware will automatically clear them and there is no need for software to clear the bits.
Wake[CSN]	0x03	A write to this port will cause all cards that have a CSN that matches the write data[7:0] to go from the ***Sleep*** state to either the ***Isolation*** state if the write data for this command is zero or the ***Config*** state if the write data is not zero. This register is write-only. Writing to this register resets the EEPROM pointer to the beginning of the Plug and Play Data Structure.
Resource Data	0x04	A read from this address reads the next byte of resource information. The Status register must be polled until bit[0] is set before this register may be read. This register is read-only.
Status	0x05	Bit[0] when set indicates it is okay to read the next data byte from the Resource Data register. This register is read-only.
Card Select Number	0x06	A write to this port sets a card's CSN. The CSN is a value uniquely assigned to each ISA card after the serial identification process so that each card may be individually selected during a Wake [CSN] command. This register is read/write.
Logical Device Number	0x07	Selects the current logical device. This register is read only. The PCnet-ISA+ controller has only 1 logical device, and this register contains a value of 0x00

Plug and Play Logical Device Configuration Registers

The PCnet-ISA+ controller supports a subset of the defined Plug and Play logical device control registers. The reason for only supporting a subset of the registers is that the PCnet-ISA+ controller does not require as many system resources as Plug and Play allows. For instance, Memory Descriptor 2 is not used, as the PCnet-ISA+ controller only requires two memory descriptors, one for the Boot PROM/Flash, and one for the SRAM in Shared Memory Mode.

Plug and Play Logical Device Control Registers

Name	Address Port Value	Definition
Activate	0x30	For each logical device there is one activate register that controls whether or not the logical device is active on the ISA bus. Bit[0], if set, activates the logical device. Bits[7:1] are reserved and must be zero. This is a read/write register. Before a logical device is activated, I/O range check must be disabled.
I/O Range Check	0x31	This register is used to perform a conflict check on the I/O port range programmed for use by a logical device. Bit[7:2] Reserved Bit 1[1] Enable I/O Range check, if set then I/O Range Check is enabled. I/O range check is only valid when the logical device is inactive. Bit[0], if set, forces the logical device to respond to I/O reads of the logical device's assigned I/O range with a 0x55 when I/O range check is in operation. If clear, the logical device drives 0xAA. This register is read/write.

Memory Space Configuration

Name	Register Index	Definition
Memory base address bits[23:16] descriptor 0	0x40	Read/write value indicating the selected memory base address bits[23:16] for memory descriptor 0. This is the Boot Prom Space.
Memory base address bits[15:08] descriptor 0	0x41	Read/write value indicating the selected memory base address bits[15:08] for memory descriptor 0.
Memory control	0x42	Bits[2:1] specifies 8/16-bit control. The encoding is identical to memory control (bits[4:3]) of the information field in the memory descriptor. Bit[0], =0, indicates the next field is used as a range length for decode (implies range length and base alignment of memory descriptor are equal). Bit[0] is read-only.
Memory upper limit address; bits[23:16] or range length; bits[23:16] for descriptor 0	0x43	Read/write value indicating the selected memory high address bits[23:16] for memory descriptor 0. If bit[0] of memory control is 0, this is the range length. If bit[0] of memory control is 1, this is considered invalid.
Memory upper limit bits[15:08] or range length; bits[15:08] for descriptor 0	0x44	Read/write value indicating the selected memory high address bits[15:08] for memory descriptor 0, either a memory address or a range length as described above.
Memory descriptor 1	0x48-0x4C	Memory descriptor 1. This is the SRAM Space for Shared Memory.

I/O Space Configuration

Name	Register Index	Definition
I/O port base address bits[15:08] descriptor 0	0x60	Read/write value indicating the selected I/O lower limit address bits[15:08] for I/O descriptor 0. If a logical device indicates it only uses 10 bit encoding, then bits[15:10] do not need to be supported.
I/O port base address bits[07:00] descriptor 0	0x61	Read/write value indicating the selected I/O lower limit address bits[07:00] for I/O descriptor 0.

I/O Interrupt Configuration

Name	Register Index	Definition
Interrupt request level select 0	0x70	Read/write value indicating selected interrupt level. Bits[3:0] select which interrupt level used for Interrupt 0. One selects IRQL 1, fifteen selects IRQL fifteen. IRQL 0 is not a valid interrupt selection and represents no interrupt selection.
Interrupt request type select 0	0x71	Read/write value indicating which type of interrupt is used for the Request Level selected above. Bit[1] : Level, 1 = high, 0 = low Bit[0] : Type, 1 = level, 0 = edge The PCnet-ISA+ controller only supports Edge High and Level Low Interrupts.

DMA Channel Configuration

Name	Register Index	Definition
DMA channel select 0	0x74	Read/write value indicating selected DMA channels. Bits[2:0] select which DMA channel is in use for DMA 0. Zero selects DMA channel 0, seven selects DMA channel 7. DMA channel 4, the cascade channel is used to indicate no DMA channel is active.

DETAILED FUNCTIONS

EEPROM

Interface

The EEPROM supported by the PCnet-ISA+ controller is an industry standard 93C56 2-Kbit EEPROM device which uses a 4-wire interface. This device directly interfaces to the PCnet-ISA+ controller through a 4-wire interface which uses 3 of the private data bus pins for Data In, Data Out, and Serial Clock. The Chip Select pin is a dedicated pin from the PCnet-ISA+ controller.

Note: *All data stored in the EEPROM is stored in bit-reversal format. Each word (16 bits) must be written into the EEPROM with bit 15 swapped with bit 0, bit 14 swapped with bit 1, etc.*

This is a 2-Kbit device organized as 128 x 16 bit words. A map of the device as used in the PCnet-ISA+ controller is below. The information stored in the EEPROM is as follows:

IEEE address	6 bytes
Reserved	10 bytes
EISA ID	4 bytes
ISACSRs	12 bytes
Plug and Play Defaults	19 bytes
8-Bit Checksum	1 byte
External Shift Chain	2 bytes
Plug and Play Config Info	192 bytes

Serial EEPROM Byte Map

The following is a byte map of the XXC56 series of EEPROMs used by the PCnet-ISA+ Ethernet Controller.

This byte map is for the case where a non-PCnet Family compatible software driver is implemented.

			Word Location
IEEE Address (Bytes 0–5)	Byte 1	Byte 0	0
	Byte 3	Byte 2	1
	Byte 5	Byte 4	2
	Byte 7	Byte 6	3
	Byte 9	Byte 8	4
	Byte 11	Byte 10	5
	Byte 13	Byte 12	6
	Byte 15	Byte 14	7
	EISA Byte 1	EISA Byte 0	8
EISA Config Reg.	EISA Byte 3	EISA Byte 2	9
	MSRDA, ISACSR0		A
	MSWRA, ISACSR1		B
	MISC Config, ISACSR2		C
Internal Registers	LED1 Config, ISACSR5		D
	LED2 Config, ISACSR6		E
	LED3 Config, ISACSR7		F
	PnP 0x61	PnP 0x60	10
	PnP 0x71	PnP 0x70	11
	Unused	PnP 0x74	12
Plug and Play Reg.	PnP 0x41	PnP 0x40	13
	PnP 0x43	PnP 0x42	14
	Unused	PnP 0x44	15
	PnP 0x49	PnP 0x48	16
	PnP 0x4b	PnP 0x4A	17
	Unused	PnP 0x4C	18
	8-bit Checksum	PnP 0xF0	19
	External Shift Chain		1A
	Unused Locations		1B : 1F
See Appendix C	Plug and Play Starting Location		20

Note:

Checksum is calculated on words 0 through 0x1Ah (first 54 Bytes).

Serial EEPROM Byte Map

The following is a byte map of the XXC56 series of EEPROMs used by the PCnet-ISA+ Ethernet Controller.

This byte map is for the case where a PCnet Family compatible software driver is implemented.

(This byte map is an application reference for use in developing AMD software devices.)

	Word Location			
	0	Byte 1	Byte 0	IEEE Address (Bytes 0–5)
	1	Byte 3	Byte 2	
	2	Byte 5	Byte 4	
	3	Reserved	Reserved	
	4	HWID (01H)	Reserved	
	5	User Space 1		
	6	16-Bit Checksum 1		
	7	ASCII W(0 x 57H)	ASCII W(0 x 57H)	
	8	EISA Byte 1	EISA Byte 0	
EISA Config Reg.	9	EISA Byte 3	EISA Byte 2	
	A	MSRDA, ISACSR0		
	B	MSWRA, ISACSR1		
	C	MISC Config, ISACSR2		
Internal Registers	D	LED1 Config, ISACSR5		
	E	LED2 Config, ISACSR6		
	F	LED3 Config, ISACSR7		
	10	PnP 0x61	PnP 0x60	I/O Ports
	11	PnP 0x71	PnP 0x70	Interrupts
	12	Unused	PnP 0x74	DMA Channels
Plug and Play Reg.	13	PnP 0x41	PnP 0x40	ROM Memory
	14	PnP 0x43	PnP 0x42	
	15	Unused	PnP 0x44	
	16	PnP 0x49	PnP 0x48	RAM Memory
	17	PnP 0x4b	PnP 0x4A	
	18	Unused	PnP 0x4C	
	19	8-bit Checksum 2	PnP 0xF0	Vendor Byte
	1A	External Shift Chain		
	1B ⋮ 1F	Unused Locations		
See Appendix C	20	Plug and Play Starting Location		See Appendix C

Note:

Checksum 1 is calculated on words 0 through 5 plus word 7.

Checksum 2 is calculated on words 0 through 0x1Ah (first 54 Bytes).

Plug and Play Register Map

The following chart and its bit descriptions show the internal configuration registers associated with the Plug and Play operation. These registers control the configuration of the PCnet-ISA+ controller.

Plug and Play Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00	READ_DATA							
0x01	SERIAL ISOLATION							
0x02	0	0	0	0	0	RST CSN	WAIT KEY	RST ALL
0x03	WAKE [CSN]							
0x04	RESOURCE_DATA							
0x05	0	0	0	0	0	0	0	READ STATUS
0x06	CSN							
0x07	0	0	0	0	0	0	0	0
0x30	0	0	0	0	0	0	0	ACTIVATE
0x31	0	0	0	0	0	0	IORNG	IORNG

READ_DATA	Address of Plug and Play READ_DATA Port.
SERIAL_ISOLATION	Used in the Serial Isolation process.
RST_CSN	Resets CSN register to zero.
WAIT_KEY	Resets Wait for Key State.
RST_ALL	Resets all logical devices.
WAKE [CSN]	Will wake up if write data matches CSN Register.
READ_STATS	Read Status of RESOURCE DATA.
RESOURCE_DATA	Next pending byte read from EEPROM.
CSN	Plug and Play CSN Value.
ACTIVATE	Indicates that the PCnet-ISA+ device should be activated.
IORNG	Bits used to enable the I/O Range Check Command.

The following chart and its bit descriptions show the internal command registers associated with the Plug and Play operation. These registers control the PCnet-ISA+ controller Plug and Play operation.

Plug and Play Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x60	0	0	0	0	0	0	1	IOAM3
0x61	IOAM2	IOAM1	IOAM0	0	0	0	0	0
0x70	0	0	0	0	IRQ3	IRQ2	IRQ1	IRQ0
0x71	0	0	0	0	0	0	IRQ_LVL	IRQ_TYPE
0x74	0	0	0	0	0	DMA2	DMA1	DMA0
0x40	0	0	0	0	1	1	0	BPAM3
0x41	BPAM2	BPAM1	BPAM0	0	0	0	0	0
0x42	0	0	0	0	0	0	BP_16B	0
0x43	1	1	1	1	1	1	1	BPSZ3
0x44	BPSZ2	BPSZ1	BPSZ0	0	0	0	0	0
0x48	0	0	0	0	1	1	0	SRAM3
0x49	SRAM2	SRAM1	SRAM0	0	0	0	0	0
0x4A	0	0	0	0	0	0	SR16B	0
0x4B	1	1	1	1	1	1	1	SRSZ3
0x4c	SRSZ2	SRSZ1	SRSZ0	0	0	0	0	0
0xF0	0	0	0	FL_SEL	BP_CS	APROM_EN	AEN_CS	IO_MODE

Plug & Play Register Locations Detailed Description (Refer to the Plug & Play Register Map above.)

IOAM[3:0] — I/O Address Address Match to bits [8:5] of SA bus (PnP 0x60–0x61). Controls the base address of PCnet-ISA+. The IOAM will be written with a value from the EEPROM.

IOAM[3:0]	Base Address (Hex)
0 0 0 0	200
0 0 0 1	220
0 0 1 0	240
0 0 1 1	260
0 1 0 0	280
0 1 0 1	2A0
0 1 1 0	2C0
0 1 1 1	2E0
1 0 0 0	300
1 0 0 1	320
1 0 1 0	340
1 0 1 1	360
1 1 0 0	380
1 1 0 1	3A0
1 1 1 0	3C0
1 1 1 1	3E0

IRQ[3:0] — IRQ selection on the ISA bus (PnP 0x70). Controls which interrupt will be asserted. ISA Edge sensitive or EISA level mode is controlled by IRQ_TYPE bit in PnP 0x71. Default is ISA Edge Sensitive. The IRQ signals will not be driven unless PnP activate register bit is set.

IRQ[3:0]	ISA IRQ Pin
0 0 1 1	IRQ3 (Default)
0 1 0 0	IRQ4
0 1 0 1	IRQ5
1 0 0 1	IRQ9
1 0 1 0	IRQ10
1 0 1 1	IRQ11
1 1 0 0	IRQ12
1 1 1 1	IRQ15

IRQ_TYPE — IRQ Type (PnP 0x71). Indicates the type of interrupt setting; Level is 1, Edge is 0.

IRQ_LVL — IRQ Level (PnP 0x71). A read-only register bit that indicates the type of setting, active high or low. Always complement of IRQ_TYPE.

DMA[2:0] DMA Channel Select (PnP 0x74). Controls the DRQ and DMA selection of PCnet-ISA+. The DMA[2:0] register will be written with a value from the EEPROM. {For Bus Master Mode Only} The DRQ signal will not be driven unless EE_VALID is set or Non-EEPROM sequential write process is complete.

DMA[2:0]			DMA Channel (DRQ/DACK Pair)
0	1	1	Channel 3
1	0	1	Channel 5
1	1	0	Channel 6
1	1	1	Channel 7

BPAM[3:0] Boot PROM Address Match to bits [23:16] of SA bus (PnP 0x40–0x41). Selects the location where the Boot PROM Address match decode is started. The BPAM will be written with a value from the EEPROM.

BPAM[3:0]				Address Location (Hex)	Size Supported (K bytes)
0	0	0	0	C0000	8, 16, 32, 64
0	0	0	1	C2000	8
0	0	1	0	C4000	8, 16
0	0	1	1	C6000	8
0	1	0	0	C8000	8, 16, 32
0	1	0	1	CA000	8
0	1	1	0	CC000	8, 16
0	1	1	1	CE000	8
1	0	0	0	D0000	8, 16, 32, 64
1	0	0	1	D2000	8
1	0	1	0	D4000	8, 16
1	0	1	1	D6000	8
1	1	0	0	D8000	8, 16, 32
1	1	0	1	DA000	8
1	1	1	0	DC000	8, 16
1	1	1	1	DE000	8

BP_16B Boot PROM 16-bit access (PnP 0x42). Is asserted if Boot PROM cycles should respond as an 16-bit device. In Bus Master mode, all boot PROM cycles will only be 8 bits in width.

BPSZ[3:0] Boot PROM Size (PnP 0x43–0x44). Selects the size of the boot PROM selected.

BPSZ[3:0]				Boot PROM Size
0	x	x	x	No Boot PROM Selected
1	1	1	1	8 K
1	1	1	0	16 K
1	1	0	0	32 K
1	0	0	0	64 K

SRAM[3:0] Static RAM Address Match to bits [16:13] of SA bus (PnP 0x48–0x49). Selects the starting location of the Shared memory by using SA[16:13] for performing address comparisons. The shared memory address match, the $\overline{\text{SMAM}}$ is asserted low. SRAM[3] value must reflect the external address match logic for SA[16].

SRAM[2:0]			SA[15:13]			SRAM Size (K bytes)
0	0	0	0	0	0	8, 16, 32, 64
0	0	1	0	0	1	8
0	1	0	0	1	0	8, 16
0	1	1	0	1	1	8
1	0	0	1	0	0	8, 16, 32
1	0	1	1	0	1	8
1	1	0	1	1	0	8, 16
1	1	1	1	1	1	8

SR_16B Static RAM 16-bit access (PnP 0x4A). Asserted if SRAM cycles should respond as an 16-bit device.

SRSZ[3:0] Static RAM Size (PnP 0x4B–0x4C). Selects the size of the static RAM selected.

SRSZ[3:0]				Shared Memory Size
0	x	x	x	No Static RAM Selected
1	1	1	1	8 K
1	1	1	0	16 K
1	1	0	0	32 K
1	0	0	0	64 K

Vendor Defined Byte (PnP 0x0F)

IO_MODE I/O Mode. When set to one, the internal selection will respond as a 16-bit port, (i.e. drive $\overline{\text{IOCS16}}$ pin). When IO_MODE is set to zero, (Default), the internal I/O

selection will respond as an 8-bit port.

AEN_CS — External Decode Logic for I/O Registers. When written with a one, the PCnet-ISA+ will use the AEN pin as I/O chip select bar, to allow for external decode logic for the upper address bit of SA [9:5]. The purpose of this pin is to allow I/O locations, not supported with the IOAM[3:0], selection, to be defined outside the range 0x200–0x3F7. When set to a zero, (Default), I/O Selection will use IOAM[3:0].

APROM_EN — External Parallel IEEE Address PROM. When set, the IRQ15 pin is reconfigured to be an Address Chip Select low, similar to $\overline{\text{APCS}}$ pin in the existing PCnet-ISA (Am79C960) device. The purpose of this bit is to allow for both a serial EEPROM and parallel PROM to coexist. When APROM_EN is set, the IEEE address located in the serial EEPROM will be ignored and parallel access will occur over the PRDB bus. When APROM_EN is cleared, default state, the IEEE address will be read in from the serial device and written to an internal RAM. When the I/O space of the IEEE PROM is selected, PCnet-ISA+, will access the contents of this RAM for I/O read cycles. I/O write cycles will be ignored.

BP_CS — Boot PROM Chip Select. When BP_CS is set to one, BALE will act as an external chip select (active low) above bit 15 of the address bus. BALE = 0, will select the boot PROM when $\overline{\text{MEMR}}$ is asserted low if the BP_CS bit is set and BPAM[2:0] match SA[15:13] and BPSZ[3:0] matches the selected size. When BP_CS is set to zero. BALE will act as the normal address latch strobe to capture the upper address bits for memory access to the boot PROM. BP_CS is by default low. The primary purpose of this bit is to allow non-ISA bus applications to support larger Boot PROMS or non-standard Boot PROM/Flash locations.

FL_SEL — Flash Memory Device Selected. When set, the Boot PROM is replaced with an external Flash memory device. In Bus Master Mode, $\overline{\text{BPCS}}$ is replaced with $\overline{\text{Flash\_OE}}$. IRQ12 becomes $\overline{\text{Flash\_WE}}$. The Flash's CS pin is grounded. In shared memory mode, $\overline{\text{BPCS}}$ is replaced with $\overline{\text{Flash\_CS}}$. IRQ12 becomes $\overline{\text{Static\_RAM\_CS}}$ pin. The $\overline{\text{SROE}}$ and $\overline{\text{SRWE}}$ signals are connected to both the SRAM and Flash memory devices. FL_SEL is cleared by a reset, which is the default.

Shared Memory Configuration Bits (Not Defined for Bus Master Mode)

In Shared Memory Mode, the address comparison above the 15th bit must be performed by external logic. All address comparisons for bit 15th and below will use the internal compare logic.

SRAM[3:0],

SR_16B, SRSZ[3:0] — These are not defined in bus-master mode. BP_16B must be written with a zero in bus-master mode.

Note: *In Bus Master Mode, the BP_16B is always considered an 8-bit device. If $\overline{\text{SBHE}}$ signal is left unconnected, in shared memory mode (i.e. 8-bit Slot), all memory and I/O access will assume 8-bit accesses. It is the responsibility of external logic to drive $\overline{\text{MEMCS16}}$ signal for the appropriate 128 Kbit segment decoded from the LA[23:17] signals. $\overline{\text{MEMCS16}}$ should be driven when accessing an 8-bit memory resource.*

Checksum Failure

After RESET, the PCnet-ISA+ controller begins reading the EEPROM and storing the information in registers inside PCnet-ISA+ controller. PCnet-ISA+ controller does a checksum on word locations 0-1Ah inclusive and if the byte checksum = 0FFh, then the data read from the EEPROM is considered good. If the checksum is not equal to 0FFh, then the PCnet-ISA+ controller enters what is called software relocatable mode.

In software relocatable mode, the device functions the same as in Plug and Play mode, except that it does not respond to the same initiation key as Plug and Play supports. Instead, a different key is used to bring PCnet-ISA+ controller out of the Wait For Key state. This key is as follows:

```
6B, 35, 9A, CD, E6, F3, 79, BC
5E, AF, 57, 2B, 15, 8A, C5, E2
F1, F8, 7C, 3E, 9F, 4F, 27, 13
09, 84, 42, A1, D0, 68, 34, 1A
```

Use Without EEPROM

In some designs, especially PC motherboard applications, it may be desirable to eliminate the EEPROM altogether. This would save money, space, and power consumption.

The operation of this mode is similar to when the PCnet-ISA+ controller encounters a checksum error, except that to enter this mode the SHFBUSY pin is left unconnected. The device will enter software relocatable mode, and the BIOS on the motherboard can wake up the device, configure it, load the IEEE address (possibly stored in Flash ROM) into the PCnet-ISA+ controller, and activate the device.

External Scan Chain

The External Scan Chain is a set of bits stored in the EEPROM which are not used in the PCnet-ISA+ controller but which can be used with external hardware to allow jumperless configuration of external devices.

After RESET, the PCnet-ISA+ controller begins reading the EEPROM and storing the information in registers inside the PCnet-ISA+ controller. SHFBUSY is held high during the read of the EEPROM. If external circuitry is added, such as a shift register, which is clocked from SCLK and is attached to DO from the EEPROM, data read out of the EEPROM will be shifted into the shift register. After reading the EEPROM to the end of the External Shift Chain, and if there is a correct checksum, SHFBUSY will go low. This will be used to latch the information from the EEPROM into the shift register. If the checksum is invalid, SHFBUSY will not go low, indicating that the EEPROM may be bad.

For more information on the use of this function, please refer to the technical reference manual.

Flash PROM

Use

Instead of using a PROM or EPROM for the Boot PROM, it may be desirable to use a Flash or EEPROM type of device for storing the Boot code. This would allow for in-system updates and changes to the information in the Boot ROM without opening up the PC. It may also be desirable to store statistics or drivers in the Flash device.

Interface

To use a Flash-type device with the PCnet-ISA+ controller, Flash Select is set in register 0F0h of the Plug and Play registers. Flash Select is cleared by RESET (default).

In bus master mode, $\overline{\text{BPCS}}$ becomes $\overline{\text{Flash\_OE}}$ and IRQ12 becomes $\overline{\text{Flash\_WE}}$. The Flash ROM devices $\overline{\text{CS}}$ pin is connected to ground.

In shared memory mode, $\overline{\text{BPCS}}$ becomes $\overline{\text{Flash\_ CS}}$ and IRQ12 becomes the static RAM Chip Select, and the $\overline{\text{SROE}}$ and $\overline{\text{SRWE}}$ signals are connected to both the SRAM and Flash devices.

Optional IEEE Address PROM

Normally, the Ethernet physical address will be stored in the EEPROM with the other configuration data. This reduces the parts count, board space requirements, and power consumption. The option to use a standard parallel 8 bit PROM is provided to manufactures who are concerned about the non-volatile nature of EEPROMs.

To use a 8 bit parallel prom to store the IEEE address data instead of storing it in the EEPROM, the APROM_EN bit is set in the Plug and Play registers by the EEPROM upon RESET. IRQ15 is redefined by the setting of this bit to be $\overline{\text{APCS}}$, or ADDRESS PROM CHIP SELECT. This pin is connected to an external 8 bit PROM, such as a 27LS19. The address pins of the PROM are connected to the lower address pins of the ISA bus, and the data lines are connected to the private data bus.

In this mode, any accesses to the IEEE address will be passed to the external PROM and the data will be passed through the PCnet-ISA+ controller to the system data bus.

EISA Configuration Registers

The PCnet-ISA+ controller has support for the 4-byte EISA Configuration Registers. These are used in EISA systems to identify the card and load the appropriate configuration file for that card. This feature is enabled using bit 10 of ISACSR2. When set to 1, the EISA Configuration registers will be enabled and will be read at I/O location 0xC80-0xC83. The contents of these 4 registers are stored in the EEPROM and are automatically read in at RESET.

Bus Interface Unit (BIU)

The bus interface unit is a mixture of a 20 MHz state machine and asynchronous logic. It handles two types of accesses; accesses where the PCnet-ISA+ controller is a slave and accesses where the PCnet-ISA+ controller is the Current Master.

In slave mode, signals like $\overline{\text{IOCS16}}$ are asserted and deasserted as soon as the appropriate inputs are received. IOCHRDY is asynchronously driven LOW if the PCnet-ISA+ controller needs a wait state. It is released synchronously when the PCnet-ISA+ controller is ready.

When the PCnet-ISA+ controller is the Current Master, all the signals it generates are synchronous to the on-chip 20 MHz clock.

DMA Transfers

The BIU will initiate DMA transfers according to the type of operation being performed. There are three primary types of DMA transfers:

1. Initialization Block DMA Transfers

Once the BIU has been granted bus mastership, it will perform four data transfer cycles (eight bytes) before relinquishing the bus. The four transfers within the mastership period will always be read cycles to contiguous addresses. There are 12 words to transfer so there will be three bus mastership periods.

2. Descriptor DMA Transfers

Once the BIU has been granted bus mastership, it will perform the appropriate number of data transfer cycles before relinquishing the bus. The transfers within the mastership period will always be of the same type (either all read or all write), but may be to non-contiguous addresses. Only the bytes which need to be read or written are accessed.

3. Burst-Cycle DMA Transfers

Once the BIU has been granted bus mastership, it will perform a series of consecutive data transfer cycles before relinquishing the bus. Each data transfer will be performed sequentially, with the issue of the address, and the transfer of the data with appropriate output signals to indicate selection of the active data bytes during the transfer. All transfers within the mastership cycle will be either read or write cycles, and will be to contiguous addresses. The number of data transfer cycles within the burst is dependent on the programming of the DMAPLUS option (CSR4, bit 14).

If DMAPLUS = 0, a maximum of 16 transfers will be performed. This may be changed by writing to the burst register (CSR80), but the default takes the same amount of time as the Am2100 family of LANCE-based boards, a little over 5 µs.

If DMAPLUS = 1, the burst will continue until the FIFO is filled to its high threshold (32 bytes in transmit operation) or emptied to its low threshold (16 bytes in receive operation). The exact number of transfer cycles in this case will be dependent on the latency of the system bus to the BIU's mastership request and the speed of bus operation.

Buffer Management Unit (BMU)

The buffer management unit is a micro-coded 20 MHz state machine which implements the initialization block and the descriptor architecture.

Initialization

PCnet-ISA+ controller initialization includes the reading of the initialization block in memory to obtain the operating parameters. The initialization block is read when the INIT bit in CSR0 is set. The INIT bit should be set before or concurrent with the STRT bit to insure correct operation. Four words at a time are read and the bus is released at the end of each block of reads, for a total of three arbitration cycles. Once the initialization block has been read in and processed, the BMU knows where the receive and transmit descriptor rings are. On completion of the read operation and after internal registers have been updated, IDON will be set in CSR0, and an interrupt generated if IENA is set.

The Initialization Block is vectored by the contents of CSR1 (least significant 16 bits of address) and CSR2 (most significant 8 bits of address). The block contains the user defined conditions for PCnet-ISA+ controller operation, together with the address and length information to allow linkage of the transmit and receive descriptor rings.

There is an alternative method to initialize the PCnet-ISA+ controller. Instead of initialization via the initialization block in memory, data can be written directly into the appropriate registers. Either method may be used at the discretion of the programmer. If the registers are written to directly, the INIT bit must not be set, or the initialization block will be read in, thus overwriting the previously written information. Please refer to Appendix D for details on this alternative method.

Reinitialization

The transmitter and receiver section of the PCnet-ISA+ controller can be turned on via the initialization block (MODE Register DTX, DRX bits; CSR15[1:0]). The state of the transmitter and receiver are monitored through CSR0 (RXON, TXON bits). The PCnet-ISA+ controller should be reinitialized if the transmitter and/or the receiver were not turned on during the original initialization and it was subsequently required to activate them, or if either section shut off due to the detection of an error condition (MERR, UFLO, TX BUFF error).

Reinitialization may be done via the initialization block or by setting the STOP bit in CSR0, followed by writing to CSR15, and then setting the START bit in CSR0. Note that this form of restart will not perform the same in the PCnet-ISA+ controller as in the LANCE. In particular, the PCnet-ISA+ controller reloads the transmit and receive descriptor pointers with their respective base addresses.This means that the software must clear the descriptor's own bits and reset its descriptor ring pointers before the restart of the PCnet-ISA controller. The reload of descriptor base addresses is performed in the LANCE only after initialization, so a restart of the LANCE without initialization leaves the LANCE pointing at the same descriptor locations as before the restart.

Buffer Management

Buffer management is accomplished through message descriptor entries organized as ring structures in memory. There are two rings, a receive ring and a transmit ring. The size of a message descriptor entry is 4 words (8 bytes).

Descriptor Rings

Each descriptor ring must be organized in a contiguous area of memory. At initialization time (setting the INIT bit in CSR0), the PCnet-ISA+ controller reads the user-defined base address for the transmit and receive descriptor rings, which must be on an 8-byte boundary, as well as the number of entries contained in the descriptor rings. By default, a maximum of 128 ring entries is permitted when utilizing the initialization block, which uses values of TLEN and RLEN to specify the transmit

and receive descriptor ring lengths. However, the ring lengths can be manually defined (up to 65535) by writing the transmit and receive ring length registers (CSR76,78) directly.

Each ring entry contains the following information:

- The address of the actual message data buffer in user or host memory
- The length of the message buffer
- Status information indicating the condition of the buffer

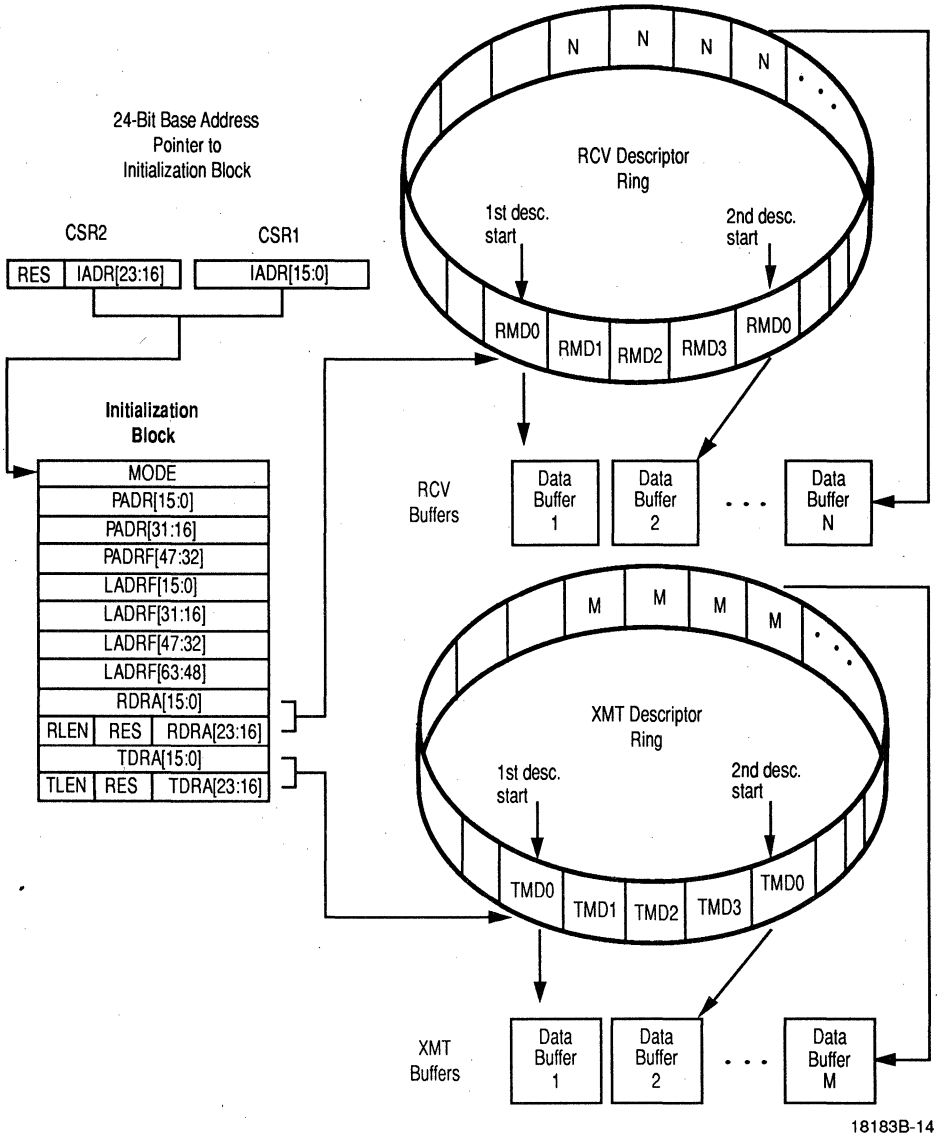
Receive descriptor entries are similar (but not identical) to transmit descriptor entries. Both are composed of four registers, each 16 bits wide for a total of 8 bytes.

To permit the queuing and de-queuing of message buffers, ownership of each buffer is allocated to either the PCnet-ISA+ controller or the host. The OWN bit within the descriptor status information, either TMD or RMD (see section on TMD or RMD), is used for this purpose. "Deadly Embrace" conditions are avoided by the ownership mechanism. Only the owner is permitted to relinquish ownership or to write to any field in the descriptor entry. A device that is not the current owner of a descriptor entry cannot assume ownership or change any field in the entry.

Descriptor Ring Access Mechanism

At initialization, the PCnet-ISA+ controller reads the base address of both the transmit and receive descriptor rings into CSRs for use by the PCnet-ISA+ controller during subsequent operation.

When transmit and receive functions begin, the base address of each ring is loaded into the current descriptor address registers and the address of the next descriptor entry in the transmit and receive rings is computed and loaded into the next descriptor address registers.



Initialization Block and Descriptor Rings

Polling

When there is no channel activity and there is no pre- or post-receive or transmit activity being performed by the PCnet-ISA+ controller then the PCnet-ISA+ controller will periodically poll the current receive and transmit descriptor entries in order to ascertain their ownership. If the DPOLL bit in CSR4 is set, then the transmit polling function is disabled.

A typical polling operation consists of the following: The PCnet-ISA+ controller will use the current receive descriptor address stored internally to vector to the appropriate Receive Descriptor Table Entry (RDTE). It will then use the current transmit descriptor address (stored internally) to vector to the appropriate Transmit Descriptor Table Entry (TDTE). These accesses will be made to RMD1 and RMD0 of the current RDTE and

TMD1 and TMD0 of the current TDTE at periodic polling intervals. All information collected during polling activity will be stored internally in the appropriate CSRs. (i.e. CSR18–19, CSR40, CSR20–21, CSR42, CSR50, CSR52). Unowned descriptor status will be internally ignored.

A typical receive poll occurs under the following conditions:

1) PCnet-ISA+ controller does not possess ownership of the current RDTE and
   the poll time has elapsed and
   RXON = 1,

or

2) PCnet-ISA+ controller does not possess ownership of the next RDTE and
   the poll time has elapsed and
   RXON = 1,

If RXON = 0, the PCnet-ISA+ controller will never poll RDTE locations.

If RXON = 1, the system should always have at least one RDTE available for the possibility of a receive event. When there is only one RDTE, there is no polling for next RDTE.

A typical transmit poll occurs under the following conditions:

1) PCnet-ISA+ controller does not possess ownership of the current TDTE and
   DPOLL = 0 and
   TXON = 1 and
   the poll time has elapsed,

or

2) PCnet-ISA+ controller does not possess ownership of the current TDTE and
   DPOLL = 0 and
   TXON = 1 and
   a packet has just been received,

or

3) PCnet-ISA+ controller does not possess ownership of the current TDTE and
   DPOLL = 0 and
   TXON = 1 and
   a packet has just been transmitted.

The poll time interval is nominally defined as 32,768 crystal clock periods, or 1.6 ms. However, the poll time register is controlled internally by microcode, so any other microcode controlled operation will interrupt the incrementing of the poll count register. For example, when a receive packet is accepted by the PCnet-ISA+ controller, the device suspends execution of the poll-time-incrementing microcode so that a receive microcode routine may instead be executed. Poll-time-incrementing code is resumed when the receive operation has completely finished. Note, however, that following the completion of any receive or transmit operation, a poll operation will always be performed. The poll time count register is never reset. Note that if a non-default is desired, then a strict sequence of setting the INIT bit in CSR0, waiting for the IDON bit in CSR0, then writing to CSR47, and then setting STRT in CSR0 must be observed, otherwise the default value will not be overwritten. See the CSR47 section for details.

Setting the TDMD bit of CSR0 will cause the microcode controller to exit the poll counting code and immediately perform a polling operation. If RDTE ownership has not been previously established, then an RDTE poll will be performed ahead of the TDTE poll.

Transmit Descriptor Table Entry (TDTE)

If, after a TDTE access, the PCnet-ISA+ controller finds that the OWN bit of that TDTE is not set, then the PCnet-ISA+ controller resumes the poll time count and reexamines the same TDTE at the next expiration of the poll time count.

If the OWN bit of the TDTE is set, but STP = 0, the PCnet-ISA+ controller will immediately request the bus in order to reset the OWN bit of this descriptor; this condition would normally be found following a LCOL or RETRY error that occurred in the middle of a transmit packet chain of buffers. After resetting the OWN bit of this descriptor, the PCnet-ISA+ controller will again immediately request the bus in order to access the next TDTE location in the ring.

If the OWN bit is set and the buffer length is 0, the OWN bit will be reset. In the LANCE the buffer length of 0 is interpreted as a 4096-byte buffer. It is acceptable to have a 0 length buffer on transmit with STP = 1 or STP = 1 *and* ENP = 1. It is not acceptable to have 0 length buffer with STP = 0 *and* ENP = 1.

If the OWN bit is set and the start of packet (STP) bit is set, then microcode control proceeds to a routine that will enable transmit data transfers to the FIFO.

If the transmit buffers are data chained (ENP=0 in the first buffer), then the PCnet-ISA+ controller will look ahead to the next transmit descriptor after it has performed at least one transmit data transfer from the first buffer. More than one transmit data transfer may possibly take place, depending upon the state of the transmitter. The transmit descriptor lookahead reads TMD0 first and TMD1 second. The contents of TMD0 and TMD1 will be stored in Next TX Descriptor Address (CSR32), Next TX Byte Count (CSR66) and Next TX Status (CSR67) regardless of the state of the OWN bit. This transmit descriptor lookahead operation is performed only once.

If the PCnet-ISA+ controller does not own the next TDTE (i.e. the second TDTE for this packet), then it will complete transmission of the current buffer and then update the status of the current (first) TDTE with the BUFF and UFLO bits being set. This will cause the transmitter to be disabled (CSR0, TXON=0). The PCnet-ISA+ controller will have to be restarted to restore the transmit function. The situation that matches this description implies that the system has not been able to stay ahead of the

PCnet-ISA+ controller in the transmit descriptor ring and therefore, the condition is treated as a fatal error. To avoid this situation, the system should always set the transmit chain descriptor own bits in reverse order.

If the PCnet-ISA+ controller does own the second TDTE in a chain, it will gradually empty the contents of the first buffer (as the bytes are needed by the transmit operation), perform a single-cycle DMA transfer to update the status (reset the OWN bit in TMD1) of the first descriptor, and then it may perform one data DMA access on the second buffer in the chain before executing another lookahead operation. (i.e. a lookahead to the third descriptor.)

The PCnet-ISA+ controller can queue up to two packets in the transmit FIFO. Call them packet "X" and packet "Y", where "Y" is after "X". Assume that packet "X" is currently being transmitted. Because the PCnet-ISA+ controller can perform lookahead data transfer over an ENP, it is possible for the PCnet-ISA+ controller to update a TDTE in a buffer belonging to packet "Y" while packet "X" is being transmitted if packet "Y" uses data chaining. This operation will result in non-sequential TDTE accesses as packet "X" completes transmission and the PCnet-ISA+ controller writes out its status, since packet "X"'s TDTE is before the TDTE accessed as part of the lookahead data transfer from packet "Y".

This should not cause any problem for properly written software which processes buffers in sequence, waiting for ownership before proceeding.

If an error occurs in the transmission before all of the bytes of the current buffer have been transferred, then TMD2 and TMD1 of the current buffer will be written; in that case, data transfers from the next buffer will not commence. Instead, following the TMD2/TMD1 update, the PCnet-ISA+ controller will go to the next transmit packet, if any, skipping over the rest of the packet which experienced an error, including chained buffers.

This is done by returning to the polling microcode where it will immediately access the next descriptor and find the condition OWN = 1 and STP = 0 as described earlier. In that case, the PCnet-ISA+ controller will reset the own bit for this descriptor and continue in like manner until a descriptor with OWN=0 (no more transmit packets in the ring) or OWN = 1 and STP = 1 (the first buffer of a new packet) is reached.

At the end of any transmit operation, whether successful or with errors, and the completion of the descriptor updates, the PCnet-ISA+ controller will always perform another poll operation. As described earlier, this poll operation will begin with a check of the current RDTE, unless the PCnet-ISA+ controller already owns that descriptor. Then the PCnet-ISA+ controller will proceed to polling the next TDTE. If the transmit descriptor OWN bit has a zero value, then the PCnet-ISA+ controller will resume poll time count incrementation. If the transmit descriptor OWN bit has a value of ONE, then the PCnet-ISA+ controller will begin filling the FIFO with transmit data and initiate a transmission. This end-of-operation poll avoids inserting poll time counts between successive transmit packets.

Whenever the PCnet-ISA+ controller completes a transmit packet (either with or without error) and writes the status information to the current descriptor, then the TINT bit of CSR0 is set to indicate the completion of a transmission. This causes an interrupt signal if the IENA bit of CSR0 has been set and the TINTM bit of CSR3 is reset.

Receive Descriptor Table Entry (RDTE)

If the PCnet-ISA+ controller does not own both the current and the next Receive Descriptor Table Entry, then the PCnet-ISA+ controller will continue to poll according to the polling sequence described above. If the receive descriptor ring length is 1, there is no next descriptor, and no look ahead poll will take place.

If a poll operation has revealed that the current and the next RDTE belongs to the PCnet-ISA+ controller, then additional poll accesses are not necessary. Future poll operations will not include RDTE accesses as long as the PCnet-ISA+ controller retains ownership to the current and the next RDTE.

When receive activity is present on the channel, the PCnet-ISA+ controller waits for the complete address of the message to arrive. It then decides whether to accept or reject the packet based on all active addressing schemes. If the packet is accepted the PCnet-ISA+ controller checks the current receive buffer status register CRST (CSR40) to determine the ownership of the current buffer.

If ownership is lacking, then the PCnet-ISA+ controller will immediately perform a (last ditch) poll of the current RDTE. If ownership is still denied, then the PCnet-ISA+ controller has no buffer in which to store the incoming message. The MISS bit will be set in CSR0 and an interrupt will be generated if IENA = 1 (CSR0) and MISSM = 0 (CSR3). Another poll of the current RDTE will not occur until the packet has finished.

If the PCnet-ISA+ controller sees that the last poll (either a normal poll or the last-ditch effort described in the above paragraph) of the current RDTE shows valid ownership, then it proceeds to a poll of the next RDTE. Following this poll, and regardless of the outcome of this poll, transfers of receive data from the FIFO may begin.

Regardless of ownership of the second receive descriptor, the PCnet-ISA+ controller will continue to perform receive data DMA transfers to the first buffer, using burst-cycle DMA transfers. If the packet length exceeds the length of the first buffer, and the PCnet-ISA+ controller does not own the second buffer, ownership of the current descriptor will be passed back to the system by writing a zero to the OWN bit of RMD1 and status will be written indicating buffer (BUFF = 1) and possibly overflow (OFLO = 1) errors.

If the packet length exceeds the length of the first (current) buffer, and the PCnet-ISA+ controller does own the

second (next) buffer, ownership will be passed back to the system by writing a zero to the OWN bit of RMD1 when the first buffer is full. Receive data transfers to the second buffer may occur before the PCnet-ISA+ controller proceeds to look ahead to the ownership of the third buffer. Such action will depend upon the state of the FIFO when the status has been updated on the first descriptor. In any case, lookahead will be performed to the third buffer and the information gathered will be stored in the chip, regardless of the state of the ownership bit. As in the transmit flow, lookahead operations are performed only once.

This activity continues until the PCnet-ISA+ controller recognizes the completion of the packet (the last byte of this receive message has been removed from the FIFO). The PCnet-ISA+ controller will subsequently update the current RDTE status with the end of packet (ENP) indication set, write the message byte count (MCNT) of the complete packet into RMD2 and overwrite the "current" entries in the CSRs with the "next" entries.

Media Access Control

The Media Access Control engine incorporates the essential protocol requirements for operation of a compliant Ethernet/802.3 node, and provides the interface between the FIFO sub-system and the Manchester Encoder/Decoder (MENDEC).

The MAC engine is fully compliant to Section 4 of ISO/IEC 8802-3 (ANSI/IEEE Standard 1990 Second Edition) and ANSI/IEEE 802.3 (1985).

The MAC engine provides programmable enhanced features designed to minimize host supervision and pre or post-message processing. These features include the ability to disable retries after a collision, dynamic FCS generation on a packet-by-packet basis, and automatic pad field insertion and deletion to enforce minimum frame size attributes.

The two primary attributes of the MAC engine are:

- Transmit and receive message data encapsulation
  - — Framing (frame boundary delimitation, frame synchronization)
  - — Addressing (source and destination address handling)
  - — Error detection (physical medium transmission errors)
- Media access management
  - — Medium allocation (collision avoidance)
  - — Contention resolution (collision handling)

Transmit And Receive Message Data Encapsulation

The MAC engine provides minimum frame size enforcement for transmit and receive packets. When APAD_XMT = 1 (bit 11 in CSR4), transmit messages will be padded with sufficient bytes (containing 00h) to ensure that the receiving station will observe an information field (destination address, source address, length/type, data and FCS) of 64-bytes. When ASTRP_RCV = 1 (bit 10 in CSR4), the receiver will automatically strip pad bytes from the received message by observing the value in the length field, and stripping excess bytes if this value is below the minimum data size (46 bytes). Both features can be independently overridden to allow illegally short (less than 64 bytes of packet data) messages to be transmitted and/or received. The use of these features reduce bus bandwidth usage because the pad bytes are not transferred to or from host memory.

Framing (frame boundary delimitation, frame synchronization)

The MAC engine will autonomously handle the construction of the transmit frame. Once the Transmit FIFO has been filled to the predetermined threshold (set by XMTSP in CSR80), and providing access to the channel is currently permitted, the MAC engine will commence the 7-byte preamble sequence (10101010b, where first bit transmitted is a 1). The MAC engine will subsequently append the Start Frame Delimiter (SFD) byte (10101011b) followed by the serialized data from the Transmit FIFO. Once the data has been completed, the MAC engine will append the FCS (most significant bit first) which was computed on the entire data portion of the message.

Note that the user is responsible for the correct ordering and content in each of the fields in the frame, including the destination address, source address, length/type and packet data.

The receive section of the MAC engine will detect an incoming preamble sequence and lock to the encoded clock. The internal MENDEC will decode the serial bit stream and present this to the MAC engine. The MAC will discard the first 8 bits of information before searching for the SFD sequence. Once the SFD is detected, all subsequent bits are treated as part of the frame. The MAC engine will inspect the length field to ensure minimum frame size, strip unnecessary pad characters (if enabled), and pass the remaining bytes through the Receive FIFO to the host. If pad stripping is performed, the MAC engine will also strip the received FCS bytes, although the normal FCS computation and checking will occur. Note that apart from pad stripping, the frame will be passed unmodified to the host. If the length field has a value of 46 or greater, the MAC engine will not attempt to validate the length against the number of bytes contained in the message.

If the frame terminates or suffers a collision before 64 bytes of information (after SFD) have been received, the MAC engine will automatically delete the frame from the Receive FIFO, without host intervention.

Addressing (source and destination address handling)

The first 6 bytes of information after SFD will be interpreted as the destination address field. The MAC engine provides facilities for physical, logical, and broadcast address reception. In addition, multiple physical addresses can be constructed (perfect address filtering) using external logic in conjunction with the EADI™ interface.

Error detection (physical medium transmission errors).

The MAC engine provides several facilities which report and recover from errors on the medium. In addition, the network is protected from gross errors due to inability of the host to keep pace with the MAC engine activity.

On completion of transmission, the following transmit status is available in the appropriate TMD and CSR areas:

- The exact number of transmission retry attempts (ONE, MORE, or RTRY).
- Whether the MAC engine had to Defer (DEF) due to channel activity.
- Loss of Carrier, indicating that there was an interruption in the ability of the MAC engine to monitor its own transmission. Repeated LCAR errors indicate a potentially faulty transceiver or network connection.
- Late Collision (LCOL) indicates that the transmission suffered a collision after the slot time. This is indicative of a badly configured network. Late collisions should not occur in a normal operating network.
- Collision Error (CERR) indicates that the transceiver did not respond with an SQE Test message within the predetermined time after a transmission completed. This may be due to a failed transceiver, disconnected or faulty transceiver drop cable, or the fact the transceiver does not support this feature (or the feature is disabled).

In addition to the reporting of network errors, the MAC engine will also attempt to prevent the creation of any network error due to the inability of the host to service the MAC engine. During transmission, if the host fails to keep the Transmit FIFO filled sufficiently, causing an underflow, the MAC engine will guarantee the message is either sent as a runt packet (which will be deleted by the receiving station) or has an invalid FCS (which will also cause the receiver to reject the message).

The status of each receive message is available in the appropriate RMD and CSR areas. FCS and Framing errors (FRAM) are reported, although the received frame is still passed to the host. The FRAM error will only be reported if an FCS error is detected and there are a non-integral number of bits in the message. The MAC engine will ignore up to seven additional bits at the end of a message (dribbling bits), which can occur under normal network operating conditions. The reception of eight additional bits will cause the MAC engine to de-serialize the entire byte, and will result in the received message and FCS being modified.

The PCnet-ISA+ controller can handle up to 7 dribbling bits when a received packet terminates. During the reception, the CRC is generated on every serial bit (including the dribbling bits) coming from the cable, although the internally saved CRC value is only updated on the eighth bit (on each byte boundary). The framing error is reported to the user as follows:

1. If the number of the dribbling bits are 1 to 7 and there is no CRC error, then there is no Framing error (FRAM = 0).
2. If the number of the dribbling bits are less than 8 and there is a CRC error, then there is also a Framing error (FRAM = 1).
3. If the number of dribbling bits = 0, then there is no Framing error. There may or may not be a CRC (FCS) error.

Counters are provided to report the Receive Collision Count and Runt Packet Count used for network statistics and utilization calculations.

Note that if the MAC engine detects a received packet which has a 00b pattern in the preamble (after the first 8 bits, which are ignored), the entire packet will be ignored. The MAC engine will wait for the network to go inactive before attempting to receive the next packet.

Media Access Management

The basic requirement for all stations on the network is to provide fairness of channel allocation. The 802.3/Ethernet protocol defines a media access mechanism which permits all stations to access the channel with equality. Any node can attempt to contend for the channel by waiting for a predetermined time (Inter Packet Gap interval) after the last activity, before transmitting on the medium. The channel is a multidrop communications medium (with various topological configurations permitted) which allows a single station to transmit and all other stations to receive. If two nodes simultaneously contend for the channel, their signals will interact, causing loss of data (defined as a collision). It is the responsibility of the MAC to attempt to avoid and recover from a collision, to guarantee data integrity for the end-to-end transmission to the receiving station.

Medium allocation (collision avoidance)

The IEEE 802.3 Standard (ISO/IEC 8802-3 1990) requires that the CSMA/CD MAC monitor the medium traffic by looking for carrier activity. When carrier is detected the medium is considered busy, and the MAC should defer to the existing message.

The IEEE 802.3 Standard also allows optional two part deferral after a receive message.

See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.1:

*"**Note**: It is possible for the PLS carrier sense indication to fail to be asserted during a collision on the media. If the deference process simply times the interpacket gap based on this indication it is possible for a short interFrame gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness the following optional measures, as specified in 4.2.8, are recommended when InterFrameSpacingPart1 is other than zero:*

(1) Upon completing a transmission, start timing the interpacket gap, as soon as transmitting and carrierSense are both false.

(2) When timing an interpacket gap following reception, reset the interpacket gap timing if carrier Sense becomes true during the first 2/3 of the interpacket gap timing interval. During the final 1/3 of the interval the timer shall not be reset to ensure fair access to the medium. An initial period shorter than 2/3 of the interval is permissible including zero."

The MAC engine implements the optional receive two part deferral algorithm, with a first part inter-frame-spacing time of 6.0 µs. The second part of the inter-frame-spacing interval is therefore 3.6 µs.

The PCnet-ISA+ controller will perform the two-part deferral algorithm as specified in Section 4.2.8 (Process Deference). The Inter Packet Gap (IPG) timer will start timing the 9.6 µs InterFrameSpacing after the receive carrier is de-asserted. During the first part deferral (InterFrameSpacingPart1 - IFS1) the PCnet-ISA+ controller will defer any pending transmit frame and respond to the receive message. The IPG counter will be reset to zero continuously until the carrier de-asserts, at which point the IPG counter will resume the 9.6 µs count once again. Once the IFS1 period of 6.0 µs has elapsed, the PCnet-ISA+ controller will begin timing the second part deferral (InterFrameSpacingPart2 - IFS2) of 3.6 µs. Once IFS1 has completed, and IFS2 has commenced, the PCnet-ISA+ controller will not defer to a receive packet if a transmit packet is pending. This means that the PCnet-ISA+ controller will not attempt to receive the receive packet, since it will start to transmit, and generate a collision at 9.6 µs. The PCnet-ISA+ controller will guarantee to complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

In addition, transmit two part deferral is implemented as an option which can be disabled using the DXMT2PD bit (CSR3). Two-part deferral after transmission is useful for ensuring that severe IPG shrinkage cannot occur in specific circumstances, causing a transmit message to follow a receive message so closely as to make them indistinguishable.

During the time period immediately after a transmission has been completed, the external transceiver (in the case of a standard AUI connected device), should generate the SQE Test message (a nominal 10 MHz burst of 5-15 bit times duration) on the CI± pair (within 0.6 µs – 1.6 µs after the transmission ceases). During the time period in which the SQE Test message is expected the PCnet-ISA+ controller will not respond to receive carrier sense.

See ANSI/IEEE Std 802.3-1990 Edition, 7.2.4.6 (1)):

"At the conclusion of the output function, the DTE opens a time window during which it expects to see the signal_quality_error *signal asserted on the Control In circuit. The time window begins when the CARRIER_STATUS becomes CARRIER_OFF. If execution of the output function does not cause CARRIER_ON to occur, no SQE test occurs in the DTE. The duration of the window shall be at least 4.0 µs but no more than 8.0 µs. During the time window the Carrier Sense Function is inhibited."*

The PCnet-ISA+ controller implements a carrier sense "blinding" period within 0 - 4.0 µs from de-assertion of carrier sense after transmission. This effectively means that when transmit two part deferral is enabled (DXMT2PD is cleared) the IFS1 time is from 4 µs to 6 µs after a transmission. However, since IPG shrinkage below 4 µs will rarely be encountered on a correctly configured network, and since the fragment size will be larger than the 4 µs blinding window, then the IPG counter will be reset by a worst case IPG shrinkage/fragment scenario and the PCnet-ISA+ controller will defer its transmission. In addition, the PCnet-ISA+ controller will not restart the "blinding" period if carrier is detected within the 4.0 µs – 6.0 µs IFS1 period, but will commence timing of the entire IFS1 period.

Contention resolution (collision handling)

Collision detection is performed and reported to the MAC engine by the integrated Manchester Encoder/Decoder (MENDEC).

If a collision is detected before the complete preamble/SFD sequence has been transmitted, the MAC Engine will complete the preamble/SFD before appending the jam sequence. If a collision is detected after the preamble/SFD has been completed, but prior to 512 bits being transmitted, the MAC Engine will abort the transmission, and append the jam sequence immediately. The jam sequence is a 32-bit all zeroes pattern.

The MAC Engine will attempt to transmit a frame a total of 16 times (initial attempt plus 15 retries) due to normal collisions (those within the slot time). Detection of collision will cause the transmission to be re-scheduled, dependent on the backoff time that the MAC Engine computes. If a single retry was required, the ONE bit will be set in the Transmit Frame Status (TMD1 in the Transmit Descriptor Ring). If more than one retry was

required, the MORE bit will be set. If all 16 attempts experienced collisions, the RTRY bit (in TMD2) will be set (ONE and MORE will be clear), and the transmit message will be flushed from the FIFO. If retries have been disabled by setting the DRTY bit in the MODE register (CSR15), the MAC Engine will abandon transmission of the frame on detection of the first collision. In this case, only the RTRY bit will be set and the transmit message will be flushed from the FIFO.

If a collision is detected after 512 bit times have been transmitted, the collision is termed a late collision. The MAC Engine will abort the transmission, append the jam sequence, and set the LCOL bit. No retry attempt will be scheduled on detection of a late collision, and the FIFO will be flushed.

The IEEE 802.3 Standard requires use of a "truncated binary exponential backoff" algorithm which provides a controlled pseudo-random mechanism to enforce the collision backoff interval, before re-transmission is attempted.

See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.5:

"At the end of enforcing a collision (jamming), the CSMA/CD sublayer delays before attempting to re-transmit the frame. The delay is an integer multiple of slotTime. The number of slot times to delay before the nth re-transmission attempt is chosen as a uniformly distributed random integer r in the range:

$0 \leq r < 2^k$, where k = min (n,10)."

The PCnet-ISA+ controller provides an alternative algorithm, which suspends the counting of the slot time/IPG during the time that receive carrier sense is detected. This algorithm aids in networks where large numbers of nodes are present, and numerous nodes can be in collision. The algorithm effectively accelerates the increase in the backoff time in busy networks, and allows nodes not involved in the collision to access the channel while the colliding nodes await a reduction in channel activity. Once channel activity is reduced, the nodes resolving the collision time out their slot time counters as normal.

Manchester Encoder/Decoder (MENDEC)

The integrated Manchester Encoder/Decoder provides the PLS (Physical Layer Signaling) functions required for a fully compliant IEEE 802.3 station. The MENDEC provides the encoding function for data to be transmitted on the network using the high accuracy on-board oscillator, driven by either the crystal oscillator or an external CMOS-level compatible clock. The MENDEC also provides the decoding function from data received from the network. The MENDEC contains a Power On Reset (POR) circuit, which ensures that all analog portions of the PCnet-ISA+ controller are forced into their correct state during power-up, and prevents erroneous data transmission and/or reception during this time.

External Crystal Characteristics

When using a crystal to drive the oscillator, the crystal specification shown in the specification table may be used to ensure less than ±0.5 ns jitter at DO±.

External Crystal Characteristics

Parameter	Min	Nom	Max	Unit
1. Parallel Resonant Frequency		20		MHz
2. Resonant Frequency Error (CL = 20 pF)	–50		+50	PPM
3. Change in Resonant Frequency With Respect To Temperature (0° – 70° C; CL = 20 pF)*	–40		+40	PPM
4. Crystal Capacitance			20	pF
5. Motional Crystal Capacitance (C1)		0.022		pF
6. Series Resistance			25	Ω
7. Shunt Capacitance			7	pF
8. Drive Level			TBD	mW

** Requires trimming crystal spec; no trim is 50 ppm total*

External Clock Drive Characteristics

When driving the oscillator from an external clock source, XTAL2 must be left floating (unconnected). An external clock having the following characteristics must be used to ensure less than ±0.5 ns jitter at DO±.

Clock Frequency:	20 MHz ±0.01%
Rise/Fall Time (tR/tF):	< 6 ns from 0.5 V to V_{DD}–0.5
XTAL1 HIGH/LOW Time (tHIGH/tLOW):	40 – 60% duty cycle
XTAL1 Falling Edge to Falling Edge Jitter:	< ±0.2 ns at 2.5 V input (V_{DD}/2)

MENDEC Transmit Path

The transmit section encodes separate clock and NRZ data input signals into a standard Manchester encoded serial bit stream. The transmit outputs (DO±) are designed to operate into terminated transmission lines. When operating into a 78 Ω terminated transmission line, the transmit signaling meets the required output levels and skew for Cheapernet, Ethernet, and IEEE-802.3.

Transmitter Timing and Operation

A 20 MHz fundamental-mode crystal oscillator provides the basic timing reference for the MENDEC portion of the PCnet-ISA+ controller. The crystal input is divided by two to create the internal transmit clock reference. Both clocks are fed into the Manchester Encoder to generate the transitions in the encoded data stream. The internal transmit clock is used by the MENDEC to internally synchronize the Internal Transmit Data (ITXDAT) from the

controller and Internal Transmit Enable (ITXEN). The internal transmit clock is also used as a stable bit-rate clock by the receive section of the MENDEC and controller.

The oscillator requires an external 0.005% crystal, or an external 0.01% CMOS-level input as a reference. The accuracy requirements, if an external crystal is used, are tighter because allowance for the on-chip oscillator must be made to deliver a final accuracy of 0.01%.

Transmission is enabled by the controller. As long as the ITXEN request remains active, the serial output of the controller will be Manchester encoded and appear at DO±. When the internal request is dropped by the controller, the differential transmit outputs go to one of two idle states, dependent on TSEL in the Mode Register (CSR15, bit 9):

TSEL LOW:	The idle state of DO± yields "zero" differential to operate transformer-coupled loads.
TSEL HIGH:	In this idle state, DO+ is positive with respect to DO– (logical HIGH).

Receive Path

The principal functions of the receiver are to signal the PCnet-ISA+ controller that there is information on the receive pair, and to separate the incoming Manchester encoded data stream into clock and NRZ data.

The receiver section (see Receiver Block Diagram) consists of two parallel paths. The receive data path is a zero threshold, wide bandwidth line receiver. The carrier path is an offset threshold bandpass detecting line receiver. Both receivers share common bias networks to allow operation over a wide input common mode range.

Input Signal Conditioning

Transient noise pulses at the input data stream are rejected by the Noise Rejection Filter. Pulse width rejection is proportional to transmit data rate which is fixed at 10 MHz for Ethernet systems but which could be different for proprietary networks. DC inputs more negative than minus 100 mV are also suppressed.

The Carrier Detection circuitry detects the presence of an incoming data packet by discerning and rejecting noise from expected Manchester data, and controls the stop and start of the phase-lock loop during clock acquisition. Clock acquisition requires a valid Manchester bit pattern of 1010b to lock onto the incoming message.

When input amplitude and pulse width conditions are met at DI±, a clock acquisition cycle is initiated.

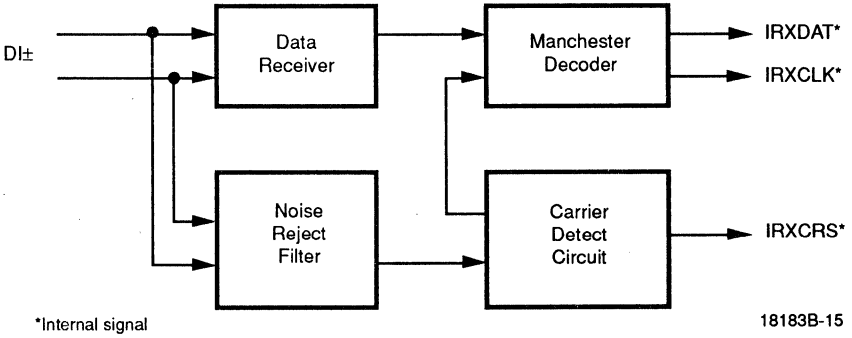
Clock Acquisition

When there is no activity at DI± (receiver is idle), the receive oscillator is phase-locked to STDCLK. The first negative clock transition (bit cell center of first valid Manchester "0") after clock acquisition begins interrupts the receive oscillator. The oscillator is then restarted at the second Manchester "0" (bit time 4) and is phase-locked to it. As a result, the MENDEC acquires the clock from the incoming Manchester bit pattern in 4 bit times with a "1010" Manchester bit pattern.

The internal receiver clock, IRXCLK, and the internal received data, IRXDAT, are enabled 1/4 bit time after clock acquisition in bit cell 5. IRXDAT is at a HIGH state when the receiver is idle (no IRXCLK). IRXDAT however, is undefined when clock is acquired and may remain HIGH or change to LOW state whenever IRXCLK is enabled. At 1/4 bit time through bit cell 5, the controller portion of the PCnet-ISA+ controller sees the first IRXCLK transition. This also strobes in the incoming fifth bit to the MENDEC as Manchester "1". IRXDAT may make a transition after the IRXCLK rising edge in bit cell 5, but its state is still undefined. The Manchester "1" at bit 5 is clocked to IRXDAT output at 1/4 bit time in bit cell 6.

PLL Tracking

After clock acquisition, the phase-locked clock is compared to the incoming transition at the bit cell center (BCC) and the resulting phase error is applied to a correction circuit. This circuit ensures that the phase-locked clock remains locked on the received signal. Individual bit cell phase corrections of the Voltage Controlled Oscillator (VCO) are limited to 10% of the phase difference between BCC and phase-locked clock.



Receiver Block Diagram

Carrier Tracking and End of Message

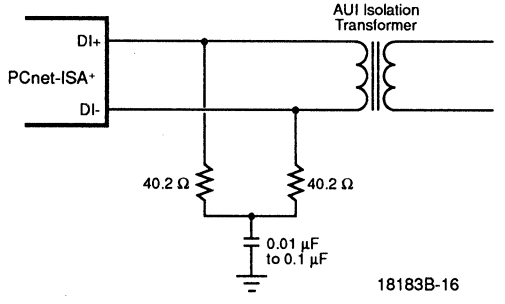
The carrier detection circuit monitors the DI± inputs after IRXCRS is asserted for an end of message. IRXCRS de-asserts 1 to 2 bit times after the last positive transition on the incoming message. This initiates the end of reception cycle. The time delay from the last rising edge of the message to IRXCRS deassert allows the last bit to be strobed by IRXCLK and transferred to the controller section, but prevents any extra bit(s) at the end of message. When IRXCRS de-asserts an IRXCRS hold off timer inhibits IRXCRS assertion for at least 2 bit times.

Data Decoding

The data receiver is a comparator with clocked output to minimize noise sensitivity to the DI± inputs. Input error is less than ± 35 mV to minimize sensitivity to input rise and fall time. IRXCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit, and clocks the data out on IRXDAT on the following IRXCLK. The data receiver also generates the signal used for phase detector comparison to the internal MENDEC voltage controlled oscillator (VCO).

Differential Input Terminations

The differential input for the Manchester data (DI±) should be externally terminated by two 40.2 Ω ±1% resistors and one optional common-mode bypass capacitor, as shown in the Differential Input Termination diagram below. The differential input impedance, ZIDF, and the common-mode input impedance, ZICM, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. If SIP devices are used, 39 Ω is the nearest usable equivalent value. The CI± differential inputs are terminated in exactly the same way as the DI± pair.



Differential Input Termination

Collision Detection

A MAU detects the collision condition on the network and generates a differential signal at the CI± inputs. This collision signal passes through an input stage which detects signal levels and pulse duration. When the signal is detected by the MENDEC it sets the internal collision signal, ICLSN, HIGH. The condition continues for approximately 1.5 bit times after the last LOW-to-HIGH transition on CI±.

Jitter Tolerance Definition

The MENDEC utilizes a clock capture circuit to align its internal data strobe with an incoming bit stream. The clock acquisition circuitry requires four valid bits with the values 1010b. Clock is phase-locked to the negative transition at the bit cell center of the second "0" in the pattern.

Since data is strobed at 1/4 bit time, Manchester transitions which shift from their nominal placement through 1/4 bit time will result in improperly decoded data. With this as the criteria for an error, a definition of "Jitter Handling" is:

> The peak deviation approaching or crossing 1/4 bit cell position from nominal input transition, for which the MENDEC section will properly decode data.

Attachment Unit Interface (AUI)

The AUI is the PLS (Physical Layer Signaling) to PMA (Physical Medium Attachment) interface which connects the DTE to a MAU. The differential interface provided by the PCnet-ISA+ controller is fully compliant with Section 7 of ISO 8802-3 (ANSI/IEEE 802.3).

After the PCnet-ISA+ controller initiates a transmission, it will expect to see data "looped-back" on the DI± pair (when the AUI port is selected). This will internally generate a "carrier sense", indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted within sometime before end of transmission. If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in the Transmit Descriptor Ring (TMD3, bit 11) after the packet has been transmitted.

Twisted Pair Transceiver (T-MAU)

The T-MAU implements the Medium Attachment Unit (MAU) functions for the Twisted Pair Medium, as specified by the supplement to IEEE 802.3 standard (Type 10BASE-T). The T-MAU provides twisted pair driver and receiver circuits, including on-board transmit digital predistortion and receiver squelch, and a number of additional features including Link Status indication, Automatic Twisted Pair Receive Polarity Detection/ Correction and Indication, Receive Carrier Sense, Transmit Active and Collision Present indication.

Twisted Pair Transmit Function

The differential driver circuitry in the TXD± and TXP± pins provides the necessary electrical driving capability and the pre-distortion control for transmitting signals over maximum length Twisted Pair cable, as specified by the 10BASE-T supplement to the IEEE 802.3 Standard. The transmit function for data output meets the propagation delays and jitter specified by the standard.

Twisted Pair Receive Function

The receiver complies with the receiver specifications of the IEEE 802.3 10BASE-T Standard, including noise immunity and received signal rejection criteria ('Smart Squelch'). Signals meeting these criteria appearing at the RXD± differential input pair are routed to the MENDEC. The receiver function meets the propagation delays and jitter requirements specified by the standard. The receiver squelch level drops to half its threshold value after unsquelch to allow reception of minimum amplitude signals and to offset carrier fade in the event of worst case signal attenuation conditions.

Note that the 10BASE-T Standard defines the receive input amplitude at the external Media Dependent Interface (MDI). Filter and transformer loss are not specified. The T-MAU receiver squelch levels are designed to account for a 1 dB insertion loss at 10 MHz for the type of receive filters and transformers usually used.

Normal 10BASE-T compatible receive thresholds are invoked when the LRT bit (CSR15, bit 9) is LOW. When the LRT bit is set, the Low Receive Threshold option is invoked, and the sensitivity of the T-MAU receiver is increased. Increasing T-MAU sensitivity allows the use of lines longer than the 100 m target distance of standard 10BASE-T (assuming typical 24 AWG cable). Increased receiver sensitivity compensates for the increased signal attenuation caused by the additional cable distance.

However, making the receiver more sensitive means that it is also more susceptible to extraneous noise, primarily caused by coupling from co-resident services (crosstalk). For this reason, end users may wish to invoke the Low Receive Threshold option on 4-pair cable only. Multi-pair cables within the same outer sheath have lower crosstalk attenuation, and may allow noise emitted from adjacent pairs to couple into the receive pair, and be of sufficient amplitude to falsely unsquelch the T-MAU.

Link Test Function

The link test function is implemented as specified by 10BASE-T standard. During periods of transmit pair inactivity, 'Link beat pulses' will be periodically sent over the twisted pair medium to constantly monitor medium integrity.

When the link test function is enabled (DLNKTST bit in CSR15 is cleared), the absence of link beat pulses and receive data on the RXD± pair will cause the TMAU to go into the Link Fail state. In the Link Fail state, data transmission, data reception, data loopback and the collision detection functions are disabled and remain disabled until valid data or greater than 5 consecutive link pulses appear on the RXD± pair. During Link Fail, the Link Status (LNKST indicated by $\overline{\text{LED0}}$) signal is inactive. When the link is identified as functional, the LNKST signal is asserted, and $\overline{\text{LED0}}$ output will be activated.

In order to inter-operate with systems which do not implement Link Test, this function can be disabled by setting the DLNKTST bit. With Link Test disabled, the Data Driver, Receiver and Loopback functions as well as Collision Detection remain enabled irrespective of the presence or absence of data or link pulses on the RXD± pair. Link Test pulses continue to be sent regardless of the state of the DLNKTST bit.

Polarity Detection and Reversal

The T-MAU receive function includes the ability to invert the polarity of the signals appearing at the RXD± pair if the polarity of the received signal is reversed (such as in the case of a wiring error). This feature allows data packets received from a reverse wired RXD± input pair to be corrected in the T-MAU prior to transfer to the MENDEC. The polarity detection function is activated following reset or Link Fail, and will reverse the receive polarity based on both the polarity of any previous link beat pulses and the polarity of subsequent packets with a valid End Transmit Delimiter (ETD).

When in the Link Fail state, the T-MAU will recognize link beat pulses of either positive or negative polarity. Exit from the Link Fail state occurs at the reception of 5–6 consecutive link beat pulses of identical polarity. On entry to the Link Pass state, the polarity of the last 5 link beat pulses is used to determine the initial receive polarity configuration and the receiver is reconfigured to subsequently recognize only link beat pulses of the previously recognized polarity.

Positive link beat pulses are defined as transmitted signal with a positive amplitude greater than 585 mV with a pulse width of 60 ns–200 ns. This positive excursion may be followed by a negative excursion. This definition is consistent with the expected received signal at a correctly wired receiver, when a link beat pulse, which fits the template of Figure 14-12 of the 10BASE-T Standard, is generated at a transmitter and passed through 100 m of twisted pair cable.

Negative link beat pulses are defined as transmitted signals with a negative amplitude greater than 585 mV with a pulse width of 60 ns–200 ns. This negative excursion may be followed by a positive excursion. This definition is consistent with the expected received signal at a reverse wired receiver, when a link beat pulse which fits the template of Figure 14-12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

The polarity detection/correction algorithm will remain "armed" until two consecutive packets with valid ETD of identical polarity are detected. When "armed," the receiver is capable of changing the initial or previous polarity configuration according to the detected ETD polarity.

On receipt of the first packet with valid ETD following reset or link fail, the T-MAU will use the inferred polarity information to configure its RXD± input, regardless of its previous state. On receipt of a second packet with a valid ETD with correct polarity, the detection/correction algorithm will "lock-in" the received polarity. If the second (or subsequent) packet is not detected as confirming the previous polarity decision, the most recently detected ETD polarity will be used as the default. Note that packets with invalid ETD have no effect on updating the previous polarity decision. Once two consecutive packets with valid ETD have been received, the T-MAU will lock the correction algorithm until either a Link Fail condition occurs or RESET is asserted.

During polarity reversal, an internal POL signal will be active. During normal polarity conditions, this internal POL signal is inactive. The state of this signal can be read by software and/or displayed by LED when enabled by the LED control bits in the ISA Bus Configuration Registers (ISACSR5, 6, 7).

Twisted Pair Interface Status

Three internal signals (XMT, RCV and COL) indicate whether the T-MAU is transmitting, receiving, or in a collision state. These signals are internal signals and the behavior of the LED outputs depends on how the LED output circuitry is programmed.

The T-MAU will power up in the Link Fail state and the normal algorithm will apply to allow it to enter the Link Pass state. In the Link Pass state, transmit or receive activity will be indicated by assertion of RCV signal going active. If T-MAU is selected using the PORTSEL bits in CSR15, when moving from AUI to T-MAU selection, the T-MAU will be forced into the Link Fail state.

In the Link Fail state, XMT, RCV and COL are inactive.

Collision Detect Function

Activity on both twisted pair signals RXD± and TXD± constitutes a collision, thereby causing the COL signal to be asserted. (COL is used by the LED control circuits) COL will remain asserted until one of the two colliding signals changes from active to idle. COL stays active for 2 bit times at the end of a collision.

Signal Quality Error (SQE) Test (Heartbeat) Function

The SQE function is disabled when the 10BASE-T port is selected and in Link Fail state.

Jabber Function

The Jabber function inhibits the twisted pair transmit function of the T-MAU if theTXD± circuit is active for an excessive period (20 ms–150 ms). This prevents any one node from disrupting the network due to a 'stuck-on' or faulty transmitter. If this maximum transmit time is exceeded, the T-MAU transmitter circuitry is disabled, the JAB bit is set (CSR4, bit 1), and the COL signal asserted. Once the transmit data stream to the T-MAU is removed, an "unjab" time of 250 ms– 750 ms will elapse before the T-MAU deasserts COL and re-enables the transmit circuitry.

Power Down

The T-MAU circuitry can be made to go into low power mode. This feature is useful in battery powered or low duty cycle systems. The T-MAU will go into power down mode when RESET is active, **coma mode** is active, or the T-MAU is not selected. Refer to the Power Down Mode section for a description of the various power down modes.

Any of the three conditions listed above resets the internal logic of the T-MAU and places the device into power down mode. In this mode, the Twisted Pair driver pins (TXD±,TXP±) are asserted LOW, and the internal T-MAU status signals (LNKST, RCVPOL, XMT, RCV and COLLISION) are inactive.

Once the $\overline{\text{SLEEP}}$ pin is deasserted, the T-MAU will be forced into the Link Fail state. The T-MAU will move to the Link Pass state only after 5–6 link beat pulses and/or a single received message is detected on the RXD± pair.

In **Snooze** mode, the T-MAU receive circuitry will remain enabled even while the $\overline{\text{SLEEP}}$ pin is driven LOW.

The T-MAU circuitry will always go into power down mode if RESET is asserted, **coma** is enabled, or the T-MAU is not selected.

EADI (EXTERNAL ADDRESS DETECTION INTERFACE)

This interface is provided to allow external address filtering. It is selected by setting the EADISEL bit in ISACSR2. This feature is typically utilized for terminal servers, bridges and/or router type products. The use of external logic is required to capture the serial bit stream from the PCnet-ISA+ controller, compare it with a table of stored addresses or identifiers, and perform the desired function.

The EADI interface operates directly from the NRZ decoded data and clock recovered by the Manchester decoder or input to the GPSI, allowing the external address detection to be performed in parallel with frame reception and address comparison in the MAC Station Address Detection (SAD) block.

SRDCLK is provided to allow clocking of the receive bit stream into the external address detection logic. SRDCLK runs only during frame reception activity. Once a received frame commences and data and clock are available, the EADI logic will monitor the alternating ("1,0") preamble pattern until the two ones of the Start Frame Delimiter ("1,0,1,0,1,0,1,1") are detected, at which point the SF/BD output will be driven HIGH.

After SF/BD is asserted the serial data from SRD should be de-serialized and sent to a content addressable memory (CAM) or other address detection device.

To allow simple serial to parallel conversion, SF/BD is provided as a strobe and/or marker to indicate the delineation of bytes, subsequent to the SFD. This provides a mechanism to allow not only capture and/or decoding of the physical or logical (group) address, it also facilitates the capture of header information to determine protocol and or inter-networking information. The $\overline{EAR}$ pin is driven LOW by the external address comparison logic to reject the frame.

If an internal address match is detected by comparison with either the Physical or Logical Address field, the frame will be accepted regardless of the condition of $\overline{EAR}$. Incoming frames which do not pass the internal address comparison will continue to be received. This allows approximately 58 byte times after the last destination address bit is available to generate the $\overline{EAR}$ signal, assuming the device is not configured to accept runt packets. $\overline{EAR}$ will be ignored after 64 byte times after the SFD, and the frame will be accepted if $\overline{EAR}$ has not been asserted before this time. If Runt Packet Accept is configured, the $\overline{EAR}$ signal must be generated prior to the receive message completion, which could be as short as 12 byte times (assuming 6 bytes for source address, 2 bytes for length, no data, 4 bytes for FCS) after the last bit of the destination address is available. $\overline{EAR}$ must have a pulse width of at least 200 ns.

Note that setting the PROM bit (CSR15, bit 15) will cause all receive frames to be received, regardless of the state of the $\overline{EAR}$ input.

If the DRCUPA bit (CSR15.B) is set and the logical address (LADRF) is set to zero, only frames which are not rejected by $\overline{EAR}$ will be received.

The EADI interface will operate as long as the STRT bit in CSR0 is set, even if the receiver and/or transmitter are disabled by software (DTX and DRX bits in CSR15 set). This situation is useful as a power down mode in that the PCnet-ISA+ controller will not perform any DMA operations; this saves power by not utilizing the ISA bus driver circuits. However, external circuitry could still respond to specific frames on the network to facilitate remote node control.

The table below summarizes the operation of the EADI features.

Internal/External Address Recognition Capabilities

PROM	$\overline{EAR}$	Required Timing	Received Messages
1	X	No timing requirements	All Received Frames
0	1	No timing requirements	All Received Frames
0	0	Low for 200 ns within 512 bits after SFD	Physical/Logical Matches

General Purpose Serial Interface (GPSI)

The PCnet-ISA+ controller contains a General Purpose Serial Interface (GPSI) designed for testing the digital portions of the chip. The MENDEC, AUI, and twisted pair interface are by-passed once the device is set up in the special "test mode" for accessing the GPSI functions. Although this access is intended only for testing the device, some users may find the non-encoded data functions useful in some special applications. Note, however, that the GPSI functions can be accessed only when the PCnet-ISA+ devices operate as a bus master.

The PCnet-ISA+ GPSI signals are consistent with the LANCE digital serial interface. Since the GPSI functions can be accessed only through a special test mode, expect some loss of functionality to the device when the GPSI is invoked. The AUI and 10BASE-T analog interfaces are disabled along with the internal MENDEC logic. The LA (unlatched address) pins are removed and become the GPSI signals, therefore, only 20 bits of address space is available. The table below shows the GPSI pin configuration:

To invoke the GPSI signals, follow the procedure below:

1. After reset or I/O read of Reset Address, write 10b to PORTSEL bits in CSR15.
2. Set the ENTST bit in CSR4
3. Set the GPSIEN bit in CSR124 (see note below)

(The pins LA17–LA23 will change function after the completion of the above three steps.)

4. Clear the ENTST bit in CSR4
5. Clear Media Select bits in ISACSR2
6. Define the PORTSEL bits in the MODE register (CSR15) to be 10b to define GPSI port. The MODE register image is in the initialization block.

Note: *LA pins will be tristated before writing to GPSIEN bit. After writing to GPSIEN, LA[17–21] will be inputs, LA[22–23] will be outputs.*

GPSI Pin Configurations

GPSI Function	GPSI I/O Type	LANCE GPSI Pin	PCnet-ISA+ GPSI Pin	PCnet-ISA+ Pin Number	PCnet-ISA+ Normal Pin Function
Receive Data	I	RX	RXDAT	5	LA17
Receive Clock	I	RCLK	SRDCLK	6	LA18
Receive Carrier Sense	I	RENA	RXCRS	7	LA19
Collision	I	CLSN	CLSN	9	LA20
Transmit Clock	I	TCLK	STDCLK	10	LA21
Transmit Enable	O	TENA	TXEN	11	LA22
Transmit Data	O	TX	TXDAT	12	LA23

Note:

The GPSI Function is available only in the Bus Master Mode of operation.

IEEE 1149.1 Test Access Port Interface

An IEEE 1149.1 compatible boundary scan Test Access Port is provided for board-level continuity test and diagnostics. All digital input, output, and input/output pins are tested. Analog pins, including the AUI differential driver (DO±) and receivers (DI±, CI±), and the crystal input (XTAL1/XTAL2) pins, are tested. The T-MAU drivers TXD±, TXP±, and receiver RXD± are also tested.

The following is a brief summary of the IEEE 1149.1 compatible test functions implemented in the PCnet-ISA+ controller.

Boundary Scan Circuit

The boundary scan test circuit requires four extra pins (TCK, TMS, TDI and TDO), defined as the Test Access Port (TAP). It includes a finite state machine (FSM), an instruction register, a data register array, and a power-on reset circuit. Internal pull-up resistors are provided for the TDI, TCK, and TMS pins. The TCK pin must not be left unconnected. The boundary scan circuit remains active during sleep.

TAP FSM

The TAP engine is a 16-state FSM, driven by the Test Clock (TCK) and the Test Mode Select (TMS) pins. This FSM is in its reset state at power-up or RESET. An independent power-on reset circuit is provided to ensure the FSM is in the TEST_LOGIC_RESET state at power-up.

Supported Instructions

In addition to the minimum IEEE 1149.1 requirements (BYPASS, EXTEST and SAMPLE instructions), three additional instructions (IDCODE, TRIBYP and SETBYP) are provided to further ease board-level testing. All unused instruction codes are reserved. See the table below for a summary of supported instructions.

Instruction Register and Decoding Logic

After hardware or software RESET, the IDCODE instruction is always invoked. The decoding logic gives signals to control the data flow in the DATA registers according to the current instruction.

Boundary Scan Register (BSR)

Each BSR cell has two stages. A flip-flop and a latch are used in the SERIAL SHIFT STAGE and the PARALLEL OUTPUT STAGE, respectively.

There are four possible operational modes in the BSR cell:

1	Capture
2	Shift
3	Update
4	System Function

Other Data Registers

(1) BYPASS REG (1 BIT)

(2) DEV ID REG (32 bits)

Bits 31–28:	Version
Bits 27–12:	Part number (2260)
Bits 11–1:	Manufacturer ID. The 11 bit manufacturer ID code for AMD is 00000000001 according to JEDEC Publication 106-A.
Bit 0:	Always a logic 1

IEEE 1149.1 Supported Instruction Summary

Instruction Name	Description	Selected Data Reg	Mode	Instruction Code
EXTEST	External Test	BSR	Test	0000
IDCODE	ID Code Inspection	ID REG	Normal	0001
SAMPLE	Sample Boundary	BSR	Normal	0010
TRIBYP	Force Tristate	Bypass	Normal	0011
SETBYP	Control Boundary to 1/0	Bypass	Test	0100
BYPASS	Bypass Scan	Bypass	Normal	1111

Power Saving Modes

The PCnet-ISA+ controller supports two hardware power-savings modes. Both are entered by asserting the SLEEP pin LOW.

In **coma** mode, the PCnet-ISA+ controller will go into deep sleep with no support to automatically wake itself up. Sleep mode is enabled when the AWAKE bit in ISACSR2 is reset. This mode is the default power down mode.

In **Snooze** mode, enabled by setting the AWAKE bit in ISACSR2 and driving the SLEEP pin LOW, the T-MAU receive circuitry will remain enabled even while the SLEEP pin is driven LOW. The LED0 output will also continue to function, indicating a good 10BASE-T link if there are link beat pulses or valid frames present. This LED0 pin can be used to drive a LED and/or external hardware that directly controls the SLEEP pin of the PCnet-ISA+ controller. This configuration effectively wakes the system when there is any activity on the 10BASE-T link.

Access Operations (Software)

We begin by describing how byte and word data are addressed on the ISA bus, including conversion cycles where 16-bit accesses are turned into 8-bit accesses because the resource accessed did not support 16-bit operations. Then we describe how registers and other resources are accessed. This section is for the device programmer, while the next section (bus cycles) is for the hardware designer.

I/O Resources

The PCnet-ISA+ controller has both I/O and memory resources. In the I/O space the resources are organized as indicated in the following table:

Offset	#Bytes	Register
0h	16	IEEE Address
10h	2	RDP
12h	2	RAP (shared by RDP and IDP)
14h	2	Reset
16h	2	IDP

The PCnet-ISA+ controller does not respond to any addresses outside of the offset range 0-17h. I/O offsets 18h and up are not used by the PCnet-ISA+ controller.

I/O Register Access

The register address port (RAP) is shared by the register data port (RDP) and the ISACSR data port (IDP) to save registers. To access the Ethernet controller's RDP or IDP, the RAP should be written first, followed by the read or write access to the RDP or IDP. I/O register accesses should be coded as 16-bit accesses, even if the PCnet-ISA+ controller is hardware configured for 8-bit I/O bus cycles. It is acceptable (and transparent) for the motherboard to turn a 16-bit software access into two separate 8-bit hardware bus cycles. The motherboard accesses the low byte before the high byte and the PCnet-ISA+ controller has circuitry to specifically support this type of access.

The reset register causes a reset when read. Any value will be accepted and the cycle may be 8 or 16 bits wide. Writes are ignored.

All PCnet-ISA+ controller register accesses should be coded as 16-bit operations.

**Note that the RAP is cleared on Reset.*

IEEE Address Access

The address PROM may be an external memory device that contains the node's unique physical Ethernet address and any other data stored by the board manufacturer. The software accesses must be 16-bit. This information may be stored in the EEPROM.

Boot PROM Access

The boot PROM is an external memory resource located by the address selected by the EEPROM or the BPAM input in shared memory mode. It may be software accessed as an 8- or 16-bit resource but the latter is recommended for best performance.

Static RAM Access

The static RAM is only present in the shared memory mode. It is located at the address selected by the SMAM input. It may be accessed as an 8- or 16-bit resource but the latter is recommended for best performance.

Bus Cycles (Hardware)

The PCnet-ISA+ controller supports both 8- and 16-bit hardware bus cycles. The following sections outline where any limitations apply based upon the architecture mode and/or the resource that is being accessed (PCnet-ISA+ controller registers, address PROM, boot PROM, or shared memory SRAM). For completeness, the following sections are arranged by architecture (Bus Master Mode or Shared Memory Mode). SRAM resources apply only to Shared Memory Mode.

All resources (registers, PROMs, SRAM) are presented to the ISA bus by the PCnet-ISA+ controller. With few exceptions, these resources can be configured for either 8-bit or 16-bit bus cycles. The I/O resources (registers, address PROM) are width configured using the EEPROM. The memory resources (boot PROM, SRAM) are width configured by external hardware.

For 16-bit memory accesses, hardware external to the PCnet-ISA+ controller asserts MEMCS16 when either of the two memory resources is selected. The ISA bus requires that all memory resources within a block of 128 Kbytes be the same width, either 8- or 16-bits. The reason for this is that the MEMCS16 signal is generally a decode of the LA17-23 address lines. 16-bit memory capability is desirable since two 8-bit accesses take the same amount of time as four 16-bit accesses.

All accesses to 8-bit resources (which do not return MEMCS16 or IOCS16) use SD0-7. If an odd byte is accessed, the Current Master swap buffer turns on. During an odd byte read the swap buffer copies the data from SD0-7 to the high byte. During an odd byte write the Current Master swap buffer copies the data from the high byte to SD0-7. The PCnet-ISA+ controller can be configured to be an 8-bit I/O resource even in a 16-bit system; this is set by the EEPROM. It is recommended that the PCnet-ISA+ controller be configured for 8-bit only I/O bus cycles for maximum compatibility with PC/AT clone motherboards.

When the PCnet-ISA+ controller is in an 8-bit system such as a PC/XT, SBHE and IOCS16 must be left unconnected (these signals do not exist in the PC/XT). This will force ALL resources (I/O and memory) to support only 8-bit bus cycles. The PCnet-ISA+ controller will function in an 8-bit system only if configured for Shared Memory Mode.

Accesses to 16-bit resources (which do return MEMCS16 or IOCS16) use either or both SD0–7 and SD8–15. A word access is indicated by A0=0 and SBHE=0 and data is transferred on all 16 data lines. An even byte access is indicated by A0=0 and SBHE=1 and data is transferred on SD0–7. An odd-byte access is indicated by A0=1 and SBHE=0 and data is transferred on SD8-15. It is illegal to have A0=1 and SBHE=1 in any bus cycle. The PCnet-ISA+ controller returns only IOCS16; MEMCS16 must be generated by external hardware if desired. The use of MEMCS16 applies only to Shared Memory Mode.

The following table describes all possible types of ISA bus accesses, including Permanent Master as Current Master and PCnet-ISA+ controller as Current Master. The PCnet-ISA+ controller will not work with 8-bit memory while it is Current Master. Any descriptions of 8-bit memory accesses are for when the Permanent Master is Current Master.

The two byte columns (D0–7 and D8–15) indicate whether the bus master or slave is driving the byte. CS16 is a shorthand for MEMCS16 and IOCS16.

Bus Master Mode

The PCnet-ISA+ controller can be configured as a Bus Master only in systems that support bus mastering. In addition, the system is assumed to support 16-bit memory (DMA) cycles (the PCnet-ISA+ controler does not use the MEMCS16 signal on the ISA bus). This does not preclude the PCnet-ISA+ controller from doing 8-bit I/O transfers. The PCnet-ISA+ controller will not function as a bus master in 8-bit platforms such as the PC/XT.

ISA Bus Accesses

R/W	A0	SBHE	CS16	D0–7	D8–15	Comments
RD	0	1	x	Slave	Float	Low byte RD
RD	1	0	1	Slave	Float*	High byte RD with swap
RD	0	0	1	Slave	Float	16-Bit RD converted to low byte RD
RD	1	0	0	Float	Slave	High byte RD
RD	0	0	0	Slave	Slave	16-Bit RD
WR	0	1	x	Master	Float	Low byte WR
WR	1	0	1	Float*	Master	High byte WR with swap
WR	0	0	1	Master	Master	16-Bit WR converted to low byte WR
WR	1	0	0	Float	Master	High byte WR
WR	0	0	0	Master	Master	16-Bit WR

**Motherboard SWAP logic drives*

Refresh Cycles

Although the PCnet-ISA+ controller is neither an originator or a receiver of refresh cycles, it does need to avoid unintentional activity during a refresh cycle in bus master mode. A refresh cycle is performed as follows: First, the REF signal goes active. Then a valid refresh address is placed on the address bus. MEMR goes active, the refresh is performed, and MEMR goes inactive. The refresh address is held for a short time and then goes invalid. Finally, REF goes inactive. During a refresh cycle, as indicated by REF being active, the PCnet-ISA+ controller ignores DACK if it goes active until it goes inactive. It is necessary to ignore DACK during a refresh because some motherboards generate a false DACK at that time.

Address PROM Cycles External PROM

The Address PROM is a small (16 bytes) 8-bit PROM connected to the PCnet-ISA+ controller Private Data Bus. The PCnet-ISA+ controller will support only 8-bit ISA I/O bus cycles for the address PROM; this limitation is transparent to software and does not preclude 16-bit software I/O accesses. An access cycle begins with the Permanent Master driving AEN LOW, driving the addresses valid, and driving IOR active. The PCnet-ISA+ controller detects this combination of signals and

arbitrates for the Private Data Bus (PRDB) if necessary. IOCHRDY is driven LOW during accesses to the address PROM.

When the Private Data Bus becomes available, the PCnet-ISA+ controller drives APCS active, releases IOCHRDY, turns on the data path from PRD0-7, and enables the SD0-7 drivers (but not SD8-15). During this bus cycle, IOCS16 is not driven active. This condition is maintained until IOR goes inactive, at which time the bus cycle ends. Data is removed from SD0-7 within 30 ns.

Address PROM Cycles Using EEPROM Data

Default mode. In this mode, the IEEE address information is stored not in an external parallel PROM but in the EEPROM along with other configuration information. PCnet-ISA+ will respond to I/O reads from the IEEE address (the first 16 bytes of the I/O map) by supplying data from an internal RAM inside PCnet-ISA+. This internal RAM is loaded with the IEEE address at RESET and is write protected.

Ethernet Controller Register Cycles

Ethernet controller registers (RAP, RDP, IDP) are naturally 16-bit resources but can be configured to operate with 8-bit bus cycles provided the proper protocol is followed. This means on a read, the PCnet-ISA+ controller will only drive the low byte of the system data bus; if an odd byte is accessed, it will be swapped down. The high byte of the system data bus is never driven by the PCnet-ISA+ controller under these conditions. On a write cycle, the even byte is placed in a holding register. An odd byte write is internally swapped up and augmented with the even byte in the holding register to provide an internal 16-bit write. This allows the use of 8-bit I/O bus cycles which are more likely to be compatible with all ISA-compatible clones, but requires that both bytes be written in immediate succession. This is accomplished simply by treating the PCnet-ISA+ controller controller registers as 16-bit software resources. The motherboard will convert the 16-bit accesses done by software into two sequential 8-bit accesses, an even byte access followed immediately by an odd byte access.

An access cycle begins with the Permanent Master driving AEN LOW, driving the address valid, and driving IOR or IOW active. The PCnet-ISA+ controller detects this combination of signals and drives IOCHRDY LOW. IOCS16 will also be driven LOW if 16-bit I/O bus cycles are enabled. When the register data is ready, IOCHRDY will be released HIGH. This condition is maintained until IOR or IOW goes inactive, at which time the bus cycle ends.

RESET Cycles

A read to the reset address causes an PCnet-ISA+ controller reset. This has the same effect as asserting the RESET pin on the PCnet-ISA+ controller, such as happens during a system power-up or hard boot. The subsequent write cycle needed in the NE2100 LANCE based family of Ethernet cards is not required but does not have any harmful effects. IOCS16 is not asserted in this cycle.

ISA Configuration Register Cycles

The ISA configuration registers are accessed by placing the address of the desired register into the RAP and reading the IDP. The ISACSR bus cycles are identical to all other PCnet-ISA+ controller register bus cycles.

Boot PROM Cycles

The Boot PROM is an 8-bit PROM connected to the PCnet-ISA+ controller Private Data Bus (PRDB) and can occupy up to 64K of address space. Since the PCnet-ISA+ controller does not generate MEMCS16, only 8-bit ISA memory bus cycles to the boot PROM are supported in Bus Master Mode; this limitation is transparent to software and does not preclude 16-bit software memory accesses. A boot PROM access cycle begins with the Permanent Master driving the addresses valid, REF inactive, and MEMR active. (AEN is not involved in memory cycles). The PCnet-ISA+ controller detects this combination of signals, drives IOCHRDY LOW, and reads a byte out of the Boot PROM. The data byte read is driven onto the lower system data bus lines and IOCHRDY is released. This condition is maintained until MEMR goes inactive, at which time the access cycle ends.

The BPCS signal generated by the PCnet-ISA+ controller is three 20 MHz clock cycles wide (300 ns). Including delays, the Boot PROM has 275 ns to respond to the BPCS signal from the PCnet-ISA+ controller. This signal is intended to be connected to the CS pin on the boot PROM, with the PROM OE pin tied to ground.

Current Master Operation

Current Master operation only occurs in the bus master mode. It does not occur in shared memory mode.

There are three phases to the use of the bus by the PCnet-ISA+ controller as Current Master, the Obtain Phase, the Access Phase, and the Release Phase.

Obtain Phase

A Master Mode Transfer Cycle begins by asserting DRQ. When the Permanent Master asserts DACK, the PCnet-ISA+ controller asserts MASTER, signifying it has taken control of the ISA bus. The Permanent Master tristates the address, command, and data lines within 60 ns of DACK going active. The Permanent Master drives AEN inactive within 71 ns of MASTER going active.

Access Phase

The ISA bus requires a wait of at least 125 ns after MASTER is asserted before the new master is allowed to drive the address, command, and data lines. The PCnet-ISA+ controller will actually wait 3 clock cycles or 150 ns.

The following signals are not driven by the Permanent Master and are simply pulled HIGH: BALE, IOCHRDY, IOCS16, MEMCS16, SRDY. Therefore, the PCnet-ISA+ controller assumes the memory which it is accessing is 16 bits wide and can complete an access in the time programmed for the PCnet-ISA+ controller MEMR and MEMW signals. Refer to the ISA Bus Configuration Register description section.

Release Phase

When the PCnet-ISA+ controller is finished with the bus, it drives the command lines inactive. 50 ns later, the controller tri-states the command, address, and data lines and drives DRQ inactive. 50 ns later, the controller drives MASTER inactive.

The Permanent Master drives AEN active within 71 ns of MASTER going inactive. The Permanent Master is allowed to drive the command lines no sooner than 60 ns after DACK goes inactive.

Master Mode Memory Read Cycle

After the PCnet-ISA+ controller has acquired the ISA bus, it can perform a memory read cycle. All timing is generated relative to the 20 MHz clock (network clock). Since there is no way to tell if memory is 8- or 16-bit or when it is ready, the PCnet-ISA+ controller by default assumes 16-bit, 1 wait state memory. The wait state assumption is based on the default value in the MSRDA register in ISACSR0.

The cycle begins with SA0-19, SBHE, and LA17-23 being presented. The ISA bus requires them to be valid for at least 28 ns before a read command and the PCnet-ISA+ controller provides one clock or 50 ns of setup time before asserting MEMR.

The ISA bus requires MEMR to be active for at least 219 ns, and the PCnet-ISA+ controller provides a default of 5 clocks, or 250 ns, but this can be tuned for faster systems with the Master Mode Read Active (MSRDA) register (see section 2.5.2). Also, if IOCHRDY is driven LOW, the PCnet-ISA+ controller will wait. The wait state counter must expire and IOCHRDY must be HIGH for the PCnet-ISA+ controller to continue.

The PCnet-ISA+ controller then accepts the memory read data. The ISA bus requires all command lines to remain inactive for at least 97 ns before starting another bus cycle and the PCnet-ISA+ controller provides at least two clocks or 100 ns of inactive time.

The ISA bus requires read data to be valid no more than 173 ns after receiving MEMR active and the PCnet-ISA+ controller requires 10 ns of data setup time. The ISA bus requires read data to provide at least 0 ns of hold time and to be removed from the bus within 30 ns after MEMR goes inactive. The PCnet-ISA+ controller requires 0 ns of data hold time.

Master Mode Memory Write Cycle

After the PCnet-ISA+ controller has acquired the ISA bus, it can perform a memory write cycle. All timing is generated relative to a 20 MHz clock which happens to be the same as the network clock. Since there is no way to tell if memory is 8- or 16-bit or when it is ready, the PCnet-ISA+ controller by default assumes 16-bit, 1 wait state memory. The wait state assumption is based on the default value in the MSWRA register in ISACSR1.

The cycle begins with SA0-19, SBHE, and LA17-23 being presented. The ISA bus requires them to be valid at least 28 ns before MEMW goes active and data to be valid at least 22 ns before MEMW goes active. The PCnet-ISA+ controller provides one clock or 50 ns of setup time for all these signals.

The ISA bus requires MEMW to be active for at least 219 ns, and the PCnet-ISA+ controller provides a default of 5 clocks, or 250 ns, but this can be tuned for faster systems with the Master Mode Write Active (MSWRA) register (ISACSR1). Also, if IOCHRDY is driven LOW, the PCnet-ISA+ controller will wait. IOCHRDY must be HIGH for the PCnet-ISA+ controller to continue.

The ISA bus requires data to be valid for at least 25 ns after MEMW goes inactive, and the PCnet-ISA+ controller provides one clock or 50 ns.

The ISA bus requires all command lines to remain inactive for at least 97 ns before starting another bus cycle. The PCnet-ISA+ controller provides at least two clocks or 100 ns of inactive time when bit 4 in ISACSR2 is set. The EISA bus requires all command lines to remain inactive for at least 170 ns before starting another bus cycle. When bit 4 in ISACSR4 is cleared, the PCnet-ISA+ controller provides 200 ns of inactive time.

Shared Memory Mode

Address PROM Cycles External PROM

The Address PROM is a small (16 bytes) 8-bit PROM connected to the PCnet-ISA+ controller Private Data Bus (PRDB). The PCnet-ISA+ controller will support only 8-bit ISA I/O bus cycles for the address PROM; this limitation is transparent to software and does not preclude 16-bit software I/O accesses. An access cycle begins with the Permanent Master driving AEN LOW, driving the addresses valid, and driving IOR active. The PCnet-ISA+ controller detects this combination of signals and arbitrates for the Private Data Bus if necessary. IOCHRDY is always driven LOW during address PROM accesses.

When the Private Data Bus becomes available, the PCnet-ISA+ controller drives APCS active, releases IOCHRDY, turns on the data path from PRD0-7, and enables the SD0-7 drivers (but not SD8-15). During this bus cycle, IOCS16 is not driven active. This condition is

maintained until IOR goes inactive, at which time the access cycle ends. Data is removed from SD0-7 within 30 ns.

The PCnet-ISA+ controller will perform 8-bit ISA bus cycle operation for all resources (registers, PROMs, SRAM) if SBHE has been left unconnected, such as in the case of an 8-bit system like the PC/XT.

Ethernet Controller Register Cycles

Ethernet controller registers (RAP, RDP, ISACSR) are naturally 16-bit resources but can be configured to operate with 8-bit bus cycles provided the proper protocol is followed. This is programmable by the EEPROM. This means on a read, the PCnet-ISA+ controller will only drive the low byte of the system data bus; if an odd byte is accessed, it will be swapped down. The high byte of the system data bus is never driven by the PCnet-ISA+ controller under these conditions. On a write, the even byte is placed in a holding register. An odd-byte write is internally swapped up and augmented with the even byte in the holding register to provide an internal 16-bit write. This allows the use of 8-bit I/O bus cycles which are more likely to be compatible with all clones, but requires that both bytes be written in immediate succession. This is accomplished simply by treating the PCnet-ISA+ controller controller registers as 16-bit software resources. The motherboard will convert the 16-bit accesses done by software into two sequential 8-bit accesses, an even- byte access followed immediately by an odd-byte access.

An access cycle begins with the Permanent Master driving AEN LOW, driving the address valid, and driving IOR or IOW active. The PCnet-ISA+ controller detects this combination of signals and drives IOCHRDY LOW. IOCS16 will also be driven LOW if 16-bit I/O bus cycles are enabled. When the register data is ready, IOCHRDY will be released HIGH. This condition is maintained until IOR or IOW goes inactive, at which time the bus cycle ends.

The PCnet-ISA+ controller will perform 8-bit ISA bus cycle operation for all resources (registers, PROMs, SRAM) if SBHE has been left unconnected, such as in the case of an 8-bit system like the PC/XT.

RESET Cycles

A read to the reset address causes an PCnet-ISA+ controller reset. This has the same effect as asserting the RESET pin on the PCnet-ISA+ controller, such as happens during a system power-up or hard boot. The subsequent write cycle needed in the NE2100 LANCE-based family of Ethernet cards is not required but does not have any harmful effects. IOCS16 is not asserted in this cycle.

ISA Configuration Register Cycles

The ISA configuration register is accessed by placing the address of the desired register into the RAP and reading the IDP. The ISACSR bus cycles are identical to all other PCnet-ISA+ controller register bus cycles.

Boot PROM Cycles

The Boot PROM is an 8-bit PROM connected to the PCnet-ISA+ controller Private Data Bus (PRDB), and can occupy up to 64 Kbytes of address space. In Shared Memory Mode, an external address comparator is responsible for asserting BPAM to the PCnet-ISA+ controller. BPAM is intended to be a perfect decode of the boot PROM address space, i.e. LA17-23, SA16. The LA bus must be latched with BALE in order to provide stable signal for BPAM. REF inactive must be used by the external logic to gate boot PROM address decoding. This same logic must assert MEMCS16 to the ISA bus if 16-bit Boot PROM bus cycles are desired.

The PCnet-ISA+ controller assumes 16-bit ISA memory bus cycles for the boot PROM. A 16-bit boot PROM bus cycle begins with the Permanent Master driving the addresses valid and MEMR active. (AEN is not involved in memory cycles). External hardware would assert BPAM and MEMCS16. The PCnet-ISA+ controller detects this combination of signals, drives IOCHRDY LOW, and reads two bytes out of the boot PROM. The data bytes read from the PROM are driven by the PCnet-ISA+ controller onto SD0-15 and IOCHRDY is released. This condition is maintained until MEMR goes inactive, at which time the access cycle ends.

The PCnet-ISA+ controller will perform 8-bit ISA bus cycle operation for all resource (registers, PROMs, SRAM) if SBHE has been left unconnected, such as in the case of an 8-bit system like the PC/XT.

The BPCS signal generated by the PCnet-ISA+ controller is three 20 MHz clock cycles wide (350 ns). Including delays, the Boot PROM has 275 ns to respond to the BPCS signal from the PCnet-ISA+ controller. This signal is intended to be connected to the CS pin on the boot PROM, with the PROM OE pin tied to ground.

Static RAM Cycles

The shared memory SRAM is an 8-bit device connected to the PCnet-ISA+ controller Private Bus, and can occupy up to 64 Kbytes of address space. In Shared Memory Mode, an external address comparator is responsible for asserting SMAM to the PCnet-ISA+ controller. SMAM is intended to be a perfect decode of the SRAM address space, i.e. LA17-23, SA16 for 64 Kbytes of SRAM. The LA signals must be latched by BALE in order to provide a stable decode for SMAM.

The PCnet-ISA+ controller assumes 16-bit ISA memory bus cycles for the SRAM, so this same logic must assert MEMCS16 to the ISA bus if 16-bit bus cycles are to be supported.

A 16-bit SRAM bus cycle begins with the Permanent Master driving the addresses valid, REF inactive, and either MEMR or MEMW active. (AEN is not involved in memory cycles). External hardware would assert SMAM and MEMCS16. The PCnet-ISA+ controller detects this combination of signals and initiates the SRAM access.

In a write cycle, the PCnet-ISA+ controller stores the data into an internal holding register, allowing the ISA bus cycle to finish normally. The data in the holding register will then be written to the SRAM without the need for ISA bus control. In the event the holding register is already filled with unwritten SRAM data, the PCnet-ISA+ controller will extend the ISA write cycle by driving IOCHRDY LOW until the unwritten data is stored in the SRAM. The current ISA bus cycle will then complete normally.

In a read cycle, the PCnet-ISA+ controller arbitrates for the Private Bus. If it is unavailable, the PCnet-ISA+ controller drives IOCHRDY LOW. The PCnet-ISA+ controller compares the 16 bits of address on the System Address Bus with that of a data word held in an internal pre-fetch register.

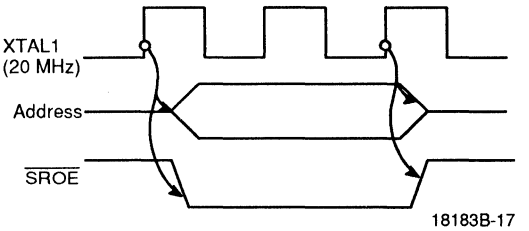
If the address does not match that of the prefetched SRAM data, then the PCnet-ISA+ controller drives IOCHRDY LOW and reads two bytes from the SRAM. The PCnet-ISA+ controller then proceeds as though the addressed data location had been prefetched.

If the internal prefetch buffer contains the correct data, then the pre-fetch buffer data is driven on the System Data bus. If IOCHRDY was previously driven LOW due to either Private Data Bus arbitration or SRAM access, then it is released HIGH. The PCnet-ISA+ controller remains in this state until MEMR is de-asserted, at which time the PCnet-ISA+ controller performs a new prefetch of the SRAM. In this way memory read wait states can be minimized.

The PCnet-ISA+ controller performs prefetches of the SRAM between ISA bus cycles. The SRAM is prefetched in an incrementing word address fashion. Prefetched data are invalidated by any other activity on the Private Bus, including Shared Memory Writes by either the ISA bus or the network interface, and also address and boot PROM reads.

The only way to configure the PCnet-ISA+ controller for 8-bit ISA bus cycles for SRAM accesses is to configure the entire PCnet-ISA+ controller to support only 8-bit ISA bus cycles. This is accomplished by leaving the SBHE pin disconnected. The PCnet-ISA+ controller will perform 8-bit ISA bus cycle operation for all resources (registers, PROMs, SRAM) if SBHE has never been driven active since the last RESET, such as in the case of an 8-bit system like the PC/XT. In this case, the external address decode logic must not assert MEMCS16 to the ISA bus, which will be the case if MEMCS16 is left unconnected. It is possible to manufacture a dual 8/16 bit PCnet-ISA+ controller adapter card, as the MEMCS16 and SBHE signals do not exist in the PC/XT environment.

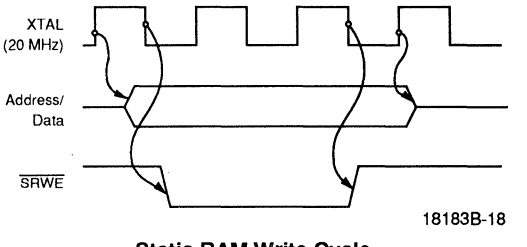
At the memory device level, each SRAM Private Bus read cycle takes two 50 ns clock periods for a maximum read access time of 75 ns. The timing looks like this:



Static RAM Read Cycle

The address and SROE go active within 20 ns of the clock going HIGH. Data is required to be valid 5 ns before the end of the second clock cycle. Address and SROE have a 0 ns hold time after the end of the second clock cycle. Note that the PCnet-ISA+ controller does not normally provide a separate SRAM CS signal; SRAM CS must always be asserted.

SRAM Private Bus write cycles require three 50 ns clock periods to guarantee non-negative address setup and hold times with regard to SRWE. The timing is illustrated as follows:



Static RAM Write Cycle

Address and data are valid 20 ns after the rising edge of the first clock period. SRWE goes active 20 ns after the falling edge of the first clock period. SRWE goes inactive 20 ns after the falling edge of the third clock period. Address and data remain valid until the end of the third clock period. Rise and fall times are nominally 5 ns.

Non-negative setup and hold times for address and data with respect to $\overline{SRWE}$ are guaranteed. $\overline{SRWE}$ has a pulse width of typically 100 ns, minimum 75 ns.

Transmit Operation

The transmit operation and features of the PCnet-ISA+ controller are controlled by programmable options.

Transmit Function Programming

Automatic transmit features, such as retry on collision, FCS generation/transmission, and pad field insertion, can all be programmed to provide flexibility in the (re-)transmission of messages.

Disable retry on collision (DRTY) is controlled by the DRTY bit of the Mode register (CSR15) in the initialization block.

Automatic pad field insertion is controlled by the APAD_XMT bit in CSR4. If APAD_XMT is set, automatic pad field insertion is enabled, the DXMTFCS feature is over-ridden, and the 4-byte FCS will be added to the transmitted frame unconditionally. If APAD_XMT is cleared, no pad field insertion will take place and runt packet transmission is possible.

The disable FCS generation/transmission feature can be programmed dynamically on a frame by frame basis. See the ADD_FCS description of TMD1.

Transmit FIFO Watermark (XMTFW in CSR80) sets the point at which the BMU (Buffer Management Unit) requests more data from the transmit buffers for the FIFO. This point is based upon how many 16-bit bus transfers (2 bytes) could be performed to the existing empty space in the transmit FIFO.

Transmit Start Point (XMTSP in CSR80) sets the point when the transmitter actually tries to go out on the media. This point is based upon the number of bytes written to the transmit FIFO for the current frame.

When the entire frame is in the FIFO, attempts at transmission of preamble will commence regardless of the value in XMTSP. The default value of XMTSP is 10b, meaning 64 bytes full.

Automatic Pad Generation

Transmit frames can be automatically padded to extend them to 64 data bytes (excluding preamble). This allows the minimum frame size of 64 bytes (512 bits) for 802.3/Ethernet to be guaranteed with no software intervention from the host/controlling process. Setting the APAD_XMT bit in CSR4 enables the automatic padding feature. The pad is placed between the LLC data field and FCS field in the 802.3 frame. FCS is always added if the frame is padded, regardless of the state of DXMTFCS. The transmit frame will be padded by bytes with the value of 00h. The default value of APAD_XMT is 0, and this will disable auto pad generation after RESET.

It is the responsibility of upper layer software to correctly define the actual length field contained in the message to correspond to the total number of LLC Data bytes encapsulated in the packet (length field as defined in the IEEE 802.3 standard). The length value contained in the message is not used by the PCnet-ISA+ controller to compute the actual number of pad bytes to be inserted. The PCnet-ISA+ controller will append pad bytes dependent on the actual number of bits transmitted onto the network. Once the last data byte of the frame has completed prior to appending the FCS, the PCnet-ISA+ controller will check to ensure that 544 bits have been transmitted. If not, pad bytes are added to extend the frame size to this value, and the FCS is then added.

The 544 bit count is derived from the following:

Minimum frame size (excluding preamble, including FCS)	64 bytes	512 bits
Preamble/SFD size	8 bytes	64 bits
FCS size	4 bytes	32 bits

To be classed as a minimum-size frame at the receiver, the transmitted frame must contain:

Preamble + (Min Frame Size + FCS) bits

At the point that FCS is to be appended, the transmitted frame should contain:

Preamble + (Min Frame Size - FCS) bits
64 + (512 - 32) bits

A minimum-length transmit frame from the PCnet-ISA+ controller will, therefore, be 576 bits after the FCS is appended.

Transmit FCS Generation

Automatic generation and transmission of FCS for a transmit frame depends on the value of DXMTFCS bit in CSR15. When DXMTFCS = 0 the transmitter will generate and append the FCS to the transmitted frame. If the automatic padding feature is invoked (APAD_XMT is SET in CSR4), the FCS will be appended by the PCnet-ISA+ controller regardless of the state of DXMTFCS. Note that the calculated FCS is transmitted most-significant bit first. The default value of DXMTFCS is 0 after RESET.

Transmit Exception Conditions

Exception conditions for frame transmission fall into two distinct categories; those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-ISA+ controller are

basically collisions within the slot time with automatic retry. The PCnet-ISA+ controller will ensure that collisions which occur within 512 bit times from the start of transmission (including preamble) will be automatically retried with no host intervention. The transmit FIFO ensures this by guaranteeing that data contained within the FIFO will not be overwritten until at least 64 bytes (512 bits) of data have been successfully transmitted onto the network.

If 16 total attempts (initial attempt plus 15 retries) fail, the PCnet-ISA+ controller sets the RTRY bit in the current transmit TDTE in host memory (TMD2), gives up ownership (sets the OWN bit to zero) for this packet, and processes the next packet in the transmit ring for transmission.

ISO 8802-3 (IEEE/ANSI 802.3) Data Frame

Abnormal network conditions include:

- Loss of carrier
- Late collision
- SQE Test Error (Does not apply to 10BASE-T port.)

These should not occur on a correctly configured 802.3 network, and will be reported if they do.

When an error occurs in the middle of a multi-buffer frame transmission, the error status will be written in the current descriptor. The OWN bit(s) in the subsequent descriptor(s) will be reset until the STP (the next frame) is found.

Loss of Carrier

A loss of carrier condition will be reported if the PCnet-ISA+ controller cannot observe receive activity while it is transmitting on the AUI port. After the PCnet-ISA+ controller initiates a transmission, it will expect to see data "looped back" on the DI± pair. This will internally generate a "carrier sense," indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted before the end of the transmission. If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in TMD2 after the frame has been transmitted. The frame will not be re-tried on the basis of an LCAR error. In 10BASE-T mode LCAR will indicate that Jabber or Link Fail state has occurred.

Late Collision

A late collision will be reported if a collision condition occurs after one slot time (512 bit times) after the transmit process was initiated (first bit of preamble commenced). The PCnet-ISA+ controller will abandon the transmit process for the particular frame, set Late Collision (LCOL) in the associated TMD3, and process the next transmit frame in the ring. Frames experiencing a late collision will not be re-tried. Recovery from this condition must be performed by upper-layer software.

SQE Test Error

During the inter packet gap time following the completion of a transmitted message, the AUI CI± pair is asserted by some transceivers as a self-test. The integral Manchester Encoder/Decoder will expect the SQE Test Message (nominal 10 MHz sequence) to be returned via the CI± pair within a 40 network bit time period after DI± pair goes inactive. If the CI± inputs are not asserted within the 40 network bit time period following the completion of transmission, then the PCnet-ISA+ controller will set the CERR bit in CSR0. CERR will be asserted in 10BASE-T mode after transmit if T-MAU is in Link Fail state. CERR will never cause INTR to be activated. It will, however, set the ERR bit in CSR0.

Host related transmit exception conditions include BUFF and UFLO as described in the Transmit Descriptor section.

Receive Operation

The receive operation and features of the PCnet-ISA+ controller are controlled by programmable options.

Receive Function Programming

Automatic pad field stripping is enabled by setting the ASTRP_RCV bit in CSR4; this can provide flexibility in the reception of messages using the 802.3 frame format.

All receive frames can be accepted by setting the PROM bit in CSR15. When PROM is set, the PCnet-ISA+ controller will attempt to receive all messages, subject to minimum frame enforcement. Promiscuous mode overrides the effect of the Disable Receive Broadcast bit on receiving broadcast frames.

The point at which the BMU will start to transfer data from the receive FIFO to buffer memory is controlled by the RCVFW bits in CSR80. The default established during reset is 10b, which sets the threshold flag at 64 bytes empty.

Automatic Pad Stripping

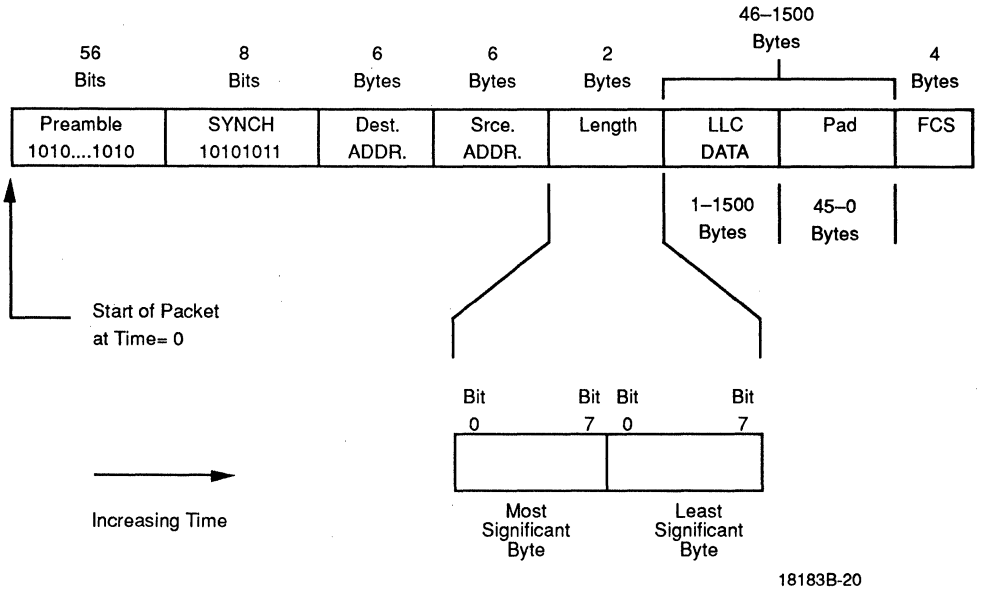
During reception of an 802.3 frame the pad field can be stripped automatically. ASTRP_RCV (bit 10 in CSR4) = 1 enables the automatic pad stripping feature. The pad field will be stripped before the frame is passed to the FIFO, thus preserving FIFO space for additional frames. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has had the pad characters stripped.

The number of bytes to be stripped is calculated from the embedded length field (as defined in the IEEE 802.3 definition) contained in the frame. The length indicates the actual number of LLC data bytes contained in the message. Any received frame which contains a length field less than 46 bytes will have the pad field stripped (if ASTRP_RCV is set). Receive frames which have a length field of 46 bytes or greater will be passed to the host unmodified.

Since any valid Ethernet Type field value will always be greater than a normal 802.3 Length field (≥46), the PCnet-ISA+ controller will not attempt to strip valid Ethernet frames.

Note that for some network protocols the value passed in the Ethernet Type and/or 802.3 Length field is not compliant with either standard and may cause problems.

The diagram below shows the byte/bit ordering of the received length field for an 802.3 compatible frame format.



IEEE/ANSI 802.3 Frame and Length Field Transmission Order

Receive FCS Checking

Reception and checking of the received FCS is performed automatically by the PCnet-ISA+ controller. Note that if the Automatic Pad Stripping feature is enabled, the received FCS will be verified against the value computed for the incoming bit stream including pad characters, but it will not be passed to the host. If a FCS error is detected, this will be reported by the CRC bit in RMD1.

Receive Exception Conditions

Exception conditions for frame reception fall into two distinct categories; those which are the result of normal

network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-ISA+ controller are basically collisions within the slot time and automatic runt packet rejection. The PCnet-ISA+ controller will ensure that collisions which occur within 512 bit times from the start of reception (excluding preamble) will be automatically deleted from the receive FIFO with no host intervention. The receive FIFO will delete any frame which is composed of fewer than 64 bytes provided that the Runt Packet Accept (RPA bit in CSR124) feature has not been enabled. This criteria will be met regardless of whether the receive frame was the first (or only) frame in the FIFO or if the receive frame was queued behind a previously received message.

Abnormal network conditions include:

- FCS errors
- Late collision

These should not occur on a correctly configured 802.3 network and will be reported if they do.

Host related receive exception conditions include MISS, BUFF, and OFLO. These are described in the Receive Descriptor section.

Loopback Operation

Loopback is a mode of operation intended for system diagnostics. In this mode, the transmitter and receiver are both operating at the same time so that the controller receives its own transmissions. The controller provides two types of internal loopback and one type of external loopback. In internal loopback mode, the transmitted data can be looped back to the receiver at one of two places inside the controller without actually transmitting any data to the external network. The receiver will move the received data to the next receive buffer, where it can be examined by software. Alternatively, in external loopback mode, data can be transmitted to and received from the external network.

There are restrictions on loopback operation. The PCnet-ISA+ controller has only one FCS generator circuit. The FCS generator can be used by the transmitter to generate the FCS to append to the frame, or it can be used by the receiver to verify the FCS of the received frame. It can not be used by the receiver and transmitter simultaneously.

If the FCS generator is connected to the receiver, the transmitter will not append an FCS to the frame, but the receiver will check for one. The user can, however, calculate the FCS value for a frame and include this four-byte number in the transmit buffer.

If the FCS generator is connected to the transmitter, the transmitter will append an FCS to the frame, but the receiver will not check for the FCS. However, the user can verify the FCS by software.

During loopback, the FCS logic can be allocated to the receiver by setting DXMTFCS = 1 in CSR15.

If DXMTFCS=0, the MAC Engine will calculate and append the FCS to the transmitted message. The receive message passed to the host will therefore contain an additional 4 bytes of FCS. In this loopback configuration, the receive circuitry cannot detect FCS errors if they occur.

If DXMTFCS=1, the last four bytes of the transmit message must contain the (software generated) FCS computed for the transmit data preceding it. The MAC Engine will transmit the data without addition of an FCS field, and the FCS will be calculated and verified at the receiver.

The loopback facilities of the MAC Engine allow full operation to be verified without disturbance to the network. Loopback operation is also affected by the state of the Loopback Control bits (LOOP, MENDECL, and INTL) in CSR15. This affects whether the internal MENDEC is considered part of the internal or external loopback path.

The multicast address detection logic uses the FCS generator circuit. Therefore, in the loopback mode(s), the multicast address detection feature of the MAC Engine, programmed by the contents of the Logical Address Filter (LADRF [63:0] in CSRs 8–11) can only be tested when DXMTFCS=1, allocating the FCS generator to the receiver. All other features operate identically in loopback as in normal operation, such as automatic transmit padding and receive pad stripping.

When performing an internal loopback, no frame will be transmitted to the network. However, when the PCnet-ISA+ controller is configured for internal loopback the receiver will not be able to detect network traffic. External loopback tests will transmit frames onto the network if the AUI port is selected, and the PCnet-PCI controller will receive network traffic while configured for external loopback when the AUI port is selected. Runt Packet Accept is automatically enabled when any loopback mode is invoked.

Loopback mode can be performed with any frame size. Runt Packet Accept is internally enabled (RPA bit in CSR124 is not affected) when any loopback mode is invoked. This is to be backwards compatible to the LANCE (Am7990) software.

When the 10BASE-T MAU is selected in external loopback mode, the collision detection is disabled. This is necessary, because a collision in a 10BASE-T system is defined as activity on the transmitter outputs and receiver inputs at the same time, which is exactly what occurs during external loopback.

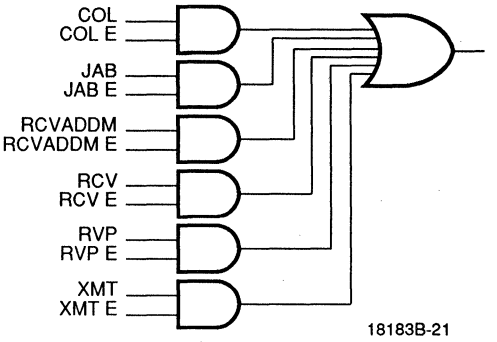
Since a 10BASE-T hub does not normally feed the station's transmitter outputs back into the station's receiver inputs, the use of external loopback in a 10BASE-T system usually requires some sort of external hardware that connects the outputs of the 10BASE-T MAU to its inputs.

LEDs

The PCnet-ISA+ controller's LED control logic allows programming of the status signals, which are displayed on 3 LED outputs. One LED ($\overline{LED0}$) is dedicated to displaying 10BASE-T Link Status. The status signals available are Collision, Jabber, Receive, Receive Polarity (active when receive polarity is okay), and Transmit. If more than one status signal is enabled, they are ORed together. An optional pulse stretcher is available for each programmable output. This allows emulation of the TPEX (Am79C98) and TPEX+ (Am79C100) LED outputs.

Signal	Behavior
LNKST	Active during Link OK Not active during Link Down
RCV	Active while receiving data
RVPOL	Active during receive polarity is OK Not active during reverse receive polarity
RCVADDM	Active during Receive with Address Match
XMT	Active while transmitting data

Each status signal is ANDed with its corresponding enable signal. The enabled status signals run to a common OR gate:



LED Control Logic

The output from the OR gate is run through a pulse stretcher, which consists of a 3-bit shift register clocked at 38 Hz. The data input of the shift register is at logic 0. The OR gate output asynchronously sets all three bits of the shift register when its output goes active. The output of the shift register controls the associated $\overline{LEDx}$ pin. Thus, the pulse stretcher provides an LED output of 52 ms to 78 ms.

Refer to the section "ISA Bus Configuration Registers" for information on LED control via the ISACSRs.

PCnet-ISA+ CONTROLLER REGISTERS

The PCnet-ISA+ controller implements all LANCE (Am7990) registers, plus a number of additional registers. The PCnet-ISA+ controller registers are compatible with the original LANCE, but there are some places where previously reserved LANCE bits are now used by the PCnet-ISA+ controller. If the reserved LANCE bits were used as recommended, there should be no compatibility problems.

Register Access

Internal registers are accessed in a two-step operation. First, the address of the register to be accessed is written into the register address port (RAP). Subsequent read or write operations will access the register pointed to by the contents of the RAP. The data will be read from (or written to) the selected register through the data port, either the register data port (RDP) for control and status registers (CSR) or the ISACSR register data port (IDP) for ISA control and status registers (ISACSR)

RAP: Register Address Port

Bit	Name	Description
15-7	RES	Reserved locations. Read and written as zeroes.
6-0	RAP	Register Address Port select. Selects the CSR or ISACSR location to be accessed. RAP is cleared by RESET.

Control and Status Registers

CSR0: PCnet-ISA+ Controller Status Register

Bit	Name	Description
15	ERR	Error is set by the ORing of BABL, CERR, MISS, and MERR. ERR remains set as long as any of the error flags are true. ERR is read only; write operations are ignored.
14	BABL	Babble is a transmitter time-out error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum length frame. BABL will be set if 1519 bytes or greater are transmitted. When BABL is set, IRQ is asserted if IENA = 1 and the mask bit BABLM (CSR3.14) is clear. BABL assertion will set the ERR bit. BABL is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. BABL is cleared by RESET or by setting the STOP bit.
13	CERR	Collision Error indicates that the collision inputs to the AUI port failed to activate within 20 network bit times after the chip terminated transmission (SQE Test). This feature is a transceiver test feature. CERR will be set in 10BASE-T mode during trasmit if in Link Fail state. CERR assertion will not result in an interrupt being generated. CERR assertion will set the ERR bit. CERR is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. CERR is cleared by RESET or by setting the STOP bit.
12	MISS	Missed Frame is set when PCnet-ISA+ controller has lost an incoming receive frame because a Receive Descriptor was not available. This bit is the only indication that receive data has been lost since there is no receive descriptor available for status information. When MISS is set, IRQ is asserted if IENA = 1 and the mask bit MISSM (CSR3.12) is clear. MISS assertion will set the ERR bit. MISS is set by the Buffer Management Unit and cleared by writing a "1". Writing a "0" has no effect. MISS is cleared by RESET or by setting the STOP bit.
11	MERR	Memory Error is set when PCnet-ISA+ controller is a bus master and has not received $\overline{\text{DACK}}$ assertion after 50 µs after DRQ assertion. Memory Error indicates that PCnet-ISA+ controller is not receiving bus mastership in time to prevent overflow/underflow conditions in the receive and transmit FIFOs. (MERR indicates a slightly different condition for the LANCE; for the LANCE MERR occurs when $\overline{\text{READY}}$ has not been asserted 25.6 µs after the address has been asserted.) When MERR is set, IRQ is asserted if IENA = 1 and the mask bit MERRM (CSR3.11) is clear.

MERR assertion will set the ERR bit.

MERR is set by the Bus Interface Unit and cleared by writing a "1". Writing a "0" has no effect. MERR is cleared by RESET or by setting the STOP bit.

10 RINT

Receive Interrupt is set after reception of a receive frame and toggling of the OWN bit in the last buffer in the Receive Descriptor Ring.

When RINT is set, IRQ is asserted if IENA = 1 and the mask bit RINTM (CSR3.10) is clear.

RINT is set by the Buffer Management Unit after the last receive buffer has been updated and cleared by writing a "1". Writing a "0" has no effect. RINT is cleared by RESET or by setting the STOP bit.

9 TINT

Transmit Interrupt is set after transmission of a transmit frame and toggling of the OWN bit in the last buffer in the Transmit Descriptor Ring.

When TINT is set, IRQ is asserted if IENA = 1 and the mask bit TINTM (CSR3.9) is clear.

TINT is set by the Buffer Management Unit after the last transmit buffer has been updated and cleared by writing a "1". Writing a "0" has no effect. TINT is cleared by RESET or by setting the STOP bit.

8 IDON

Initialization Done indicates that the initialization sequence has completed. When IDON is set, PCnet-ISA+ controller has read the Initialization block from memory.

When IDON is set, IRQ is asserted if IENA = 1 and the mask bit IDONM (CSR3.8) is clear.

IDON is set by the Buffer Management Unit after the initialization block has been read from memory and cleared by writing a "1". Writing a "0" has no effect. IDON is cleared by RESET or by setting the STOP bit.

7 INTR

Interrupt Flag indicates that one or more of the following interrupt causing conditions has occurred: BABL, MISS, MERR, MPCO, RCVCCO, RINT, TINT, IDON, JAB or TXSTRT; and its associated mask bit is clear. If IENA = 1 and INTR is set, IRQ will be active.

INTR is cleared automatically when the condition that caused interrupt is cleared.

INTR is read only. INTR is cleared by RESET or by setting the STOP bit.

6 IENA

Interrupt Enable allows IRQ to be active if the Interrupt Flag is set. If IENA = "0" then IRQ will be disabled regardless of the state of INTR.

IENA is set by writing a "1" and cleared by writing a "0". IENA is cleared by RESET or by setting the STOP bit.

5 RXON

Receive On indicates that the Receive function is enabled. RXON is set if DRX (CSR15.0) = "0" after the START bit is set. If INIT and START are set together, RXON will not be set until after the initialization block has been read in.

RXON is read only. RXON is cleared by RESET or by setting the STOP bit.

4 TXON

Transmit On indicates that the Transmit function is enabled. TXON is set if DTX (CSR15.1) = "0" after the START bit is set. If INIT and START are set together, TXON will not be set until after the initialization block has been read in.

TXON is read only. TXON is cleared by RESET or by setting the STOP bit.

3 TDMD

Transmit Demand, when set, causes the Buffer Management Unit to access the Transmit Descriptor Ring without waiting for the poll-time counter to elapse. If TXON is not enabled, TDMD bit will be reset and no Transmit Descriptor Ring access will occur. TDMD is required to be set if the DPOLL bit in CSR4 is set; setting TDMD while DPOLL = 0 merely hastens the PCnet-ISA+ controller's response to a Transmit Descriptor Ring Entry.

TDMD is set by writing a "1". Writing a "0" has no effect. TDMD will be cleared by the Buffer Management Unit when it fetches a Transmit Descriptor. TDMD is cleared by RESET or by setting the STOP bit.

2	STOP	STOP assertion disables the chip from all external activity. The chip remains inactive until either STRT or INIT are set. If STOP, STRT and INIT are all set together, STOP will override STRT and INIT. STOP is set by writing a "1" or by RESET. Writing a "0" has no effect. STOP is cleared by setting either STRT or INIT.
1	STRT	STRT assertion enables PCnet-ISA+ controller to send and receive frames, and perform buffer management operations. Setting STRT clears the STOP bit. If STRT and INIT are set together, PCnet-ISA+ controller initialization will be performed first. STRT is set by writing a "1". Writing a "0" has no effect. STRT is cleared by RESET or by setting the STOP bit.
0	INIT	INIT assertion enables PCnet-ISA+ controller to begin the initialization procedure which reads in the initialization block from memory. Setting INIT clears the STOP bit. If STRT and INIT are set together, PCnet-ISA+ controller initialization will be performed first. INIT is not cleared when the initialization sequence has completed. INIT is set by writing a "1". Writing a "0" has no effect. INIT is cleared by RESET or by setting the STOP bit.

CSR1: IADR[15:0]

Bit	Name	Description
15-0	IADR [15:0]	Lower address of the Initialization address register. Bit location 0 must be zero. Whenever this register is written, CSR16 is updated with CSR1's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR2: IADR[23:16]

Bit	Name	Description
15-8	RES	Reserved locations. Read and written as zero.
7-0	IADR [23:16]	Upper 8 bits of the address of the Initialization Block. Bit locations 15-8 must be written with zeros. Whenever this register is written, CSR17 is updated with CSR2's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR3: Interrupt Masks and Deferral Control

Bit	Name	Description
15	RES	Reserved location. Written as zero and read as undefined.
14	BABLM	Babble Mask. If BABLM is set, the BABL bit in CSR0 will be masked and will not set INTR flag in CSR0. BABLM is cleared by RESET and is not affected by STOP.
13	RES	Reserved location. Written as zero and read as undefined.
12	MISSM	Missed Frame Mask. If MISSM is set, the MISS bit in CSR0 will be masked and will not set INTR flag in CSR0. MISSM is cleared by RESET and is not affected by STOP.
11	MERRM	Memory Error Mask. If MERRM is set, the MERR bit in CSR0 will be masked and will not set INTR flag in CSR0. MERRM is cleared by RESET and is not affected by STOP.
10	RINTM	Receive Interrupt Mask. If RINTM is set, the RINT bit in CSR0 will be masked and will not set INTR flag in CSR0. RINTM is cleared by RESET and is not affected by STOP.
9	TINTM	Transmit Interrupt Mask. If TINTM is set, the TINT bit in CSR0 will be masked and will not set INTR flag in CSR0. TINTM is cleared by RESET and is not affected by STOP.
8	IDONM	Initialization Done Mask. If IDONM is set, the IDON bit in CSR0 will be masked and will not set INTR flag in CSR0. IDONM is cleared by RESET and is not affected by STOP.
7-6	RES	Reserved locations. Written as zero and read as undefined.

5	LAPPEN	Look Ahead Packet Processing (LAPPEN) . When set to a one, the LAPPEN bit will cause the PCnet-ISA+ controller to generate an interrupt following the descriptor write operation to the first buffer of a receive packet. This interrupt will be generated in addition to the interrupt that is generated following the descriptor write operation to the last buffer of a receive packet. The interrupt will be signaled through the RINT bit of CSR0. Setting LAPPEN to a one also enables the PCnet-ISA+ controller to read the STP bit of the receive descriptors. PCnet-ISA+ controller will use STP information to determine where it should begin writing a receive packet's data. Note that while in this mode, the PCnet-ISA+ controller can write intermediate packet data to buffers whose descriptors do not contain STP bits set to one. Following the write to the last descriptor used by a packet, the PCnet-ISA+ controller will scan through the next descriptor entries to locate the next STP bit that is set to a one. The PCnet-ISA+ controller will begin writing the next packet's data to the buffer pointed to by that descriptor. Note that because several descriptors may be allocated by the host for each packet, and not all messages may need all of the descriptors that are allocated between descriptors that contain STP = one, then some descriptors/buffers may be skipped in the ring. While performing the search for the next STP bit that is set to one, the PCnet-ISA+ controller will advance through the receive descriptor ring regardless of the state of ownership bits. If any of the entries that are examined during this search indicate PCnet-ISA+ will RESET the OWN bit to zero in these entries. If a scanned entry indicates host ownership with STP="0", then the PCnet-ISA+ controller will not alter the entry, but will advance to the next entry. When the STP bit is found to be true, but the descriptor that contains this setting is not owned by the PCnet-ISA+ controller, then the PCnet-ISA+ controller will stop advancing through the ring entries and begin periodic polling of this entry. When the STP bit is found to be true, and the descriptor that contains this setting is owned by the PCnet-ISA+ controller, then the PCnet-ISA+ controller will stop advancing through the ring entries, store the descriptor information that is has just read, and wait for the next receive to arrive. This behavior allows the host software to pre-assign buffer space in such a manner that the "header" portion of a receive packet will always be written to a particular memory area, and the "data" portion of a receive packet will always be written to a separate memory area. The interrupt is generated when the "header" bytes have been written to the "header" memory area. Read/Write accessible always. The LAPPEN bit will be reset zero by RESET and will unaffected by the STOP. See Appendix E for more information on LAPP.
4	DXMT2PD	Disable Transmit Two Part Deferral. (Described in the Media Access Management section). If DXMT2PD is set, Transmit Two Part Deferral will be disabled. DXMT2PD is cleared by RESET and is not affected by STOP.
3	EMBA	Enable Modified Back-off Algorithm. If EMBA is set, a modified back-off algorithm is implemented as described in the Media Access Management section. Read/Write accessible. EMBA is cleared by RESET and is not affected by STOP.
2-0	RES	Reserved locations. Written as zero and read as undefined.

CSR4: Test and Features Control

Bit	Name	Description
15	ENTST	Enable Test Mode operation. When ENTST is set, writing to test mode registers CSR124 and CSR126 is allowed, and other

register test functions are enabled. In order to set ENTST, it must be written with a "1" during the first write access to CSR4 after RESET. Once a "0" is written to this bit location, ENTST cannot be set until after the PCnet-ISA+ controller is reset.

ENTST is cleared by RESET.

14 DMAPLUS — When DMAPLUS = "1", the burst transaction counter in CSR80 is disabled. If DMAPLUS = "0", the burst transaction counter is enabled.

DMA-PLUS is cleared by RESET.

13 TIMER — Timer Enable Register. If TIMER is set, the Bus Timer Register, CSR82, is enabled. If TIMER is set, CSR82 must be written with a value. If TIMER is cleared, the Bus Timer Register is disabled.

TIMER is cleared by RESET.

12 DPOLL — Disable Transmit Polling. If DPOLL is set, the Buffer Management Unit will disable transmit polling. Likewise, if DPOLL is cleared, automatic transmit polling is enabled. If DPOLL is set, TDMD bit in CSR0 must be periodically set in order to initiate a manual poll of a transmit descriptor. Transmit descriptor polling will not take place if TXON is reset.

DPOLL is cleared by RESET.

11 APAD_XMT — Auto Pad Transmit. When set, APAD_XMT enables the automatic padding feature. Transmit frames will be padded to extend them to 64 bytes, including FCS. The FCS is calculated for the entire frame (including pad) and appended after the pad field. APAD_XMT will override the programming of the DXMTFCS bit (CSR15.3).

APAD_ XMT is reset by activation of the RESET pin.

10 ASTRP_RCV — ASTRP_RCV enables the automatic pad stripping feature. The pad and FCS fields will be stripped from receive frames and not placed in the FIFO.

ASTRP_ RCV is reset by activation of the RESET pin.

9 MFCO — Missed Frame Counter Overflow Interrupt.

This bit indicates the MFC (CSR112) has overflowed. Can be cleared by writing a "1" to this bit. Also cleared by RESET or setting the STOP bit. Writing a "0" has no effect.

8 MFCOM — Missed Frame Counter Overflow Mask.

If MFCOM is set, MFCO will not set INTR in CSR0.

MFCOM is set by Reset and is not affected by STOP.

7-6 RES — Reserved locations. Read and written as zero.

5 RCVCCO — Receive Collision Counter Overflow.

This bit indicates the Receive Collision Counter (CSR114) has overflowed. It can be cleared by writing a 1 to this bit. Also cleared by RESET or setting the STOP bit. Writing a 0 has no effect.

4 RCVCCOM — Receive Collision Counter Overflow Mask.

If RCVCCOM is set, RCVCCO will not set INTR in CSR0.

RCVCCOM is set by RESET and is not affected by STOP.

3 TXSTRT — Transmit Start status is set whenever PCnet-ISA+ controller begins trans- mission of a frame.

When TXSTRT is set, IRQ is asserted if IENA = 1 and the mask bit TXSTRTM (CSR4.2) is clear.

TXSTRT is set by the MAC Unit and cleared by writing a "1", setting RESET or setting the STOP bit. Writing a "0" has no effect.

2 TXSTRTM — Transmit Start Mask. If TXSTRTM is set, the TXSTRT bit in CSR4 will be masked and will not set INTR flag in CSR0.

TXS-TRTM is set by RESET and is not affected by STOP.

1 JAB — Jabber Error is set when the PCnet-ISA+ controller Twisted-pair MAU function exceeds an allowed transmission limit. Jabber is set by the TMAU cell and can only be asserted in 10BASE-T mode.

When JAB is set, IRQ is asserted if IENA = 1 and the mask bit JABM (CSR4.4) is clear.

		The JAB bit can be reset even if the jabber condition is still present. JAB is set by the TMAU circuit and cleared by writing a "1". Writing a "0" has no effect. JAB is also cleared by RESET or setting the STOP bit.
0	JABM	Jabber Error Mask. If JABM is set, the JAB bit in CSR4 will be masked and will not set INTR flag in CSR0. JABM is set by RESET and is not affected by STOP.

CSR6: RCV/XMT Descriptor Table Length

Bit	Name	Description
15-12	TLEN	Contains a copy of the transmit encoded ring length (TLEN) field read from the initialization block during PCnet-ISA+ controller initialization. This field is written during the PCnet-ISA+ controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. TLEN is only defined after initialization.
11-8	RLEN	Contains a copy of the receive encoded ring length (RLEN) read from the initialization block during PCnet-ISA+ controller initialization. This field is written during the PCnet-ISA+ controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. RLEN is only defined after initialization.
7-0	RES	Reserved locations. Read as zero. Write operations should not be performed.

CSR8: Logical Address Filter, LADRF[15:0]

Bit	Name	Description
15-0	LADRF[15:0]	Logical Address Filter, LADRF [15:0]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set.

CSR9: Logical Address Filter, LADRF[31:16]

Bit	Name	Description
15-0	LADRF[31:16]	Logical Address Filter, LADRF[31:16]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set.

CSR10: Logical Address Filter, LADRF[47:32]

Bit	Name	Description
15-0	LADRF[47:32]	Logical Address Filter, LADRF[47:32]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set.

CSR11: Logical Address Filter, LADRF[63:48]

Bit	Name	Description
15-0	LADRF[63:48]	Logical Address Filter, LADRF[63:48]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. Read/write accessible only when STOP bit is set.

CSR12: Physical Address Register, PADR[15:0]

Bit	Name	Description
15-0	PADR[15:0]	Physical Address Register, PADR[15:0]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set.

CSR13: Physical Address Register, PADR[31:16]

Bit	Name	Description
15-0	PADR[31:16]	Physical Address Register, PADR[31:16]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set.

CSR14: Physical Address Register, PADR[47:32]

Bit	Name	Description
15-0	PADR[47:32]	Physical Address Register, PADR[47:32]. Undefined until initialized either automatically by loading the initialization block or directly by an I/O write to this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set.

CSR15: Mode Register

Bit	Name	Description
		This register's fields are loaded during the PCnet-ISA+ controller initialization routine with the corresponding Initialization Block values. The register can also be loaded directly by an I/O write. Activating the RESET pin clears all bits of CSR15 to zero.
15	PROM	Promiscuous Mode. When PROM = "1", all incoming receive frames are accepted. Read/write accessible only when STOP bit is set.
14	DRCVBC	Disable Receive Broadcast. When set, disables the PCnet-ISA+ controller from receiving broadcast messages. Used for protocols that do not support broadcast addressing, except as a function of multicast. DRCVBC is cleared by activation of the RESET pin (broadcast messages will be received). Read/write accessible only when STOP bit is set.
13	DRCVPA	Disable Receive Physical Address. When set, the physical address detection (Station or node ID) of the PCnet-ISA+ controller will be disabled. Frames addressed to the nodes individual physical address will not be recognized (although the frame may be accepted by the EADI mechanism). Read/write accessible only when STOP bit is set.
12	DLNKTST	Disable Link Status. When DLNKTST = "1", monitoring of Link Pulses is disabled. When DLNKTST = "0", monitoring of Link Pulses is enabled. This bit only has meaning when the 10BASE-T network interface is selected. Read/write accessible only when STOP bit is set.
11	DAPC	Disable Automatic Polarity Correction. When DAPC = "1", the 10BASE-T receive polarity reversal algorithm is disabled. Likewise, when DAPC = "0", the polarity reversal algorithm is enabled. This bit only has meaning when the 10BASE-T network interface is selected. Read/write accessible only when STOP bit is set.
10	MENDECL	MENDEC Loopback Mode. See the description of the LOOP bit in CSR15. Read/write accessible only when STOP bit is set.
9	LRT/TSEL	Low Receive Threshold (T-MAU Mode only) Transmit Mode Select (AUI Mode only)
	LRT	Low Receive Threshold. When LRT = "1", the internal twisted pair receive thresholds are reduced by 4.5 dB below the standard 10BASE-T value (approximately 3/5) and the unsquelch threshold for the RXD circuit will be 180-312 mV peak. When LRT = "0", the unsquelch threshold for the RXD circuit will be the standard 10BASE-T value, 300-520 mV peak. In either case, the RXD circuit post squelch threshold will be one half of the unsquelch threshold. This bit only has meaning when the 10BASE-T network interface is selected.

Read/write accessible only when STOP bit is set. Cleared by RESET.

TSEL — Transmit Mode Select. TSEL controls the levels at which the AUI drivers rest when the AUI transmit port is idle. When TSEL = 0, DO+ and DO- yield "zero" differential to operate transformer coupled loads (Ethernet 2 and 802.3). When TSEL = 1, the DO+ idles at a higher value with respect to DO- , yielding a logical HIGH state (Ethernet 1).

This bit only has meaning when the AUI network interface is selected. Not available under Auto-Select Mode.

Read/write accessible only when STOP bit is set. Cleared by RESET.

8-7 PORTSEL [1:0] — Port Select bits allow for software controlled selection of the network medium. PORTSEL active only when Media-Select Bit set to 0 in ISACSR2.

Read/write accessible only when STOP bit is set. Cleared by RESET.

The network port configuration are as follows:

PORTSEL[1:0]	Network Port
0 0	AUI
0 1	10BASE-T
1 0	GPSI*
1 1	Reserved

**Refer to the section on General Purpose Serial Interface for detailed information on accessing GPSI.*

6 INTL — Internal Loopback. See the description of LOOP, CSR15.2.

Read/write accessible only when STOP bit is set.

5 DRTY — Disable Retry. When DRTY = "1", PCnet-ISA+ controller will attempt only one transmission. If DRTY = "0", PCnet-ISA+ controller will attempt to transmit 16 times before signaling a retry error.

Read/write accessible only when STOP bit is set.

4 FCOLL — Force Collision. This bit allows the collision logic to be tested. PCnet-ISA+ controller must be in internal loopback for FCOLL to be valid. If FCOLL = "1", a collision will be forced during loopback transmission attempts; a Retry Error will ultimately result. If FCOLL = "0", the Force Collision logic will be disabled.

Read/write accessible only when STOP bit is set.

3 DXMTFCS — Disable Transmit CRC (FCS). When DXMTFCS = 0, the transmitter will generate and append a FCS to the transmitted frame. When DXMTFCS = 1, the FCS logic is allocated to the receiver and no FCS is generated or sent with the transmitted frame.

See also the ADD_FCS bit in TMD1. If DXMTFCS is set, no FCS will be generated. If both DXMTFCS is set and ADD_FCS is clear for a particular frame, no FCS will be generated. If ADD_FCS is set for a particular frame, the state of DXMTFCS is ignored and a FCS will be appended on that frame by the transmit circuitry.

In loopback mode, this bit determines if the transmitter appends FCS or if the receiver checks the FCS.

This bit was called DTCR in the LANCE (Am7990).

Read/write accessible only when STOP bit is set.

2 LOOP — Loopback Enable allows PCnet-ISA+ controller to operate in full duplex mode for test purposes. When LOOP = "1", loopback is enabled. In combination with INTL and MENDECL, various loopback modes are defined as follows:

LOOP	INTL	MENDECL	Loopback Mode
0	X	X	Non-loopback
1	0	X	External Loopback
1	1	0	Internal Loopback Include MENDEC
1	1	1	Internal Loopback Exclude MENDEC

Read/write accessible only when STOP bit is set. LOOP is cleared by RESET.

1 DTX — Disable Transmit. If this bit is set, the PCnet-ISA+ controller will not access the Transmit Descriptor Ring and, therefore, no transmissions will occur. DTX = "0" will set TXON bit (CSR0.4) after STRT (CSR0.1) is asserted. DTX is defined after the initialization block is read.

Read/write accessible only when STOP bit is set.

0 DRX — Disable Receiver. If this bit is set, the PCnet-ISA+ controller will not access the Receive Descriptor Ring and, therefore, all receive frame data are ignored. DRX = "0" will set RXON bit (CSR0.5) after STRT (CSR0.1) is asserted. DRX is defined after the initialization block is read.

Read/write accessible only when STOP bit is set.

CSR16: Initialization Block Address Lower

Bit	Name	Description
15-0	IADR	Lower 16 bits of the address of the Initialization Block. Bit location 0 must be zero. This register is an alias of CSR1. Whenever this register is written, CSR1 is updated with CSR16's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR17: Initialization Block Address Upper

Bit	Name	Description
15-8	RES	Reserved locations. Written as zero and read as undefined.
7-0	IADR	Upper 8 bits of the address of the Initialization Block. Bit locations 15-8 must be written with zeros. This register is an alias of CSR2. Whenever this register is written, CSR2 is updated with CSR17's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR18-19: Current Receive Buffer Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	CRBA	Contains the current receive buffer address to which the PCnet-ISA+ controller will store incoming frame data. Read/write accessible only when STOP bit is set.

CSR20-21: Current Transmit Buffer Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	CXBA	Contains the current transmit buffer address from which the PCnet-ISA+ controller is transmitting. Read/write accessible only when STOP bit is set.

CSR22-23: Next Receive Buffer Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	NRBA	Contains the next receive buffer address to which the PCnet-ISA+ controller will store incoming frame data. Read/write accessible only when STOP bit is set.

CSR24-25: Base Address of Receive Ring

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	BADR	Contains the base address of the Receive Ring. Read/write accessible only when STOP bit is set.

CSR26-27: Next Receive Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	NRDA	Contains the next RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR28-29: Current Receive Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	CRDA	Contains the current RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR30-31: Base Address of Transmit Ring

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	BADX	Contains the base address of the Transmit Ring. Read/write accessible only when STOP bit is set.

CSR32-33: Next Transmit Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	NXDA	Contains the next TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR34-35: Current Transmit Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	CXDA	Contains the current TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR36-37: Next Next Receive Descriptor Address

Bit	Name	Description
31-0	NNRDA	Contains the next next RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR38-39: Next Next Transmit Descriptor Address

Bit	Name	Description
31-0	NNXDA	Contains the next next TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR40-41: Current Receive Status and Byte Count

Bit	Name	Description
31-24	CRST	Current Receive Status. This field is a copy of bits 15:8 of RMD1 of the current receive descriptor. Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined.
11-0	CRBC	Current Receive Byte Count. This field is a copy of the BCNT field of RMD2 of the current receive descriptor. Read/write accessible only when STOP bit is set.

CSR42-43: Current Transmit Status and Byte Count

Bit	Name	Description
31-24	CXST	Current Transmit Status. This field is a copy of bits 15:8 of TMD1 of the current transmit descriptor. Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined.
11-0	CXBC	Current Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the current transmit descriptor.

Read/write accessible only when STOP bit is set.

CSR44-45: Next Receive Status and Byte Count

Bit	Name	Description
31-24	NRST	Next Receive Status. This field is a copy of bits 15:8 of RMD1 of the next receive descriptor. Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined.
11-0	NRBC	Next Receive Byte Count. This field is a copy of the BCNT field of RMD2 of the next receive descriptor. Read/write accessible only when STOP bit is set.

CSR46: Poll Time Counter

Bit	Name	Description
15-0	POLL	Poll Time Counter. This counter is incremented by the PCnet-ISA+ controller microcode and is used to trigger the descriptor ring polling operation of the PCnet-ISA+ controller. Read/write accessible only when STOP bit is set.

CSR47: Polling Interval

Bit	Name	Description
31-16	RES	Reserved locations. Written as zero and read as undefined.
15-0	POLLINT	Polling Interval. This register contains the time that the PCnet-ISA+ controller will wait between successive polling operations. The POLLINT value is expressed as the two's complement of the desired interval, where each bit of POLLINT represents one-half of an XTAL1 period of time. POLLINT[3:0] are ignored. (POLINT[16] is implied to be a one, so POLLINT[15] is significant, and does not represent the sign of the two's complement POLLINT value.) The default value of this register is 0000. This corresponds to a polling interval of 32,768 XTAL1 periods. The POLINT value of 0000 is created during the microcode initialization routine, and therefore might not be seen when reading CSR47 after RESET. If the user desires to program a value for POLLINT other than the default, then the correct procedure is to first set INIT only in CSR0. Then, when the initialization sequence is complete, the user must set STOP in CSR0. Then the user may write to CSR47 and then set STRT in CSR0. In this way, the default value of 0000 in CSR47 will be overwritten with the desired user value. Read/write accessible only when STOP bit is set.

CSR48-49: Temporary Storage

Bit	Name	Description
31-0	TMP0	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR50-51: Temporary Storage

Bit	Name	Description
31-0	TMP1	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR52-53: Temporary Storage

Bit	Name	Description
31-0	TMP2	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR54-55: Temporary Storage

Bit	Name	Description
31-0	TMP3	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR56-57: Temporary Storage

Bit	Name	Description
31-0	TMP4	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR58-59: Temporary Storage

Bit	Name	Description
31-0	TMP5	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR60-61: Previous Transmit Descriptor Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	PXDA	Contains the previous TDRE address pointer. The PCnet-ISA+ controller has the capability to stack multiple transmit frames. Read/write accessible only when STOP bit is set.

CSR62-63: Previous Transmit Status and Byte Count

Bit	Name	Description
31-24	PXST	Previous Transmit Status. This field is a copy of bits 15:8 of TMD1 of the previous transmit descriptor. Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined. Accessible only when STOP bit is set.
11-0	PXBC	Previous Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the previous transmit descriptor. Read/write accessible only when STOP bit is set.

CSR64-65: Next Transmit Buffer Address

Bit	Name	Description
31-24	RES	Reserved locations. Written as zero and read as undefined.
23-0	NXBA	Contains the next transmit buffer address from which the PCnet-ISA+ controller will transmit an outgoing frame. Read/write accessible only when STOP bit is set.

CSR66-67: Next Transmit Status and Byte Count

Bit	Name	Description
31-24	NXST	Next Transmit Status. This field is a copy of bits 15:8 of TMD1 of the next transmit descriptor. Read/write accessible only when STOP bit is set.
23-12	RES	Reserved locations. Written as zero and read as undefined. Accessible only when STOP bit is set.
11-0	NXBC	Next Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the next transmit descriptor. Read/write accessible only when STOP bit is set.

CSR68-69: Transmit Status Temporary Storage

Bit	Name	Description
31-0	XSTMP	Transmit Status Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR70-71: Temporary Storage

Bit	Name	Description
31-0	TMP8	Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR72: Receive Ring Counter

Bit	Name	Description
15-0	RCVRC	Receive Ring Counter location. Contains a Two's complement binary number used to number the current receive descriptor. This counter interprets the value in CSR76 as pointing to the first descriptor; a two's complement value of -1 (FFFFh) corresponds to the last descriptor in the ring. Read/write accessible only when STOP bit is set.

CSR74: Transmit Ring Counter

Bit	Name	Description
15-0	XMTRC	Transmit Ring Counter location. Contains a Two's complement binary number used to number the current transmit descriptor. This counter interprets the value in CSR78 as pointing to the first descriptor; a two's complement value of -1 (FFFFh) corresponds to the last descriptor in the ring. Read/write accessible only when STOP bit is set.

CSR76: Receive Ring Length

Bit	Name	Description
15-0	RCVRL	Receive Ring Length. Contains the Two's complement of the receive descriptor ring length. This register is initialized during the PCnet-ISA+ controller initialization routine based on the value in the RLEN field of the initialization block. This register can be manually altered; the actual receive ring length is defined by the current value in this register. Read/write accessible only when STOP bit is set.

CSR78: Transmit Ring Length

Bit	Name	Description
15-0	XMTRL	Transmit Ring Length. Contains the two's complement of the transmit descriptor ring length. This register is initialized during the PCnet-ISA+ controller initialization routine based on the value in the TLEN field of the initialization block. This register can be manually altered; the actual transmit ring length is defined by the current value in this register. Read/write accessible only when STOP bit is set.

CSR80: Burst and FIFO Threshold Control

Bit	Name	Description
15-14	RES	Reserved locations. Read as ones. Written as zero.
13-12	RCVFW[1:0]	Receive FIFO Watermark. RCVFW controls the point at which ISA bus receive DMA is requested in relation to the number of received bytes in the receive FIFO. RCVFW specifies the number of bytes which must be present (once the frame has been verified as a non-runt) before receive DMA is requested. Note however that in order for receive DMA to be performed for a new frame, at least 64 bytes must have been received. This effectively avoids having to react to receive frames which are runts or suffer a collision during the slot time (512 bit times). If the Runt Packet Accept feature is enabled, receive DMA will be requested as soon as either the RCVFW threshold is reached, or a complete valid receive frame is detected (regardless of length). RCVFW is set to a value of 10b (64 bytes) after RESET. Read/write accessible only when STOP bit is set.

RCVFW[1:0]	Bytes Received
00	16
01	32
10	64
11	Reserved

11-10	XMTSP[1:0]	Transmit Start Point. XMTSP controls the point at which preamble transmission attempts commence in relation to the number of bytes written to the transmit FIFO for the current transmit frame. When the entire frame is in the FIFO, transmission will start regardless of the value in XMTSP. XMTSP is given a value of 10b (64 bytes) after RESET. Regardless of XMTSP, the FIFO will not internally over

write its data until at least 64 bytes (or the entire frame if <64 bytes) have been transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be re-written to the transmit FIFO, and re-tries will be handled autonomously by the MAC. This bit is read/write accessible only when the STOP bit is set.

XMTSP[1:0]	Bytes Written
00	4
01	16
10	64
11	112

9-8 XMTFW[1:0] Transmit FIFO Watermark. XMTFW specifies the point at which transmit DMA stops, based upon the number of write cycles that could be performed to the transmit FIFO without FIFO overflow. Transmit DMA is allowed at any time when the number of write cycles specified by XMTFW could be executed without causing transmit FIFO overflow. XMTFW is set to a value of 00b (8 cycles) after hardware RESET. Read/write accessible only when STOP bit is set.

XMTFW[1:0]	Write Cycles
00	8
01	16
10	32
11	Reserved

7-0 DMABR DMA Burst Register. This register contains the maximum allowable number of transfers to system memory that the Bus Interface will perform during a single DMA cycle. The Burst Register is not used to limit the number of transfers during Descriptor transfers. A value of zero will be interpreted as one transfer. During RESET a value of 16 is loaded in the BURST register. If DMAPLUS (CSR4.14) is set, the DMA Burst Register is disabled.

When the Bus Activity Timer register (CSR82: DMABAT) is enabled, the PCnet-ISA+ controller will relinquish the bus when either the time specified in DMABAT has elapsed or the number of transfers specified in DMABR have occured.

Read/write accessible only when STOP bit is set.

CSR82: Bus Activity Timer

Bit	Name	Description
15-0	DMABAT	Bus Activity Timer. If the TIMER bit in CSR4 is set, this register contains the maximum allowable time that the PCnet-ISA+ controller will take up on the system bus during FIFO data transfers in each bus mastership period. The DMABAT starts counting upon receipt of $\overline{DACK}$ from the host system. The DMABAT Register does not limit the number of transfers during Descriptor transfers. A value of zero will limit the PCnet-ISA+ controller to one bus cycle per mastership period. A non-zero value is interpreted as an unsigned number with a resolution of 100 ns. For instance, a value of 51 µs would be programmed with a value of 510. When the TIMER bit in CSR4 is set, DMABAT is enabled and must be initialized by the user. The DMABAT register is undefined until written. When the Bus Activity Timer register (CSR82: DMABAT) is enabled, the PCnet-ISA+ control- ler will relinquish the bus when either the time specified in DMABAT has elapsed or the number of transfers specified in DMABR have occured. When ENTST (CSR4.15) is asserted, all writes to this register will automatically perform a decrement cycle. Read/write accessible only when STOP bit is set.

CSR84-85: DMA Address

Bit	Name	Description
31-0	DMABA	DMA Address Register. This register contains the address of system memory for the current DMA cycle. The Bus Interface Unit controls the Address

Register by issuing increment commands to increment the memory address for sequential operations. The DMABA register is undefined until the first PCnet-ISA+ controller DMA operation.

This register has meaning only if the PCnet-ISA+ controller is in Bus Master Mode.

Read/write accessible only when STOP bit is set.

CSR86: Buffer Byte Counter

Bit	Name	Description
15-12	RES	Reserved, Read and written with ones.
11-0	DMABC	DMA Byte Count Register. Contains the Two's complement of the current size of the remaining transmit or receive buffer in bytes. This register is incremented by the Bus Interface Unit. The DMABC register is undefined until written. Read/write accessible only when STOP bit is set.

CSR88-89: Chip ID

Bit	Name	Description
31-28		Version. This 4-bit pattern is silicon revision dependent.
27-12		Part number. The 16-bit code for the PCnet-ISA+ controller is 0010001001100000b.
11-1		Manufacturer ID. The 11-bit manufacturer code for AMD is 00000000001b. This code is per the JEDEC Publication 106-A.
0		Always a logic 1. This register is exactly the same as the Chip ID register in the JTAG description.

CSR92: Ring Length Conversion

Bit	Name	Description
15-0	RCON	Ring Length Conversion Register. This register performs a ring length conversion from an encoded value as found in the initialization block to a Two's complement value used for internal counting. By writing bits 15–12 with an encoded ring length, a Two's complemented value is read. The RCON register is undefined until written. Read/write accessible only when STOP bit is set.

CSR94: Transmit Time Domain Reflectometry Count

Bit	Name	Description
15-10	RES	Reserved locations. Read and written as zero.
9-0	XMTTDR	Time Domain Reflectometry reflects the state of an internal counter that counts from the start of transmission to the occurrence of loss of carrier. TDR is incremented at a rate of 10 MHz. Read accessible only when STOP bit is set. Write operations are ignored. XMTTDR is cleared by RESET.

CSR96-97: Bus Interface Scratch Register 0

Bit	Name	Description
31-0	SCR0	This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. The SCR0 register is undefined until written. Read/write accessible only when STOP bit is set.

CSR98-99: Bus Interface Scratch Register 1

Bit	Name	Description
31-0	SCR1	This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. Read/write accessible only when STOP bit is set.

CSR104-105: SWAP

Bit	Name	Description
31-0	SWAP	This register performs word and byte swapping depending upon if

32-bit or 16-bit internal write operations are performed. This register is used internally by the BIU/BMU as a word or byte swapper. The swap register can perform 32-bit operations that the PC can not; the register is externally accessible for test reasons only. CSR104 holds the lower 16 bits and CSR105 holds the upper 16 bits.

The swap function is defined as follows:

Internal Write Operation	SWAP Register Result
32-Bit word	SRC[31:16] → SWAP[15:0] SRC[15:0] → SWAP[31:16]
Lower 16-Bit (CSR104)	SRC[15:8] → SWAP[7: 0] SRC[7:0] → SWAP[15:8]

Read/write accessible only when STOP bit is set.

CSR108-109: Buffer Management Scratch

Bit	Name	Description
31-0	BMSCR	The Buffer Management Scratch register is used for assembling Receive and Transmit Status. This register is also used as the primary scan register for Buffer Management Test Modes. BMSCR register is undefined until written. Read/write accessible only when STOP bit is set.

CSR112: Missed Frame Count

Bit	Name	Description
15-0	MFC	Counts the number of missed frames. This register is always readable and is cleared by STOP. A write to this register performs an increment when the ENTST bit in CSR4 is set. When MFC is all 1's (65535) and a missed frame occurs, MFC increments to 0 and sets MFC0 bit (CSR4.9).

CSR114: Receive Collision Count

Bit	Name	Description
15-0	RCVCC	Counts the number of Receive collisions seen, regular and late. This register is always readable and is cleared by STOP. A write to this register performs an increment when the ENTST bit in CSR4 is set. When RCVCC is all 1's (65535) and a receive collision occurs, RCVCC increments to 0 and sets RCVCC0 bit (CSR4.5)

CSR124: Buffer Management Unit Test

Bit	Name	Description
		This register is used to place the BMU/BIU into various test modes to support Test/Debug. This register is writeable when the ENTST bit in CSR4 is set.
15-5	RES	Reserved locations. Written as zero and read as undefined.
4	GPSIEN	This mode places the PCnet-ISA+ controller in the GPSI Mode. This mode will reconfigure the External Address Pins so that the GPSI port is exposed. This allows bypassing the MENDEC- TMAU logic. This bit should only be set if the external logic supports GPSI operation. Damage to the device may occur in a non-GPSI configuration. Refer to the GPSI section.
3	RPA	Runt Packet Accept. This bit forces the CORE receive logic to accept Runt Packets. This bit allows for faster testing.
2-0	RES	For test purposes only. Reserved locations. Written as zero and read as undefined.

ISA Bus Configuration Registers

The ISA Bus Data Port (IDP) allows access to registers which are associated with the ISA bus. These registers are called ISA Bus Configuration Registers (ISACSRs), and are indexed by the value in the Register Address Port (RAP). The table below defines the ISACSRs which

can be accessed. All registers are 16 bits. The "Default" value is the value in the register after reset and is hexadecimal.

Refer to the section "LEDs" for information on LED control logic.

ISACSR	MNEMONIC	Default	Name
0	MSRDA	0005H	Master Mode Read Active
1	MSWRA	0005H	Master Mode Write Active
2	MC	0002H	Miscellaneous Configuration
3	EC	8000H*	EEPROM Configuration
4	LED0	0000H	Link Integrity
5	LED1	0084H	Default: RCV
6	LED2	0008H	Default: RCVPOL
7	LED3	0090H	Default: XMT
8	SC	0000H	Software Configuration (Read-Only register)

**This value can be 0000H for systems that do not support EEPROM option.*

ISACSR0: Master Mode Read Active

Bit	Name	Description
3-0	MSRDA	This register is used to tune the $\overline{\text{MEMR}}$ command signal active time. The value stored in MSRDA defines the number of 50 ns periods that the command signal is active. The default value of 5h indicates 250 ns pulse widths. A value of 0 or 1 will generate 50 ns wide commands.
15-4	RES	Reserved locations. Written as zero and read as undefined.

ISACSR1: Master Mode Write Active

Bit	Name	Description
3-0	MSWRA	This register is used to tune the $\overline{\text{MEMW}}$ command signal active time. The value stored in MSWRA defines the number of 50 ns periods that the command signal is active. The default value of 5h indicates 250 ns pulse widths. A value of 0 or 1 will generate 50 ns wide commands.
15-4	RES	Reserved locations. Written as zero and read as undefined.

ISACSR2: Miscellaneous Configuration

Bit	Name	Description
15	MODE_STATUS	Mode Status. This is a read-only register which indicates whether the PCnet-ISA+ is configured in shared memory mode. A set condition indicates shared-memory while a clear condition indicates bus-master condition.
14	TMAU_LOOPE	10BASE-T External Loop back Enable. This bit is usable only when 10BASE-T is selected AND PCnet-ISA+ is in external loop back. External loop back is set during initialization via the MODE register. When TMAU_LOOPE is set, a board level test is enabled via a loop back clip which ties the 10BASE-T RJ45 transmit pair to the receiver pair. This will test all external components (i.e. transformers, resistors, etc.) of the 10BASE-T path. TMAU_LOOPE assertion is not suitable for live network tests. When TMAU_LOOPE is deasserted, default condition, external loop back in 10BASE-T is allowed.
13	Reserved	Written with zero and read as undefined.
12	SLOT_ID	Slot Identification. This is a read-only register bit which indicates if PCnet-ISA+ is either in an 16 or 8 bit slot. Reading a one indicates an 8 bit slot. Zero indicates a 16-bit slot. (SLOT_ID bit is not valid after the INIT bit is set in CSR0.)
11	ISA_PROTECT	ISA Protect. When set, the ISACSR's 0-2 and 4-7 are protected from being written over by software drivers. When ISA_PROTECT is cleared, ISACSR's 0-7 are allowed to be written over by software and reset by reading the Software reset I/O location. (Default is zero)
10	EISA_DECODE	EISA Decode. This control bit allows EISA product identifier registers 12-bit decode xC80 - xC83 (4 Bytes). Default is zero.
9	P&P_ACT	Plug and Play Active. When this bit is set, PCnet-ISA+ will become active after serially reading the EEPROM. If check sum failure exist, PCnet-ISA+ will not beome active and alternate

access method to Plug and Play registers will occur. Default is zero.

8 APWEN — Address PROM Write Enable. It is reset to zero by RESET. When asserted, this pin allows write access to the internal Address PROM RAM. APWEN is used also to protect the Flash device from write cycles. When programming of the Flash device is required, the APWEN bit needs to be set. When reset, this pin protects the internal Address PROM RAM, and external Flash device from being overwritten.

7 EISA_LVL — EISA Level. This bit is a read-only register. It indicates if the level or edge sensitive interrupts have been selected. A set condition indicates level sensitive interrupts. A clear condition indicates ISA edge.

6 DSDBUS — Disable Staggered Data Bus. When this bit is a zero, the data bus driver timing is staggered from the address bus driver timing in Bus Master mode. When this bit is a one, the data bus is not staggered. It is similar to the PCnet-ISA (Am79C960) timing. This bit is reset to zero. For most applications, this bit should not have to be set.

5 10BASE5_SEL — 10BASE5 Select. When this bit is a one, the DC to DC converter will be deselected via the DXCVR pin. When 10BASE5_SEL is a zero, the DC to DC converter will be selected via the DXCVR bit when the AUI port is selected to support a DC-DC converter for 10BASE2 MUAs. When 10BASE-T port is selected by whatever means, DXCR pin will high independent of the bit selected by the driver software mode register, MEDSEL bits, and Auto Selection process. 10BASE5_SEL is reset to zero.

4 ISAINACT — ISAINACT allows for reduced inactive timing appropriate for modern ISA machines. ISAINACT is cleared when RESET is asserted. When ISAINACT is a zero, tMMR3 and tMMW3 parameters are nominally 200 ns, which is compatible with EISA system. When ISAINACT is set by writing a one, tMMR3 and tMMW3 are nominally set to 100 ns.

3 EADISEL — EADI Select. Enables EADI match mode.

When EADI mode is selected, the pins named LED1, LED2, and LED3 change in function while LED0 continues to indicate 10BASE-T Link Status.

LED	EADI Function
1	SF/BD
2	SRD
3	SRDCLK

2 AWAKE — Auto-Wake. If AWAKE = "1", the 10BASE-T receive circuitry is active during sleep and listens for Link Pulses. LED0 indicates Link Status and goes active if the 10BASE-T port comes of out of "link fail" state. This LED0 pin can be used by external circuitry to re-enable the PCnet-ISA+ controller and/or other devices.

When AWAKE = "0", the Auto-Wake circuity is disabled. This bit only has meaning when the 10BASE-T network interface is selected.

1,0 MEDSEL — Media Select. It was previously defined as ASEL (Auto Select) and XMAUSEL (External MAU Select) in the PCnet-ISA. They are now combined together and defined to be software compatible with ASEL and XMAUSEL in the PCnet-ISA (Am79C960).

MEDSEL (1:0)	Function
0 0	Software Select (Mode Reg, CSR15)
0 1	10BASE-T Port
1 0	Auto Selection (Default)
1 1	AUI Port

ISACSR3: EEPROM Configuration

Bit	Name	Description
15	EE_VALID	EEPROM Valid. This bit is a read-only register. When a one is read, EE_PROM has a valid checksum. The sum of the total bytes reads should equals FF hex. When a zero is read, checksum failed, or SHFTBUSY pin was sampled with a zero which indicates no EEPROM present.
14	EE_LOAD	EEPROM Load. When written with a one, the device will load the EE_PROM into the

		PCnet-ISA+, performing self configuration. This command must be last write to ISACSR3 Register. PCnet-ISA+ will not respond to any slave commands while loading the EE_PROM register. EE_LOAD will be reset with a zero after EE_PROM is read. It takes approximately, 1.4 ms for serial EEPROM load process to complete.
13–5	N/A	Reserved. Read and written as zeros.
4	EE_EN	EEPROM Enable. When EE_EN is written with a one, the lower three bits of PRDB becomes SK, DI and DO, respectively. EECS and SHFBUSY are controlled by the software select bits. This bit must be written with a one to write to or read from the EEPROM. PCnet-ISA+ should be in the STOP state when EE_EN is written. When EN_EN is cleared, DI/DO, SK, EECS and SHFBUSY have no control.
3	SHFBUSY	Shift Busy. SHFBUSY allows for the control of the SHFBUSY pin. When a one is written, SHFBUSY goes high provided EE_EN is a 1. When a zero is written, SHFBUSY is held to a zero. When EE_EN is cleared, SHFBUSY will maintain the last value programmed. (Refer to Bit 4 above, EE_EN, for detailed use of this bit.)
2	EECS	EEPROM Chip Select. EECS asserts the chip select to the Serial EEPROM. (Refer to Bit 4 above, EE_EN, for detailed use of this bit.)
1	SK	Serial Shift Clock. SK controls the SK input to the Serial EEPROM and the optional External Shift Logic. (Refer to Bit 4 above, EE_EN, for detailed use of this bit.)
0	DI/DO	Serial Shift Data In and Serial Shift Data Out. When written, this bit controls the DI input of the serial EEPROM. When read, this bit represents the DO value of the serial EEPROM. (Refer to Bit 4 above, EE_EN, for detailed use of this bit.)

ISACSR4: LED0 Status (Link Integrity)

Bit	Name	Description
		ISACSR4 is a non-programmable register that uses one bit to reflect the status of the $\overline{\text{LED0}}$ pin. This pin defaults to twisted pair MAU Link Status (LNKST) and is not programmable.
15	LNKST	LNKST is a read-only register bit that indicates whether the Link Status LED is asserted. When LNKST is read as zero, the Link Status LED is not asserted. When LNKST is read as one, the Link Status LED is asserted, indicating good 10BASE-T integrity.
14-0	RES	Reserved locations. Written as zero, read as undefined.

ISACSR5: $\overline{\text{LED1}}$ Status

Bit	Name	Description
		ISACSR5 controls the function(s) that the $\overline{\text{LED1}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. ISACSR5 defaults to Receive Status (RCV) with pulse stretcher enabled (PSE = 1) and is fully programmable.
15	LEDOUT	Indicates the current (non-stretched) state of the function(s) generated. Read only.
14-8	RES	Reserved locations. Read and written as zero.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination for each enabled function occurrence. 0 is disabled, 1 is enabled.
6	RES	Reserved locations. Read and written as zero.
5	RCVADDM	Receive Address Match. This bit when set allows for LED control of only receive packets which match internal address match.
4	XMT E	Enable Transmit Status Signal. Indicates PCnet-ISA+ controller transmit activity .

0 disables the signal, 1 enables the signal.

3 RVPOL E — Enable Receive Polarity Signal.

Enables LED pin assertion when receive polarity is correct on the 10BASE-T port. Clearing the bit indicates this function is to be ignored.

2 RCV E — Enable Receive Status Signal. Indicates receive activity on the network.

0 disables the signal, 1 enables the signal.

1 JAB E — Enable Jabber Signal. Indicates the PCnet-ISA+ controller is jabbering on the network.

0 disables the signal, 1 enables the signal.

0 COL E — Enable Collision Signal. Indicates collision activity on the network.

0 disables the signal, 1 enables the signal.

ISACSR6: $\overline{\text{LED2}}$ Status

Bit	Name	Description
		ISACSR6 controls the function(s) that the $\overline{\text{LED2}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. ISACSR6 defaults to twisted pair MAU Receive Polarity (RCVPOL) with pulse stretcher enabled (PSE = 1) and is fully programmable.
15	LEDOUT	Indicates the current (non-stretched) state of the function(s) generated. Read only.
14	LEDXOR	This bit when set causes LED2 to be an active high signal when asserted. When this bit is cleared, LED2 will be active low when asserted. (***Note:*** *This bit when used in conjunction with the RVPOLE bit (Bit 3) of ISACSR5, ISACSR6, and ISACSR7 can be used to create a "Polarity Bad" LED.*)

RVPOLE	LEDXOR	Result
0	X	10BASE-T polarity function ignored
1	0	LED1 pin low with "Good" 10BASE-T polarity (LED on)
1	1	LED1 pin high with "Good" 10BASE-T polarity (LED off)

13-8 RES — Reserved locations. Read and written as zero.

7 PSE — Pulse Stretcher Enable. Extends the LED illumination for each enabled function occurrence.

0 is disabled, 1 is enabled.

6 RES — Reserved locations. Read and written as zero.

5 RCVADDM — Receive Address Match. This bit when set allows for LED control of only receive packets that match internal address match.

4 XMT E — Enable Transmit Status Signal. Indicates PCnet-ISA+ controller transmit activity .

0 disables the signal, 1 enables the signal.

3 RVPOL E — Enable Receive Polarity Signal.

Enables LED pin assertion when receive polarity is correct on the 10BASE-T port. Clearing the bit indicates this function is to be ignored.

2 RCV E — Enable Receive Status Signal. Indicates receive activity on the network.

0 disables the signal, 1 enables the signal.

1 JAB E — Enable Jabber Signal. Indicates the PCnet-ISA+ controller is jabbering on the network.

0 disables the signal, 1 enables the signal.

0 COL E — Enable Collision Signal. Indicates collision activity on the network.

0 disables the signal, 1 enables the signal.

ISACSR7: $\overline{\text{LED3}}$ Status

Bit	Name	Description
		ISACSR7 controls the function(s) that the $\overline{\text{LED3}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. ISACSR7 defaults to Transmit Status (XMT) with pulse stretcher enabled (PSE = 1) and is fully programmable.
15	LEDOUT	Indicates the current (non-stretched) state of the function(s) generated. Read only.
14-8	RES	Reserved locations. Read and written as zero.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination for each enabled function occurrence. 0 is disabled, 1 is enabled.
6	RES	Reserved locations. Read and written as zero.
5	RCVADDM	Receive Address Match. This bit when set allows for LED control of only receive packets that match internal address match.
4	XMT E	Enable Transmit Status Signal. Indicates PCnet-ISA+ controller transmit activity . 0 disables the signal, 1 enables the signal. Enable Receive Polarity Signal.
3	RVPOL E	Enables LED pin assertion when receive polarity is correct on the 10BASE-T port. Clearing the bit indicates this function is to be ignored.
2	RCV E	Enable Receive Status Signal. Indicates receive activity on the network. 0 disables the signal, 1 enables the signal.
1	JAB E	Enable Jabber Signal. Indicates the PCnet-ISA+ controller is jabbering on the network. 0 disables the signal, 1 enables the signal.
0	COL E	Enable Collision Signal. Indicates collision activity on the network. 0 disables the signal, 1 enables the signal.

ISACSR8: Software Configuration Register (Read-Only Register)

Bit	Description
15–12	Read-only image of SR_AM(3:0) of P&P register 0x48 - 0x49.
11–8	Read-only image of BP_AM(3:0) of P&P register 0x40 - 0x41.
7–4	Read-only image of IRQSEL(3:0) of P&P register 0x70.
3	Reserved, written with zero, read as undefined.
2–0	Read-only image of DMASEL(2:0) of P&P register 0x74.

Initialization Block

The figure below shows the Initialization Block memory configuration. Note that the Initialization Block must be based on a word (16-bit) boundary.

Address	Bits 15–12	Bits 11–8	Bits 7–4	Bits 3–0
IADR+00	MODE 15-00			
IADR+02	PADR 15-00			
IADR+04	PADR 31-16			
IADR+06	PADR 47-32			
IADR+08	LADRF 15-00			
IADR+10	LADRF 31-16			
IADR+12	LADRF 47-32			
IADR+14	LADRF 63-48			
IADR+16	RDRA 15-00			
IADR+18	RLEN	RES	RDRA 23-16	
IADR+20	TDRA 15-00			
IADR+22	TLEN	RES	TDRA 23-16	

RLEN and TLEN

The TLEN and RLEN fields in the initialization block are 3 bits wide, occupying bits 15,14, and 13, and the value in these fields determines the number of Transmit and Receive Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is as follows:

R/TLEN	# of DREs
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

If a value other than those listed in the above table is desired, CSR76 and CSR78 can be written after initialization is complete. See the description of the appropriate CSRs.

RDRA and TDRA

TDRA and RDRA indicate where the transmit and receive descriptor rings, respectively, begin. Each DRE must be located on an 8-byte boundary.

LADRF

The Logical Address Filter (LADRF) is a 64-bit mask that is used to accept incoming Logical Addresses. If the first bit in the incoming address (as transmitted on the wire) is a "1", the address is deemed logical. If the first bit is a "0", it is a physical address and is compared against the physical address that was loaded through the initialization block.

A logical address is passed through the CRC generator, producing a 32-bit result. The high order 6 bits of the CRC are used to select one of the 64 bit positions in the Logical Address Filter. If the selected filter bit is set, the address is accepted and the frame is placed into memory.

The Logical Address Filter is used in multicast addressing schemes. The acceptance of the incoming frame based on the filter value indicates that the message may be intended for the node. It is the node's responsibility to determine if the message is actually intended for the node by comparing the destination address of the stored message with a list of acceptable logical addresses.

If the Logical Address Filter is loaded with all zeroes and promiscuous mode is disabled, all incoming logical addresses except broadcast will be rejected.

The Broadcast address, which is all ones, does not go through the Logical Address Filter and is handled as follows:

1) If the Disable Broadcast Bit is cleared, the broadcast address is accepted.
2) If the Disable Broadcast Bit is set and promiscuous mode is enabled, the broadcast address is accepted.
3) If the Disable Broadcast Bit is set and promiscous mode is disabled, the broadcast address is rejected.

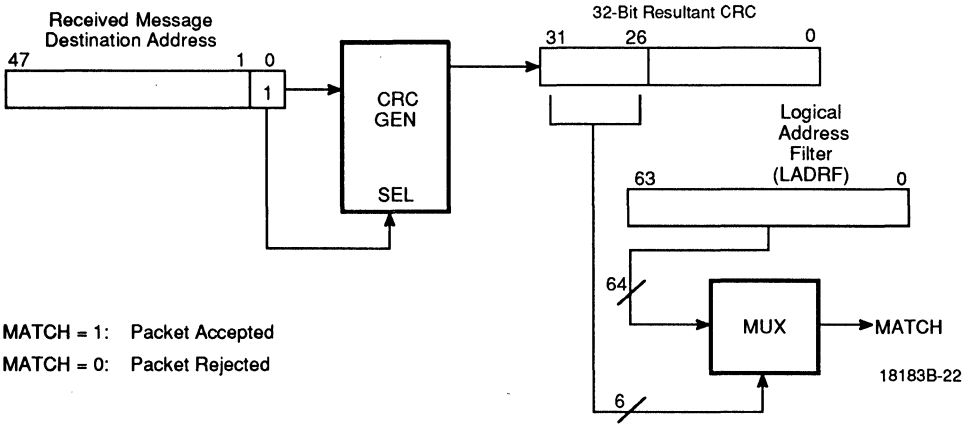
If external loopback is used, the FCS logic must be allocated to the receiver (by setting the DXMTFCS bit in CSR15, and clearing the ADD_FCS bit in TMD1) when using multicast addressing.

PADR

This 48-bit value represents the unique node address assigned by the IEEE and used for internal address comparison. PADR[0] is the first address bit transmitted on the wire, and must be zero. The six-hex-byte nomenclature used by the IEEE maps to the PCnet-ISA+ controller PADR register as follows: the first byte comprises PADR[7:0], with PADR[0] being the least significant bit of the byte. The second IEEE byte maps to PADR[15:8], again from LSbit to MSbit, and so on. The sixth byte maps to PADR[47:40], the LSbit being PADR[40].

MODE

The mode register in the initialization block is copied into CSR15 and interpreted according to the description of CSR15.



Address Match Logic

Receive Descriptors

The Receive Descriptor Ring Entries (RDREs) are composed of four receive message fields (RMD0-3). Together they contain the following information:

- The address of the actual message data buffer in user (host) memory
- The length of that message buffer
- Status information indicating the condition of the buffer. The eight most significant bits of RMD1 (RMD1[15:0]) are collectively termed the STATUS of the receive descriptor.

RMD0

Holds LADRF [15:0]. This is combined with HADR [7:0] in RMD1 to form the 24-bit address of the buffer pointed to by this descriptor table entry. There are no restrictions on buffer byte alignment or length.

RMD1

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-ISA+ controller (OWN=1). The PCnet-ISA+ controller clears the OWN bit after filling the buffer pointed to by the descriptor entry. The host sets the OWN bit after emptying the buffer. Once the PCnet-ISA+ controller or host has relinquished ownership of a buffer, it must not change any field in the descriptor entry.
14	ERR	ERR is the OR of FRAM, OFLO, CRC, or BUFF. ERR is written by the PCnet-ISA+ controller.
13	FRAM	FRAMING ERROR indicates that the incoming frame contained a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is written by the PCnet-ISA+ controller.
12	OFLO	OVERFLOW error indicates that the receiver has lost all or part of the incoming frame, due to an inability to store the frame in a memory buffer before the internal FIFO overflowed. OFLO is valid only when ENP is not set. OFLO is written by the PCnet-ISA+ controller.
11	CRC	CRC indicates that the receiver has detected a CRC (FCS) error on the incoming frame. CRC is valid only when ENP is set and OFLO is not. CRC is written by the PCnet-ISA+ controller.
10	BUFF	BUFFER ERROR is set any time the PCnet-ISA+ controller does not own the next buffer while data chaining a received frame. This can occur in either of two ways: 1) The OWN bit of the next buffer is zero 2) FIFO overflow occurred before the PCnet-ISA+ controller polled the next descriptor

		If a Buffer Error occurs, an Overflow Error may also occur internally in the FIFO, but will not be reported in the descriptor status entry unless both BUFF and OFLO errors occur at the same time. BUFF is written by the PCnet-ISA+ controller.
9	STP	START OF PACKET indicates that this is the first buffer used by the PCnet-ISA+ controller for this frame. It is used for data chaining buffers. STP is written by the PCnet-ISA+ controller in normal operation. In SRPINT Mode (CSR3.5 set to 1) this bit is written by the driver.
8	ENP	END OF PACKET indicates that this is the last buffer used by the PCnet-ISA+ controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is written by the PCnet-ISA+ controller.
7-0	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and is not changed by the PCnet-ISA+ controller.

RMD2

Bit	Name	Description
15-12	ONES	MUST BE ONES. This field is written by the host and unchanged by the PCnet-ISA+ controller.
11-0	BCNT	BUFFER BYTE COUNT is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and is not changed by the PCnet-ISA+ controller.

RMD3

Bit	Name	Description
15-12	RES	RESERVED and read as zeros.
11-0	MCNT	MESSAGE BYTE COUNT is the length in bytes of the received message, expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the PCnet-ISA+ controller and cleared by the host.

Transmit Descriptors

The Transmit Descriptor Ring Entries (TDREs) are composed of four transmit message fields (TMD0-3). Together they contain the following information:

- The address of the actual message data buffer in user or host memory
- The length of the message buffer
- Status information indicating the condition of the buffer. The eight most significant bits of TMD1 (TMD1[15:8]) are collectively termed the STATUS of the transmit descriptor.

Note that bit 13 of TMD1, which was formerly a reserved bit in the LANCE (Am7990), is assigned a new meaning, ADD_FCS.

TMD0

Holds LADR [15:0]. This is combined with HADR [7:0] in TMD1 to form a 24-bit address of the buffer pointed to by this descriptor table entry. There are no restrictions on buffer byte alignment or length.

TMD1

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-ISA+ controller (OWN=1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The PCnet-ISA+ controller clears the OWN bit after transmitting the contents of the buffer. Both the PCnet-ISA+ controller and the host must not alter a descriptor entry after it has relinquished ownership.
14	ERR	ERR is the OR of UFLO, LCOL, LCAR, or RTRY. ERR is written by the PCnet-ISA+ controller. This bit is set in the current descriptor when the error occurs, and therefore may be set in any descriptor of a chained buffer transmission.
13	ADD_FCS	ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. It is valid only if the STP bit is set. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS = 0, FCS generation is controlled by DXMTFCS. ADD_FCS is written

		by the host, and unchanged by the PCnet-ISA+ controller. This was a reserved bit in the LANCE (Am7990).
12	MORE	MORE indicates that more than one re-try was needed to transmit a frame. MORE is written by the PCnet-ISA+ controller. This bit has meaning only if the ENP or the ERR bit is set.
11	ONE	ONE indicates that exactly one re-try was needed to transmit a frame. ONE flag is not valid when LCOL is set. ONE is written by the PCnet-ISA+ controller. This bit has meaning only if the ENP or the ERR bit is set.
10	DEF	DEFERRED indicates that the PCnet-ISA+ controller had to defer while trying to transmit a frame. This condition occurs if the channel is busy when the PCnet-ISA+ controller is ready to transmit. DEF is written by the PCnet-ISA+ controller. This bit has meaning only if the ENP or ERR bits are set.
9	STP	START OF PACKET indicates that this is the first buffer to be used by the PCnet-ISA+ controller for this frame. It is used for data chaining buffers. The STP bit must be set in the first buffer of the frame, or the PCnet-ISA+ controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is written by the host and is not changed by the PCnet-ISA+ controller.
8	ENP	END OF PACKET indicates that this is the last buffer to be used by the PCnet-ISA+ controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is written by the host and is not changed by the PCnet-ISA+ controller.
7-0	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and is not changed by the PCnet-ISA+ controller.

TMD2

Bit	Name	Description
15-12	ONES	MUST BE ONES. This field is written by the host and unchanged by the PCnet-ISA+ controller.
11-0	BCNT	BUFFER BYTE COUNT is the length of the buffer pointed to by this descriptor, expressed as the two's com- plement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the PCnet-ISA+ controller. This field is written by the host and is not changed by the PCnet-ISA+ controller. There are no minimum buffer size restrictions. Zero length buffers are allowed for protocols which require it.

TMD3

Bit	Name	Description
15	BUFF	BUFFER ERROR is set by the PCnet-ISA+ controller during transmission when the PCnet-ISA+ controller does not find the ENP flag in the current buffer and does not own the next buffer. This can occur in either of two ways: 1) The OWN bit of the next buffer is zero. 2) FIFO underflow occurred before the PCnet-ISA+ controller obtained the next STATUS byte (TMD1[15:8]). BUFF error will turn off the transmitter (CSR0, TXON = 0). If a Buffer Error occurs, an Underflow Error will also occur. BUFF is not valid when LCOL or RTRY error is set during transmit data chaining. BUFF is written by the PCnet-ISA+ controller.
14	UFLO	UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that the FIFO has emptied before the end of the frame was

reached. Upon UFLO error, the transmitter is turned off (CSR0, TXON = 0). UFLO is written by the PCnet-ISA+ controller.

13 RES — RESERVED bit. The PCnet-ISA+ controller will write this bit with a "0".

12 LCOL — LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The PCnet-ISA+ controller does not re-try on late collisions. LCOL is written by the PCnet-ISA+ controller.

11 LCAR — LOSS OF CARRIER is set in AUI mode when the carrier is lost during an PCnet-ISA+ controller-initiated transmission. The PCnet-ISA+ controller does not stop transmission upon loss of carrier. It will continue to transmit the whole frame until done. LCAR is written by the PCnet-ISA+ controller.

In 10BASE-T mode, LCAR will be set when the T-MAU is in link fail state.

10 RTRY — RETRY ERROR indicates that the transmitter has failed after 16 attempts to successfully transmit a message, due to repeated collisions on the medium. If DRTY = 1 in the MODE register, RTRY will set after one failed transmission attempt. RTRY is written by the PCnet-ISA+ controller.

09-00 TDR — TIME DOMAIN REFLECTOMETRY reflects the state of an internal PCnet-ISA+ controller counter that counts at a 10 MHz rate from the start of a transmission to the occurrence of a collision or loss of carrier. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the PCnet-ISA+ controller and is valid only if RTRY is set.

Note that 10 MHz gives very low resolution and in general has not been found to be particularly useful. This feature is here primarily to maintain full compatibility with the LANCE.

Register Summary

Ethernet Controller Registers (Accessed via RDP Port)

RAP Addr	Symbol	Width	User Register	Comments
00	CSR0	16-bit	Y	PCnet-ISA+ controller status
01	CSR1	16-bit	Y	Lower IADR: maps to location 16
02	CSR2	16-bit	Y	Upper IADR: maps to location 17
03	CSR3	16-bit	Y	Mask Register
04	CSR4	16-bit	Y	Miscellaneous Register
05	CSR5	16-bit		Reserved
06	CSR6	16-bit		RXTX: RX/TX Encoded Ring Lengths
07	CSR7	16-bit		Reserved
08	CSR8	16-bit	Y	LADR0: LADRF[15:0]
09	CSR9	16-bit	Y	LADR1: LADRF[31:16]
10	CSR10	16-bit	Y	LADR2: LADRF[47:32]
11	CSR11	16-bit	Y	LADR3: LADRF[63:48]
12	CSR12	16-bit	Y	PADR0: PADR[15:0]
13	CSR13	16-bit	Y	PADR1: PADR[31:16]
14	CSR14	16-bit	Y	PADR2: PADR[47:32]
15	CSR15	16-bit	Y	MODE: Mode Register
16-17	CSR16	32-bit		IADR: Base Address of INIT Block
18-19	CSR18	32-bit		CRBA: Current RCV Buffer Address
20-21	CSR20	32-bit		CXBA: Current XMT Buffer Address
22-23	CSR22	32-bit		NRBA: Next RCV Buffer Address
24-25	CSR24	32-bit	Y	BADR: Base Address of RCV Ring
26-27	CSR26	32-bit		NRDA: Next RCV Descriptor Address
28-29	CSR28	32-bit		CRDA: Current RCV Descriptor Address
30-31	CSR30	32-bit	Y	BADX: Base Address of XMT Ring
32-33	CSR32	32-bit		NXDA: Next XMT Descriptor Address
34-35	CSR34	32-bit		CXDA: Current XMT Descriptor Address
36-37	CSR36	32-bit		Next Next Receive Descriptor Address
38-39	CSR38	32-bit		Next Next Transmit Descriptor Address
40-41	CSR40	32-bit		CRBC: Current RCV Stat and Byte Count
42-43	CSR42	32-bit		CXBC: Current XMT Status and Byte Count
44-45	CSR44	32-bit		NRBC: Next RCV Stat and Byte Count
46	CSR46	16-bit		POLL: Poll Time Counter
47	CSR47	32-bit	Y	Polling Interval
48-49	CSR48	32-bit		TMP0: Temporary Storage
50-51	CSR50	32-bit		TMP1: Temporary Storage
52-53	CSR52	32-bit		TMP2: Temporary Storage
54-55	CSR54	32-bit		TMP3: Temporary Storage
56-57	CSR56	32-bit		TMP4: Temporary Storage
58-59	CSR58	32-bit		TMP5: Temporary Storage
60-61	CSR60	32-bit		PXDA: Previous XMT Descriptor Address
62-63	CSR62	32-bit		PXBC: Previous XMT Status and Byte Count

Register Summary

Ethernet Controller Registers (Accessed via RDP Port) (continued)

RAP Addr	Symbol	Width	User Register	Comments
64-65	CSR64	32-bit		NXBA: Next XMT Buffer Address
66-67	CSR66	32-bit		NXBC: Next XMT Status and Byte Count
68-69	CSR68	32-bit		XSTMP: XMT Status Temporary
70-71	CSR70	32-bit		RSTMP: RCV Status Temporary
72	CSR72	16-bit		RCVRC: RCV Ring Counter
74	CSR74	16-bit		XMTRC: XMT Ring Counter
76	CSR76	16-bit	Y	RCVRL: RCV Ring Length
78	CSR78	16-bit	Y	XMTRL: XMT Ring Length
80	CSR80	16-bit	Y	DMABR: Burst Register
82	CSR82	16-bit	Y	DMABAT: Bus Activity Timer
84-85	CSR84	32-bit		DMABA: Address Register
86	CSR86	16-bit		DMABC: Byte Counter/Register
88-89	CSR88	32-bit	Y	Chip ID Register
92	CSR92	16-bit		RCON: Ring Length Conversion Register
94	CSR94	16-bit		XMTTDR: Transmit Time Domain Reflectometry
96-97	CSR96	32-bit		SCR0: BIU Scratch Register 0
98-99	CSR98	32-bit		SCR1: BIU Scratch Register 1
104-105	CSR104	32-bit		SWAP:16-bit word/byte Swap Register
108-109	CSR108	32-bit		BMSCR: BMU Scratch Register
112	CSR112	16-bit	Y	Missed Frame Count
114	CSR114	16-bit	Y	Receive Collision Count
124	CSR124	16-bit	Y	BMU Test Register
126	CSR126	16-bit		Reserved

Note: *Although the PCnet-ISA+ controller has many registers that can be accessed by software, most of these registers are intended for debugging and production testing purposes only. The registers with a "Y" are the only registers that should be accessed by network software.*

Register Summary

ISACSR—ISA Bus Configuration Registers (Accessed via IDP Port)

RAP Addr	Mnemonic	Default	Name
0	MSRDA	0005H	Master Mode Read Active
1	MSWRA	0005H	Master Mode Write Active
2	MC	0002H	Miscellaneous Configuration
3	EC	8000*H	EEPROM Configuration
4	LED0	0000H	$\overline{\text{LED0}}$ Status (Link Integrity)
5	LED1	0084H	$\overline{\text{LED1}}$ Status (Default: RCV)
6	LED2	0008H	$\overline{\text{LED2}}$ Status (Default: RCVPOL)
7	LED3	0090H	$\overline{\text{LED3}}$ Status (Default: XMT)
8	SC	0000H	Software Configuration (Read-Only Register)

**This value can be 0000H for systems that do not support EEPROM option*

I/O Address Offset

Offset	#Bytes	Register
0h	16	Address PROM
10h	2	RDP
12h	2	RAP (shared by RDP and IDP)
14h	2	Reset
16h	2	IDP

SYSTEM APPLICATION

ISA Bus Interface

Compatibility Considerations

Although 8 MHz is now widely accepted as the standard speed at which to run the ISA bus, many machines have been built which operate at higher speeds with non-standard timing. Some machines do not correctly support 16-bit I/O operations with wait states. Although the PCnet-ISA+ controller is quite fast, some operations still require an occasional wait state. The PCnet-ISA+ controller moves data through memory accesses, therefore, I/O operations do not affect performance. By configuring the PCnet-ISA+ controller as an 8-bit I/O device, compatibility with PC/AT-class machines is obtained at virtually no cost in performance. To treat the PCnet-ISA+ controller as an 8-bit software resource (for non-ISA applications), the even-byte must be accessed first, followed by an odd-byte access.

Memory cycle timing is an area where some tradeoffs may be necessary. Any slow down in a memory cycle translates directly into lower bandwidth. The PCnet-ISA+ controller starts out with much higher bandwidth than most slave type controllers and should continue to be superior even if an extra 50 or 100 ns are added to memory cycles.

The memory cycle active time is tunable in 50 ns increments with a default of 250 ns. The memory cycle idle time defaults to 200 ns and can be reprogrammed to 100 ns. See register description for ISACS42. Most machines should not need tuning.

The PCnet-ISA+ controller is compatible with NE2100 and NE1500T software drivers. All the resources such as address PROM, boot PROM, RAP, and RDP are in the same location with the same semantics. An additional set of registers (ISA CSR) is available to configure on board resources such as ISA bus timing and LED operation. However, loopback frames for the PCnet-ISA+ controller must contain more than 64 bytes of data if the Runt Packet Accept feature is not enabled; this size limitation does not apply to LANCE (Am7990) based boards such as the NE2100 and NE1500T.

Bus Master

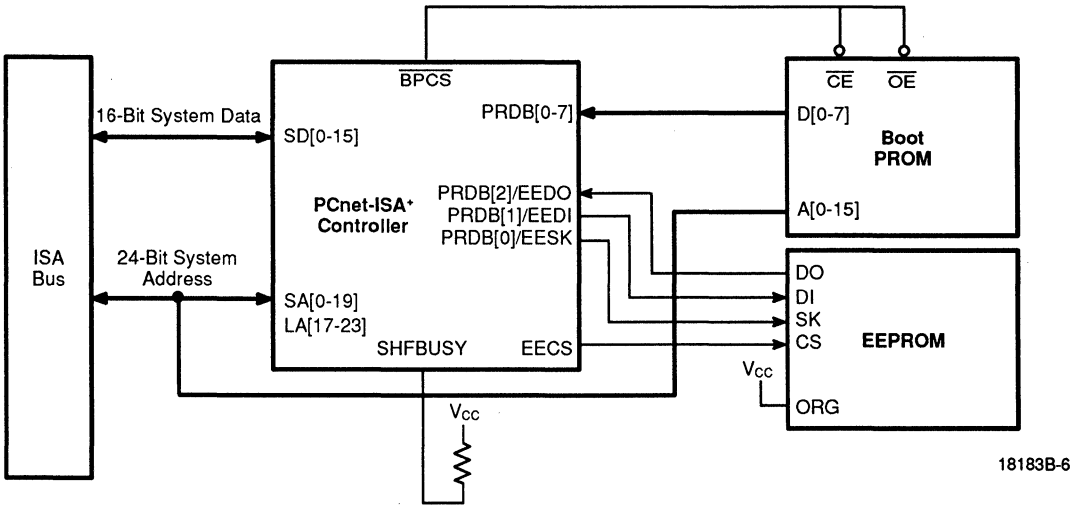
Bus Master mode is the preferred mode for client applications on PC/AT or similar machines supporting 16-bit DMA with its unsurpassed combination of high performance and low cost.

Shared Memory

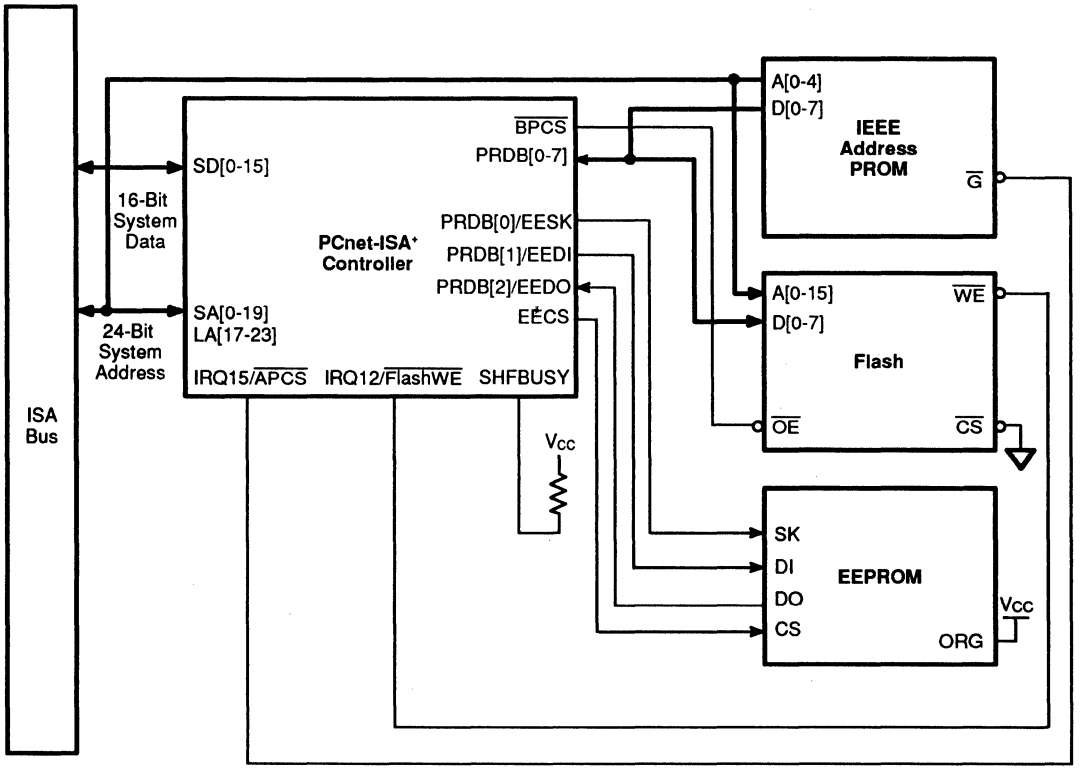
The shared memory mode is recommended for file servers or other applications where there is very high, average or peak latency.

The address compare circuit has the following functions. It receives the 7 LA signals, generates $\overline{\text{MEMCS16}}$, and compares them to the desired shared memory and boot PROM addresses. The logic latches the address compare result when BALE goes inactive and uses the appropriate SA signals to generate $\overline{\text{SMAM}}$ and $\overline{\text{BPAM}}$.

All these functions can be performed in one PAL device. To operate in an 8-bit PC/XT environment, the LA signals should have weak pull-down resistors connected to them to present a logic 0 level when not driven.

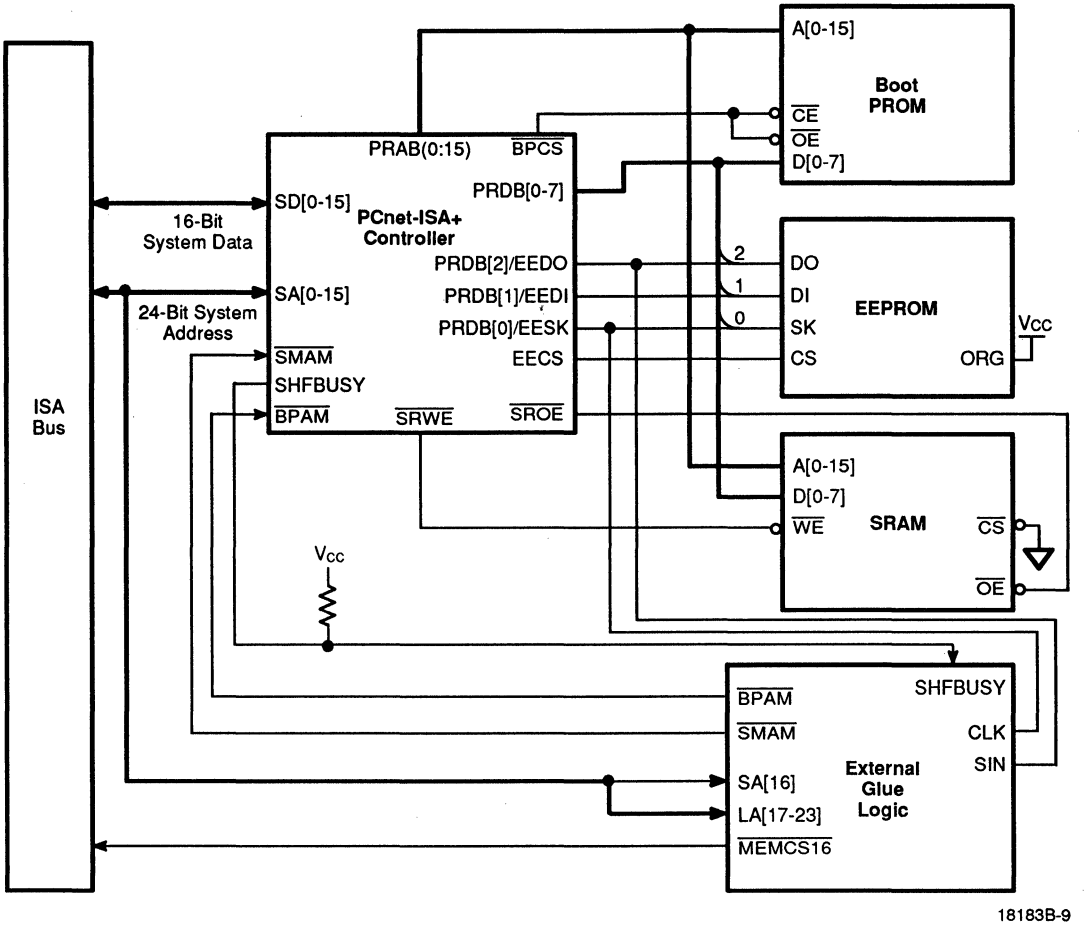


Bus Master Block Diagram
Plug and Play Compatible

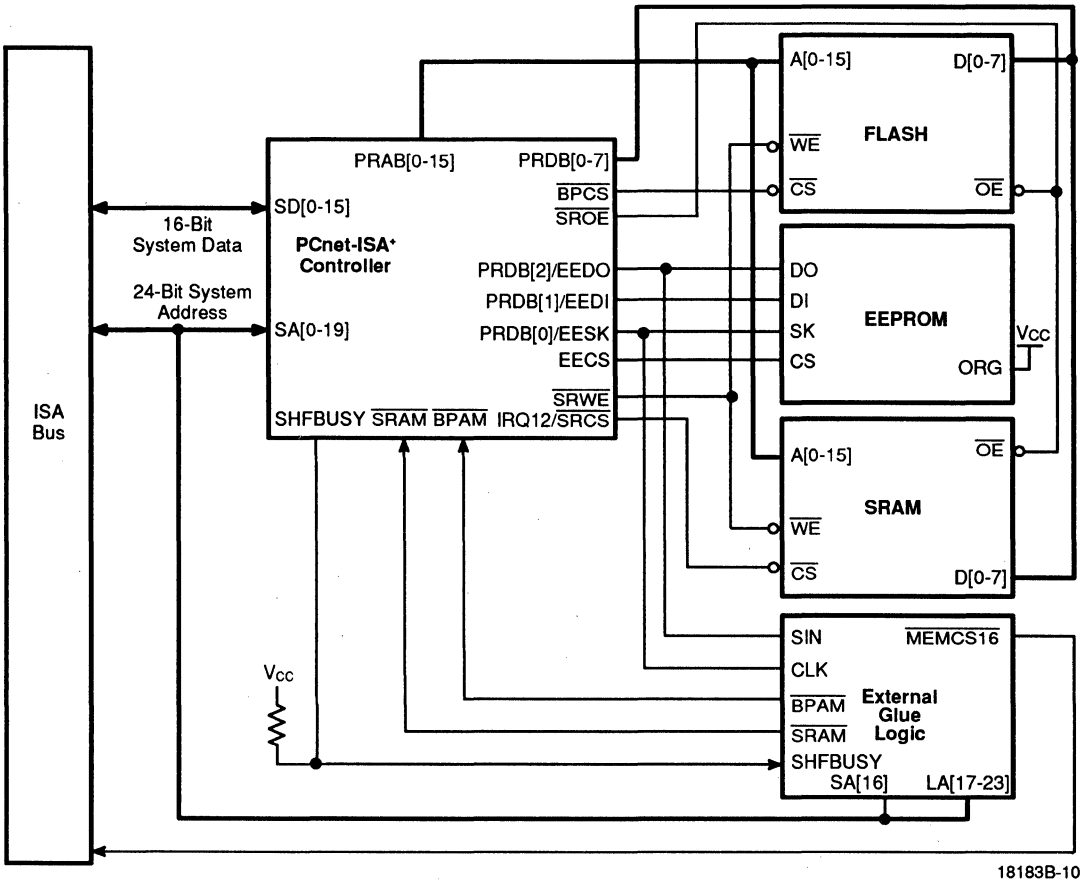


18183B-7

Bus Master Block Diagram
Plug and Play Compatible with Flash Support



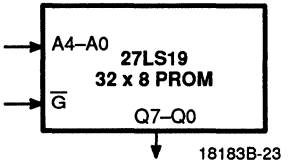
Shared Memory Block Diagram
Plug and Play Compatible



Shared Memory Block Diagram
Plug and Play Compatible with Flash Memory Support

Optional Address PROM Interface

The suggested address PROM is the Am27LS19, a 32x8 device. $\overline{APCS}$ should be connected directly to the device's $\overline{G}$ input.

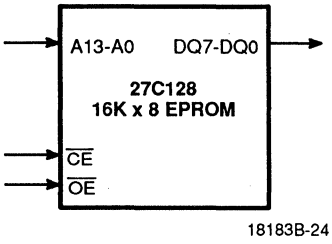


Address PROM Example

Boot PROM Interface

The boot PROM is a 8K–64K EPROM. Its $\overline{OE}$ pin should be tied to ground, and chip enable $\overline{CE}$ to $\overline{BPCS}$ to minimize power consumption at the expense of speed. Shown below is a 27C128.

Higher density EPROMs place an address line on the pin that is defined for lower density EPROMs as the V_{PP} (programming voltage) pin. For READ only operation on an EPROM, the V_{PP} pin can assume any logic level, as long as the voltage on the V_{PP} pin does not exceed the programming voltage threshold (typically 7 V to 12 V). Therefore, a socket with a 27512 pinout will also support 2764 and 27128 EPROM devices.



Boot PROM Example

Static RAM Interface (for Shared Memory Only)

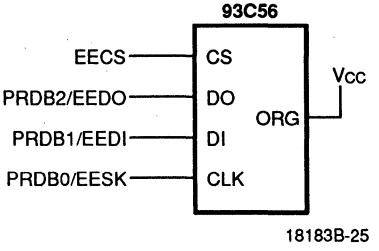
The SRAM is an 8Kx8 or 32Kx8 device. The PCnet-ISA+ controller can support 64 Kbytes of SRAM address space. The PCnet-ISA+ controller provides $\overline{SROE}$ and $\overline{SRWE}$ outputs which can go directly to the $\overline{OE}$ and $\overline{WE}$ pins of the SRAM, respectively. The address lines are connected as described in the shared memory section and the data lines go to the Private Data Bus.

AUI

The PCnet-ISA+ controller drives the AUI through a set of transformers. The DI and CI inputs should each be terminated with a pair of matched 39 Ω or 40.2 Ω resistors connected in series with the middle node bypassed to ground with a .01 µF to 0.1 µF capacitor. Refer to the PCnet-ISA Technical Manual (PID #16850B) for network interface design and refer to Appendix A for a list of compatible AUI isolation transformers.

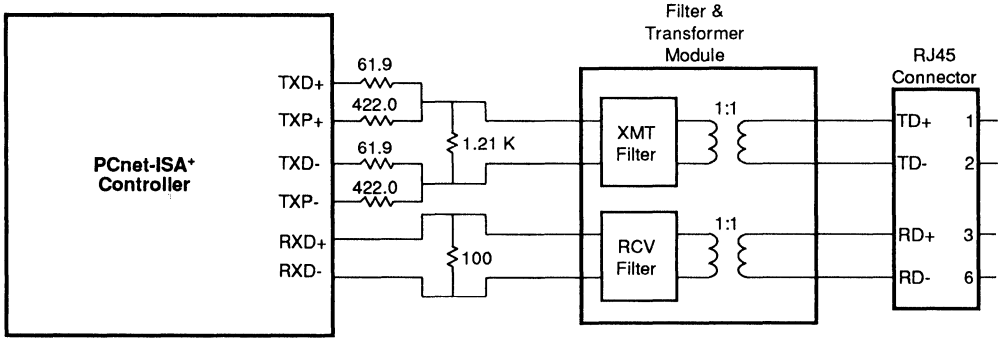
EEPROM Interface

The suggested EEPROM is the industry standard 93C56 2 Kbit serial EEPROM. This is used in the 16-bit mode to provide 128 x 16-bit EEPROM locations to store configuration information as well as the Plug and Play information.



10BASE-T Interface

The diagram below shows the proper 10BASE-T network interface design. Refer to the *PCnet Family Technical Manual* (PID #18216A) for more design details, and refer to Appendix A for a list of compatible 10BASE-T filter/transformer modules.



Note: *All resistors are ±1%*

18183B-26

10BASE-T External Components and Hookup

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C

Ambient Temperature
Under Bias ..... 0°C to +70°C

Supply Voltage to AVss
or DVss (AV_{DD}, DV_{DD}) ..... −0.3 V to +6.0 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A) ..... 0°C to +70°C

Supply Voltages
(AV_{DD}, DV_{DD}) ..... 5 V ±5%

All inputs within the range: .. $AV_{SS} - 0.5\ V \le V_{in} \le AV_{DD} + 0.5\ V$, or $DV_{SS} - 0.5\ V \le V_{in} \le DV_{DD} + 0.5\ V$

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (refer to page 19 for driver types)

Parameter Symbol	Parameter Description	Test Conditions		Min	Max	Unit
Digital Input Voltage						
V_{IL}	Input LOW Voltage				0.8	V
V_{IH}	Input HIGH Voltage			2.0	DV_{DD} + 0.5	V
Digital Ouput Voltage						
V_{OL}	Output LOW Voltage				0.5	V
V_{OH}	Output HIGH Voltage	(Note 1)		2.4		V
Digital Input Leakage Current						
I_{IX}	Input Leakage Current	V_{DD} = 5 V, V_{IN} = 0 V (Note 2)		−10	10	µA
Digital Ouput Leakage Current						
I_{OZL}	Output Low Leakage Current (Note 3)	V_{OUT} = 0 V		−10		µA
I_{OZH}	Output High Leakage Current (Note 3)	V_{OUT} = V_{DD}			10	µA
Crystal Input Current						
V_{ILX}	XTAL1 Input LOW Threshold Voltage	V_{IN} = External Clock		−0.5	0.8	V
V_{ILHX}	XTAL1 Input HIGH Threshold Voltage	V_{IN} = External Clock		3.5	V_{DD} + 0.5	V
I_{ILX}	XTAL1 Input LOW Current	VIN = DVSS	Active	−120	0	µA
			Sleep	−10	+10	µA
I_{IHX}	XTAL1 Input HIGH Current	VIN = VDD	Active	0	120	µA
			Sleep		400	µA
Attachment Unit Interface						
I_{IAXD}	Input Current at DI+ and DI−	AV_{SS} < VIN < AV_{DD}		−500	+500	µA
I_{IAXC}	Input current at CI+ and CI−	AV_{SS} < VIN < AV_{DD}		−500	+500	µA
V_{AOD}	Differential Output Voltage |(DO+)−(DO−)|	R_L = 78 Ω		630	1200	mV
V_{AODOFF}	Transmit Differential Output Idle Voltage	R_L = 78 Ω (Note 5)		−40	+40	mV

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Attachment Unit Interface (continued)					
I_{AODOFF}	Transmit Differential Output Idle Current	R_L = 78 Ω (Note 4)	–1	+1	mA
V_{CMT}	Transmit Output Common Mode Voltage	R_L = 78 Ω	2.5	AV_{DD}	V
V_{ODI}	DO± Transmit Differential Output Voltage Imbalance	R_L = 78 Ω (Note 5)		25	mV
V_{ATH}	Receive Data Differential Input Threshold	(Note 5)	–35	35	mV
V_{ASQ}	DI± and CI± Differential Input Threshold (Squelch)		–275	–160	mV
V_{IRDVD}	DI± and CI± Differential Mode Input Voltage Range		–1.5	+1.5	V
V_{ICM}	DI± and CI± Input Bias Voltage	I_{IN} = 0 mA	AV_{DD}–3.0	AV_{DD}–1.0	V
V_{OPD}	DO± Undershoot Voltage at Zero Differential on Transmit Return to Zero (ETD)	(Note 5)		–100	mV
Twisted Pair Interface					
I_{IRXD}	Input Current at RXD±	$AV_{SS} < V_{IN} < AV_{DD}$	–500	500	µA
R_{RXD}	RXD± Differential Input Resistance	(Note 5)	10		KΩ
V_{TIVB}	RXD+, RXD– Open Circuit Input Voltage (Bias)	I_{IN} = 0 mA	AV_{DD} – 3.0	AV_{DD} – 1.5	V
V_{TIDV}	Differential Mode Input Voltage Range (RXD±)	AV_{DD} = +5 V	–3.1	+3.1	V
V_{TSQ+}	RXD Positive Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	300	520	mV
V_{TSQ-}	RXD Negative Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	–520	–300	mV
V_{THS+}	RXD Post-Squelch Positive Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	150	293	mV
V_{THS-}	RXD Post-Squelch Negative Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	–293	–150	mV
V_{LTSQ+}	RXD Positive Squelch Threshold (Peak)	LRT = 1 (Note 6)	180	312	mV
V_{LTSQ-}	RXD Negative Squelch Threshold (Peak)	LRT = 1 (Note 6)	–312	–180	mV
V_{LTHS+}	RXD Post-Squelch Positive Threshold (Peak)	LRT = 1 (Note 6)	90	156	mV
V_{LTHS-}	RXD Post-Squelch Negative Threshold (Peak)	LRT = 1 (Note 6)	–156	–90	mV

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Twisted Pair Interface (continued)					
V_{RXDTH}	RXD Switching Threshold	(Note 5)	–35	35	mV
V_{TXH}	TXD± and TXP± Output HIGH Voltage	DV_{SS} = 0 V	DV_{DD} – 0.6	DV_{DD}	V
V_{TXL}	TXD± and TXP± Output LOW Voltage	DV_{DD} = +5 V	DV_{SS}	DV_{SS} + 0.6	V
V_{TXI}	TXD± and TXP± Differential Output Voltage Imbalance		–40	+40	mV
V_{TXOFF}	TXD± and TXP± Idle Output Voltage	DV_{DD} = +5 V	–40	+40	mV
R_{TX}	TXD± Differential Driver Output Impedance	(Note 5)		40	Ω
	TXP± Differential Driver Output Impedance	(Note 5)		80	Ω
IEEE 1149.1 (JTAG) Test Port					
V_{IL}	TCK, TMS, TDI			0.8	V
V_{IH}	TCK, TMS, TDI		2.0		V
V_{OL}	TDO	I_{OL} = 2.0 mA		0.4	V
V_{OH}	TDO	I_{OH} = –0.4 mA	2.4		V
I_{IL}	TCK, TMS, TDI	V_{DD} = 5.5 V, V_I = 0.5 V		–200	µA
I_{IH}	TCK, TMS, TDI	V_{DD} =5.5 V, V_I = 2.7 V		–100	µA
I_{OZ}	TDO	0.4 V < V_{OUT} < V_{DD}	–10	+10	µA
Power Supply Current					
I_{DD}	Active Power Supply Current	XTAL1 = 20 MHz		75	mA
I_{DDCOMA}	Coma Mode Power Supply Current	$\overline{SLEEP}$ active		200	µA
$I_{DDSNOOZE}$	Snooze Mode Mall Power Supply Current	Awake bit set active		10	mA

Notes:

1. *V_{OH} does not apply to open-drain output pins.*
2. *I_{IX} applies to all input only pins except DI±, CI±, XTAL1 and PRDB[7:0].*
3. *I_{OZL} applies to all three-state output pins and bi-directional pins, except PRDB[7:0]. I_{OZH} applies to pins PRDB[7:0].*
4. *Correlated to other tested parameters—not tested directly.*
5. *Parameter not tested.*
6. *LRT is bit 9 of Mode register (CSR15)*

SWITCHING CHARACTERISTICS: BUS MASTER MODE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Input/Output Write Timing					
t_{IOW1}	AEN, $\overline{SBHE}$, SA0–9 Setup to ↓ $\overline{IOW}$		10		ns
t_{IOW2}	AEN, $\overline{SBHE}$,SA0–9 Hold After ↑ $\overline{IOW}$		5		ns
t_{IOW3}	$\overline{IOW}$ Assertion		100		ns
t_{IOW4}	$\overline{IOW}$ Inactive		55		ns
t_{IOW5}	SD Setup to ↑ $\overline{IOW}$		10		ns
t_{IOW6}	SD Hold After ↑ $\overline{IOW}$		10		ns
t_{IOW7}	↓ IOCHRDY Delay From ↓ $\overline{IOW}$		0	35	ns
t_{IOW8}	IOCHRDY Inactive		125		ns
t_{IOW9}	↑IOCHRDY to ↑ $\overline{IOW}$		0		ns
Input/Output Read Timing					
t_{IOR1}	AEN, $\overline{SBHE}$, SA0–9 Setup to ↓ $\overline{IOR}$		15		ns
t_{IOR2}	AEN, $\overline{SBHE}$,SA0–9 Hold After ↑ $\overline{IOR}$		5		ns
t_{IOR3}	$\overline{IOR}$ Inactive		55		ns
t_{IOR4}	SD Hold After ↑ $\overline{IOR}$		0	20	ns
t_{IOR5}	SD Valid From ↓ $\overline{IOR}$		0	110	ns
t_{IOR6}	↓ IOCHRDY Delay From ↓ $\overline{IOR}$		0	35	ns
t_{IOR7}	IOCHRDY Inactive		125		ns
t_{IOR8}	SD Valid From ↑ IOCHRDY		–130	10	ns
I/O To Memory Command Inactive					
t_{IOM1}	↑ $\overline{IOW}$/$\overline{MEMW}$ to ↓ (S)$\overline{MEMR}$/$\overline{IOR}$		55		ns
t_{IOM2}	↑ (S)$\overline{MEMR}$/$\overline{IOR}$ to ↓ $\overline{IOW}$/$\overline{MEMW}$		55		ns
$\overline{IOCS16}$ Timing					
t_{IOCS1}	AEN, $\overline{SBHE}$, SA0–9 to ↓ $\overline{IOCS16}$		0	35	ns
t_{IOCS2}	AEN, $\overline{SBHE}$, SA0–9 to $\overline{IOCS16}$ Tristated		0	25	ns
Master Mode Bus Acquisition					
t_{MMA1}	$\overline{REF}$ Inactive to ↓ $\overline{DACK}$		5		ns
t_{MMA2}	↑ DRQ to ↓ $\overline{DACK}$		0		ns
t_{MMA3}	$\overline{DACK}$ Inactive		55		ns
t_{MMA4}	↓ $\overline{DACK}$ to ↓ $\overline{MASTER}$			35	ns
t_{MMA5}	↓ $\overline{MASTER}$ to Active Command, $\overline{SBHE}$, SA0–19, LA17–23		125	185	ns

SWITCHING CHARACTERISTICS: BUS MASTER MODE (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Master Mode Bus Release					
t_{MMBR1}	Command Deassert to ↓ DRQ		45	65	ns
t_{MMBR2}	↓ DRQ to ↑ $\overline{DACK}$		0		ns
t_{MMBR3}	↓ DRQ to ↑ $\overline{MASTER}$		40	60	ns
t_{MMBR4}	↓ DRQ to Command, $\overline{SBHE}$, SA0–19, LA17–23 Tristated		–15	0	ns
Master Write Cycles					
t_{MMW1}	$\overline{SBHE}$, SA0–19, LA17–23, Active to ↓ $\overline{MEMW}$	(Note 1)	EXTIME + 45	EXTIME + 65	ns
t_{MMW2}	$\overline{MEMW}$ Active	(Note 2)	MSWRA – 10	MSWRA + 5	ns
t_{MMW3}	$\overline{MEMW}$ Inactive	(Note 1)	EXTIME + 97	EXTIME + 105	ns
t_{MMW4}	↑ $\overline{MEMW}$ to $\overline{SBHE}$, SA0–19, LA17–23,SD Inactive		45	55	ns
t_{MMW5}	$\overline{SBHE}$, SA0–19, LA17–23, SD Hold After ↑ $\overline{MEMW}$		45	60	ns
t_{MMW6}	$\overline{SBHE}$, SA0–19, LA17–23, SD Setup to ↓ $\overline{MEMW}$	(Note 1)	EXTIME + 45	EXTIME + 55	ns
t_{MMW7}	↓ IOCHRDY Delay From ↓ $\overline{MEMW}$		t_{MMW2} – 175		ns
t_{MMW8}	IOCHRDY Inactive		55		ns
t_{MMW9}	↑ IOCHRDY to ↑ $\overline{MEMW}$		130		ns
t_{MMW10}	SD Active to ↓ $\overline{MEMW}$	(Note 1)	EXTIME + 20	EXTIME + 60	ns
t_{MMW11}	SD Setup to ↓ $\overline{MEMW}$	(Note 1)	EXTIME + 20	EXTIME + 60	ns
Master Read Cycles					
t_{MMR1}	$\overline{SBHE}$, SA0–19, LA17–23, Active to ↓ $\overline{MEMR}$	(Note 1)	EXTIME + 45	EXTIME + 60	ns
t_{MMR2}	$\overline{MEMR}$ Active	(Note 2)	MSRDA – 10	MSRDA + 5	ns
t_{MMR3}	$\overline{MEMR}$ Inactive	(Note 1)	EXTIME + 97	EXTIME + 105	ns
t_{MMR4}	↑ $\overline{MEMR}$ to $\overline{SBHE}$, SA0–19, LA17–23 Inactive		45	55	ns
t_{MMR5}	$\overline{SBHE}$, SA0–19, LA17–23 Hold After ↑ $\overline{MEMW}$		45	55	ns
t_{MMR6}	$\overline{SBHE}$, SA0–19, LA17–23 Setup to ↓ $\overline{MEMR}$	(Note 1)	EXTIME + 45	EXTIME + 55	ns
t_{MMR7}	↓ IOCHRDY Delay From ↓ $\overline{MEMR}$		t_{MMR2} – 175		ns
t_{MMR8}	IOCHRDY Inactive		55		ns
t_{MMR9}	↑ IOCHRDY to ↑ $\overline{MEMR}$		130		ns
t_{MMR10}	SD Setup to ↑ $\overline{MEMR}$		30		ns
t_{MMR11}	SD Hold After ↑ $\overline{MEMR}$		0		ns

SWITCHING CHARACTERISTICS: BUS MASTER MODE (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Master Mode Address PROM Read					
t_{MA1}	↓ $\overline{IOR}$ to ↓ $\overline{APCS}$		125	260	ns
t_{MA2}	$\overline{APCS}$ Active		140	155	ns
t_{MA3}	PRDB Setup to ↑ $\overline{APCS}$		20		ns
t_{MA4}	PRDB Hold After ↑ $\overline{APCS}$		0		ns
t_{MA5}	↑ $\overline{APCS}$ to ↑ IOCHRDY		45	65	ns
t_{MA6}	SD Valid From ↑ IOCHRDY		0	10	ns
Master Mode Boot PROM Read					
t_{MB1}	$\overline{REF}$, $\overline{SBHE}$,SA0–19 Setup to ↓ $\overline{SMEMR}$		10		ns
t_{MB2}	$\overline{REF}$, $\overline{SBHE}$,SA0–19 Hold ↑ $\overline{SMEMR}$		5		ns
t_{MB3}	↓ IOCHRDY Delay From ↓ $\overline{SMEMR}$		0	35	ns
t_{MB4}	$\overline{SMEMR}$ Inactive		55		ns
t_{MB5}	↓ $\overline{SMEMR}$ to ↓ $\overline{BPCS}$		125	260	ns
t_{MB6}	$\overline{BPCS}$ Active		290	305	ns
t_{MB7}	↑ $\overline{BPCS}$ to ↑ IOCHRDY		45	65	ns
t_{MB8}	PRDB Setup to ↑ $\overline{BPCS}$		20		ns
t_{MB9}	PRDB Hold After ↑ $\overline{BPCS}$		0		ns
t_{MB10}	SD Valid From ↑ IOCHRDY		0	10	ns
t_{MB11}	SD Hold After ↑ $\overline{SMEMR}$		0	20	ns
t_{MB12}	LA20–23 Hold From ↓ BALE		10		ns
t_{MB13}	LA20–23 Setup to ↓ $\overline{MEMR}$		10		ns
t_{MB14}	↑ BALE Setup to ↓ $\overline{MEMR}$		10		ns

Notes:

1. *EXTIME is 100 ns when ISACSR2, bit 4, is cleared (default). EXTIME is 0 ns when ISACSR2, bit 4, is set.*
2. *MSRDA and MSWDA are parameters which are defined in registers ISACSR0 and ISACSR1, respectively.*

SWITCHING CHARACTERISTICS: BUS MASTER MODE—FLASH READ CYCLE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
t_{MFR1}	REF, SBHE,SA0–19 Setup to ↓ $\overline{MEMR}$		10		ns
t_{MFR2}	REF, SBHE,SA0–19 Hold From ↑ $\overline{MEMR}$		5		ns
t_{MFR3}	↓ IOCHRDY to $\overline{MEMR}$		0	35	ns
t_{MFR4}	↓ $\overline{MEMR}$ Inactive		55		ns
t_{MFR5}	↓ $\overline{MEMR}$ to ↓ $\overline{BPCS}$		125	260	ns
t_{MFR6}	$\overline{BPCS}$ Active		190	205	ns
t_{MFR7}	↑ $\overline{BPCS}$ to ↑ IOCHRDY		45	65	ns
t_{MFR8}	PRDB Setup to ↑ of $\overline{BPCS}$		20		ns
t_{MFR9}	PRDB Hold to ↑ of $\overline{BPCS}$		0		ns
t_{MFR10}	SD Valid From ↑ IOCHRDY		0	10	ns
t_{MFR11}	SD Tristate to ↑ $\overline{MEMR}$		0	20	ns
t_{MFR12}	LA20–23 Hold From ↓ BALE		10		ns
t_{MFR13}	LA20–23 Setup to ↓ $\overline{MEMR}$		10		ns
t_{MFR14}	↑ BALE Setup to ↓ $\overline{MEMR}$		15		ns

SWITCHING CHARACTERISTICS: BUS MASTER MODE—FLASH WRITE CYCLE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
t_{MFW1}	$\overline{SBHE}$, SA0–19 Setup to ↓ $\overline{MEMW}$		10		ns
t_{MFW2}	$\overline{SBHE}$, SA0–19 Hold From ↑ $\overline{MEMW}$		5		ns
t_{MFW3}	↓ IOCHRDY to ↓ $\overline{MEMW}$		0	35	ns
t_{MFW4}	$\overline{MEMW}$ Inactive		50		ns
t_{MFW5}	↑ FL_WE to ↑ IOCHRDY		20	90	ns
t_{MFW6}	↑ $\overline{MEMW}$ Hold From ↑ IOCHRDY		0		ns
t_{MFW7}	SD Valid From ↓ $\overline{MEMW}$			175	ns
t_{MFW8}	SD Hold From ↑ $\overline{MEMW}$		0		ns
t_{MFW9}	PRDB Valid From ↓ $\overline{MEMW}$			175	ns
t_{MFW10}	PRDB Setup to ↓ FL_WE		15		ns
t_{MFW11}	FL_WE Active		140	155	ns
t_{MFW12}	PRDB Hold From ↑ FL_WE		15		
t_{MFW13}	LA20–23 Hold From ↓ BALE		10		ns
t_{MFW14}	LA20–23 Setup to ↓ $\overline{MEMW}$		10		ns
t_{MFW15}	↑ BALE Setup to ↓ $\overline{MEMW}$		15		ns

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Input/Output Write Timing					
t_{IOW1}	AEN, $\overline{SBHE}$, SA0–9 Setup to ↓ $\overline{IOW}$		10		ns
t_{IOW2}	AEN, $\overline{SBHE}$,SA0–9 Hold From ↑ $\overline{IOW}$		5		ns
t_{IOW3}	$\overline{IOW}$ Assertion		150		ns
t_{IOW4}	$\overline{IOW}$ Inactive		55		ns
t_{IOW5}	SD Setup to ↑ $\overline{IOW}$		10		ns
t_{IOW6}	SD Hold After ↑ $\overline{IOW}$		10		ns
t_{IOW7}	↓ IOCHRDY Delay From ↓ $\overline{IOW}$		0	35	ns
t_{IOW8}	IOCHRDY Inactive		125		ns
t_{IOW9}	↑IOCHRDY to ↑ $\overline{IOW}$		0		ns
Input/Output Read Timing					
t_{IOR1}	AEN, $\overline{SBHE}$, SA0–9 Setup to ↓ $\overline{IOR}$		15		ns
t_{IOR2}	AEN, $\overline{SBHE}$,SA0–9 Hold After ↑ $\overline{IOR}$		5		ns
t_{IOR3}	$\overline{IOR}$ Inactive		55		ns
t_{IOR4}	SD Hold From ↑ $\overline{IOR}$		0	20	ns
t_{IOR5}	SD Valid From ↓ $\overline{IOR}$		0	110	ns
t_{IOR6}	↓ IOCHRDY Delay From ↓ $\overline{IOR}$		0	35	ns
t_{IOR7}	IOCHRDY Inactive		125		ns
t_{IOR8}	SD Valid From ↑ IOCHRDY		–130	10	ns
Memory Write Timing					
t_{MW1}	SA0–15, $\overline{SBHE}$, ↓ $\overline{SMAM}$ Setup to ↓ $\overline{MEMW}$		10		ns
t_{MW2}	SA0–15, SBHE, ↑ $\overline{SMAM}$ Hold From ↑ $\overline{MEMW}$		5		ns
t_{MW3}	$\overline{MEMW}$ Assertion		150		ns
t_{MW4}	$\overline{MEMW}$ Inactive		55		ns
t_{MW5}	SD Setup to ↑ $\overline{MEMW}$		10		ns
t_{MW6}	SD Hold From ↑ $\overline{MEMW}$		10		ns
t_{MW7}	↓ IOCHRDY Delay From ↓ $\overline{MEMW}$		0	35	ns
t_{MW8}	IOCHRDY Inactive		125		ns
t_{MW9}	↑ $\overline{MEMW}$ to ↑ IOCHRDY		0		ns

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Memory Read Timing					
t_{MR1}	SA0–15, $\overline{SBHE}$, ↓ $\overline{SMAM/BPAM}$ Setup to ↓ $\overline{MEMR}$		10		ns
t_{MR2}	SA0–15, $\overline{SBHE}$, ↑ $\overline{SMAM/BPAM}$ Hold From ↑ $\overline{MEMR}$		5		ns
t_{MR3}	$\overline{MEMR}$ Inactive		55		ns
t_{MR4}	SD Hold From ↑ $\overline{MEMR}$		0	20	ns
t_{MR5}	SD Valid From ↓ $\overline{MEMR}$		0	110	ns
t_{MR6}	↓ IOCHRDY Delay From ↓ $\overline{MEMR}$		0	35	ns
t_{MR7}	IOCHRDY Inactive		125		ns
t_{MR8}	SD Valid From ↑ IOCHRDY		–130	10	ns
I/O To Memory Command Inactive					
t_{IOM1}	↓ $\overline{IOW/MEMW}$ to ↓ $\overline{(S)MEMR/IOR}$		55		ns
t_{IOM2}	↓ $\overline{(S)MEMR/IOR}$ to ↓ $\overline{IOW/MEMW}$		55		ns
IOCS16 Timing					
t_{IOCS1}	AEN, $\overline{SBHE}$, SA0–9 to ↓ $\overline{IOCS16}$		0	35	ns
t_{IOCS2}	AEN, $\overline{SBHE}$, SA0–9 to $\overline{IOCS16}$ Tristated		0	25	ns
SRAM Read/Write, Boot PROM Read, Address PROM Read on Private Bus					
t_{PR4}	PRAB Change to PRAB Change, SRAM Access		95	105	ns
t_{PR5}	PRDB Setup to PRAB Change, SRAM Access		20		ns
t_{PR6}	PRDB Hold From PRAB Change, SRAM Access		0		ns
t_{PR7}	PRAB Change to PRAB Change, APROM Access		145	155	ns

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
SRAM Read/Write, Boot PROM Read, Address PROM Read on Private Bus (continued)					
t_{PR8}	PRDB Setup to PRAB Change, APROM Access		20		ns
t_{PR9}	PRDB Hold After PRAB Change, APROM Access		0		ns
t_{PR10}	PRAB Change to PRAB Change, BPROM Access		290	305	ns
t_{PR11}	PRDB Setup to PRAB Change, BPROM Access		20		ns
t_{PR12}	PRDB Hold After PRAB Change, BPROM Access		0		ns
t_{PR13}	PRAB Change to PRAB Change, SRAM Write		145	155	ns
t_{PR14}	PRAB Change to $\downarrow \overline{SRWE}$		20	30	ns
t_{PR15}	PRAB Change to $\uparrow \overline{SRWE}$		120	130	ns
t_{PR16}	PRAB Change to PRAB Change, Flash Access		190	205	ns
t_{PR17}	PRAB Change to PRAB Change, Flash Write		190	205	ns
t_{PR18}	PRAB Change to $\uparrow \overline{SRWE}$		170	180	ns

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE—FLASH READ CYCLE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
t_{MFR1}	BPAM, REF, SBHE, SA0–19 Setup to ↓ $\overline{MEMR}$		10		ns
t_{MFR2}	BPAM, REF, SBHE, SA0–19 Hold From ↑ $\overline{MEMR}$		5		ns
t_{MFR3}	↓ IOCHRDY to ↓ $\overline{MEMR}$		0	35	ns
t_{MFR4}	$\overline{MEMR}$ Inactive		55		ns
t_{MFR5}	↓ $\overline{MEMR}$ to ↓ $\overline{BPCS/SROE}$		125	260	ns
t_{MFR6}	$\overline{BPCS/SROE}$ Active		190	205	ns
t_{MFR7}	↑ $\overline{BPCS/SROE}$ to ↑ IOCHRDY		45	65	ns
t_{MFR8}	PRDB Setup to ↑ of $\overline{BPCS/SROE}$		20		ns
t_{MFR9}	PRDB Hold to ↑ of $\overline{BPCS/SROE}$		0		ns
t_{MFR10}	SD Valid From ↑ IOCHRDY		0	10	ns
t_{MFR11}	SD Tristate to ↑ $\overline{MEMR}$		0	20	ns

SWITCHING CHARACTERISTICS: SHARED MEMORY MODE—FLASH WRITE CYCLE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
t_{MFW1}	BPAM, SBHE, SA0–19 Setup to ↓ $\overline{MEMW}$		10		ns
t_{MFW2}	BPAM, SBHE, SA0–19 Hold After ↑ $\overline{MEMW}$		5		ns
t_{MFW3}	↓ IOCHRDY to ↓ $\overline{MEMW}$		0	35	ns
t_{MFW4}	$\overline{MEMW}$ Inactive		50		ns
t_{MFW5}	↑ $\overline{SRWE}$ to ↑ IOCHRDY		20	90	ns
t_{MFW6}	↑ $\overline{MEMW}$ Hold From ↑ IOCHRDY		0		ns
t_{MFW7}	SD Valid From ↓ $\overline{MEMW}$			175	ns
t_{MFW8}	SD Hold From ↑ $\overline{MEMW}$		0		ns
t_{MFW9}	$\overline{BPCS}$/PRDB Valid From ↓ $\overline{MEMW}$			175	ns
t_{MFW10}	$\overline{BPCS}$/PRDB Setup to ↓ $\overline{SRWE}$		15		ns
t_{MFW11}	$\overline{SRWE}$ Active		140	155	ns
t_{MFW12}	$\overline{BPCS}$/PRDB Hold From ↑ $\overline{SRWE}$		15		ns

SWITCHING CHARACTERISTICS: EADI

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
tEAD1	SRD Setup to ↑ SRDCLK		40		ns
tEAD2	SRD Hold to ↑ SRDCLK		40		ns
tEAD3	$\overline{SF/BD}$ Change to ↓ SRDCLK		–15	+15	ns
tEAD4	$\overline{EAR}$ Deassertion to ↑ SRDCLK (First Rising Edge)		50		ns
tEAD5	$\overline{EAR}$ Assertion From SFD Event (Packet Rejection)		0	51,090	ns
tEAD6	$\overline{EAR}$ Assertion		110		ns

Note: *External Address Detection interface is invoked by setting bit 3 in ISACSR2 and resetting bit 0 in ISACSR2. External MAU select is not available when EADISEL bit is set.*

SWITCHING CHARACTERISTICS: JTAG (IEEE 1149.1) INTERFACE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
tJTG1	TCK HIGH Assertion		20		ns
tJTG2	TCK Period		50		ns
tJTG3	TDI Setup to ↑ TCK		5		ns
tJTG4	TDI, TMS Hold From ↑ TCK		5		ns
tJTG5	TMS Setup to ↑ TCK		8		ns
tJTG6	TDO Active From ↓ TCK		0	30	ns
tJTG7	TDO Change From ↓ TCK		0	30	ns
tJTG8	TDO Tristate From ↓ TCK		0	25	ns

Note: *JTAG logic is reset with an internal Power-On Reset circuit independent of Sleep Modes.*

SWITCHING CHARACTERISTICS: GPSI

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Transmit Timing					
t_{GPT1}	STDCLK Period (802.3 Compliant)		99.99	100.01	ns
t_{GPT2}	STDCLK HIGH Time		40	60	ns
t_{GPT3}	TXDAT and TXEN Delay from ↑ TCLK		0	70	ns
t_{GPT4}	RXCRS Setup to ↑ STDCLK(Last Bit)		210		ns
t_{GPT5}	RXCRS Hold From ↓ TENA		0		ns
t_{GPT6}	CLSN Active Time to Trigger Collision	(Note 1)	110		ns
t_{GPT7}	CLSN Active to ↓ RXCRS to Prevent LCAR Assertion		0		ns
t_{GPT8}	CLSN Active to ↓ RXCRS for SQE Hearbeat Window		0	4.0	µs
t_{GPT9}	CLSN Active to ↑ RXCRS for Normal Collision		0	51.2	µs
Receive Timing					
t_{GPR1}	SRDCLK Period	(Note 2)	80	120	ns
t_{GPR2}	SRDCLK High Time	(Note 2)	30	80	ns
t_{GPR3}	SRDCLK Low Time	(Note 2)	30	80	ns
t_{GPR4}	RXDAT and RXCRS Setup to ↑ SRDCLK		15		ns
t_{GPR5}	RXDAT Hold From ↑ RCLK		15		ns
t_{GPR6}	RXCRS Hold From ↓ SRDCLK		0		ns
t_{GPR7}	CLSN Active to First ↑ SRDCLK (Collision Recognition)		0		ns
t_{GPR8}	CLSN Active to ↑ SRDCLKfor Address Type Designation Bit	(Note 3)	51.2		µs
t_{GPR9}	CLSN Setup to last ↑ SRDCLKfor Collision Recognition		210		ns
t_{GPR10}	CLSN Active		110		ns
t_{GPR11}	CLSN Inactive Setup to First ↑ RCLK		300		ns
t_{GPR12}	CLSN Inactive Hold to Last ↑ RCLK		300		ns

Notes:

1. *CLSN must be asserted for a continuous period of 110 ns or more. Assertion for less than 110 ns period may or may not result in CLSN recognition.*
2. *RCLK should meet jitter requirements of IEEE 802.3 specification.*
3. *CLSN assertion before 51.2 µs will be indicated as a normal collision. CLSN assertion after 51.2 µs will be considered as a Late Receive Collision.*

SWITCHING CHARACTERISTICS: AUI

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
AUI Port					
t_{DOTR}	DO+,DO- Rise Time (10% to 90%)		2.5	5.0	ns
t_{DOTF}	DO+,DO- Fall Time (90% to 10%)		2.5	5.0	ns
t_{DORM}	DO+,DO- Rise and fall Time Mismatch		–	1.0	ns
t_{DOETD}	DO+/- End of Transmission		200	375	ns
t_{PWODI}	DI Pulse Width Accept/Reject Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 1)	15	45	ns
t_{PWKDI}	DI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 2)	136	200	ns
t_{PWOCI}	CI Pulse Width Accept/Reject Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 3)	10	26	ns
t_{PWKCI}	CI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 4)	90	160	ns
Internal MENDEC Clock Timing					
t_{X1}	XTAL1 Period	V_{IN} = External Clock	49.995	50.005	ns
t_{X1H}	XTAL1 HIGH Pulse Width	V_{IN} = External Clock	20		ns
t_{X1L}	XTAL1 LOW Pulse width	V_{IN} = External Clock	20		ns
t_{X1R}	XTAL1 Rise Time	V_{IN} = External Clock		5	ns
t_{X1F}	XTAL1 Fall Time	V_{IN} = External Clock		5	ns

Notes:

1. *DI pulses narrower than t_{PWODI} (min) will be rejected; pulses wider than t_{PWODI} (max) will turn internal DI carrier sense on.*
2. *DI pulses narrower than t_{PWKDI} (min) will maintain internal DI carrier sense on; pulses wider than t_{PWKDI} (max) will turn internal DI carrier sense off.*
3. *CI pulses narrower than t_{PWOCI} (min) will be rejected; pulses wider than t_{PWOCI} (max) will turn internal CI carrier sense on.*
4. *CI pulses narrower than t_{PWKCI} (min) will maintain internal CI carrier sense on; pulses wider than t_{PWKCI} (max) will turn internal CI carrier sense off.*

SWITCHING CHARACTERISTICS: 10BASE-T INTERFACE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Transmit Timing					
t_{TETD}	Transmit Start of Idle		250	350	ns
t_{TR}	Transmitter Rise Time	(10% to 90%)		5.5	ns
t_{TF}	Transmitter Fall Time	(90% to 10%)		5.5	ns
t_{TM}	Transmitter Rise and Fall Time Mismatch			2	ns
t_{PERLP}	Idle Signal Period		8	24	ms
t_{PWLP}	Idle Link Pulse Width	(Note 1)	75	120	ns
t_{PWPLP}	Predistortion Idle Link Pulse Width	(Note 1)	45	55	ns
t_{JA}	Transmit Jabber Activation Time		20	150	ms
t_{JR}	Transmit Jabber Reset Time		250	750	ms
Receive Timing					
t_{PWNRD}	RXD Pulse Width Not to Turn Off Internal Carrier Sense	VIN > VTHS (min)	136	–	ns
t_{PWROFF}	RXD Pulse Width to Turn Off	VIN > VTHS (min)		200	ns

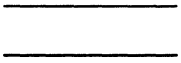
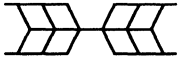
Note:

1. Not tested; parameter guaranteed by characterization.

SWITCHING CHARACTERISTICS: SERIAL EEPROM

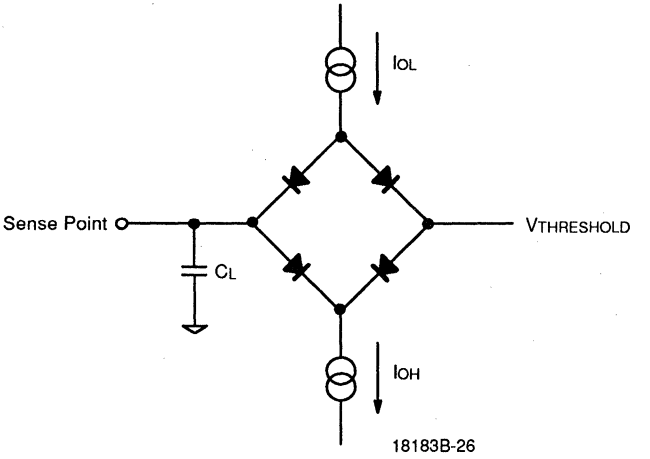
Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
t_{SR1}	EESK High Time		790		ns
t_{SR2}	EESK Low Time		790		ns
t_{SR3}	↑ EECS EEDI From ↓ EESK		-15	15	ns
t_{SR4}	↓ EECS, EEDI and SHFBUSY From ↓ EESK		-15	15	ns
t_{SR5}	EECS Low Time		1590		ns
t_{SR6}	EEDO Setup to ↑ EESK		35		ns
t_{SR7}	EEDO Hold From ↑ EESK		0		ns
t_{SL1}	EEDO Setup to ↓ $\overline{IOR}$		95		ns
t_{SL2}	EEDO Setup to ↑ IOCHRDY		140		ns
t_{SL3}	EESK, EEDI, EECS and SHFBUSY Delay From ↑ $\overline{IOW}$		160	235	ns

KEY TO SWITCHING WAVEFORMS

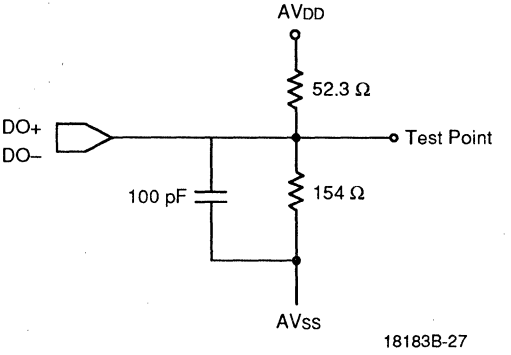
WAVEFORM	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off" State

KS000010

SWITCHING TEST CIRCUITS

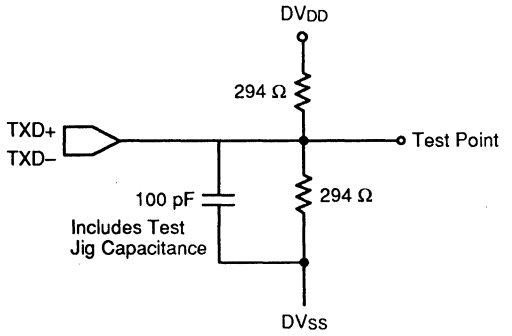


Normal and Three-State Outputs

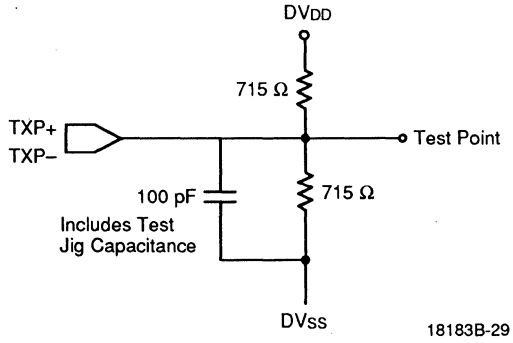


AUI DO Switching Test Circuit

SWITCHING TEST CIRCUITS

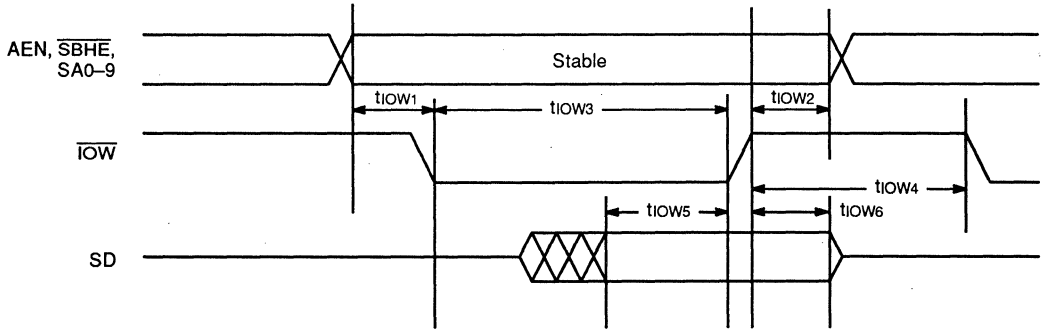


TXD Switching Test Circuit



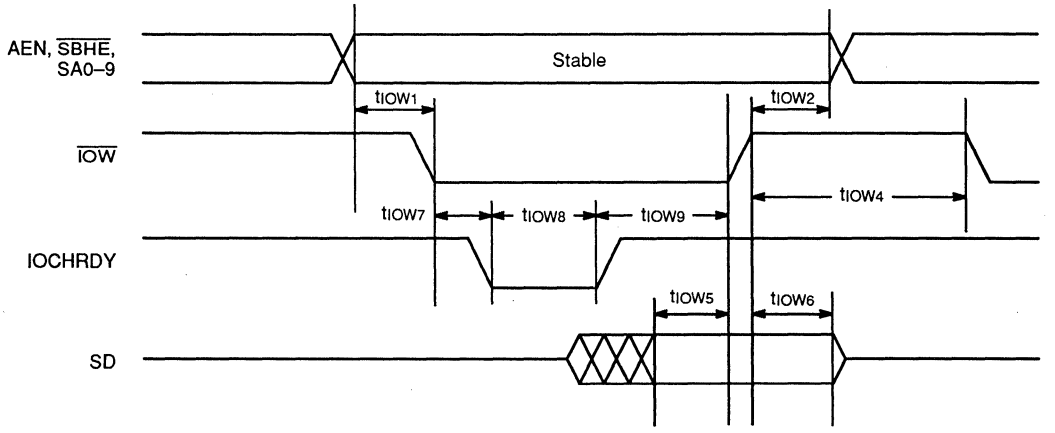
TXP Outputs Test Circuit

SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-30

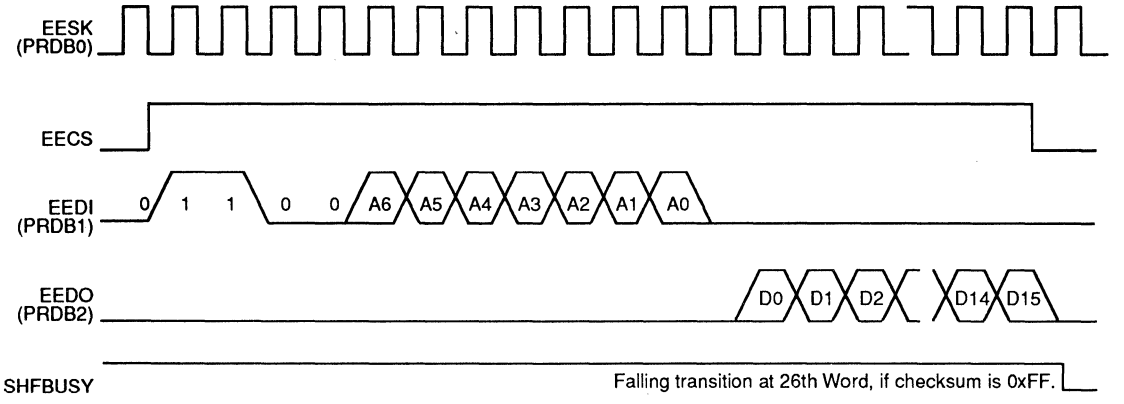
I/O Write without Wait States



18183B-31

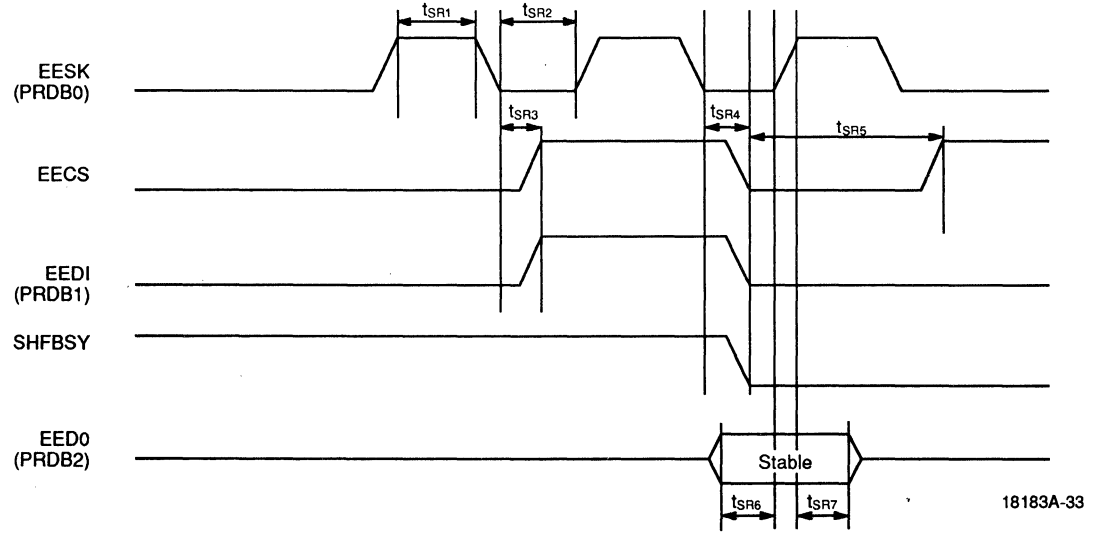
I/O Write with Wait States

SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-32

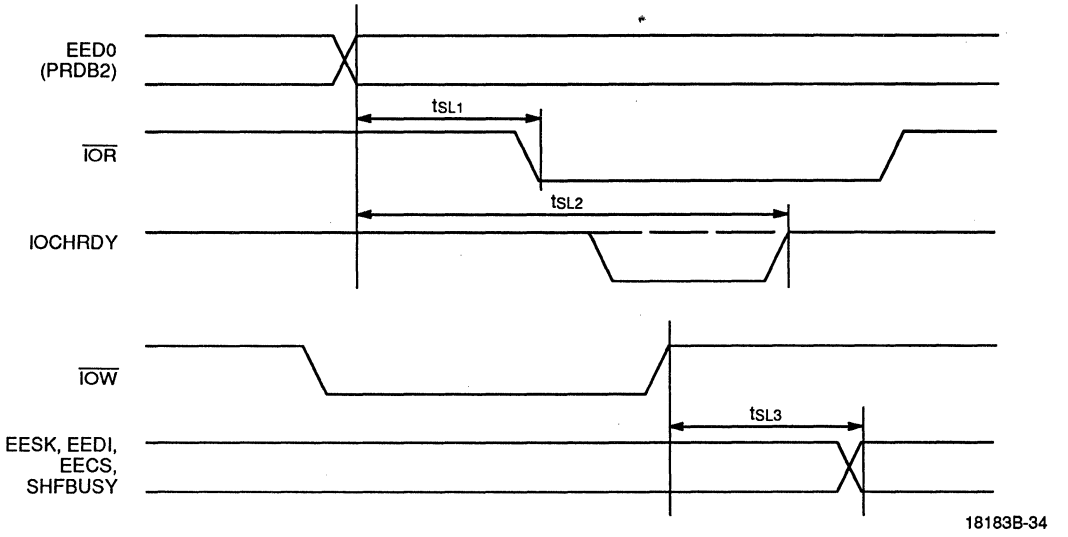
Serial Shift EEPROM Interface Read Timing



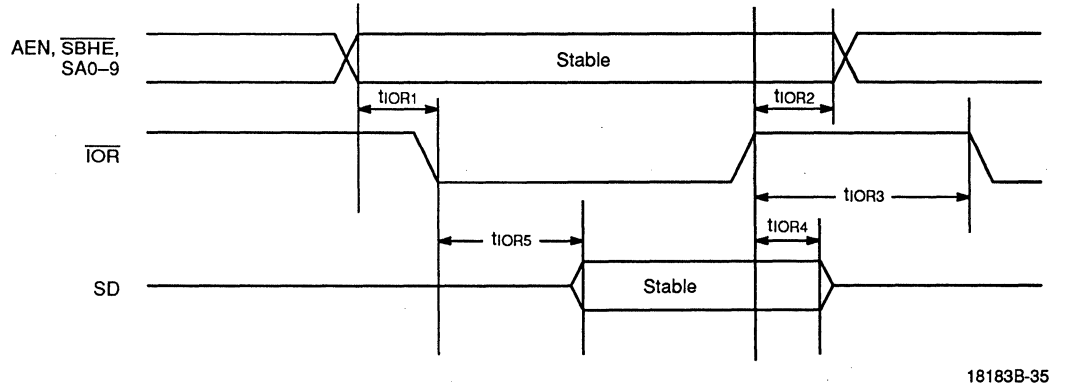
18183A-33

Serial EEPROM Control Timing

SWITCHING WAVEFORMS: BUS MASTER MODE

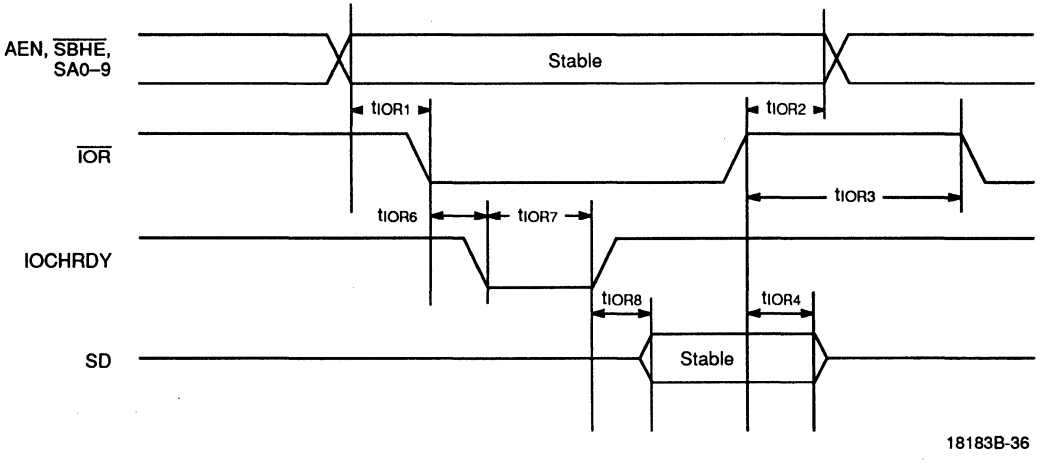


Slave Serial EEPROM Latency Timing

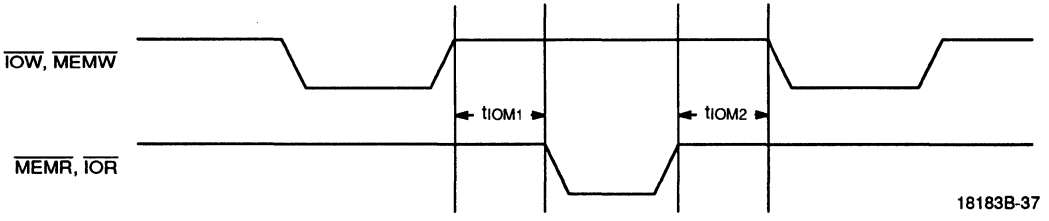


I/O Read without Wait States

SWITCHING WAVEFORMS: BUS MASTER MODE

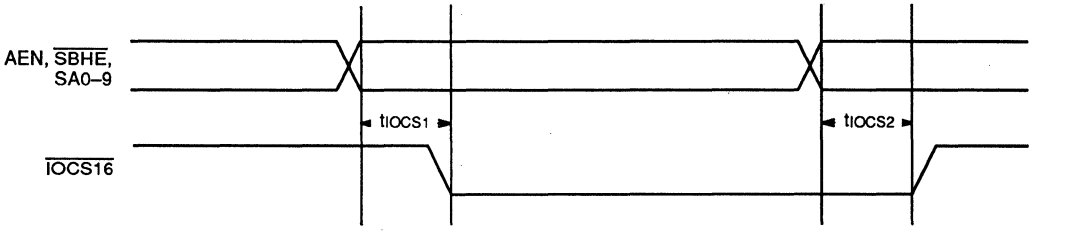


I/O Read with Wait States

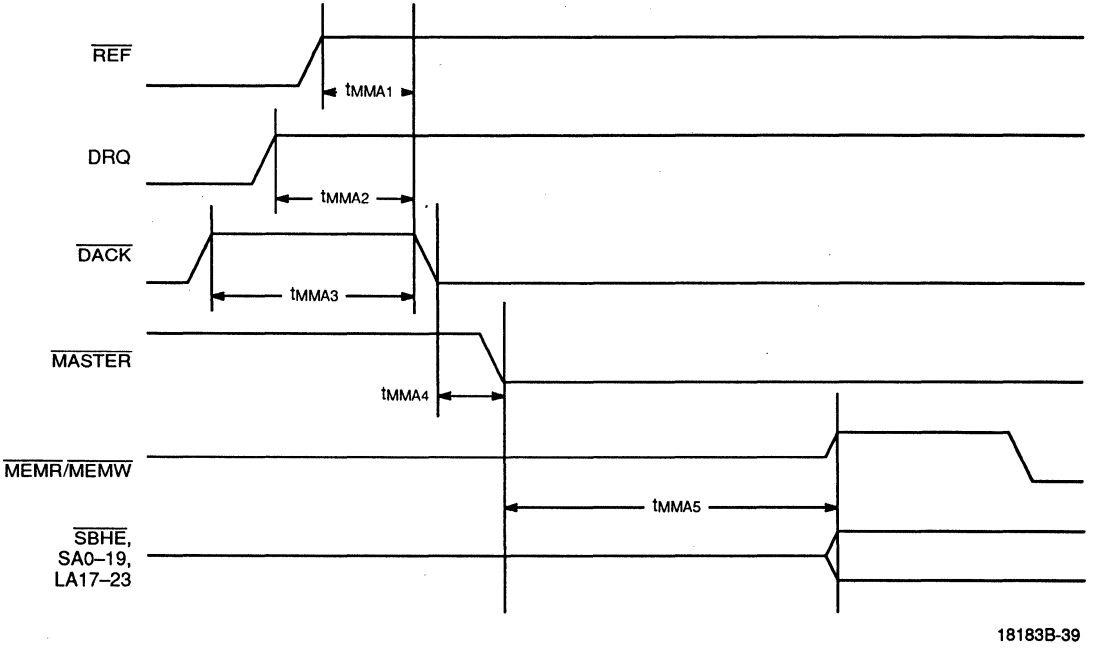


I/O to Memory Command Inactive Time

SWITCHING WAVEFORMS: BUS MASTER MODE

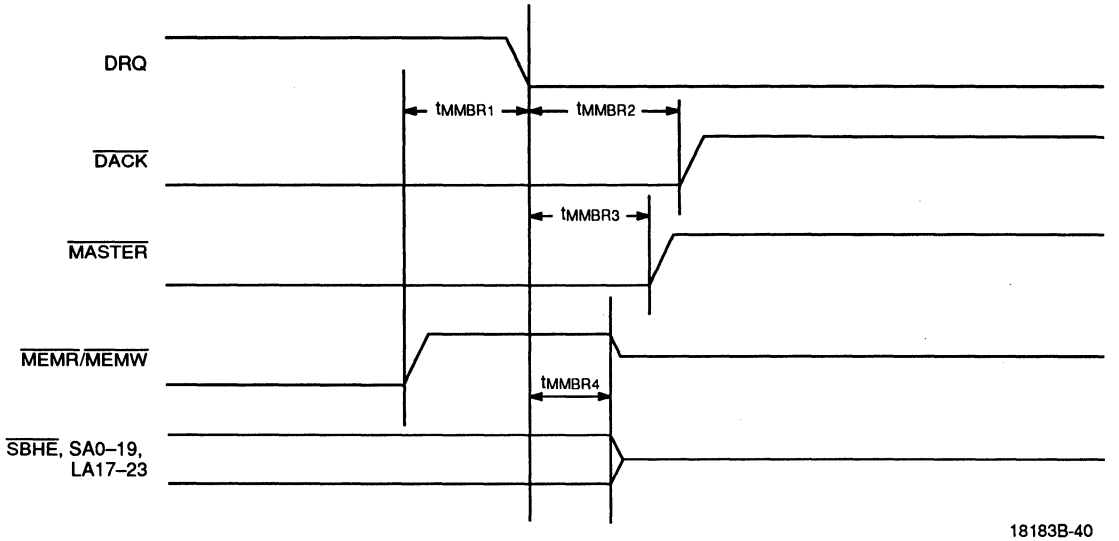


IOCS16 Timings



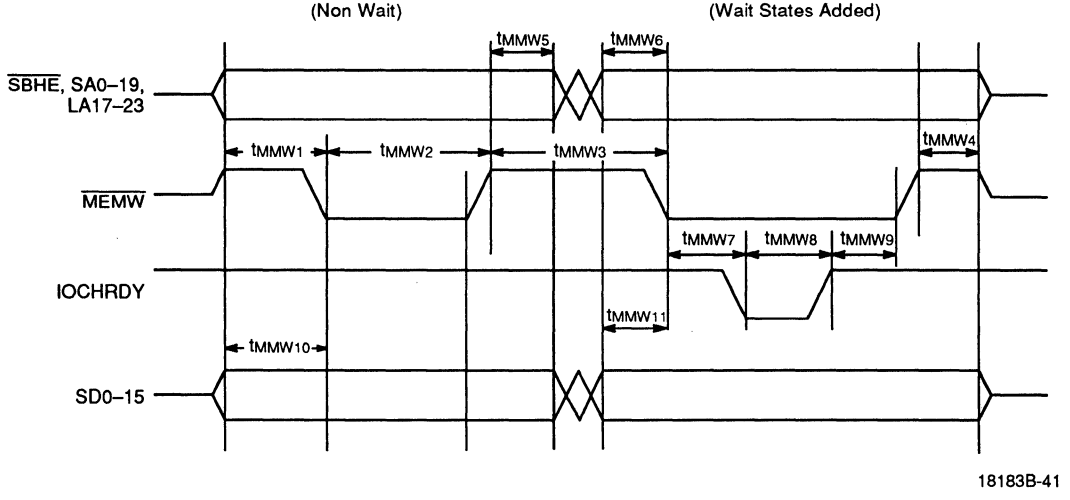
Bus Acquisition

SWITCHING WAVEFORMS: BUS MASTER MODE



18183B-40

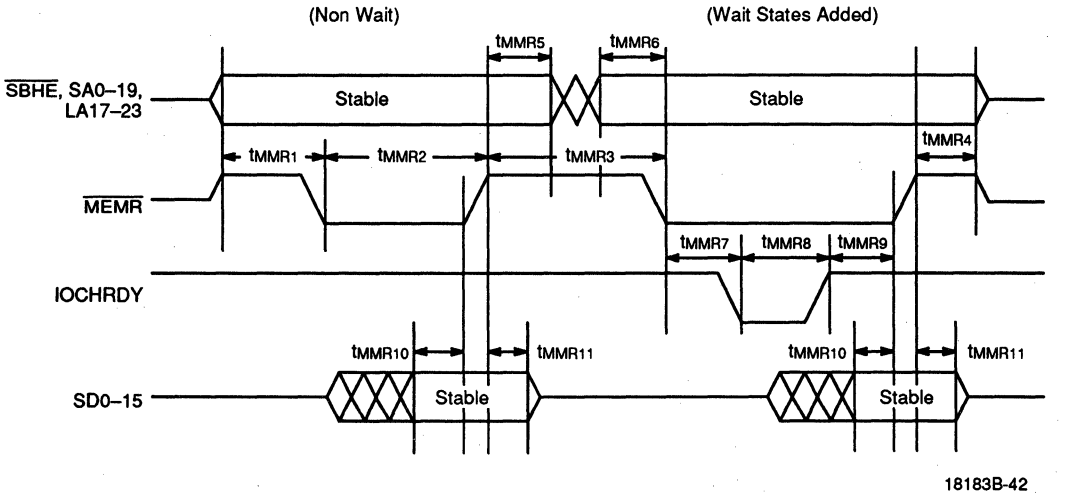
Bus Release



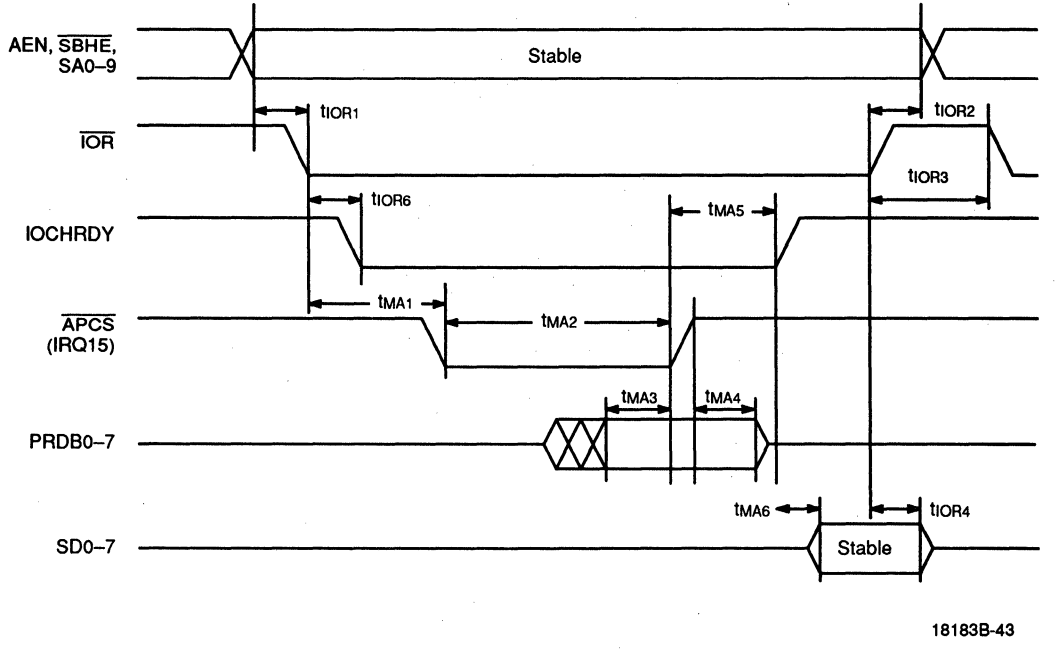
18183B-41

Write Cycles

SWITCHING WAVEFORMS: BUS MASTER MODE

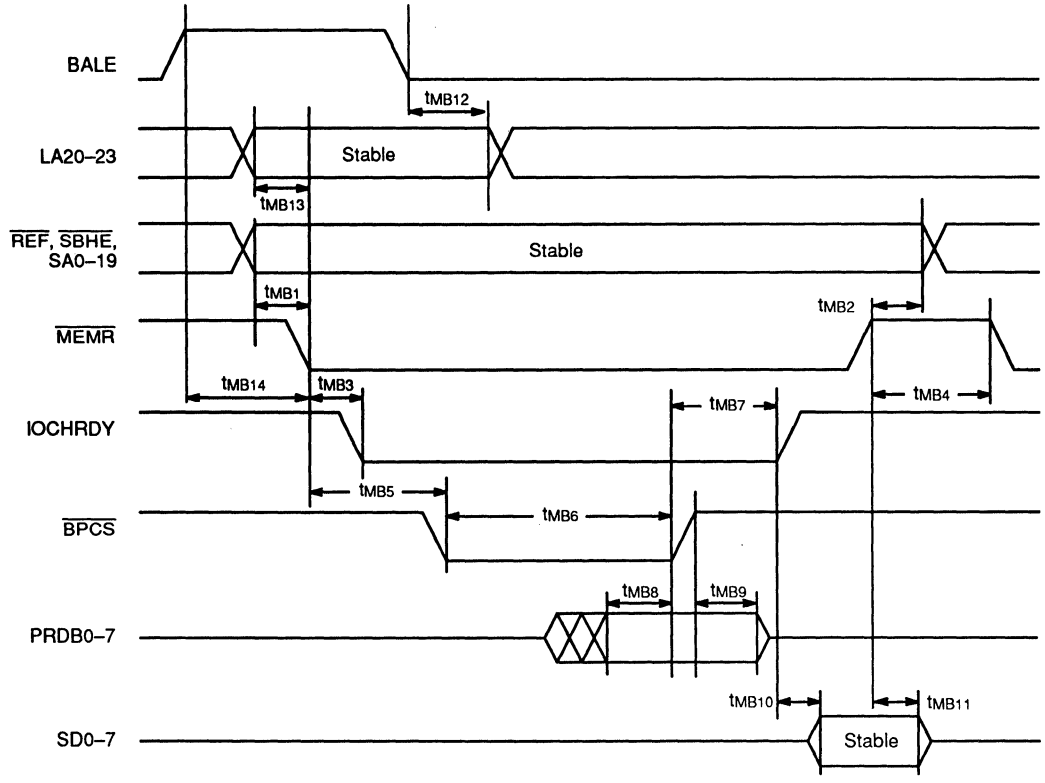


Read Cycles



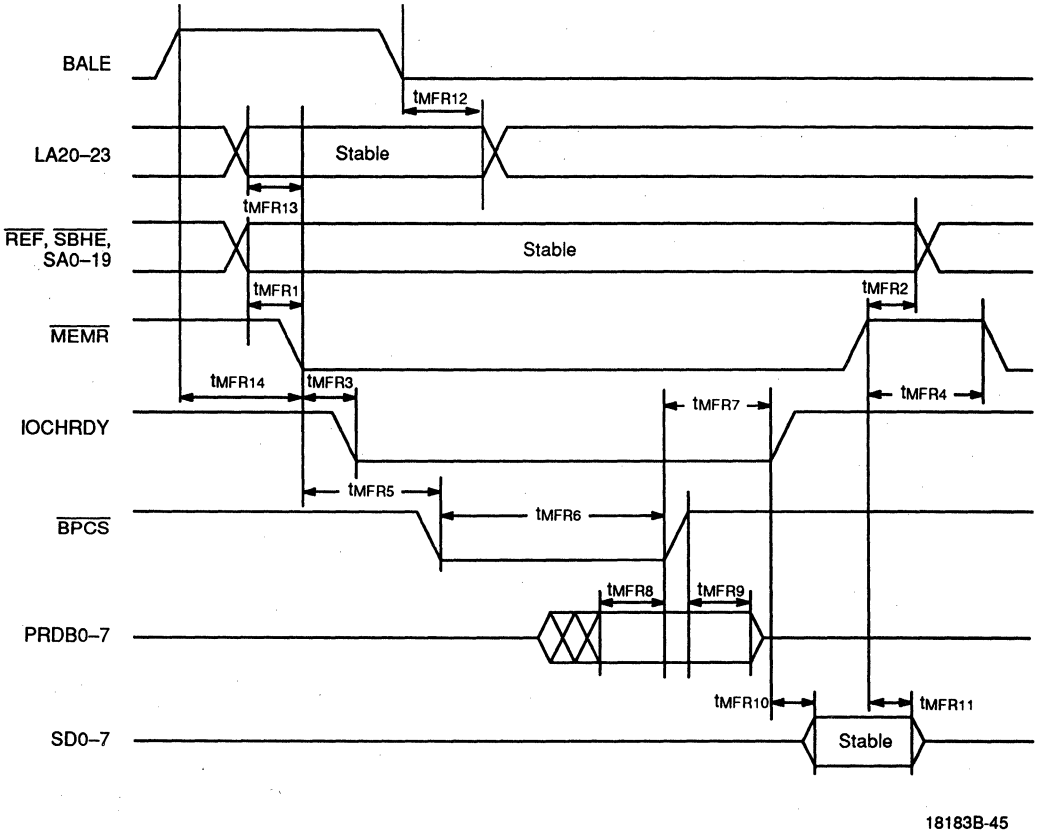
External Address PROM Read Cycle

SWITCHING WAVEFORMS: BUS MASTER MODE



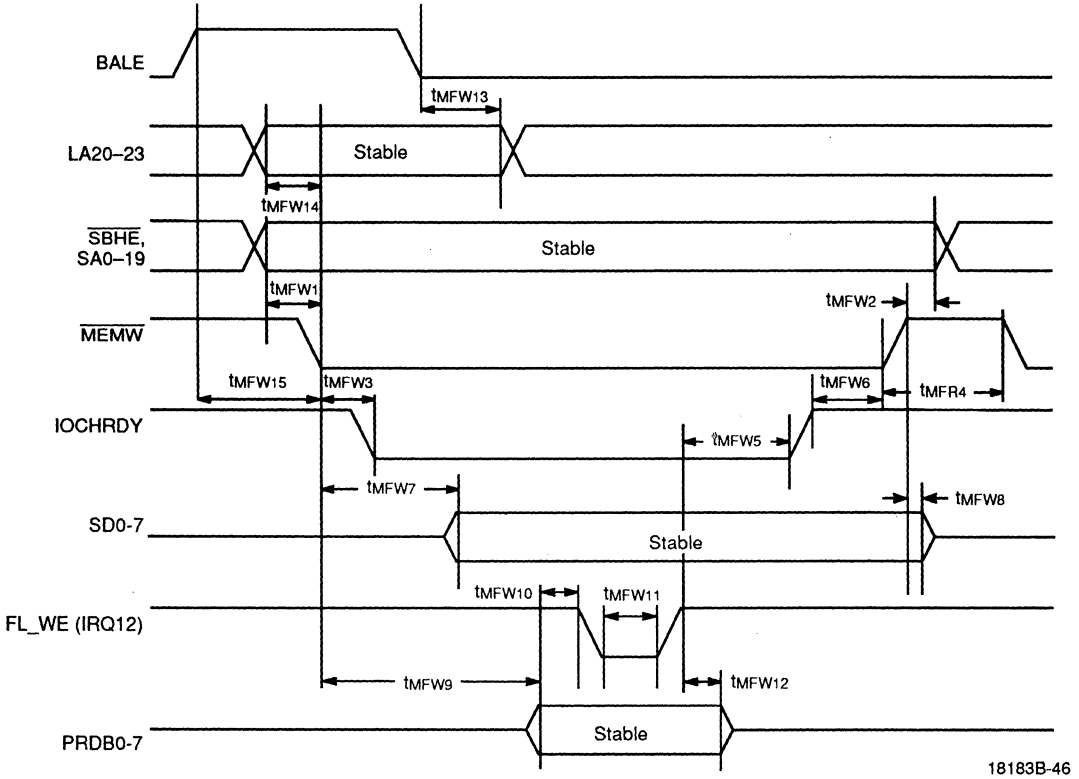
Boot PROM Read Cycle

SWITCHING WAVEFORMS: BUS MASTER MODE



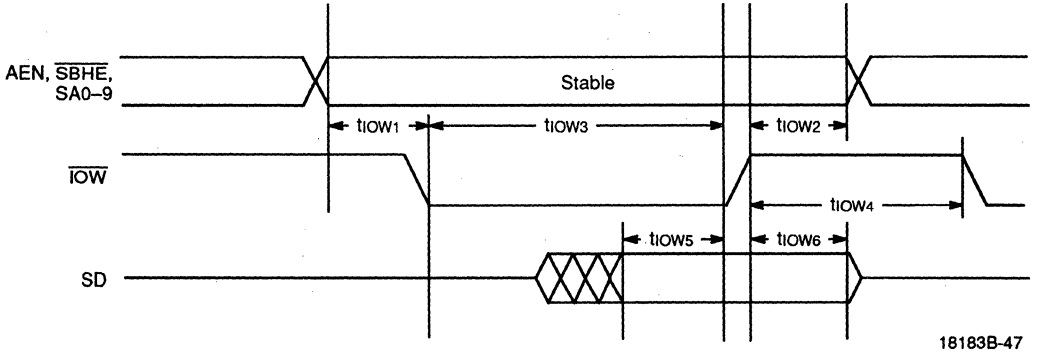
Flash Read Cycle

SWITCHING WAVEFORMS: BUS MASTER MODE

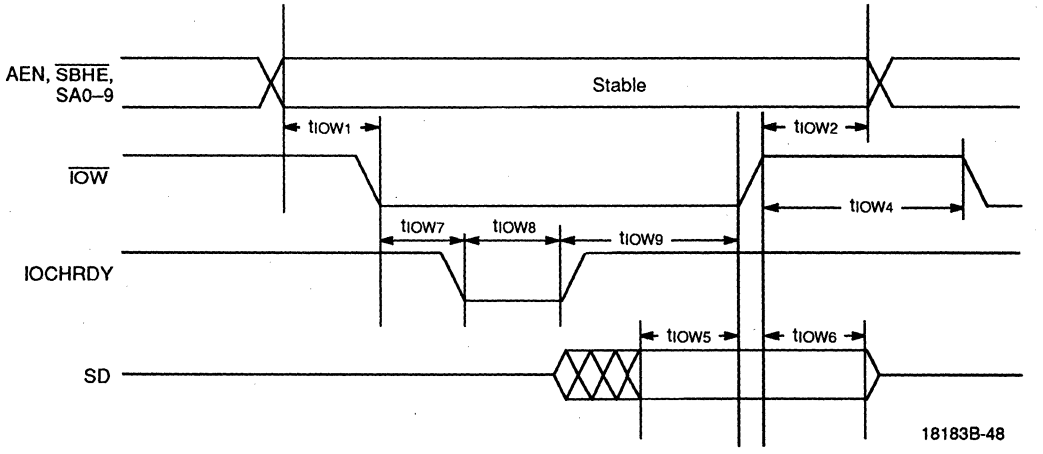


Flash Write Cycle

SWITCHING WAVEFORMS: SHARED MEMORY MODE

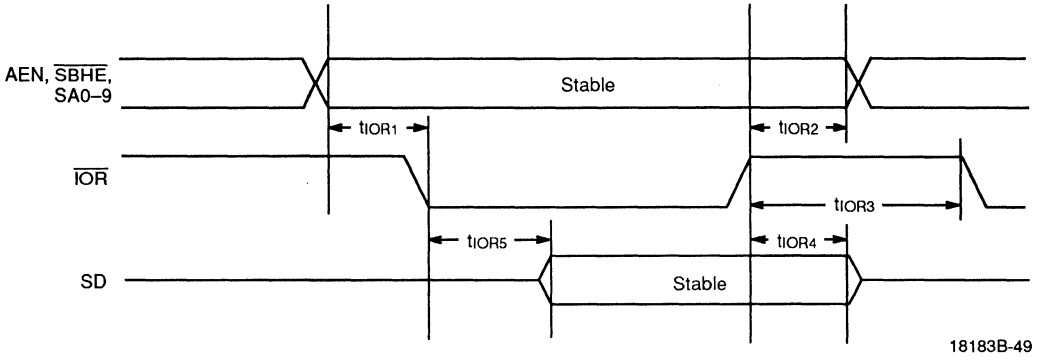


I/O Write without Wait States

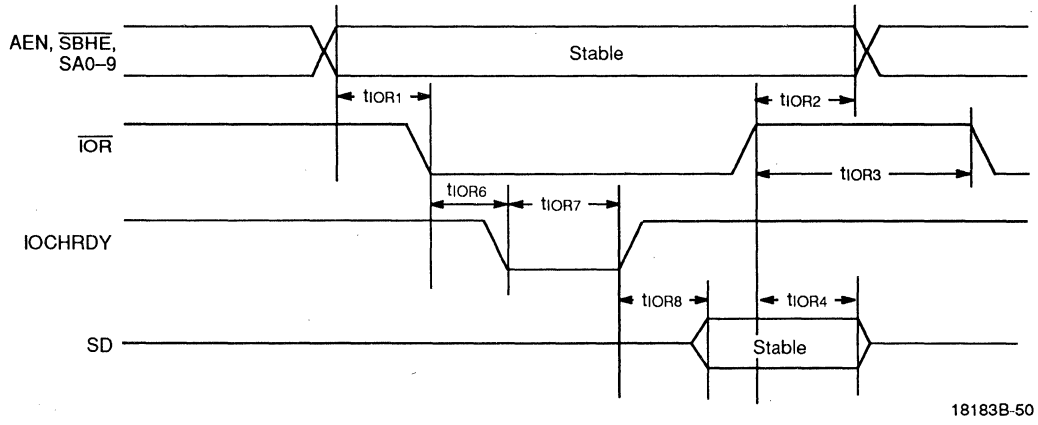


I/O Write with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE

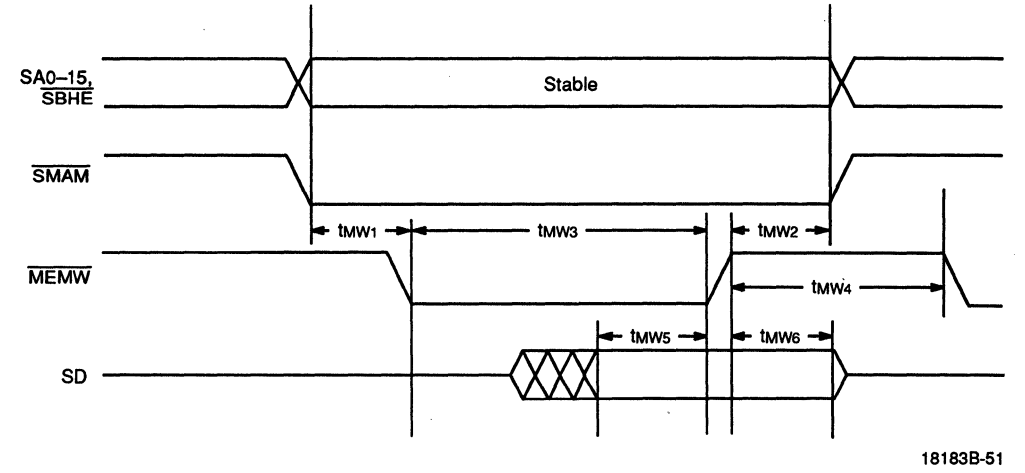


I/O Read without Wait States

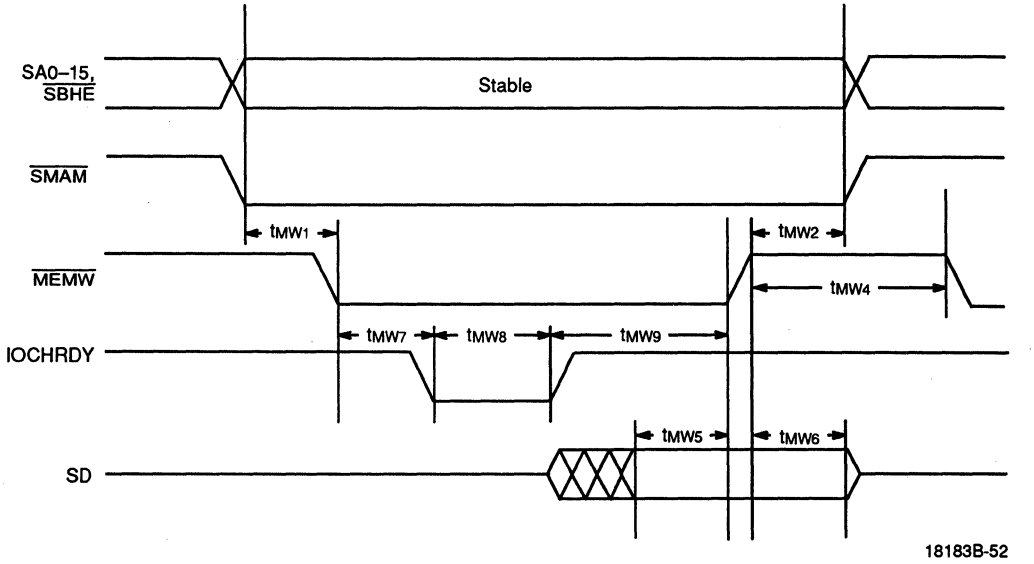


I/O Read with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE

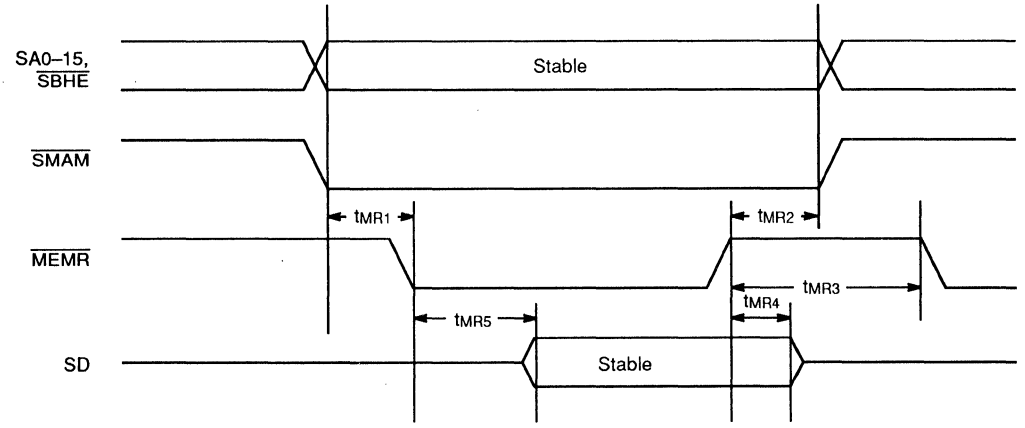


Memory Write without Wait States



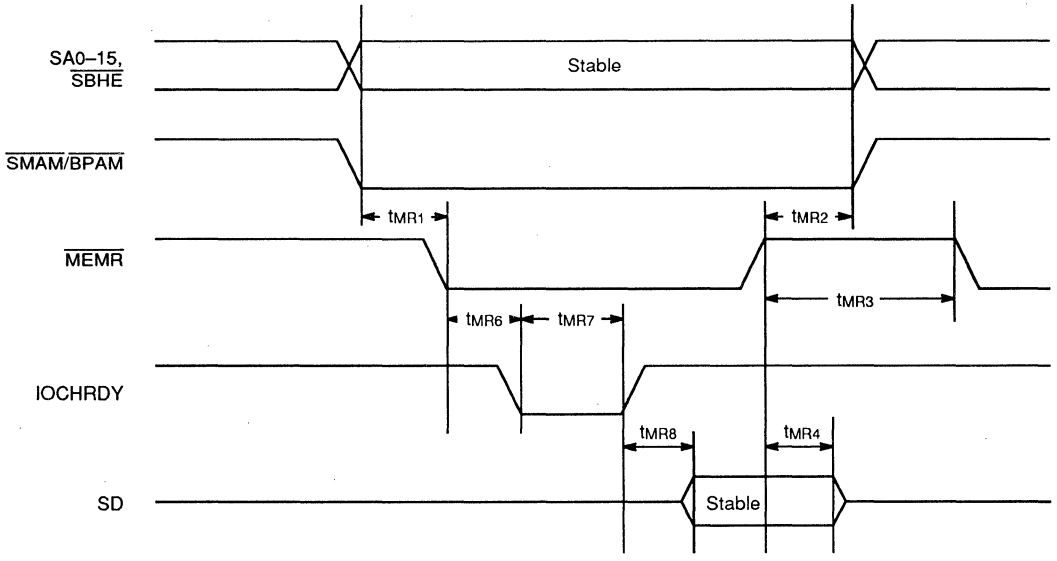
Memory Write with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-53

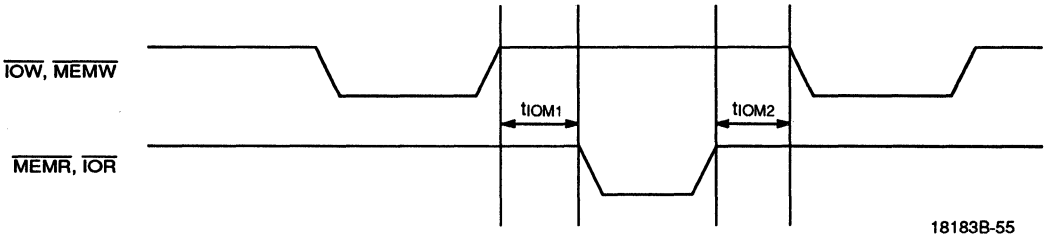
Memory Read without Wait States



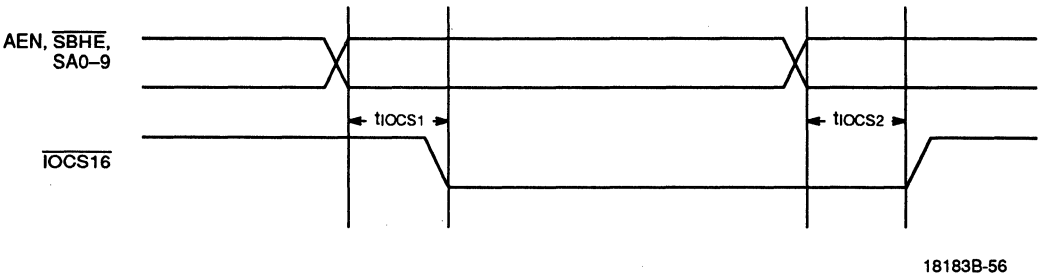
18183B-54

Memory Read with Wait States

SWITCHING WAVEFORMS: SHARED MEMORY MODE

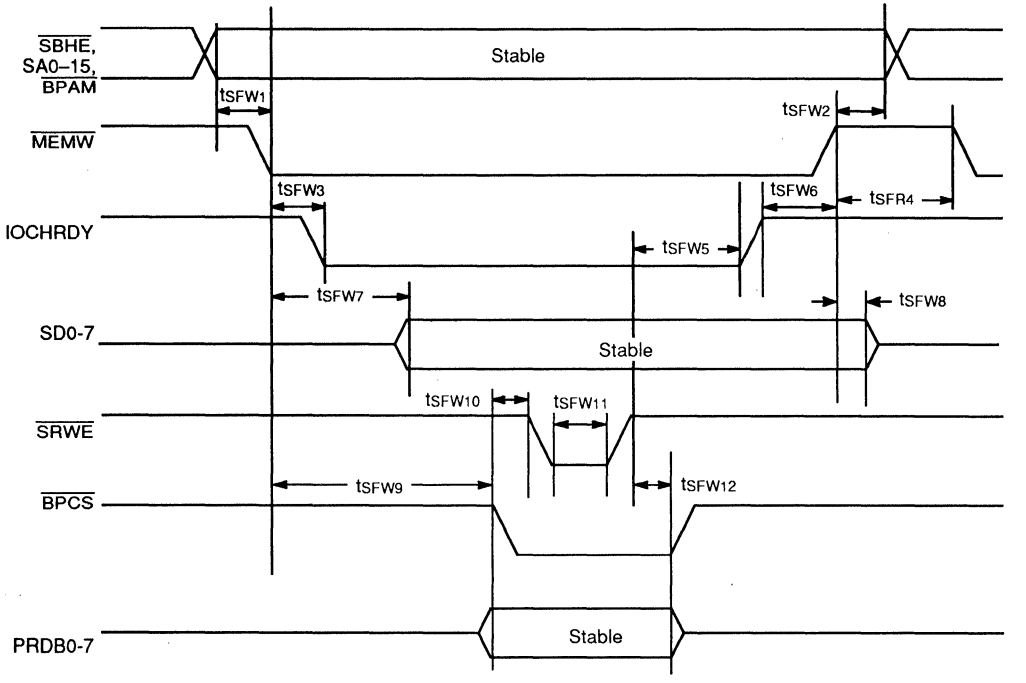


I/O to Memory Command Inactive Time



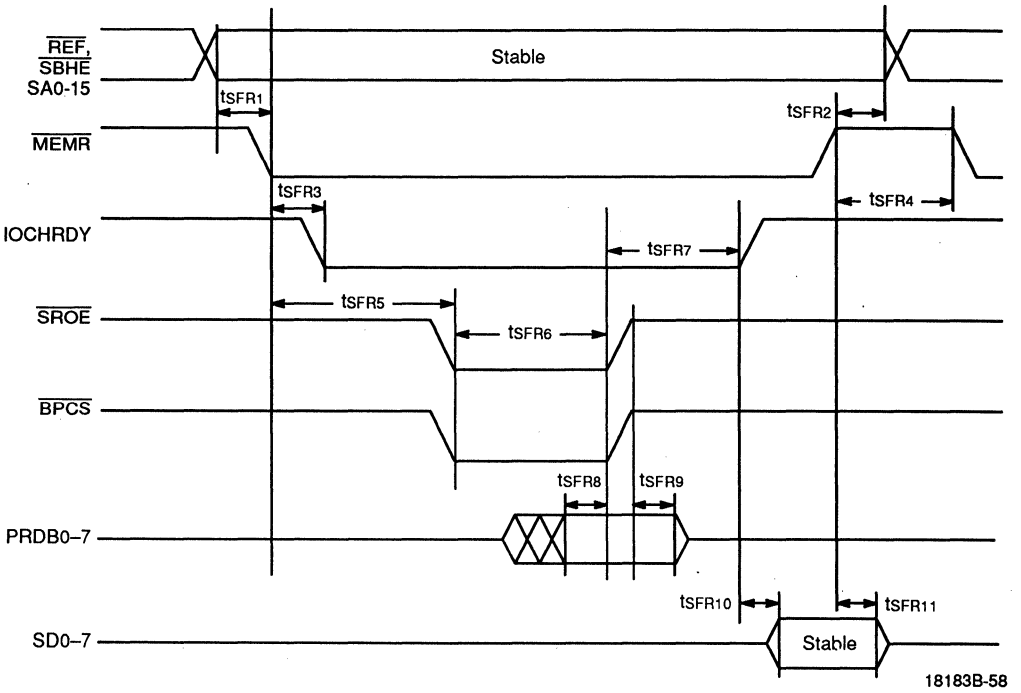
IOCS16 Timings

SWITCHING WAVEFORMS: SHARED MEMORY MODE



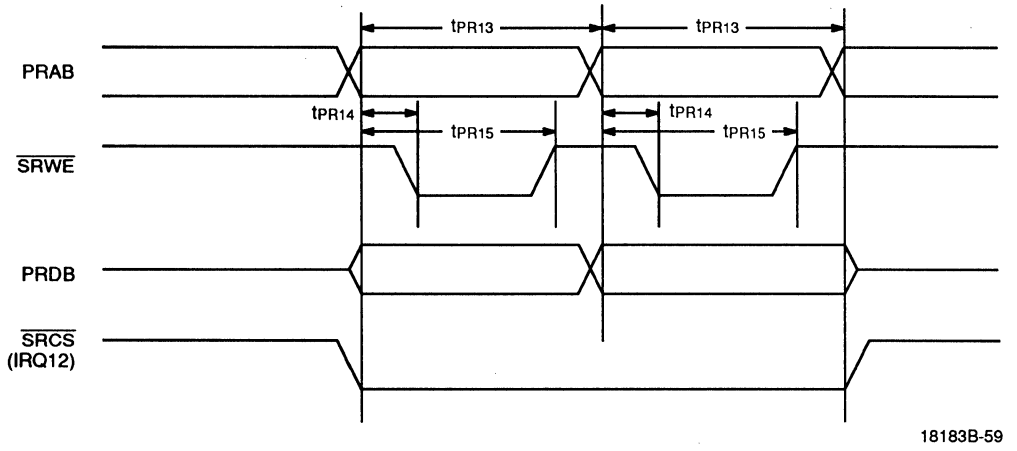
Flash Write Cycle

SWITCHING WAVEFORMS: SHARED MEMORY MODE

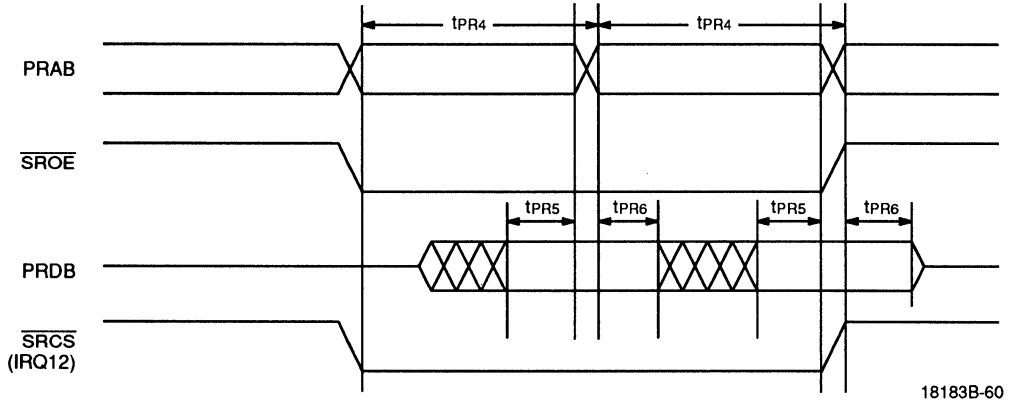


Flash Read Cycle

SWITCHING WAVEFORMS: SHARED MEMORY MODE

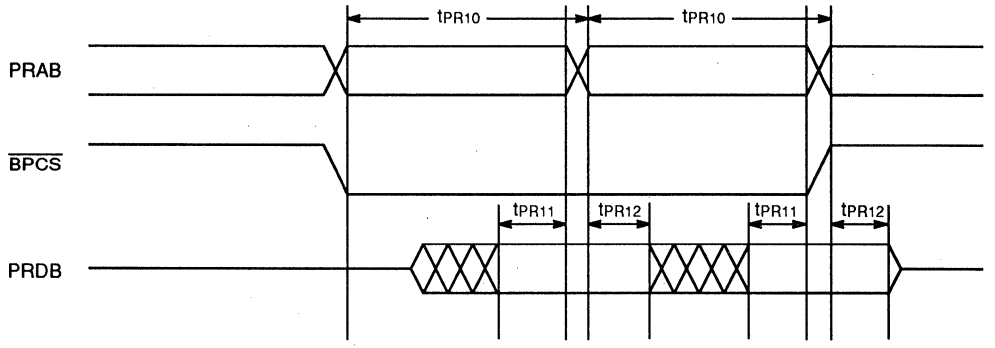


SRAM Write on Private Bus (When FL_Sel is Enabled)



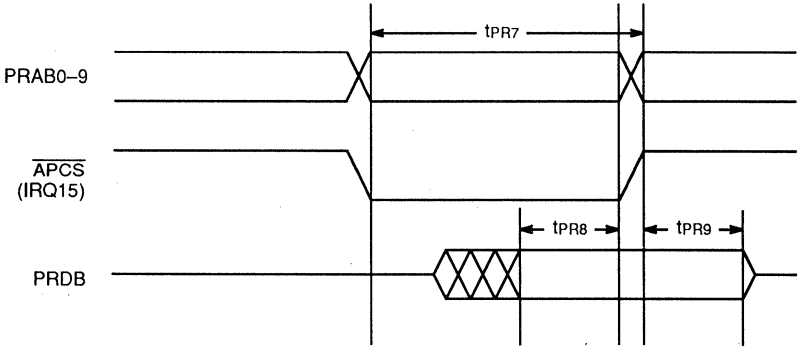
SRAM Read on Private Bus (When FL_Sel is Enabled)

SWITCHING WAVEFORMS: SHARED MEMORY MODE



18183B-61

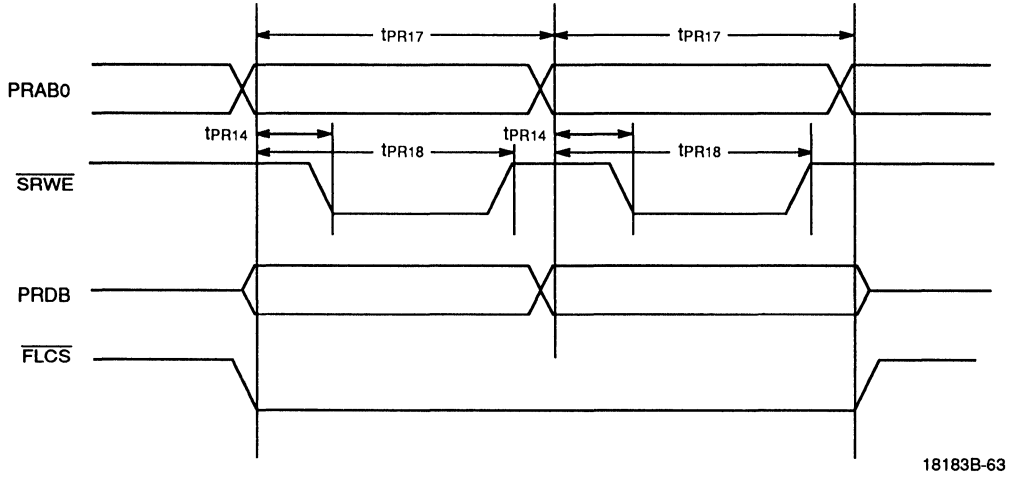
Boot PROM Read on Private Bus



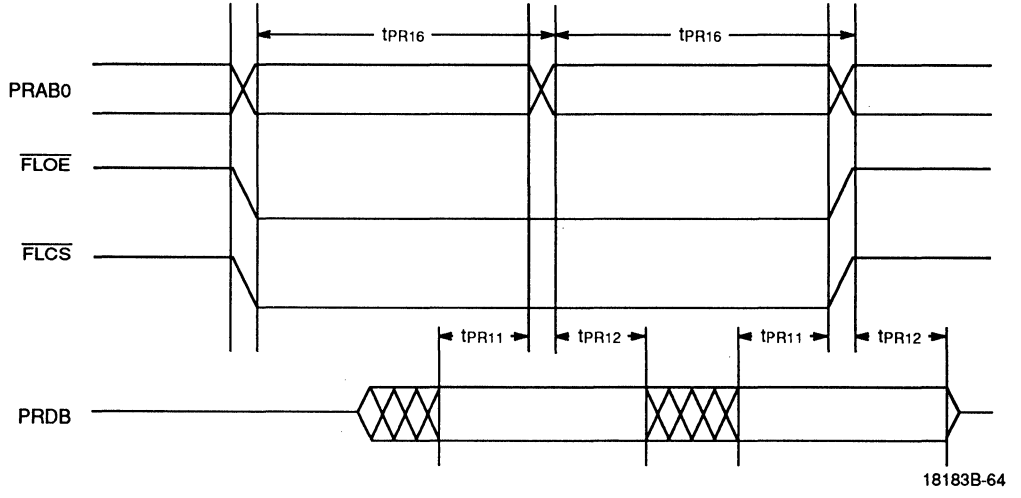
18183B-62

Address PROM Read on Private Bus

SWITCHING WAVEFORMS: SHARED MEMORY MODE

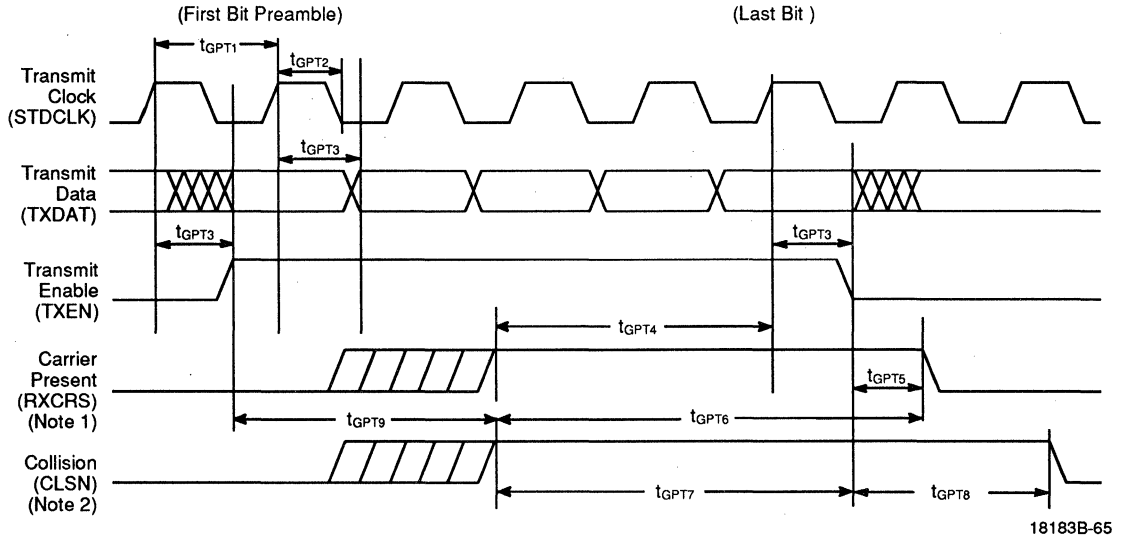


Flash Write on Private Bus



Flash Read on Private Bus

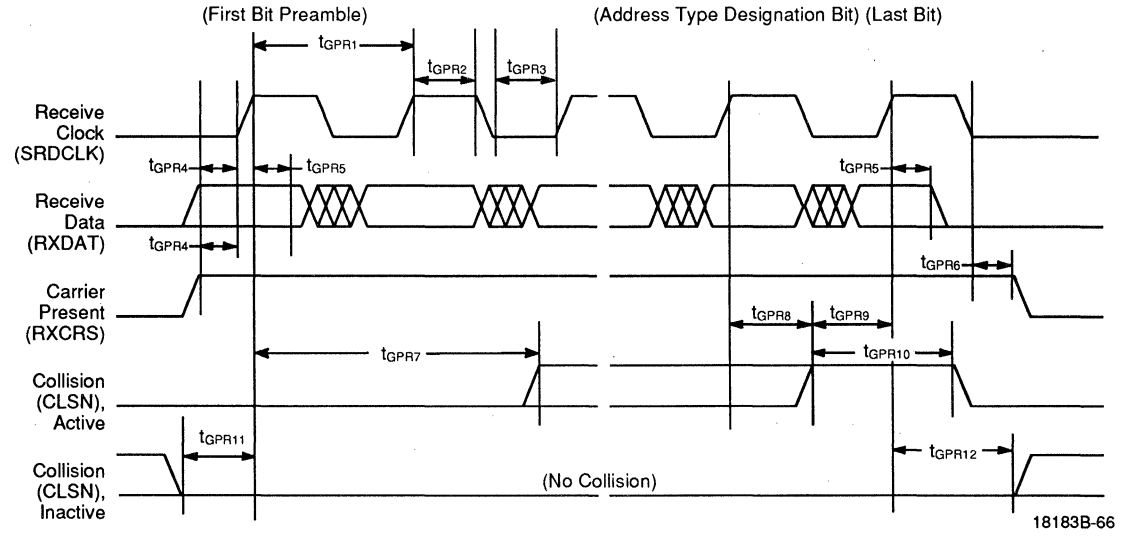
SWITCHING WAVEFORMS: GPSI



Notes:

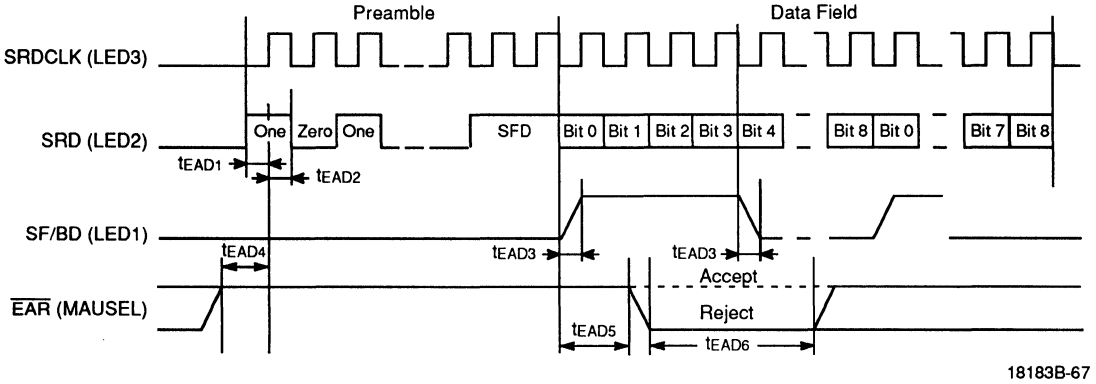
1. *If RXCRS is not present during transmission, LCAR bit in TMD3 will be set.*
2. *If CLSN is not present during or shortly after transmission, CERR in CSR0 will be set.*

Transmit Timing



Receive Timing

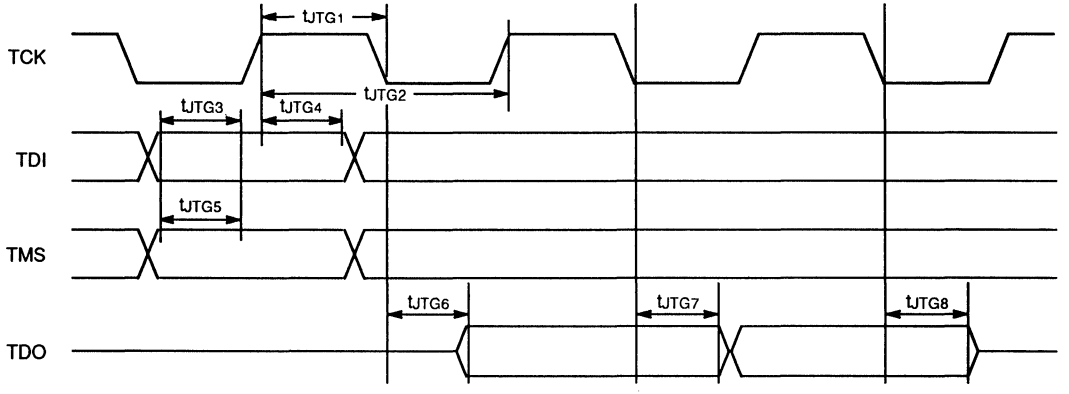
SWITCHING WAVEFORMS: EADI



18183B-67

EADI Reject Timing

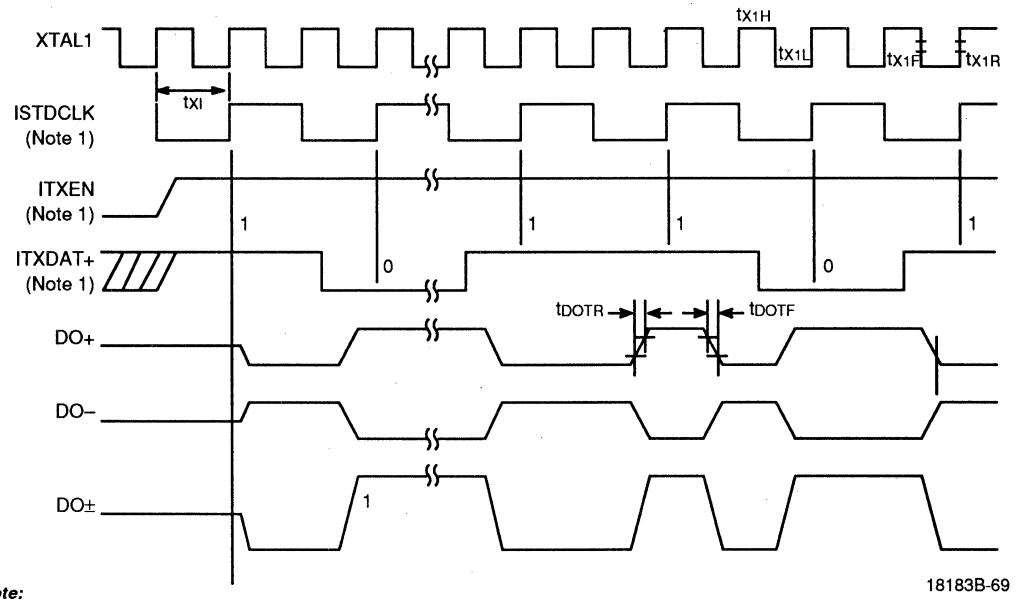
SWITCHING WAVEFORMS: JTAG (IEEE 1149.1) INTERFACE



18183B-68

Test Access Port Timing

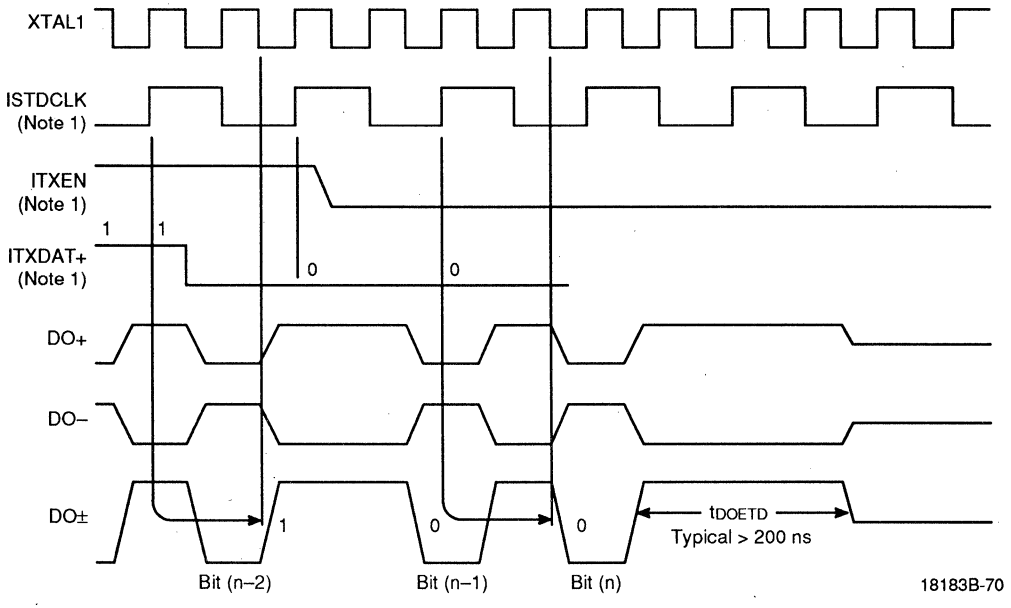
SWITCHING WAVEFORMS: AUI



Note:

1. Internal signal and is shown for clarification only.

Transmit Timing—Start of Packet

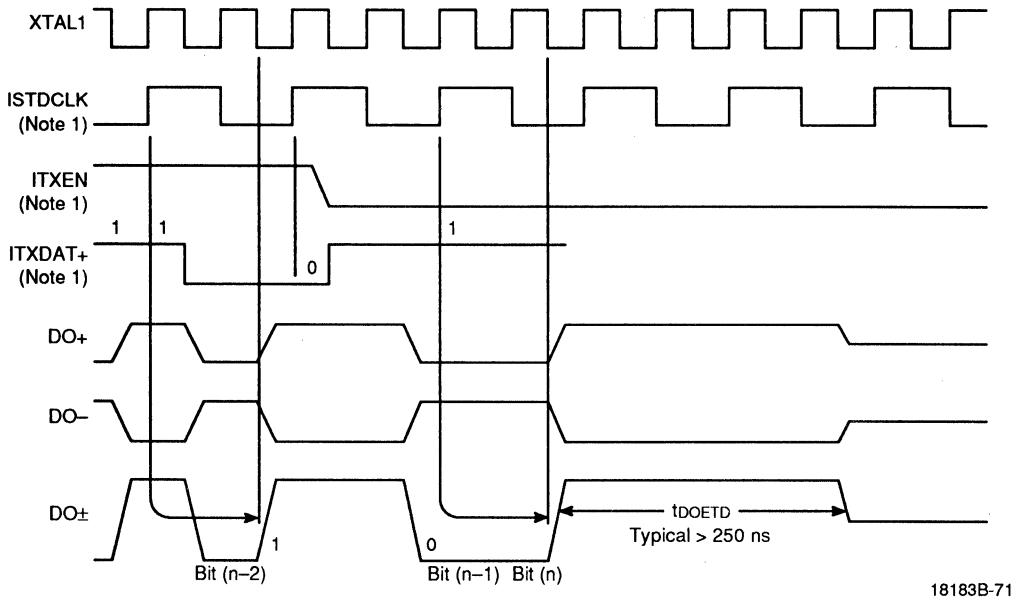


Note:

1. Internal signal and is shown for clarification only.

Transmit Timing—End of Packet (Last Bit = 0)

SWITCHING WAVEFORMS: AUI

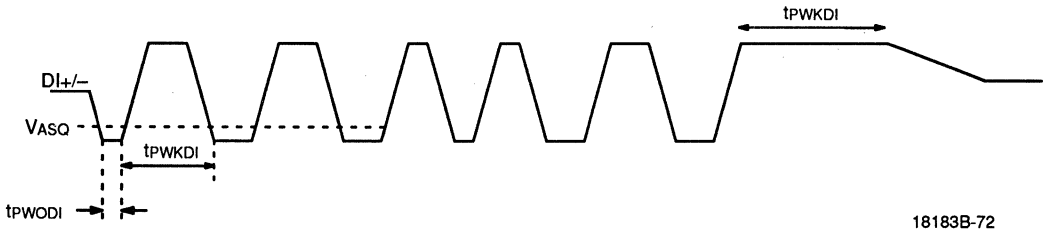


Note:

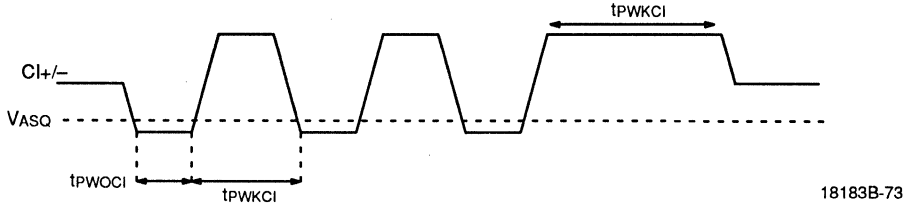
1. Internal signal and is shown for clarification only.

Transmit Timing—End of Packet (Last Bit = 1)

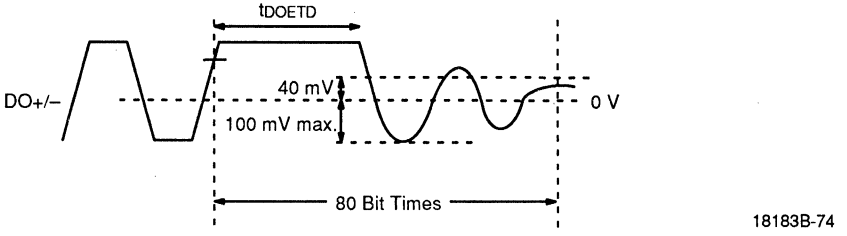
SWITCHING WAVEFORMS: AUI



Receive Timing Diagram

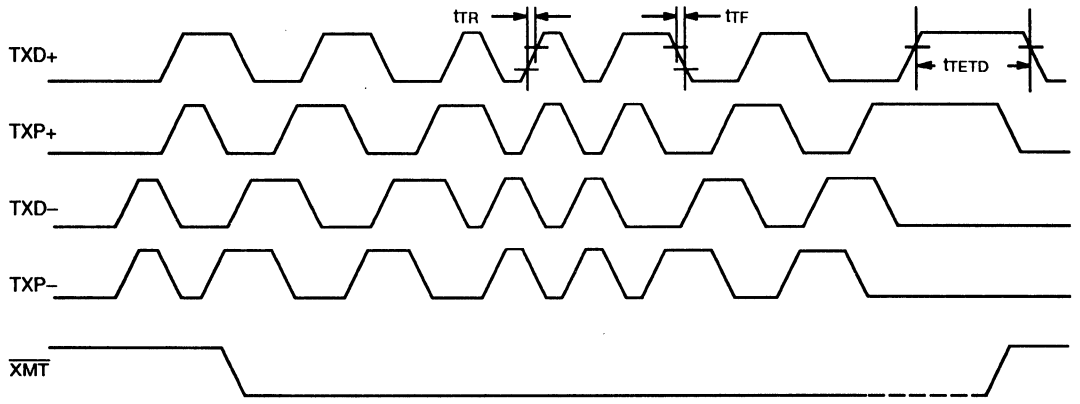


Collision Timing Diagram



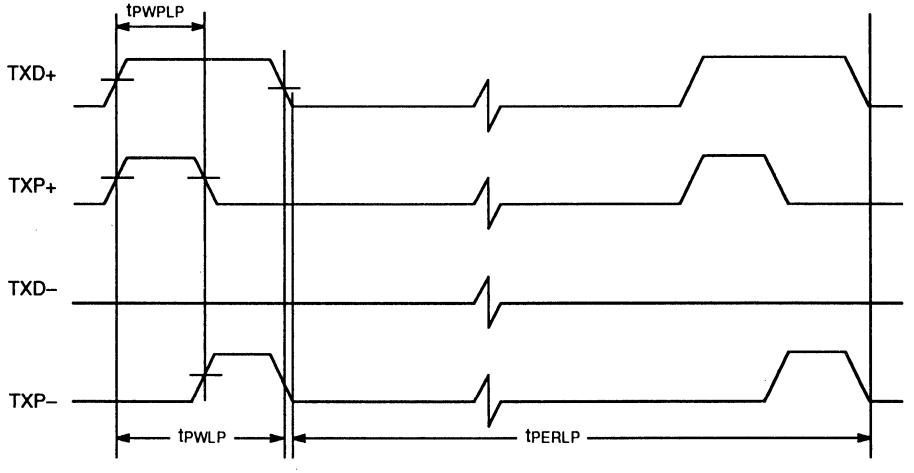
Port DO ETD Waveform

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



18183B-75

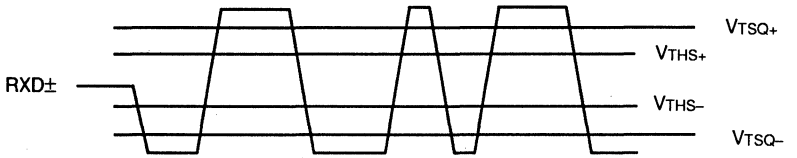
Transmit Timing



18183B-76

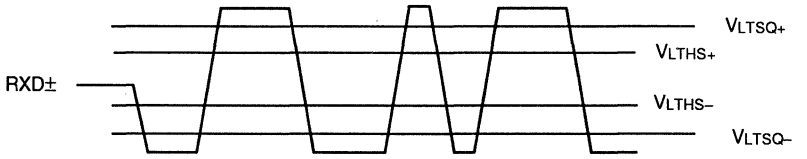
Idle Link Test Pulse

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



18183B-77

Receive Thresholds (LRT = 0 in CSR15 bit 9)



18183B-78

Receive Thresholds (LRT = 1 in CSR15 bit 9)

PCnet-ISA+ Compatible Media Interface Modules

PCnet-ISA+ COMPATIBLE 10BASE-T FILTERS AND TRANSFORMERS

The table below provides a sample list of PCnet-ISA+ compatible 10BASE-T filter and transformer modules available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Filters and Transformers	Filters Transformers and Choke	Filters Transformers Dual Choke	Filters Transformers Dual Chokes
Bel Fuse	A556-2006-DE	16-pin 0.3" DIL	√			
Bel Fuse	0556-2006-00	14-pin SIP	√			
Bel Fuse	0556-2006-01	14-pin SIP			√	
Bel Fuse	0556-6392-00	16-pin 0.5" DIL			√	
Halo Electronics	FD02-101G	16-pin 0.3" DIL	√			
Halo Electronics	FD12-101G	16-pin 0.3" DIL		√		
Halo Electronics	FD22-101G	16-pin 0.3" DIL			√	
PCA Electronics	EPA1990A	16-pin 0.3" DIL	√			
PCA Electronics	EPA2013D	16-pin 0.3" DIL		√		
PCA Electronics	EPA2162	16-pin 0.3" SIP			√	
Pulse Engineering	PE-65421	16-pin 0.3" DIL	√			
Pulse Engineering	PE-65434	16-pin 0.3" SIL			√	
Pulse Engineering	PE-65445	16-pin 0.3" DIL			√	
Pulse Engineering	PE-65467	12-pin 0.5" SMT				√
Valor Electronics	PT3877	16-pin 0.3" DIL	√			
Valor Electronics	FL1043	16-pin 0.3" DIL			√	

PCnet-ISA+ Compatible AUI Isolation Transformers

The table below provides a sample list of PCnet-ISA+ compatible AUI isolation transformers available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3" DIL	50 µH
Bel Fuse	S553-0756-AE	16-pin 0.3" SMD	75 µH
Halo Electronics	TD01-0756K	16-pin 0.3" DIL	75 µH
Halo Electronics	TG01-0756W	16-pin 0.3" SMD	75 µH
PCA Electronics	EP9531-4	16-pin 0.3" DIL	50 µH
Pulse Engineering	PE64106	16-pin 0.3" DIL	50 µH
Pulse Engineering	PE65723	16-pin 0.3" SMT	75 µH
Valor Electronics	LT6032	16-pin 0.3" DIL	75 µH
Valor Electronics	ST7032	16-pin 0.3" SMD	75 µH

PCnet-ISA+ Compatible DC/DC Converters

The table below provides a sample list of PCnet-ISA+ compatible DC/DC converters available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Voltage	Remote On/Off
Halo Electronics	DCU0-0509D	24-pin DIP	5/-9	No
Halo Electronics	DCU0-0509E	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1007P	24-pin DIP	5/-9	No
PCA Electronics	EPC1054P	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1078	24-pin DIP	5/-9	Yes
Valor Electronics	PM7202	24-pin DIP	5/-9	No
Valor Electronics	PM7222	24-pin DIP	5/-9	Yes

MANUFACTURER CONTACT INFORMATION

Contact the following companies for further information on their products:

Company		U.S. and Domestic	Asia	Europe
Bel Fuse	Phone: FAX:	(201) 432-0463 (201) 432-9542	852-328-5515 852-352-3706	33-1-69410402 33-1-69413320
Halo Electronics	Phone: FAX:	(415) 969-7313 (415) 367-7158	65-285-1566 65-284-9466	
PCA Electronics (HPC in Hong Kong)	Phone: FAX:	818-892-0761 818-894-5791	852-553-0165 852-873-1550	33-1-44894800 33-1-42051579
Pulse Engineering	Phone: FAX:	(619) 674-8100 (619) 675-8262	852-425-1651 852-480-5974	353-093-24107 353-093-24459
Valor Electronics	Phone: FAX:	(619) 537-2500 (619) 537-2525	852-513-8210 852-513-8214	49-89-6923122 49-89-6926542

APPENDIX B

Layout Recommendations for Reducing Noise

DECOUPLING LOW-PASS R/C FILTER DESIGN

The PCnet-ISA+ controller is an integrated, single-chip Ethernet controller, which contains both digital and analog circuitry. The analog circuitry contains a high speed Phase-Locked Loop (PLL) and Voltage Controlled Oscillator (VCO). Because of the mixed signal characteristics of this chip, some extra precautions must be taken into account when designing with this device.

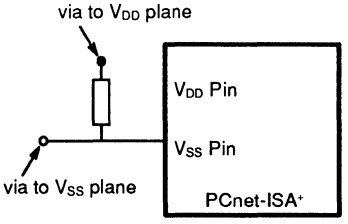
Described in this section is a simple decoupling low-pass R/C filter that can significantly increase noise immunity of the PLL circuit, thus, prevent noise from disrupting the VCO. Bit error rate, a common measurement of network performance, as a result can be drastically reduced. In certain cases the bit error rate can be reduced by orders of magnitude.

Implementation of this filter is not necessary to achieve a functional product that meets the IEEE 802.3 specification and provides adequate performance. However, this filter will help designers meet those specifications with more margin.

Digital Decoupling

The DVSS pins that are sinking the most current are those that provide the ground for the ISA bus output signals since these outputs require 24 mA drivers. The DVSS10 and DVSS12 pins provide the ground for the internal digital logic. In addition, DVSS11 provides ground for the internal digital and for the Input and I/O pins.

The CMOS technology used in fabricating the PCnet-ISA+ controller employs an n-type substrate. In this technology, all V_{DD} pins are electrically connected to each other internally. Hence, in a four-layer board, when decoupling between V_{DD} and critical V_{SS} pins, the specific V_{DD} pin that you connect to is not critical. In fact, the V_{DD} connection of the decoupling capacitor can be made directly to the power plane, near the closest V_{DD} pin to the V_{SS} pin of interest. However, we recommend that the V_{SS} connection of the decoupling capacitor be made directly to the V_{SS} pin of interest as shown.



AMD recommends that at least one low-frequency bulk decoupling capacitor be used in the area of the PCnet-ISA+ controller. 22 µF capacitors have worked well for this. In addition, a total of four or five 0.1 µF capacitors have proven sufficient around the DV_{SS} and DV_{DD} pins that supply the drivers of the ISA bus output pins.

Analog Decoupling

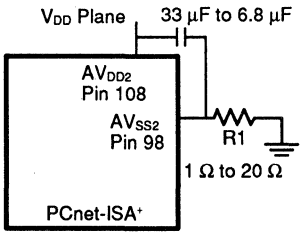
The most critical pins are the analog supply and ground pins. All of the analog supply and ground pins are located in one corner of the device. Specific requirements of the analog supply pins are listed below.

AVSS1 and AVDD3

These pins provide the power and ground for the Twisted Pair and AUI drivers. Hence, they are very noisy. A dedicated 0.1 µF capacitor between these pins is recommended.

AVSS2 and AVDD2

These pins are the most critical pins on the PCnet-ISA+ controller because they provide the power and ground for the PLL portion of the chip. The V_{CO} portion of the PLL is sensitive to noise in the 60 kHz–200 kHz. range. To prevent noise in this frequency range from disrupting the VCO, AMD strongly recommends that the low-pass filter shown below be implemented on these pins. Tests using this filter have shown significantly increased noise immunity and reduced Bit Error Rate (BER) statistics in designs using the PCnet-ISA+ controller.



To determine the value for the resistor and capacitor, the formula is:

$$R * C \geq 88$$

Where R is in ohms and C is in microfarads. Some possible combinations are given below. To minimize the voltage drop across the resistor, the R value should not be more than 20 Ω.

R	C
2.7 Ω	33 µF
4.3 Ω	22 µF
6.8 Ω	15 µF
10 Ω	10 µF
20 Ω	6.8 µF

AVSS2 and AVDD2/AVDD4

These pins provide power and ground for the AUI and twisted pair receive circuitry. No specific decoupling has been necessary on these pins.

Sample Configuration File

SAMPLE CONFIGURATION FILE

The following is a sample configuration file for the PCnet-ISA+ device used in an AMD Ethernet card. This card requires one DMA channel, one interrupt, one I/O port in the 0x200-0x3FF range (0x20 bytes aligned). The vendor ID of AMD is AMD. The vendor assigned part number for this card is 2100 and the serial number is 0x12345678. The card has only one logical device, that is an ethernet controller. There are no compatible devices with this logical device. The following record should be returned by the card during the identification process.

Note: *All data stored in the EEPROM is stored in bit-reversal format. Each word (16 bits) must be written into the EEPROM with bit 15 swapped with bit 0, bit 14 swapped with bit 1, etc.*

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Plug and Play Header
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x04                                  ; Vendor EISA ID Byte 0
DB 0x43                                  ; Vendor EISA ID Byte 1
DB 0x00                                  ; Vendor Assigned ID Byte 0
DB 0x21                                  ; Vendor Assigned ID Byte 1
DB 0x78                                  ; Serial Number byte 0
DB 0x56                                  ; Serial Number byte 1
DB 0x34                                  ; Serial Number byte 2
DB 0x12                                  ; Serial Number byte 3
DB Checksum                              ; Checksum calculated on above bits
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Plug and Play Version
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x0A                                  ; Small Item, Plug and Play version
DB 0x10                                  ; BCD major version [7:4] = 1
                                         ; BCD minor version [3:0] = 0
DB 0x00                                  ; Vendor specific version number
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Identifier String
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
DB 0x82                                  ; Large Item, Type Identifier string (ANSI)
DB 0x1c                                  ; Length Byte 0 (28 bytes)
DB 0x00                                  ; Length Byte 1
DB "AMD Ethernet Network Adapter"        ; Identifier String
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Logical Device ID
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

DB 0x15                         ; Small Item, Type Logical Device ID
DB 0x11                         ; Logical Device ID byte 0
DB 0x11                         ; Logical Device ID byte 1
DB 0x22                         ; Logical Device ID byte 2
DB 0x22                         ; Logical Device ID byte 3
DB 0x01                         ; Logical Device Flags [0] - required for boot

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; I/O Port Descriptor
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

DB 0x47                         ; Small Item, type I/O Port
DB 0x00                         ; Information, [0] = 0, 10 bit Decode
DB 0x00                         ; Minimum Base Address [07:00]
DB 0x02                         ; Minimum Base Address [15:08]
DB 0xE0                         ; Maximum Base Address [07:00]
DB 0x03                         ; Maximum Base Address [15:08]
DB 0x20                         ; Base Address Increment (32 ports)
DB 0x18                         ; Number of ports required

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; DMA Descriptor
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

DB 0x2A                         ; Small Item, type DMA Format
DB 0xE8                         ; DMA channel mask ch 3, 5, 6, 7
DB 0x06                         ; 16-Bit only, Bus Master

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;IRQ Format
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

DB 0x23                         ; Small Item, type IRQ Format
DB 0x38                         ; IRQs supported [7:0]  } 3, 4, 5
DB 0x9E                         ; IRQs supported [15:8] } 9, 10, 11, 12, 15
DB 0x01                         ; Information: High true, edge

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; End Tag
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

DB 0x78                         ; Small item, type END TAG
DB Checksum                     ; Checksum
```

APPENDIX D

Alternative Method for Initialization

The PCnet-ISA+ controller may be initialized by performing I/O writes only. That is, data can be written directly to the appropriate control and status registers (CSR) instead of reading from the Initialization Block in memory. The registers that must be written are shown in the table below. These are followed by writing the START bit in CSR0.

Control and Status Register	Comment
CSR8	LADRF[15:0]
CSR9	LADRF[31:16]
CSR10	LADRF[47:32]
CSR11	LADRF[63:48]
CSR12	PADR[15:0]
CSR13	PADR[31:16]
CSR14	PADR[47:32]
CSR15	Mode
CSR24-25	BADR
CSR30-31	BADX
CSR47	POLLINT
CSR76	RCVRL
CSR78	XMTRL

Note: *The INIT bit must not be set or the initialization block will be accessed instead.*

APPENDIX E

Introduction of the Look Ahead Packet Processing (LAPP) Concept

A driver for the PCnet-ISA+ controller would normally require that the CPU copy receive frame data from the controller's buffer space to the application's buffer space *after* the entire frame has been received by the controller. For applications that use a ping-pong windowing style, the traffic on the network will be halted until the current frame has been completely processed by the entire application stack. This means that the time between last byte of a receive frame arriving at the client's Ethernet controller and the client's transmission of the first byte of the next outgoing frame will be separated by:

1) the time that it takes the client's CPU's interrupt procedure to pass software control from the current task to the driver
2) plus the time that it takes the client driver to pass the header data to the application and request an application buffer
3) plus the time that it takes the application to generate the buffer pointer and then return the buffer pointer to the driver
4) plus the time that it takes the client driver to transfer all of the frame data from the controller's buffer space into the application's buffer space and then call the application again to process the complete frame
5) plus the time that it takes the application to process the frame and generate the next outgoing frame
6) plus the time that it takes the client driver to set up the descriptor for the controller and then write a TDMD bit to CSR0

The sum of these times can often be about the same as the time taken to actually transmit the frames on the wire, thereby yielding a network utilization rate of less than 50%.

An important thing to note is that the PCnet-ISA+ controller's data transfers to its buffer space are such that the system bus is needed by the PCnet-ISA+ controller for approximately 4% of the time. This leaves 96% of the sytem bus bandwidth for the CPU to perform some of the inter–frame operations *in advance of the completion of network receive activity*, if possible. The question then becomes: how much of the tasks that need to be performed between reception of a frame and transmission of the next frame can be performed *before* the reception of the frame actually ends at the network, and how can the CPU be instructed to perform these tasks during the network reception time?

The answer depends upon exactly what is happening in the driver and application code, but the steps that can be performed at the same time as the receive data are arriving include as much as the first three steps and part of the fourth step shown in the sequence above. By performing these steps before the entire frame has arrived, the frame throughput can be substantially increased.

A good increase in performance can be expected when the first three steps are performed before the end of the network receive operation. A much more significant performance increase could be realized if the PCnet-ISA+ controller could place the frame data directly into the application's buffer space; (i.e. eliminate the need for step four.) In order to make this work, it is necessary that the application buffer pointer be determined before the frame has completely arrived, then the buffer pointer in the next desriptor for the receive frame would need to be modified in order to direct the PCnet-ISA+ controller to write directly to the application buffer. More details on this operation will be given later.

An alternative modification to the existing system can gain a smaller, but still significant improvement in performance. This alternative leaves step four unchanged in that the CPU is still required to perform the copy operation, but it allows a large portion of the copy operation to be done before the frame has been completely received by the controller, (i.e. the CPU can perform the copy operation of the receive data from the PCnet-ISA+ controller's buffer space into the application buffer space *before* the frame data has completely arrived from the network.) This allows the copy operation of step four to be performed concurrently with the arrival of network data, rather than sequentially, following the end of network receive activity.

Outline of the LAPP Flow:

This section gives a suggested outline for a driver that utilizes the LAPP feature of the PCnet-ISA+ controller.

Note: *The labels in the following text are used as references in the timeline diagram that follows.*

SETUP:

The driver should set up descriptors in groups of 3, with the OWN and STP bits of each set of three descriptors to read as follows: 11b, 10b, 00b.

An option bit (LAPPEN) exists in CSR3, bit position 5. The software should set this bit. When set, the LAPPEN bit directs the PCnet-ISA+ to generate an INTERRUPT when STP has been written to a receive descriptor by the PCnet-ISA+ controller.

FLOW:

The PCnet-ISA+ controller polls the current receive descriptor at some point in time before a message arrives. The PCnet-ISA+ controller determines that this receive buffer is OWNed by the PCnet-ISA+ controller and it stores the descriptor information to be used when a message does arrive.

N0: Frame preamble appears on the wire, followed by SFD and destination address.

N1: The 64th byte of frame data arrives from the wire. This causes the PCnet-ISA+ controller to begin frame data DMA operations to the first buffer.

C0: When the 64th byte of the message arrives, the PCnet-ISA+ controller performs a lookahead operation to the next receive descriptor. This descriptor should be owned by the PCnet-ISA+ controller.

C1: The PCnet-ISA+ controller intermittently requests the bus to transfer frame data to the first buffer as it arrives on the wire.

S0: The driver remains idle.

C2: When the PCnet-ISA+ controller has completely filled the first buffer, it writes status to the first descriptor.

C3: When the first descriptor for the frame has been written, changing ownership from the PCnet-ISA+ controller to the CPU, the PCnet-ISA+ controller will generate an SRP INTERRUPT. (This interrupt appears as a RINT interrupt in CSR0.)

S1: The SRP INTERRUPT causes the CPU to switch tasks to allow the PCnet-ISA+ controller's driver to run.

C4: During the CPU interrupt-generated task switching, the PCnet-ISA+ controller is performing a lookahead operation to the third descriptor. At this point in time, the third descriptor is owned by the CPU. [***Note:*** *Even though the third buffer is not owned by the PCnet-ISA+ controller, existing AMD Ethernet controllers will continue to perform data DMA into the buffer space that the controller already owns (i.e. buffer number 2). The controller does not know if buffer space in buffer number 2 will be sufficient or not, for this frame, but it has no way to tell except by trying to move the entire message into that space. Only when the message does not fit will it signal a buffer error condition—there is no need to panic at the point that it discovers that it does not yet own descriptor number 3.*]

S2: The first task of the driver's interrupt service routine is to collect the header information from the PCnet-ISA+ controller's first buffer and pass it to the application.

S3: The application will return an application buffer pointer to the driver. The driver will add an offset to the application data buffer pointer, since the PCnet-ISA+ controller will be placing the first portion of the message into the first and second buffers. (The modified application data buffer pointer will only be directly used by the PCnet-ISA+ controller when it reaches the third buffer.) The driver will place the modified data buffer pointer into the final descriptor of the group (#3) and will grant ownership of this descriptor to the PCnet-ISA+ controller.

C5: Interleaved with S2, S3 and S4 driver activity, the PCnet-ISA+ controller will write frame data to buffer number 2.

S4: The driver will next proceed to copy the contents of the PCnet-ISA+ controller's first buffer to the *beginning* of the application space. This copy will be to the exact (unmodified) buffer pointer that was passed by the application.

S5: After copying all of the data from the first buffer into the beginning of the application data buffer, the driver will begin to poll the ownership bit of the second descriptor. The driver is waiting for the PCnet-ISA+ controller to finish filling the second buffer.

C6: At this point, knowing that it had not previously owned the third descriptor, and knowing that the current message has not ended (there is more data in the fifo), the PCnet-ISA+ controller will make a "last ditch lookahead" to the final (third) descriptor; This time, the ownership will be TRUE (i.e. the descriptor belongs to the controller), because the driver wrote the application pointer into this descriptor and then changed the ownership to give the descriptor to the PCnet-ISA+ controller back at S3. Note that if steps S1, S2 and S3 have not completed at this time, a BUFF error will result.

C7: After filling the second buffer and performing the last chance lookahead to the next descriptor, the PCnet-ISA+ controller will write the status and change the ownership bit of descriptor number 2.

S6: After the ownership of descriptor number 2 has been changed by the PCnet-ISA+ controller, the *next* **driver** poll of the 2nd descriptor will show ownership granted to the CPU. The driver now copies the data from buffer number 2 into the "middle section" of the application buffer space. This

operation is interleaved with the C7 and C8 operations.

C8: The PCnet-ISA+ controller will perform data DMA to the last buffer, whose pointer is pointing to application space. Data entering the last buffer will not need the infamous "double copy" that is required by existing drivers, since it is being placed directly into the application buffer space.

N2: The message on the wire ends.

S7: When the driver completes the copy of buffer number 2 data to the application buffer space, it begins polling descriptor number 3.

C9: When the PCnet-ISA+ controller has finished all data DMA operations, it writes status and changes ownership of descriptor number 3.

S8: The driver sees that the ownership of descriptor number 3 has changed, and it calls the application to tell the application that a frame has arrived.

S9: The application processes the received frame and generates the next TX frame, placing it into a TX buffer.

S10: The driver sets up the TX descriptor for the PCnet-ISA+ controller.

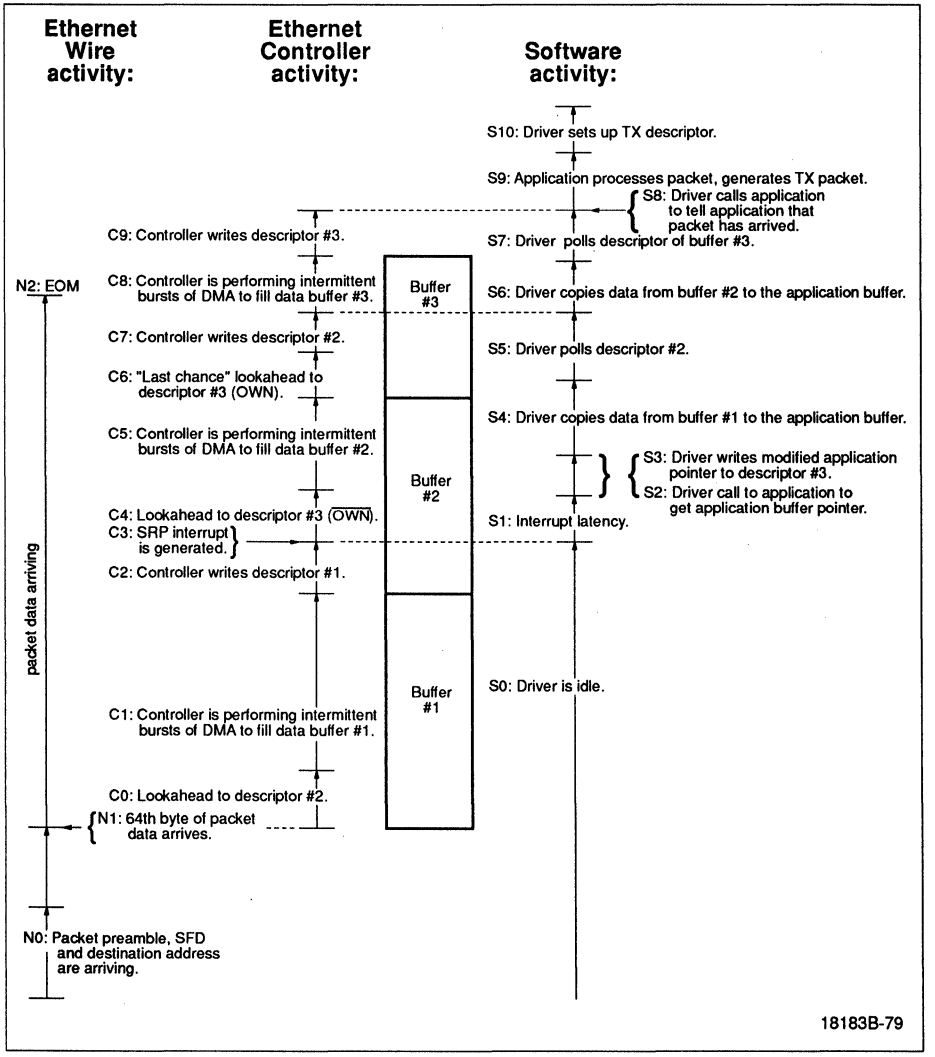


Figure 1. Look Ahead Packet Processing (LAPP) Timeline

LAPP Enable Software Requirements

Software needs to set up a receive ring with descriptors formed into groups of 3. The first descriptor *of each group* should have OWN = 1 and STP = 1, the second descriptor *of each group* should have OWN = 1 and STP = 0. The third descriptor *of each group* should have OWN = 0 and STP = 0. The size of the first buffer (as indicated in the first descriptor), should be ***at least*** equal to the largest expected header size; However, for maximum efficiency of CPU utilization, the first buffer size should be larger than the header size. It should be equal to the expected number of message bytes, minus the time needed for Interrupt latency and minus the application call latency, minus the time needed for the driver to write to the third descriptor, minus the time needed for the driver to copy data from buffer #1 to the application buffer space, and minus the time needed for the driver to copy data from buffer #2 to the application buffer space. Note that the time needed for the copies performed by the driver depends upon the sizes of the 2nd and 3rd buffers, and that the sizes of the second and third buffers need to be set accoring to the time needed for the data copy operations! This means that an iterative self–adjusting mechanism needs to be placed into the software to determine the correct buffer sizing for optimal operation. Fixed values for buffer sizes may be used; In such a case, the LAPP method will still provide a significant performance increase, but the performance increase will not be maximized.

The following diagram illustrates this setup for a receive ring size of 9:

LAPP Enable Rules for Parsing of Descriptors

When using the LAPP method, software must use a modified form of descriptor *parsing* as follows:

Software will examine OWN and STP to determine where a RCV frame begins. RCV frames will only begin in buffers that have OWN = 0 and STP = 1.

Software shall assume that a frame continues until it finds *either* ENP = 1 or ERR= 1.

Software must discard all descriptors with OWN = 0 and STP = 0 and move to the next descriptor when searching for the beginning of a new frame; ENP and ERR should be ignored by software during this search.

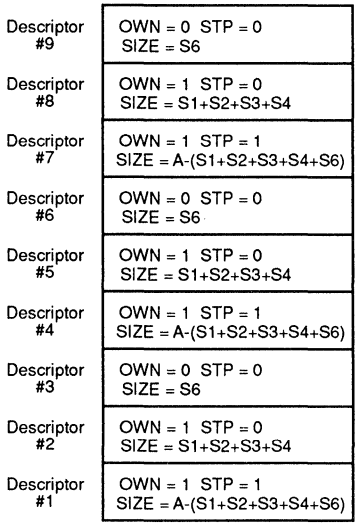
Software cannot change an STP value in the receive descriptor ring after the initial setup of the ring is complete, even if software has ownership of the STP descriptor *unless* the previous STP descriptor in the ring is *also* OWNED by the software.

When LAPPEN = 1, then hardware will use a modified form of descriptor *parsing* as follows:

The controller will examine OWN and STP to determine where to begin placing a RCV frame. A new RCV frame will only begin in a buffer that has OWN = 1 and STP = 1.

The controller will always obey the OWN bit for determining whether or not it may use the next buffer for a chain.

The controller will always mark the end of a frame with *either* ENP = 1 or ERR= 1.



A = Expected message size in bytes
S1 = Interrupt latency
S2 = Application call latency
S3 = Time needed for driver to write to third descriptor
S4 = Time needed for driver to copy data from buffer #1 to application buffer space
S6 = Time needed for driver to copy data from buffer #2 to application buffer space

Note that the times needed for tasks S1, S2, S3, S4, and S6 should be divided by 0.8 ms to yield an equivalent number of network byte times before subtracting these quantities from the expected message size A.

18183B-80

Figure 2. LAPP 3 Buffer Grouping

The controller will *discard* all descriptors with **OWN = 1** and STP = 0 and move to the next descriptor ***when searching for a place to begin a new frame***. It discards these desciptors by simply changing the ownership bit from OWN=1 to OWN = 0. Such a descriptor is unused for receive purposes by the controller, and the driver must recognize this. (The driver will recognize this if it follows the software rules.)

The controller will *ignore* all descriptors with **OWN = 0** and STP = 0 and move to the next descriptor ***when searching for a place to begin a new frame***. In other words, the controller is allowed to skip entries in the ring that it does not own, but only when it is looking for a place to begin a new frame.

Some Examples of LAPP Descriptor Interaction

Choose an expected frame size of 1060 bytes.

Choose buffer sizes of 800, 200 and 200 bytes.

1) Assume that a 1060 byte frame arrives correctly, and that the timing of the early interrupt and the software is smooth. The descriptors will have changed from:

Descriptor Number	Before the Frame Arrived			After the Frame Has Arrived			Comments (After Frame Arrival)
	OWN	STP	ENP*	OWN	STP	ENP*	
1	1	1	X	0	1	0	Bytes 1–800
2	1	0	X	0	0	0	Bytes 801–1000
3	0	0	X	0	0	1	Bytes 1001–1060
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Not yet used

**ENP or ERR*

2) Assume that instead of the expected 1060 byte frame, a 900 byte frame arrives, either because there was an error in the network, or because this is the last frame in a file transmission sequence.

Descriptor Number	Before the Frame Arrived			After the Frame Has Arrived			Comments (After Frame Arrival)
	OWN	STP	ENP*	OWN	STP	ENP*	
1	1	1	X	0	1	0	Bytes 1–800
2	1	0	X	0	0	1	Bytes 801–900
3	0	0	X	0	0	?**	Discarded buffer
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Not yet used

**ENP or ERR*

*** Note that the PCnet-ISA+ controller might write a ZERO to ENP location in the 3rd descriptor. Here are the two possibilities:*

1) If the controller finishes the data transfers into buffer number 2 after the driver writes the application's modified buffer pointer into the third descriptor, then the controller will write a ZERO to ENP for this buffer and will write a ZERO to OWN and STP.

2) If the controller finishes the data transfers into buffer number 2 before the driver writes the application's modified buffer pointer into the third descriptor, then the controller will complete the frame in buffer number two and then skip the then un-owned third buffer. In this case, the PCnet-ISA+ controller will not have had the opportunity to RESET the ENP bit in this descriptor, and it is possible that the software left this bit as ENP=1 from the last time through the ring. Therefore, the software ***must*** *treat the location as a don't care; The rule is, after finding ENP=1 (or ERR=1) in descriptor number 2, the software* must *ignore ENP bits until it finds the next STP=1.*

3) Assume that instead of the expected 1060 byte frame, a 100 byte frame arrives, because there was an error in the network, or because this is the last frame in a file transmission sequence, or perhaps because it is an acknowledge frame.

Descriptor Number	Before the Frame Arrived			After the Frame Has Arrived			Comments (After Frame Arrival)
	OWN	STP	ENP*	OWN	STP	ENP*	
1	1	1	X	0	1	1	Bytes 1–100
2	1	0	X	0	0	0***	Discarded buffer
3	0	0	X	0	0	?**	Discarded buffer
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Not yet used

**ENP or ERR*

*** Same as note in case 2 above, except that in this case, it is very unlikely that the driver can respond to the interrupt and get the pointer from the application before the PCnet-ISA+ controller has completed its poll of the next descriptors. This means that for almost all occurrences of this case, the PCnet-ISA+ controller will not find the OWN bit set for this descriptor and therefore, the ENP bit will almost always contain the old value, since the PCnet-ISA+ controller will not have had an opportunity to modify it.*

**** Note that even though the PCnet-ISA+ controller will write a ZERO to this ENP location, the software should treat the location as a don't care, since after finding the ENP=1 in descriptor number 2, the software should ignore ENP bits until it finds the next STP=1.*

Buffer Size Tuning

For maximum performance, buffer sizes should be adjusted depending upon the expected frame size and the values of the interrupt latency and application call latency. The best driver code will minimize the CPU utilization while also minimizing the latency from frame end on the network to frame sent to application from driver (frame latency). These objectives are aimed at increasing throughput on the network while decreasing CPU utilization.

Note that the buffer sizes in the ring may be altered at any time that the CPU has ownership of the corresponding descriptor. The best choice for buffer sizes will maximize the time that the driver is swapped out, while minimizing the time from the last byte written by the PCnet-ISA+ controller to the time that the data is passed from the driver to the application. In the diagram, this corresponds to maximizing S0, while minimizing the time between C9 and S8. (The timeline happens to show a minimal time from C9 to S8.)

Note that by increasing the size of buffer number 1, we increase the value of S0. However, when we increase the size of buffer number 1, we also increase the value of S4. If the size of buffer number 1 is too large, then the driver will not have enough time to perform tasks S2, S3, S4, S5 and S6. The result is that there will be delay from the execution of task C9 until the execution of task S8. A perfectly timed system will have the values for S5 and S7 at a minimum.

An average increase in performance can be achieved if the general guidelines of buffer sizes in Figure 2 is followed. However, as was noted earlier, the correct sizing for buffers will depend upon the expected message size. There are two problems with relating expected message size with the correct buffer sizing:

1) Message sizes cannot always be accurately predicted, since a single application may expect different message sizes at different times, therefore, the buffer sizes chosen will not always maximize throughput.

2) Within a single application, message sizes might be somewhat predicatable, but when the same driver is to be shared with multiple applications, there may not be a common predictable message size.

Additional problems occur when trying to define the correct sizing because the correct size also depends upon the interrupt latency, which may vary from system to system, depending upon both the hardware and the software installed in each system.

In order to deal with the unpredictable nature of the message size, the driver can implement a self tuning

mechanism that examines the amount of time spent in tasks S5 and S7 as such: While the driver is polling for each descriptor, it could count the number of poll operations performed and then adjust the number 1 buffer size to a larger value, by adding "t" bytes to the buffer count, if the number of poll operations was greater than "x". If fewer than "x" poll operations were needed for each of S5 and S7, then the software should adjust the buffer size to a smaller value by, subtracting "y" bytes from the buffer count. Experiments with such a tuning mechanism must be performed to determine the best values for "X" and "y."

Note whenever the size of buffer number 1 is adjusted, buffer sizes for buffer number 2 and buffer 3 should also be adjusted.

In some systems the typical mix of receive frames on a network for a client application consists mostly of large data frames, with very few small frames. In this case, for maximum efficiency of buffer sizing, *when a frame arrives under a certain size limit, the driver should* ***not*** *adjust the buffer sizes in response to the short frame.*

An Alternative LAPP Flow – the TWO Interrupt Method

An alternative to the above suggested flow is to use two interrupts, one at the start of the Receive frame and the other at the end of the receive frame, instead of just looking for the SRP interupt as was described above. This alternative attempts to reduce the amount of time that the software "wastes" while polling for descriptor own bits. This time would then be available for other CPU tasks. It also minimizes the amount of time the CPU needs for data copying. This savings can be applied to other CPU tasks.

The time from the end of frame arrival on the wire to delivery of the frame to the application is labeled as frame latency. For the one-interrupt method, frame latency is minimized, while CPU utilization increases. For the two-interrupt method, frame latency becomes greater, while CPU utilization decreases.

Note that some of the CPU time that can be applied to non-Ethernet tasks is used for task switching in the CPU. One task switch is required to swap a non-Ethernet task into the CPU (after S7A) and a second task switch is needed to swap the Ethernet driver back in again (at S8A). If the time needed to perform these task switches exceeds the time saved by not polling descriptors, then there is a net loss in performance with this method. Therefore, the NEW WORD method implemented should be carefully chosen.

Figure 3 shows the event flow for the two-interrupt method.

Figure 4 shows the buffer sizing for the two-interrupt method. Note that the second buffer size will be about the same for each method.

There is another alternative which is a marriage of the two previous methods. This third possibility would use the buffer sizes set by the two-interrupt method, but would use the polling method of determining frame end. This will give good frame latency but at the price of very high CPU utilization.

And still, there are even more compromise positions that use various fixed buffer sizes and effectively, the flow of the one-interrupt method. All of these compromises will reduce the complexity of the one-interrupt method by removing the heuristic buffer sizing code, but they all become less efficient than heuristic code would allow.

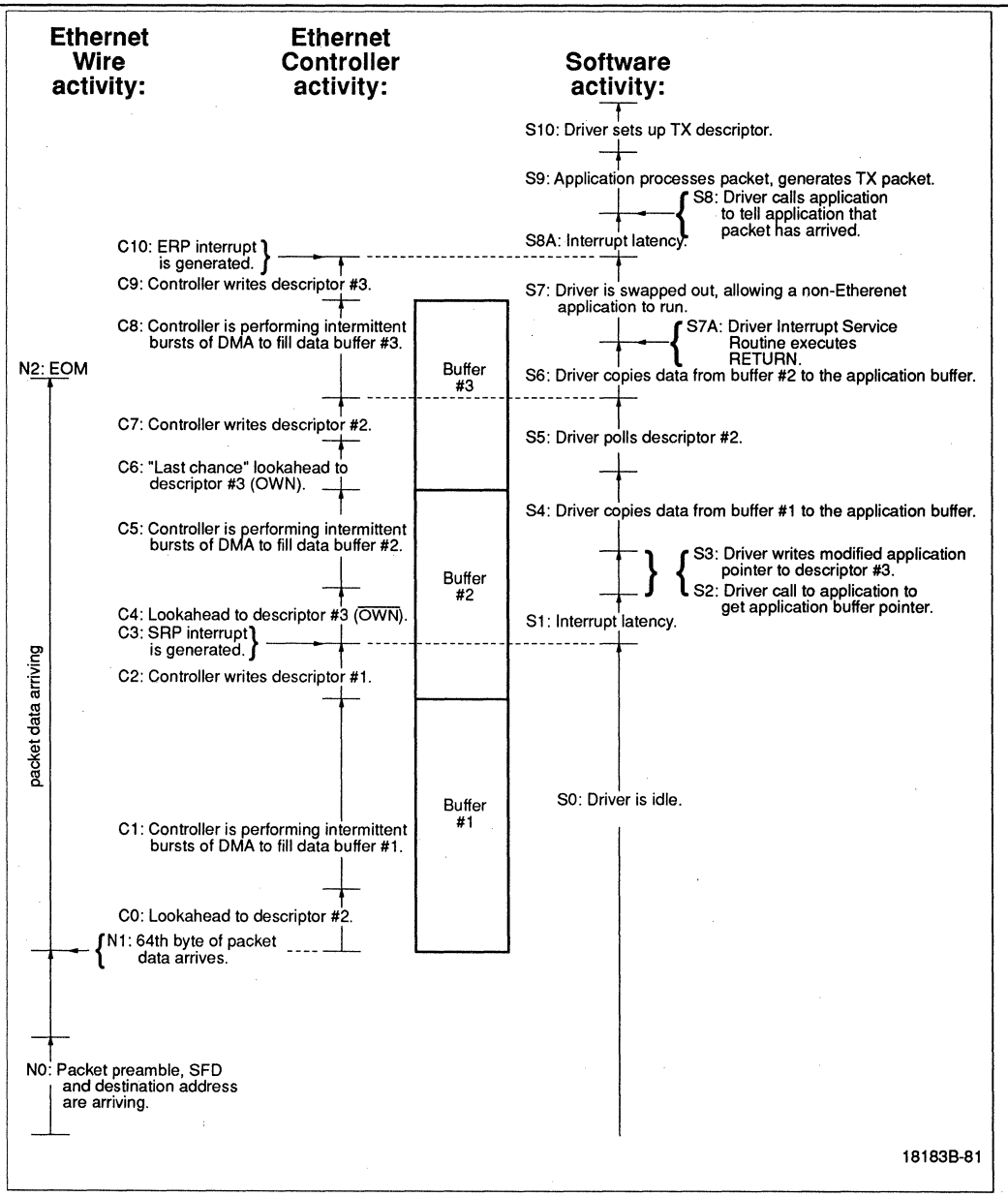
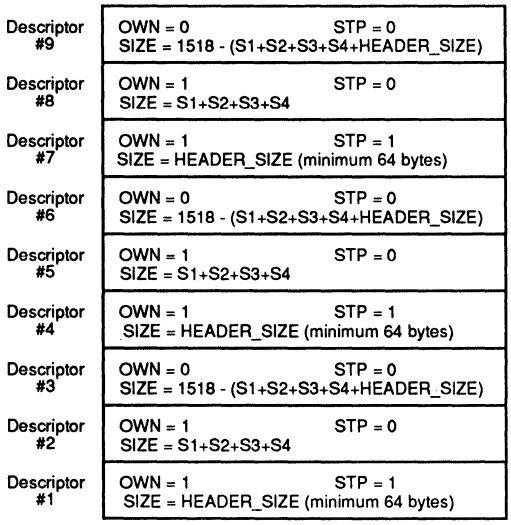


Figure 3. LAPP Timeline for TWO-INTERRUPT Method



A = Expected message size in bytes
S1 = Interrupt latency
S2 = Application call latency
S3 = Time needed for driver to write to third descriptor
S4 = Time needed for driver to copy data from buffer #1 to application buffer space
S6 = Time needed for driver to copy data from buffer #2 to application buffer space

Note that the times needed for tasks S1, S2, S3, S4, and S6 should be divided by 0.8 ms to yield an equivalent number of network byte times before subtracting these quantities from the expected message size A.

18183B-82

Figure 4. LAPP 3 Buffer Grouping for TWO-INTERRUPT Method

Some Characteristics of the XXC56 Serial EEPROMs

SWITCHING CHARACTERISTICS of a TYPICAL XXC56 SERIAL EEPROM INTERFACE

Applicable over recommended operating range from T_A = –40°C to +85°C, V_{CC} = +1.8 V to +5.5 V, C_L = 1 TTL Gate and 100 pF (unless otherwise noted)

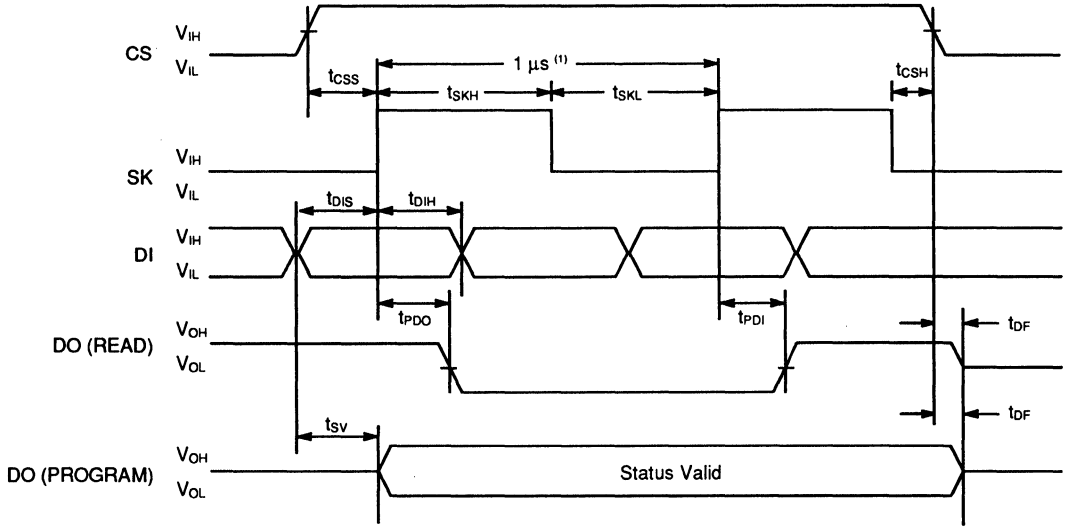
Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
f_{SK}	SK Clock Frequency		0	0.5	MHz
t_{SKH}	SK High Time	(Note 1)	500		ns
t_{SKL}	SK Low Time	(Note 1)	500		ns
t_{CS}	Minimum CS Low Time	(Note 2)	500		ns
t_{CSS}	CS Setup Time	Relative to SK	100		ns
t_{DIS}	DI Setup Time	Relative to SK	200		ns
t_{CSH}	CS Hold Time	Relative to SK	0		ns
t_{DIH}	DI Hold Time	Relative to SK	200		ns
t_{PD1}	Output Delay to '1'	AC Test		1000	ns
t_{PD0}	Output Delay to '0'	AC Test		1000	ns
t_{SV}	CS to Status Valid	AC Test		1000	ns
t_{DF}	CS to DO in High Impedance	AC Test; CS = V_{IL}		200	ns
t_{WP}	Write Cycle Time			10	ms
	Endurance	Number of Data Changes per Bit	Typical 100,000		Cycles

Notes:

1. *The SK frequency specifies a minimum SK clock period of 2 µs, therefore in an SK clock cycle t_{SKH} + t_{SKL} must be greater than or equal to 2 µs. For example, if the t_{SKL} = 500 ns then the minimum t_{SKH} = 1.5 µs in order to meet the SK frequency specification.*
2. *CS must be brought low for a minimum of 500 ns (t_{CS}) between consecutive instruction cycles.*

INSTRUCTION SET FOR THE XXC56 SERIES OF EEPROMs

Instruction	SB	Op Code	Address x8	Address x16	Data x8	Data x16	Comments
READ	1	10	A8–A0	A7–A0			Reads data stored in memory, at specified address
EWEN	1	00	11XXXXXXX	11XXXXXX			Write enable must precede all programming modes
ERASE	1	11	A8–A0	A7–A0			Erases memory location An–A0
WRITE	1	01	A0–A0	A7–A0	D7–D0	D15–D0	Writes memory location An–A0
ERAL	1	00	10XXXXXXX	10XXXXXX			Erases all memory locations. Valid only at V_{CC} = 4.5 V to 5.5 V
WRAL	1	00	01XXXXXXX	01XXXXXX	D7–D0	D15–D0	Writes all memory locations. Valid when V_{CC} = 5.0 V ± 10% and Disable Register cleared
EWDS	1	00	00XXXXXXX	00XXXXXX			Disables all programming instructions



Note:

1. This is the minimum SK period.

18183B-57

Typical XXC56 Series
Serial EEPROM Control Timing

PRELIMINARY

Am79C965

PCnet™-32 Single-Chip 32-Bit Ethernet Controller

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Single-chip Ethernet controller for 486 and Video Electronics Standards Association (VESA) local buses**
- **Supports ISO 8802-3 (IEEE/ANSI 802.3) and Ethernet standards**
- **Direct interface to the 486 local bus or VESA VL-Bus**
- **Enhanced burst mode with support for Am486™ burst read/write operations**
- **Software compatible with AMD's Am7990 LANCE, Am79C90 C-LANCE, Am79C960 PCnet-ISA, Am79C961 PCnet-ISA+, Am79C970 PCnet-PCI, and Am79C900 ILACC™ register and descriptor architecture**
- **Compatible with Am2100/Am1500T and Novell® NE2100/NE1500 driver software**
- **High performance Bus Master architecture with integrated DMA Buffer Management Unit for low CPU and bus utilization**
- **Built-in byte-swap logic supports both Big and Little Endian byte alignment**
- **Microwire™ EEPROM interface supports jumperless design**
- **Single +5 V power supply operation**
- **Low power, CMOS design with sleep modes allows reduced power consumption for critical battery powered applications and Green PCs**
- **Look-Ahead Packet Processing (LAPP) allows protocol analysis to begin before end of receive frame**
- **Integrated Manchester Encoder/Decoder**
- **Individual 136-byte transmit and 128-byte receive FIFOs provide frame buffering for increased system latency tolerance and support the following features:**
  - — Automatic retransmission with no FIFO reload
  - — Automatic receive stripping and transmit padding (individually programmable)
  - — Automatic runt packet rejection
  - — Automatic deletion of received collision frames
- **JTAG Boundary Scan (IEEE 1149.1) test access port interface for board level production test**
- **Provides integrated Attachment Unit Interface (AUI) and 10BASE-T transceiver with automatic port selection**
- **Automatic Twisted Pair receive polarity detection and automatic correction of the receive polarity**
- **Optional byte padding to long-word boundary on receive**
- **Dynamic transmit FCS generation programmable on a frame-by-frame basis**
- **Internal/external loopback capabilities**
- **Supports the following types of network interfaces:**
  - — AUI to external 10BASE2, 10BASE5, 10BASE-T or 10BASE-F MAU
  - — Internal 10BASE-T transceiver with Smart Squelch to Twisted Pair medium
- **Supports LANCE/C-LANCE/PCnet-ISA General Purpose Serial Interface (GPSI)**
- **160-pin PQFP package**

GENERAL DESCRIPTION

The PCnet-32 single-chip 32-bit Ethernet controller is a highly integrated Ethernet system solution designed to address high-performance system application requirements. It is a flexible bus-mastering device that can be used in any networking application, including network-ready PCs, printers, fax modems, and bridge/router designs. The bus-master architecture provides high data throughput in the system and low CPU and bus utilization. The PCnet-32 controller is fabricated with AMD's advanced low-power CMOS process to provide low operating and standby current for power sensitive applications.

The PCnet-32 controller is a complete Ethernet node integrated into a single VLSI device. It contains a bus interface unit, a DMA buffer management unit, an ISO 8802-3 (IEEE/ANSI 802.3) defined Media Access Control (MAC) function, individual 136-byte transmit and 128-byte receive FIFOs, an ISO 8802-3 (IEEE/ANSI 802.3) defined Attachment Unit Interface (AUI) and

Publication# 18219 Rev. B Amendment/0
Issue Date: May 1994

Twisted-Pair Transceiver Media Attachment Unit (10BASE-T MAU), and a microwire EEPROM interface. The PCnet-32 controller is also register compatible with the LANCE (Am7990) Ethernet controller, the C-LANCE (Am79C90) Ethernet controller, the ILACC (Am79C900) Ethernet controller, and all Ethernet controllers in the PCnet Family, including the PCnet-ISA controller (Am79C960), the PCnet-ISA+ controller (Am79C961), and the PCnet-PCI controller (Am79C970). The buffer management unit supports the LANCE, ILACC, and PCnet descriptor software models. The PCnet-32 controller is software compatible with Novell NE2100 and NE1500 Ethernet adapter card architectures. In addition, a Sleep function has been incorporated to provide low standby current which is essential for notebooks and Green PCs.

The 32-bit demultiplexed bus interface unit provides a direct interface to the VESA VL-Bus and 486 series microprocessors, simplifying the design of an Ethernet node in a PC system. With its built-in support for both little and big endian byte alignment, this controller also addresses proprietary non-PC applications.

Key PCnet-32 configuration parameters, including the unique IEEE physical address, can be read from an external non-volatile memory (serial EEPROM) immediately following system reset. In addition, the I/O location at which the internal registers are accessed may be stored in the EEPROM, allowing the software model of the device to be located appropriately in system I/O space during system initialization.

The controller has the capability to automatically select either the AUI port or the Twisted-Pair transceiver. Only one interface is active at any one time. The individual transmit and receive FIFOs reduce system overhead, providing sufficient latency during frame transmission and reception, and minimizing intervention during normal network error recovery. The integrated Manchester encoder/decoder (MENDEC) eliminates the need for an external Serial Interface Adapter (SIA) in the node system. The built-in General Purpose Serial Interface (GPSI) allows the MENDEC to be by-passed. In addition, the device provides programmable on-chip LED drivers for transmit, receive, collision, receive polarity, link integrity, or jabber status. The PCnet-32 controller also provides an External Address Detection Interface (EADI) to allow external hardware address filtering in internetworking applications.

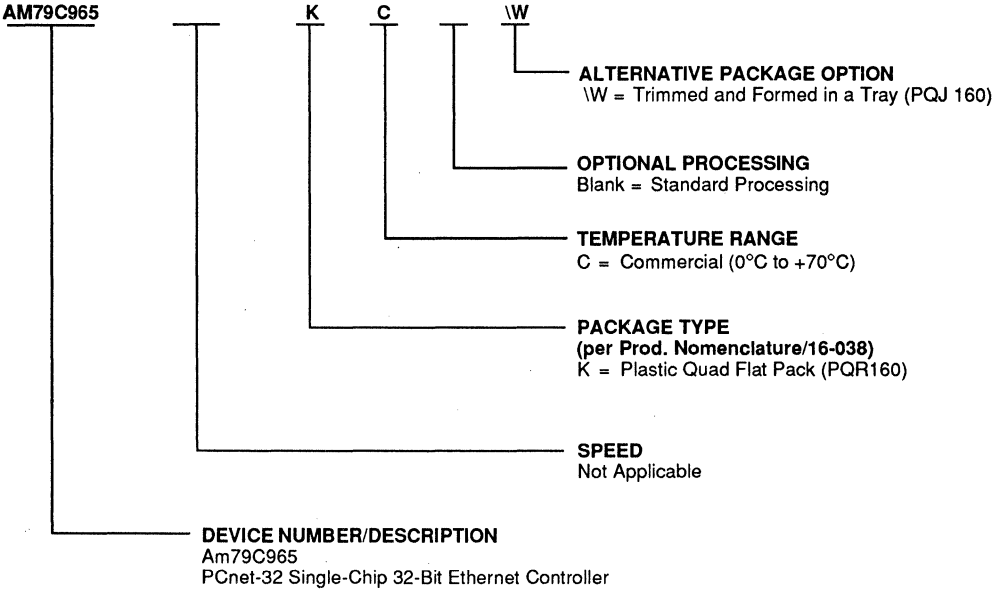
RELATED PRODUCTS

Part No	Description
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted Pair Ethernet Transceiver Plus (TPEX+)
Am7996	IEEE 802.3/Ethernet/Cheapernet Tap Transceiver
Am79C981	Integrated Multiport Repeater Plus (IMR+)
Am79C987	Hardware Implemented Management Information Base (HIMIB)
Am79C940	Media Access Controller for Ethernet (MACE)
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C900	Integrated Local Area Communications Controller (ILACC)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA+ Single-Chip Ethernet Controller (with Microsoft *Plug n' Play* support)
Am79C970	PCnet-PCI Single-Chip Ethernet Controller (for PCI bus)
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



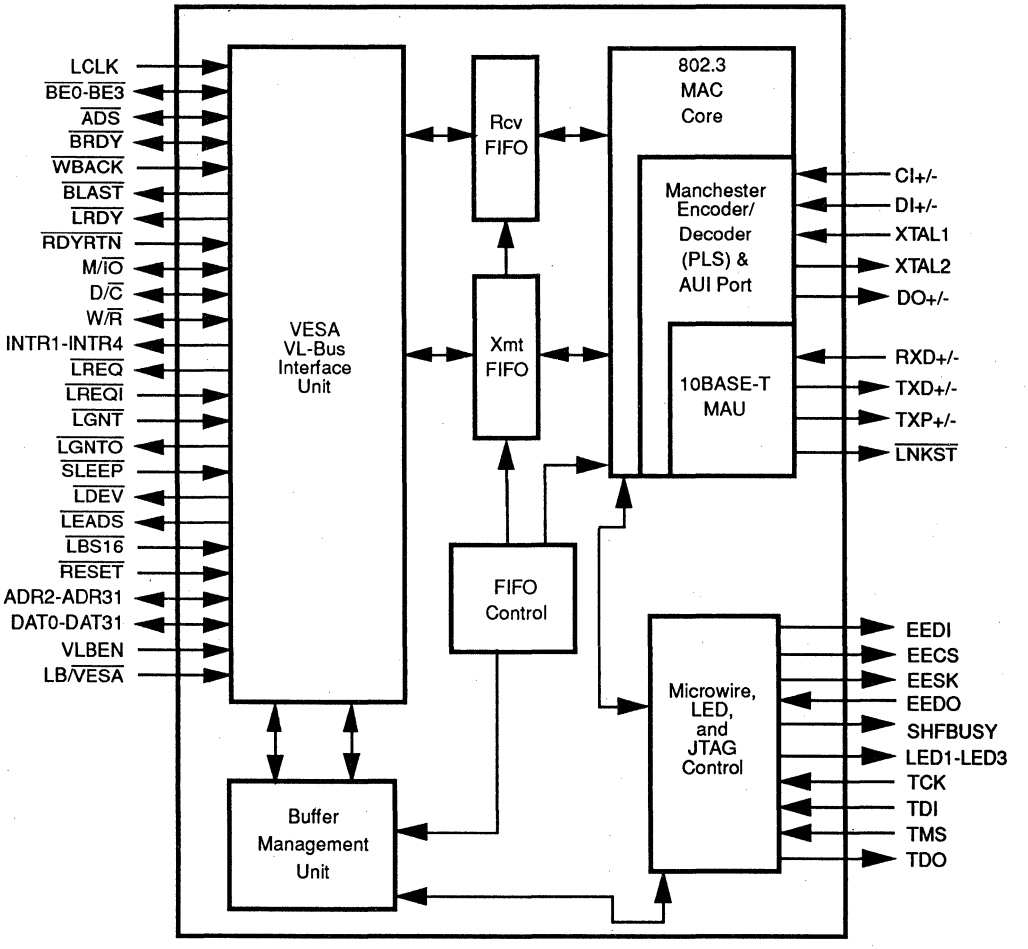
Valid Combinations	
AM79C965	KC, KC\W

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

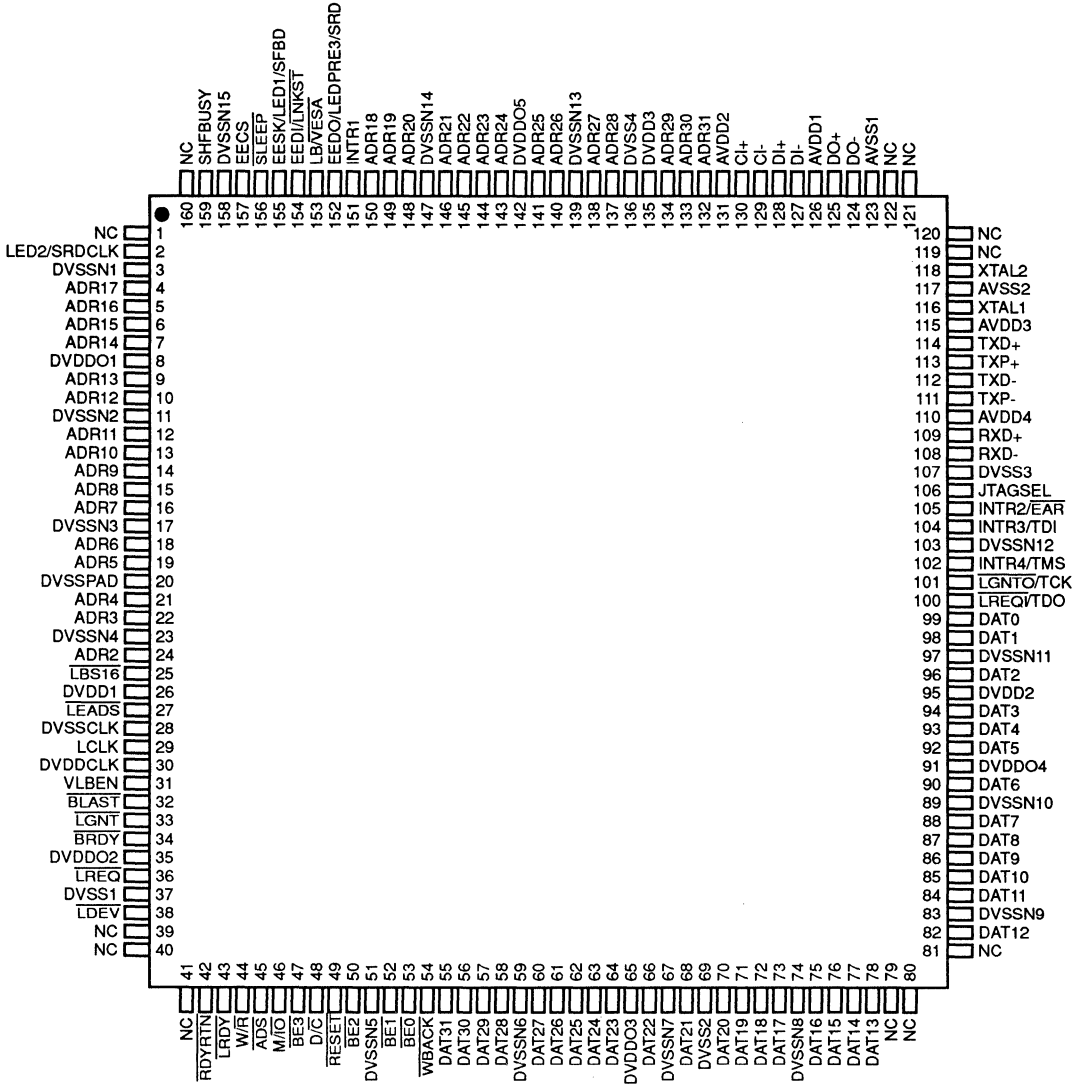
TABLE OF CONTENTS

BLOCK DIAGRAM: VESA VL-BUS MODE



18219B-1

CONNECTION DIAGRAM: VESA VL-BUS MODE



18219B-2

PIN DESIGNATIONS: VESA VL-BUS MODE
Listed by Pin Number

Pin #	Name	Pin #	Name	Pin #	Name	Pin #	Name
1	NC	41	NC	81	NC	121	NC
2	LED2/SRDCLK	42	RDYRTN	82	DAT12	122	NC
3	DVSSN1	43	LRDY	83	DVSSN9	123	AVSS1
4	ADR17	44	W/R	84	DAT11	124	DO-
5	ADR16	45	ADS	85	DAT10	125	DO+
6	ADR15	46	M/IO	86	DAT9	126	AVDD1
7	ADR14	47	BE3	87	DAT8	127	DI-
8	DVDDO1	48	D/C	88	DAT7	128	DI+
9	ADR13	49	RESET	89	DVSSN10	129	CI-
10	ADR12	50	BE2	90	DAT6	130	CI+
11	DVSSN2	51	DVSSN5	91	DVDDO4	131	AVDD2
12	ADR11	52	BE1	92	DAT5	132	ADR31
13	ADR10	53	BE0	93	DAT4	133	ADR30
14	ADR9	54	WBACK	94	DAT3	134	ADR29
15	ADR8	55	DAT31	95	DVDD2	135	DVDD3
16	ADR7	56	DAT30	96	DAT2	136	DVSS4
17	DVSSN3	57	DAT29	97	DVSSN11	137	ADR28
18	ADR6	58	DAT28	98	DAT1	138	ADR27
19	ADR5	59	DVSSN6	99	DAT0	139	DVSSN13
20	DVSSPAD	60	DAT27	100	LREQI/TDO	140	ADR26
21	ADR4	61	DAT26	101	LGNTO/TCK	141	ADR25
22	ADR3	62	DAT25	102	INTR4/TMS	142	DVDDO5
23	DVSSN4	63	DAT24	103	DVSSN12	143	ADR24
24	ADR2	64	DAT23	104	INTR3/TDI	144	ADR23
25	LBS16	65	DVDDO3	105	INTR2/EAR	145	ADR22
26	DVDD1	66	DAT22	106	JTAGSEL	146	ADR21
27	LEADS	67	DVSSN7	107	DVSS3	147	DVSSN14
28	DVSSCLK	68	DAT21	108	RXD-	148	ADR20
29	LCLK	69	DVSS2	109	RXD+	149	ADR19
30	DVDDCLK	70	DAT20	110	AVDD4	150	ADR18
31	VLBEN	71	DAT19	111	TXP-	151	INTR1
32	BLAST	72	DAT18	112	TXD-	152	EEDO/LEDPRE3/SRD
33	LGNT	73	DAT17	113	TXP+	153	LB/VESA
34	BRDY	74	DVSSN8	114	TXD+	154	EEDI/LNKST
35	DVDDO2	75	DAT16	115	AVDD3	155	EESK/LED1/SFBD
36	LREQ	76	DAT15	116	XTAL1	156	SLEEP
37	DVSS1	77	DAT14	117	AVSS2	157	EECS
38	LDEV	78	DAT13	118	XTAL2	158	DVSSN15
39	NC	79	NC	119	NC	159	SHFBUSY
40	NC	80	NC	120	NC	160	NC

PIN DESIGNATIONS: VESA VL-BUS MODE
Listed by Pin Name

Name	Pin #	Name	Pin #	Name	Pin #	Name	Pin #
ADR2	24	$\overline{BE3}$	47	DO+	125	JTAGSEL	106
ADR3	22	$\overline{BLAST}$	32	DO-	124	LB/$\overline{VESA}$	153
ADR4	21	$\overline{BRDY}$	34	DVDD1	26	$\overline{LBS16}$	25
ADR5	19	CI+	130	DVDD2	95	LCLK	29
ADR6	18	CI-	129	DVDD3	135	$\overline{LDEV}$	38
ADR7	16	D/$\overline{C}$	48	DVDDCLK	30	$\overline{LEADS}$	27
ADR8	15	DAT0	99	DVDDO1	8	LED2/SRDCLK	2
ADR9	14	DAT1	98	DVDDO2	35	$\overline{LGNT}$	33
ADR10	13	DAT2	96	DVDDO3	65	$\overline{LGNTO}$/TCK	101
ADR11	12	DAT3	94	DVDDO4	91	$\overline{LRDY}$	43
ADR12	10	DAT4	93	DVDDO5	142	$\overline{LREQ}$	36
ADR13	9	DAT5	92	DVSS1	37	$\overline{LREQI}$/TDO	100
ADR14	7	DAT6	90	DVSS2	69	M/$\overline{IO}$	46
ADR15	6	DAT7	88	DVSS3	107	NC	1
ADR16	5	DAT8	87	DVSS4	136	NC	39
ADR17	4	DAT9	86	DVSSCLK	28	NC	40
ADR18	150	DAT10	85	DVSSN1	3	NC	41
ADR19	149	DAT11	84	DVSSN2	11	NC	79
ADR20	148	DAT12	82	DVSSN3	17	NC	80
ADR21	146	DAT13	78	DVSSN4	23	NC	81
ADR22	145	DAT14	77	DVSSN5	51	NC	119
ADR23	144	DAT15	76	DVSSN6	59	NC	120
ADR24	143	DAT16	75	DVSSN7	67	NC	121
ADR25	141	DAT17	73	DVSSN8	74	NC	122
ADR26	140	DAT18	72	DVSSN9	83	NC	160
ADR27	138	DAT19	71	DVSSN10	89	$\overline{RDYRTN}$	42
ADR28	137	DAT20	70	DVSSN11	97	$\overline{RESET}$	49
ADR29	134	DAT21	68	DVSSN12	103	RXD+	109
ADR30	133	DAT22	66	DVSSN13	139	RXD-	108
ADR31	132	DAT23	64	DVSSN14	147	SHFBUSY	159
$\overline{ADS}$	45	DAT24	63	DVSSN15	158	$\overline{SLEEP}$	156
AVDD1	126	DAT25	62	DVSSPAD	20	TXD+	114
AVDD2	131	DAT26	61	EECS	157	TXD-	112
AVDD3	115	DAT27	60	EEDI/$\overline{LNKST}$	154	TXP+	113
AVDD4	110	DAT28	58	EEDO/LEDPRE3/SRD	152	TXP-	111
AVSS1	123	DAT29	57	EESK/LED1/SFBD	155	VLBEN	31
AVSS2	117	DAT30	56	INTR1	151	W/$\overline{R}$	44
$\overline{BE0}$	53	DAT31	55	INTR2/$\overline{EAR}$	105	$\overline{WBACK}$	54
$\overline{BE1}$	52	DI+	128	INTR3/TDI	104	XTAL1	116
$\overline{BE2}$	50	DI-	127	INTR4/TMS	102	XTAL2	118

PIN DESIGNATIONS: VESA VL-BUS MODE
Listed by Group

Pin Name	Pin Function	Type	Driver	No. of Pins
VESA VL-Bus Interface				
ADR2–ADR31	Address Bus	IO	TS	30
$\overline{ADS}$	Address Status	I/O	TS	1
$\overline{BE0}$–$\overline{BE3}$	Byte Enable	I/O	TS	4
$\overline{BLAST}$	Burst Last	O	TS	1
$\overline{BRDY}$	Burst Ready	I/O	TS	1
D/$\overline{C}$	Data/Control Select	I/O	TS	1
DAT0–DAT31	Data Bus	I/O	TS	32
INTR1	Interrupt Number 1	O	TS	1
INTR2	Interrupt Number 2	I/O	TS	1
INTR3	Interrupt Number 3	I/O	TS	1
INTR4	Interrupt Number 4	I/O	TS	1
JTAGSEL	JTAG Select	I		1
LB/$\overline{VESA}$	Local Bus/VESA VL-Bus Select pin	I		1
$\overline{LBS16}$	Local Bus Size 16	I		1
LCLK	Local Clock	I		1
$\overline{LDEV}$	Local Device	O	O4	1
$\overline{LEADS}$	Local External Address Strobe	O	TS	1
$\overline{LGNT}$	Local Bus Grant	I		1
$\overline{LGNTO}$	Local Grant Out	I/O	TS	1
$\overline{LRDY}$	Local Ready	O	TS	1
$\overline{LREQ}$	Local Bus Request	O	O8	1
$\overline{LREQI}$	Local Bus Request In	I/O	TS	1
M/$\overline{IO}$	Memory/I/O Select	I/O	TS	1
$\overline{RDYRTN}$	Ready Return	I		1
$\overline{RESET}$	Reset	I		1
VLBEN	Burst Enable	I		1
W/$\overline{R}$	Write/Read Select	I/O	TS	1
$\overline{WBACK}$	Write Back	I		1
Board Interface				
EECS	Microwire Serial EEPROM Chip Select	O	O8	1
EEDI/$\overline{LNKST}$	Microwire Serial EEPROM Data In/Link Status	O	LED	1
EEDO/LEDPRE3	Microwire Address EEPROM Data Out/LED3 predriver	I/O	LED	1
EESK/LED1	Microwire Serial EEPROM Clock/LED1	O	LED	1
LED2	LED Output Number 2	O	LED	1
SHFBUSY	Shift Busy (for external EEPROM-programmable logic)	O	O8	1
$\overline{SLEEP}$	Sleep Mode	I		1
XTAL1	Crystal Input	I		1
XTAL2	Crystal Output	O		1

PIN DESIGNATIONS: VESA VL-BUS MODE (continued)
Listed by Group

Pin Name	Pin Function	Type	Driver	No. of Pins
Attachment Unit Interface (AUI)				
CI+/CI-	AUI Collision Differential Pair	I		2
DI+/DI-	AUI Data In Differential Pair	I		2
DO+/DO-	AUI Data Out Differential Pair	O	DO	2
Twisted Pair Transceiver Interface (10BASE-T)				
RXD+/RXD-	Receive Differential Pair	I		2
TXD+/TXD-	Transmit Differential Pair	O	TDO	2
TXP+/TXP-	Transmit Pre-Distortion Differential Pair	O	TPO	2
$\overline{\text{LNKST}}$/EEDI	Link Status/Microwire Serial EEPROM Data In	O	LED	1
IEEE 1149.1 Test Access Port Interface (JTAG)				
TCK	Test Clock	I/O	TS	1
TDI	Test Data In	I/O	TS	1
TDO	Test Data Out	I/O	TS	1
TMS	Test Mode Select	I/O	TS	1
External Address Detection Interface (EADI)				
$\overline{\text{EAR}}$	External Address Reject Low	I/O	TS	1
SRD	Serial Receive Data	I/O	TS	1
SRDCLK	Serial Receive Data Clock	I/O	TS	1
SFBD	Start Frame-Byte Delimiter	O	LED	1
Power Supplies				
AVDD	Analog Power	P		4
AVSS	Analog Ground	P		2
DVDD	Digital Power	P		3
DVDDCLK	Digital Power Clock	P		1
DVDDO	I/O Buffer Digital Power	P		5
DVSS	Digital Ground	P		4
DVSSCLK	Digital Ground Clock	P		1
DVSSN	I/O Buffer Digital Ground	P		15
DVSSPAD	Digital Ground Pad	P		1

PIN DESIGNATIONS: VESA VL-BUS MODE

Driver Type

Table 1. Output Driver Types

Name	Type	I_{OL} (mA)	I_{OH} (mA)	pF
TS	Tri-State	8	–0.4	50
O8	Totem Pole	8	–0.4	50
O4	Totem Pole	4	–0.4	50
OD	Open Drain	8		50
LED	LED	12	–0.4	50

Table 2. Pins with Pull-Ups

Signal	Pull-up
TDI	≥ 10 KΩ
TMS	≥ 10 KΩ
TCK	≥ 10 KΩ

PIN DESCRIPTION: VESA VL-BUS MODE

Configuration Pins

JTAGSEL

JTAG Function Select *Input*

The value of this pin will asynchronously select between JTAG Mode and Multi-Interrupt Mode.

The value of this pin will asynchronously affect the function of the JTAG–INTR–Daisy chain arbitration pins, regardless of the state of the RESET pin and regardless of the state of the LCLK pin. If the value is a "1", then the PCnet-32 controller will be programmed for JTAG mode. If the value is a "0", then the PCnet-32 controller will be programmed for Multi-Interrupt Mode.

When programmed for JTAG mode, four pins of the PCnet-32 controller will be configured as a JTAG (IEEE 1149.1) Test Access Port. When programmed for Multi-Interrupt Mode, two of the JTAG pins will become interrupts and two JTAG pins will be used for daisy chain arbitration support. Table 3 below outlines the pin changes that will occur by programming the JTAGSEL pin.

Table 3. JTAG Pin Changes

Pin	JTAGSEL=1 JTAG mode	JTAGSEL=0 Multi-Interrupt Mode
LGNT0/TCK	TCK	LGNTO
LGNT1/TDO	TDO	LREQI
LGNT2/TDI	TDI	INTR3
LGNT3/TMS	TMS	INTR4

The JTAGSEL pin may be tied directly to V_{DD} or V_{SS}. A series resistor may be used but is not necessary.

LB/VESA

Local Bus/VESA VL-Bus Select *Input*

The value of this pin will asynchronously determine the operating mode of the PCnet-32 controller, regardless of the state of the RESET pin and regardless of the state of the LCLK pin. If the LB/VESA pin is a tied to V_{DD}, then the PCnet-32 controller will be programmed for Local Bus Mode. If the LB/VESA pin is tied to V_{SS}, then the PCnet-32 controller will be programmed for VESA-VL Bus Mode.

Note that the setting of LB/VESA determines the functionality of the following pins (names in parentheses are pins in 486 local bus mode): VLBEN (Am486), RESET (RESET), LBS16 (AHOLD), LREQ (HOLD), LGNT (HLDA), LREQI (HOLDI) and LGNTO (HLDAO).

VLBEN

Burst Enable *Input*

This pin is used to determine whether or not bursting is supported by the PCnet-32 device in VESA VL-Bus mode. The VLBEN pin is sampled at every rising edge of LCLK while the RESET pin is asserted.

In VESA-VL mode (the LB/VESA pin is tied to V_{SS}), if the sampled value of VLBEN is low, then the BREADE and BWRITE bits in BCR18 will be forced low, and the PCnet-32 controller will never attempt to perform linear burst reads or writes. If the sampled value of VLBEN is high, linear burst accesses are permitted, consistent with the values programmed into BREADE and BWRITE.

Because of byte-duplication conventions within a 32-bit Am386 system, the PCnet-32 controller will always produce the correct bytes in the correct byte lanes in accordance with the Am386DX data sheet. This byte duplication will automatically occur, regardless of the operating mode selected by the LB/VESA pin.

The VLBEN pin may be tied directly to V_{DD} or V_{SS}. A series resistor may be used but is not necessary.

The VLBEN pin need only be valid when the RESET pin is active (regardless of the connection of the LB/VESA pin) and may be tied to ID(3) in a VESA VL-Bus version 1.0 system, or to the logical AND of ID(4), ID(3), ID(1), and ID(0) in a VESA-VL-Bus version 1.1 or 2.0 system.

Note: *This pin needs to be tied low when the LB/VESA pin has been tied to V_{DD}. See the pin description for the Am486 pin in the Local Bus Mode section.*

Configuration Pin Settings Summary

Table 4 below shows the possible pin configurations that may be invoked with the PCnet-32 controller configuration pins.

Table 4. Configuration Pin Settings

LB/VESA	VLBEN	JTAGSEL	Mode Selected
0	X*	0	VL Bus mode with 4 interrupts and daisy chain arbitration
0	X*	1	VL Bus mode with 2 interrupts and JTAG
1	0	0	Am486 mode with 4 interrupts and daisy chain arbitration
1	0	1	Am486 mode with 2 interrupts and JTAG
1	1	X	Reserved

**X = Don't care*

Pin Connections to V_{DD} or V_{SS}

Several pins may be connected to V_{DD} or V_{SS} for various application options. Some pins are required to be connected to V_{DD} or V_{SS} in order to set the controller into a particular mode of operation, while other pins might be connected to V_{DD} or V_{SS} if that pin's function is not implemented in a specific application. Table 5 shows which pins *require* a connection to V_{DD} or V_{SS}, and which pins may *optionally* be connected to V_{DD} or V_{SS} because the application does not support that pin's function. The table also shows whether or not the connections need to be resistive.

VESA VL-Bus Interface

ADR2–ADR31

Address Bus *Input/Output*

Address information which is stable during a bus operation, regardless of the source. When the PCnet-32 controller is Current Master, A2–A31 will be driven. When the PCnet-32 controller is not Current Master, the A2–A31 lines are continuously monitored to determine if an address match exists for I/O slave transfers.

$\overline{ADS}$

Address Status *Input/Output*

When driven LOW, this signal indicates that a valid bus cycle definition and address are available on the M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$ and A2–A31 pins of the local bus interface. At that time, the PCnet-32 controller will examine the combination of M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$, and the A2–A31 pins to determine if the current access is directed toward the PCnet-32 controller.

$\overline{ADS}$ will be driven LOW when the PCnet-32 controller performs a bus master access on the local bus.

$\overline{BE0}$–$\overline{BE3}$

Byte Enable *Input/Output*

These signals indicate which bytes on the data bus are active during read and write cycles. When $\overline{BE3}$ is active, the byte on DAT31–DAT24 is valid. $\overline{BE2}$–$\overline{BE0}$ active indicate valid data on pins DAT23–DAT16, DAT15–DAT8, DAT7–DAT0, respectively. The byte enable signals are outputs for bus master and inputs for bus slave operations.

$\overline{BLAST}$

Burst Last *Output*

When the $\overline{BLAST}$ signal is asserted, then the next time that $\overline{BRDY}$ or $\overline{RDYRTN}$ is asserted, the burst cycle is complete.

$\overline{BRDY}$

Burst Ready *Input/Output*

$\overline{BRDY}$ functions as an input to the PCnet-32 controller during bus master cycles. When $\overline{BRDY}$ is asserted during a master cycle, it indicates to the PCnet-32 controller that the target device is accepting burst transfers. It also serves the same function as $\overline{RDYRTN}$ does for non-burst accesses. That is, it indicates that the target device has accepted the data on a master write cycle, or that the target device has presented valid data onto the bus during master read cycles.

Table 5. Pin Connections to Power/Ground

Pin Name	Pin No	Supply Strapping	Resistive Connection to Supply	Recommended Resistor Size
LED2/SRDCLK	2	Required	Required	324 Ω in series with LED, or 10 KΩ without LED
$\overline{LBS16}$	25	Optional	Required	10 KΩ
VLBEN	31	Required	Optional	NA
$\overline{WBACK}$	54	Optional	Required	10 KΩ
$\overline{LREQI}$/TDO	100	Optional	Required	10 KΩ
JTAGSEL	106	Required	Optional	NA
EEDO/LEDPRE3/SRD	152	Optional	Required	10 KΩ
LB/$\overline{VESA}$	153	Required	Optional	NA
EEDI/$\overline{LNKST}$	154	Optional	Required	324 Ω in series with LED, or 10 KΩ without LED
EESK/LED1/SFBD	155	Required	Required	324 Ω in series with LED, or 10 KΩ without LED
$\overline{SLEEP}$	156	Optional	Required	10 KΩ
All Other Pins	—	Optional	Required	10 KΩ

If $\overline{BRDY}$ and $\overline{RDYRTN}$ are sampled active in the same cycle, then $\overline{RDYRTN}$ takes precedence, causing the next transfer cycle to begin with a T1 cycle.

$\overline{BRDY}$ functions as an output during PCnet-32 controller slave cycles and is always driven inactive (HIGH).

$\overline{BRDY}$ is floated if the PCnet-32 controller is not being accessed as the current slave device on the local bus.

D/$\overline{C}$

Data/Control Select *Input/Output*

During slave accesses to the PCnet-32 controller, the D/$\overline{C}$ pin, along with M/$\overline{IO}$ and W/$\overline{R}$, indicates the type of cycle that is being performed. PCnet-32 controller will only respond to local bus accesses in which D/$\overline{C}$ is driven HIGH by the local bus master.

During PCnet-32 controller bus master accesses, the D/$\overline{C}$ pin is an output and will always be driven HIGH.

D/$\overline{C}$ is floated if the PCnet-32 controller is not the current master on the local bus.

DAT0–DAT31

Data Bus *Input/Output*

Used to transfer data to and from the PCnet-32 controller to system resources via the local bus. DAT31–DAT0 are driven by the PCnet-32 controller when performing bus master writes and slave read operations. Data on DAT31–DAT0 is latched by the PCnet-32 controller when performing bus master reads and slave write operations.

The PCnet-32 controller will always follow Am386DX byte lane conventions. This means that for word and byte accesses in which PCnet-32 controller drives the data bus (i.e. master write operations and slave read operations), the PCnet-32 controller will produce duplicates of the active bytes on the unused half of the 32-bit data bus. Table 6 illustrates the cases in which duplicate bytes are created.

Table 6. Byte Duplication on Data Bus

$\overline{BE3}$–$\overline{BE0}$	DAT [31:24]	DAT [23:16]	DAT [15:8]	DAT [7:0]
1110	Undef	Undef	Undef	A
1101	Undef	Undef	A	Undef
1011	Undef	A	Undef	Copy A
0111	A	Undef	Copy A	Undef
1100	Undef	Undef	B	A
1001	Undef	C	B	Undef
0011	D	C	Copy D	Copy C
1000	Undef	C	B	A
0001	D	C	B	Undef
0000	D	C	B	A

INTR1–INTR4

Interrupt Request *Output*

An attention signal which indicates that one or more of the following status flags is set: BABL, MISS, MERR, RINT, IDON, MFCO, RCVCCO, TXSTRT, or JAB. Each of these status flags has a mask bit which allows for suppression of INTR assertion. These flags have the meaning shown in Table 7.

Table 7. Status Flags

BABL	Babble (CSR0, bit 14)
MISS	Missed Frame (CSR0, bit 12)
MERR	Memory Error (CSR0, bit 11)
RINT	Receive Interrupt (CSR0, bit 10)
IDON	Initialization Done (CSR0, bit 8)
MFCO	Missed Packet Count Overflow (CSR4, bit 9)
RCVCCO	Receive Collision Count Overflow (CSR4, bit 5)
TXSTRT	Transmit Start (CSR4, bit 3)
JAB	Jabber (CSR4, bit 1)

Note that there are four possible interrupt pins, depending upon the mode that has been selected with the JTAGSEL pin. Only one interrupt pin may be used at one time. The active interrupt pin is selected by programming the interrupt select register (BCR21). The default setting of BCR121will select interrupt INTR1 as the active interrupt. Note that BCR21 is EEPROM-programmable. Inactive interrupt pins are floated.

The polarity of the interrupt signal is determined by the INTLEVEL bit of BCR2. The interrupt pins may be programmed for level-sensitive or edge-sensitive operation.

PCnet-32 controller interrupt pins will be floated at H_RESET and will remain floated until either the EEPROM has been successfully read, or, following an EEPROM read failure, a Software Relocatable Mode sequence has been successfully executed.

$\overline{LBS16}$

Local Bus Size 16 *Input*

$\overline{LBS16}$ is sampled during PCnet-32 controller bus mastering activity to determine if the target device on the VL-Bus is 32 or 16 bits in width. If the $\overline{LBS16}$ signal is sampled active at least one clock period before the assertion of $\overline{LRDY}$ during a PCnet-32 controller bus master transfer, then the PCnet-32 controller will convert a single 32-bit transfer into two 16-bit transfers. Not all 32-bit transfers need to be split into two 16-bit transfers. Table 8 shows the sequence of transfers that will be executed for each possible 32-bit bus transfer that encounters a proper assertion of the $\overline{LBS16}$ signal.

Table 8. Data Transfer Sequence from 32-Bit Wide to 16-Bit Wide

Current Access				Next with $\overline{LBS16}$			
$\overline{BE3}$	$\overline{BE2}$	$\overline{BE1}$	$\overline{BE0}$	$\overline{BE3}$	$\overline{BE2}$	$\overline{BE1}$	$\overline{BE0}$
1	1	1	0	NR			
1	1	0	0	NR			
1	0	0	0	1	0	1	1
0	0	0	0	0	0	1	1
1	1	0	1	NR			
1	0	0	1	1	0	1	1
0	0	0	1	0	0	1	1
1	0	1	1	NR			
0	0	1	1	NR			
0	1	1	1	NR			

NR = No second access Required for these cases

During accesses in which PCnet-32 controller is acting as the VL-Bus target device, the $\overline{LBS16}$ signal will not be driven. In this case, it is expected that the VL-Bus required pull-up device will bring the $\overline{LBS16}$ signal to an inactive level and the PCnet-32 controller will be seen by the VL-Bus master as a 32-bit peripheral.

LCLK

Local Clock *Input*

LCLK is a 1x clock that follows the same phase as a 486-type CPU clock. LCLK is always driven by the system logic or the VL-Bus controller to all VL-Bus masters and targets. The rising edge of the clock signifies the change of CPU states, and hence, the change of PCnet-32 controller states.

$\overline{LDEV}$

Local Device *Output*

$\overline{LDEV}$ is driven by the PCnet-32 controller when it recognizes an access to PCnet-32 controller I/O space. Such recognition is dependent upon a valid sampled $\overline{ADS}$ strobe plus valid M/$\overline{IO}$, D/$\overline{C}$ and ADR31–ADR5 values.

$\overline{LEADS}$

Local External Address Strobe *Output*

During VL-Bus master write and read accesses the $\overline{LEADS}$ pin will be asserted on every T1 cycle as is specified in the VESA VL-Bus specification, regardless of the settings of the GCIC bit of BCR18 and the CLL bits of BCR18.

$\overline{LGNT}$

Local Bus Grant *Input*

When $\overline{LGNT}$ is asserted and $\overline{LREQ}$ is being asserted by the PCnet-32 controller, the PCnet-32 controller assumes ownership of the VL bus.

Note that this pin changes polarity when Local Bus mode has been selected (see pin description of HLDA in 486 Local Bus Interface section).

$\overline{LGNTO}$

Local Grant Out *Output*

This signal is multiplexed with the TCK pin, and is available only when the Multi-Interrupt mode has been selected with the JTAGSEL pin.

An additional local bus master may daisy-chain its $\overline{LGNT}$ signal through the PCnet-32 controller $\overline{LGNTO}$ pin. The PCnet-32 controller will deliver a $\overline{LGNTO}$ signal to the additional local bus master whenever the PCnet-32 controller receives a $\overline{LGNT}$ from the arbitration logic, but is not simultaneously requesting the bus internally. The second local bus master must connect its $\overline{LREQ}$ output to the $\overline{LREQI}$ input of the PCnet-32 controller in order to complete the local bus daisy-chain arbitration control.

When $\overline{SLEEP}$ is not asserted, daisy chain arbitration signals that pass through the PCnet-32 controller will experience a one-clock delay from input to output (i.e. $\overline{LREQI}$ to $\overline{LREQ}$ and $\overline{LGNT}$ to $\overline{LGNTO}$).

While $\overline{SLEEP}$ is asserted (either in **snooze** mode or **coma** mode), if the PCnet-32 controller is configured for a daisy chain ($\overline{LREQI}$ and $\overline{LGNTO}$ signals have been selected with the JTAGSEL pin), then the system arbitration signal $\overline{LGNT}$ will be passed directly to the daisy-chain signal $\overline{LGNTO}$ without experiencing a one-clock delay. However, some combinatorial delay will be introduced in this path.

Note that this pin changes polarity when Local Bus mode has been selected (see pin description of HLDAO in 486 Local Bus Interface section).

$\overline{LRDY}$

Local Ready *Output*

$\overline{LRDY}$ functions as an output from the PCnet-32 controller during PCnet-32 controller slave cycles. During PCnet-32 controller slave read cycles, $\overline{LRDY}$ is asserted to indicate that valid data has been presented on

the data bus. During PCnet-32 controller slave write cycles, LRDY is asserted to indicate that the data on the data bus has been internally latched. LRDY will be asserted low for one clock period when the PCnet-32 controller wishes to terminate the cycle. LRDY is then driven high for one-half of one clock period before being released.

LRDY is floated if the PCnet-32 controller is not the current slave on the local bus.

LREQ

Local Bus Request *Output*

LREQ is used by the PCnet-32 controller to gain control of the VL-Bus and become the active VL Bus Master. LREQ is active low. Once asserted, LREQ remains active until LGNT has become active, independent of subsequent assertion of SLEEP or setting of the STOP bit or access to the S_RESET port (offset 14h).

Note that this pin changes polarity when Local Bus mode has been selected (see pin description of HOLD in 486 Local Bus Interface section).

LREQI

Local Bus Request In *Input*

This signal is multiplexed with the TDO pin, and is available only when the Multi-Interrupt mode has been selected with the JTAGSEL pin.

An additional local bus master may daisy-chain its bus hold request signal through the PCnet-32 controller LREQI pin. The PCnet-32 controller will convey the LREQI request to the arbitration logic via the PCnet-32 controller LREQ output. The second local bus master must connect its LGNT input to the LGNTO output of the PCnet-32 controller in order to complete the local bus daisy-chain arbitration control.

When SLEEP is not asserted, daisy chain arbitration signals that pass through the PCnet-32 controller will experience a one-clock delay from input to output (i.e. LREQI to LREQ and LGNT to LGNTO).

While SLEEP is asserted (either in **snooze** mode or **coma** mode), if the PCnet-32 controller is configured for a daisy chain (LREQI and LGNTO signals have been selected with the JTAGSEL pin), then the daisy-chain signal LREQI will be passed directly to the system arbitration signal LREQ without experiencing a one-clock delay. However, some combinatorial delay will be introduced in this path.

If Multi-Interrupt mode has been selected and the daisy-chain arbitration feature is not used, then the LREQI input should be tied to V_{DD}.

Note that this pin changes polarity when Local Bus mode has been selected (see pin description of HOLDI in 486 Local Bus Interface section).

M/IO

Memory I/O Select *Input/Output*

During slave accesses to the PCnet-32 controller, the M/IO pin, along with D/C and W/R, indicates the type of cycle that is being performed. PCnet-32 controller will only respond to local bus accesses in which M/IO is sampled as a zero by the PCnet-32 controller.

During PCnet-32 controller bus master accesses, the M/IO pin is an output and will always be driven high.

M/IO is floated if the PCnet-32 controller is not the current master on the local bus.

RDYRTN

Ready Return *Input*

RDYRTN functions as an input to the PCnet-32 controller. RDYRTN is used to terminate all master accesses performed by the PCnet-32 controller, except that linear burst transfers may also be terminated with the BRDY signal. RDYRTN is used to terminate slave read accesses to PCnet-32 controller I/O space.

When asserted during slave read accesses to PCnet-32 controller I/O space, RDYRTN indicates that the bus mastering device has seen the LRDY that was generated by the PCnet-32 controller and has accepted the PCnet-32 controller slave read data. Therefore, PCnet-32 controller will hold slave read data on the bus until it synchronously samples the RDYRTN input as active low. The PCnet-32 controller will *not* hold LRDY valid asserted during this time. The duration of the LRDY pulse generated by the PCnet-32 controller will always be a single LCLK cycle.

RDYRTN is ignored during slave write accesses to PCnet-32 controller I/O space. Slave write accesses to PCnet-32 controller I/O space are considered terminated by the PCnet-32 controller at the end of the cycle during which the PCnet-32 controller issues an active RDY.

In systems where both a LRDY and RDYRTN (or equivalent) signals are provided, then LRDY must *not* be tied to RDYRTN. Most systems now provide for a local device ready input to the memory controller that is separate from the CPU READY signal. This second READY signal is usually labeled as READYIN. This signal should be connected to the PCnet-32 controller LRDY signal. The CPU READY signal should be connected to the PCnet-32 controller RDYRTN pin.

In systems where only one READY signal is provided, then the PCnet-32 controller LRDY output may be tied to the PCnet-32 controller RDYRTN input.

RESET

System Reset *Input*

When RESET is asserted low and the LB/VESA pin has been tied to VSS, then the PCnet-32 controller performs an internal system reset of the H_RESET type (HARDWARE_RESET). The RESET pin must be held for a minimum of 30 LCLK periods when VL mode has been selected. While in the H_RESET state, the PCnet-32 controller will float or deassert all outputs.

W/R

Write/Read Select *Input/Output*

During slave accesses to the PCnet-32 controller, the W/R pin, along with D/C and M/IO, indicates the type of cycle that is being performed.

During PCnet-32 controller bus master accesses, the W/R pin is an output.

W/R is floated if the PCnet-32 controller is not the current master on the local bus.

WBACK

Write Back *Input*

WBACK is monitored as in input during VL-Bus Master Accesses. When PCnet-32 controller is current VL-Bus master, the PCnet-32 controller will float all appropriate bus mastering signals within 1 clock period of the assertion of WBACK. When WBACK is deasserted, PCnet-32 controller will re-execute any accesses that were suspended due to the assertion of WBACK and then will proceed with other scheduled accesses, if any.

Register access cannot be performed to the PCnet-32 device while WBACK is asserted.

Board Interface

LED1

LED1 *Output*

This pin is shared with the EESK function. When operating as LED1, the function and polarity on this pin are programmable through BCR5. The LED1 output from the PCnet-32 controller is capable of sinking the necessary 12 mA of current to drive an LED directly.

The LED1 pin is also used during EEPROM Auto-detection to determine whether or not an EEPROM is present at the PCnet-32 controller microwire interface. At the trailing edge of RESET, this pin is sampled to determine the value of the EEDET bit in BCR19. A sampled HIGH value means that an EEPROM is present, and EEDET will be set to ONE. A sampled LOW value means that an EEPROM is not present, and EEDET will be set to ZERO. See the "EEPROM Auto-detection" section for more details.

If no LED circuit is to be attached to this pin, then a pull-up or pull-down resistor must be attached instead, in order to resolve the EEDET setting.

LED2

LED2 *Output*

This pin is shared with the SRDCLK function. When operating as LED2, the function and polarity on this pin are programmable through BCR6. The LED2 output from the PCnet-32 controller is capable of sinking the necessary 12 mA of current to drive an LED directly.

This pin also selects address width for Software Relocatable Mode. When this pin is HIGH during Software Relocatable Mode, then the device will be programmed to use 32 bits of addressing while snooping accesses on the bus during Software Relocatable Mode. When this pin is LOW during Software Relocatable Mode, then the device will be programmed to use 24 bits of addressing while snooping accesses on the bus during Software Relocatable Mode. The upper 8 bits of address will be assumed to match during the snooping operation when LED2 is LOW. The 24-bit addressing mode is intended for use in systems that employ the GPSI signals. For more information on the GPSI function see section *General Purpose Serial Interface.*

If no LED circuit is to be attached to this pin, then a pull up or pull down resistor must be attached instead, in order to resolve the Software Relocatable Mode address width setting.

LEDPRE3

LEDPRE3 *Output*

This pin is shared with the EEDO function. When operating as LEDPRE3, the function and polarity on this pin are programmable through BCR7. This signal is labeled as LED *"PRE"*3 because of the multi-function nature of this pin. If an LED circuit were directly attached to this pin, it would create an IOL requirement that could not be met by the serial EEPROM that would also be attached to this pin. Therefore, if this pin is to be used as an additional LED output while an EEPROM is used in the system, then buffering is required between the LEDPRE3 pin and the LED circuit. If no EEPROM is included in the system design, then the LEDPRE3 signal may be directly connected to an LED without buffering. The LEDPRE3 output from the PCnet-32 controller is capable of sinking the necessary 12 mA of current to drive an LED in this case. For more details regarding LED connection, see the section on LEDs.

LNKST

Link Status *Output*

This pin provides 12 mA for driving an LED. It indicates an active link connection on the 10BASE-T interface. The function and polarity are programmable through BCR4. Note that this pin is multiplexed with the EEDI function.

This pin remains active in snooze mode.

SHFBUSY

Shift Busy *Output*

The function of the SHFBUSY signal is to indicate when the last byte of the EEPROM contents has been shifted out of the EEPROM on the EEDO signal line. This information is useful for *external EEPROM-programmable registers* that do not use the microwire protocol, as is described herein: When the PCnet-32 controller is performing a serial read of the EEPROM through the microwire interface, the SHFBUSY signal will be driven HIGH. SHFBUSY can serve as a serial shift enable to allow the EEPROM data to be serially shifted into an external device or series of devices. The SHFBUSY signal will remain actively driven HIGH until the end of the EEPROM read operation. If the EEPROM checksum was verified, then the SHFBUSY signal will be driven LOW at the end of the EEPROM read operation. If the EEPROM checksum verification failed, then the SHFBUSY signal will remain HIGH. This function effectively demarcates the end of a successful EEPROM read operation and therefore is useful as a programmable-logic *low-active output enable* signal. For more details on external EEPROM-programmable registers, see the *EEPROM Microwire Access* section under *Hardware Access.*

This pin can be controlled by the host system by writing to BCR19, bit 3 (EBUSY).

$\overline{\text{SLEEP}}$

Sleep *Input*

When $\overline{\text{SLEEP}}$ input is asserted (active LOW), the PCnet-32 controller performs an internal system reset of the S_RESET type and then proceeds into a power savings mode. (The reset operation caused by $\overline{\text{SLEEP}}$ assertion will not affect BCR registers.) All outputs will be placed in their normal S_RESET condition. During sleep mode, all PCnet-32 controller inputs will be ignored except for the $\overline{\text{SLEEP}}$ pin itself. De-assertion of $\overline{\text{SLEEP}}$ results in wake-up. The system must refrain from starting the network operations of the PCnet-32 controller for 0.5 seconds following the deassertion of the $\overline{\text{SLEEP}}$ signal in order to allow internal analog circuits to stabilize.

Both LCLK and XTAL1 inputs must have valid clock signals present in order for the $\overline{\text{SLEEP}}$ command to take effect.

If $\overline{\text{SLEEP}}$ is asserted while $\overline{\text{LREQ}}$ is asserted, then the PCnet-32 controller will perform an internal system S_RESET and then wait for the assertion of $\overline{\text{LGNT}}$. When $\overline{\text{LGNT}}$ is asserted, the $\overline{\text{LREQ}}$ signal will be deasserted and then the PCnet-32 controller will proceed to the power savings mode. Note that the internal system S_RESET will not cause the $\overline{\text{LREQ}}$ signal to be deasserted.

The $\overline{\text{SLEEP}}$ pin should not be asserted during power supply ramp-up. If it is desired that $\overline{\text{SLEEP}}$ be asserted at power up time, then the system must delay the assertion of $\overline{\text{SLEEP}}$ until three LCLK cycles after the completion of a valid pin $\overline{\text{RESET}}$ operation.

XTAL1–XTAL2

Crystal Oscillator Inputs *Input/Output*

The crystal frequency determines the network data rate. The PCnet-32 controller supports the use of quartz crystals to generate a 20 MHz frequency compatible with the ISO 8802-3 (IEEE/ANSI 802.3) network frequency tolerance and jitter specifications. See the section *External Crystal Characteristics* (in section Manchester Encoder/Decoder) for more detail.

The network data rate is one-half of the crystal frequency. XTAL1 may alternatively be driven using an external CMOS level source, in which case XTAL2 must be left unconnected. Note that when teh PCnet-32 controller is in coma mode, there is an internal 22 KΩ resistor from XTAL1 to ground. If an external source drives XTAL1, some power will be consumed driving this resistor. If XTAL1 is driven LOW at this time power consumption will be minimized. In this case, XTAL1 must remain active for at least 30 cycles after the assertion of $\overline{\text{SLEEP}}$ and deassertion of $\overline{\text{LREQ}}$.

Microwire EEPROM Interface

EESK

EEPROM Serial Clock *Output*

The EESK signal is used to access the external ISO 8802-3 (IEEE/ANSI 802.3) address PROM. This pin is designed to directly interface to a serial EEPROM that uses the microwire interface protocol. EESK is connected to the microwire EEPROM's Clock pin. It is controlled by either the PCnet-32 controller directly during a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 1.

EESK can be used during programming of *external EEPROM-programmable registers* that do not use the microwire protocol as follows:

When the PCnet-32 controller is performing a serial read of the IEEE Address EEPROM through the microwire interface, the SHFBUSY signal will serve as a serial shift enable to allow the EEPROM data to be serially shifted into an external device or series of devices. This same signal can be used to gate the *output* of the programmed logic to avoid the problem of releasing intermediate values to the rest of the system board logic. The EESK signal can serve as the clock, and EEDO will serve as the input data stream to the programmable shift register.

EEDO

EEPROM Data Out *Input*

The EEDO signal is used to access the external ISO 8802-3 (IEEE/ANSI 802.3) address PROM. This pin is designed to directly interface to a serial EEPROM that uses the microwire interface protocol. EEDO is connected to the microwire EEPROM's Data Output pin. It

is controlled by the EEPROM during reads. It may be read by the host system by reading BCR19, bit 0.

EEDO can be used during programming of *external EEPROM-programmable registers* that do not use the microwire protocol as follows:

When the PCnet-32 controller is performing a serial read of the IEEE Address EEPROM through the microwire interface, the SHFBUSY signal will serve as a serial shift enable to allow the EEPROM data to be serially shifted into an external device or series of devices. This same signal can be used to gate the *output* of the programmed logic to avoid the problem of releasing intermediate values to the rest of the system board logic. The EESK signal can serve as the clock, and EEDO will serve as the input data stream to the programmable shift register.

EECS

EEPROM Chip Select *Output*

The function of the EECS signal is to indicate to the microwire EEPROM device that it is being accessed. The EECS signal is active high. It is controlled by either the PCnet-32 controller during a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 2.

EEDI

EEPROM Data In *Output*

The EEDI signal is used to access the external ISO 8802-3 (IEEE/ANSI 802.3) address PROM. EEDI functions as an output. This pin is designed to directly interface to a serial EEPROM that uses the microwire interface protocol. EEDI is connected to the microwire EEPROM's Data Input pin. It is controlled by either the PCnet-32 controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 0.

Attachment Unit Interface

CI±

Collision In *Input*

A differential input pair signaling the PCnet-32 controller that a collision has been detected on the network media, indicated by the CI± inputs being driven with a 10 MHz pattern of sufficient amplitude and pulse width to meet ISO 8802-3 (IEEE/ANSI 802.3) standards. Operates at pseudo ECL levels.

DI±

Data In *Input*

A differential input pair to the PCnet-32 controller carrying Manchester encoded data from the network. Operates at pseudo ECL levels.

DO±

Data Out *Output*

A differential output pair from the PCnet-32 controller for transmitting Manchester encoded data to the network. Operates at pseudo ECL levels.

Twisted Pair Interface

RXD±

10-BASE-T Receive Data *Input*

10BASE-T port differential receivers.

TXD±

10BASE-T Transmit Data *Output*

10BASE-T port differential drivers.

TXP±

10BASE-T Pre-distortion Control *Output*

These outputs provide transmit predistortion control in conjunction with the 10BASE-T port differential drivers.

External Address Detection Interface

The EADI interface is enabled through bit 3 of BCR2 (EADISEL).

$\overline{\text{EAR}}$

External Address Reject Low *Input*

An EADI input signal. The incoming frame will be checked against the internally active address detection mechanisms and the result of this check will be OR'd with the value on the $\overline{\text{EAR}}$ pin. The $\overline{\text{EAR}}$ pin is defined as $\overline{\text{REJECT}}$.

See the EADI section for details regarding the function and timing of this signal.

Note that this pin is multiplexed with the INTR2 pin.

SFBD

Start Frame–Byte Delimiter *Output*

Start Frame–Byte Delimiter Enable. EADI output signal. An initial rising edge on this signal indicates that a start of frame delimiter has been detected. The serial bit stream will follow on the SRD signal, commencing with the destination address field. SFBD will go high for 4 bit times (400 ns) after detecting the second "1" in the SFD (Start of Frame Delimiter) of a received frame. SFBD will subsequently toggle every 400 ns (1.25 MHz frequency) with each rising edge indicating the first bit of each subsequent byte of the received serial bit stream. SFBD will be inactive during frame transmission.

Note that this pin is multiplexed with the LED1 pin.

SRD

Serial Receive Data *Output*

An EADI output signal. SRD is the decoded NRZ data from the network. This signal can be used for external address detection. Note that when the 10BASE-T port is selected, transitions on SRD will only occur during receive activity. When the AUI port is selected, transitions on SRD will occur during both transmit and receive activity.

Note that this pin is multiplexed with the LEDPRE3 pin.

SRDCLK

Serial Receive Data Clock *Output*

An EADI output signal. Serial Receive Data is synchronous with reference to SRDCLK. Note that when the 10BASE-T port is selected, transitions on SRDCLK will only occur during receive activity. When the AUI port is selected, transitions on SRDCLK will occur during both transmit and receive activity.

Note that this pin is multiplexed with the LED2 pin.

General Purpose Serial Interface

The GPSI interface is selected through the PORTSEL bits of the Mode register (CSR15) and enabled through the TSTSHDW[1] bit (BCR18) or the CORETEST bit (CSR124).

Note that when GPSI test mode is invoked, slave address decoding must be restricted to the lower 24 bits of the address bus by setting the IOAW24 bit in BCR2 and by pulling LED2 LOW during Software Relocatable Mode. The upper 8 bits of the address bus will always be considered matched when examining incoming I/O addresses. During master accesses while in GPSI mode, the PCnet-32 controller will not drive the upper 8 bits of the address bus with address information. See the GPSI section for more detail.

TXDAT

Transmit Data *Input/Output*

TXDAT is an output, providing the serial bit stream for transmission, including preamble, SFD data and FCS field, if applicable.

Note that the TxDAT pin is multiplexed with the A31 pin.

TXEN

Transmit Enable *Input/Output*

TXEN is an output, providing an enable signal for transmission. Data on the TXDAT pin is not valid unless the TXEN signal is HIGH.

Note that the TXEN pin is multiplexed with the A30 pin.

STDCLK

Serial Transmit Data Clock *Input*

STDCLK is an input, providing a clock signal for MAC activity, both transmit and receive. Rising edges of the STDCLK can be used to validate TXDAT output data.

The STDCLK pin is multiplexed with the A29 pin.

Note that this signal must meet the frequency stability requirement of the ISO 8802-3 (IEEE/ANSI 802.3) specification for the crystal.

CLSN

Collision *Input/Output*

CLSN is an input, indicating to the core logic that a collision has occurred on the network.

Note that the CLSN pin is multiplexed with the A28 pin.

RXCRS

Receive Carrier Sense *Input/Output*

RXCRS is an input. When this signal is HIGH, it indicates to the core logic that the data on the RXDAT input pin is valid.

Note that the RXCRS pin is multiplexed with the A27 pin.

SRDCLK

Serial Receive Data Clock *Input/Output*

SRDCLK is an input. Rising edges of the SRDCLK signal are used to sample the data on the RXDAT input whenever the RXCRS input is HIGH.

Note that the SRDCLK pin is multiplexed with the A26 pin.

RXDAT

Receive Data *Input/Output*

RXDAT is an input. Rising edges of the SRDCLK signal are used to sample the data on the RXDAT input whenever the RXCRS input is HIGH.

Note that the RXDAT pin is multiplexed with the A25 pin.

IEEE 1149.1 Test Access Port Interface

TCK

Test Clock *Input*

The clock input for the boundary scan test mode operation. TCK can operate up to 10 MHz. If left unconnected, this pin has a default value of HIGH.

TDI

Test Data Input *Input*

The test data input path to the PCnet-32 controller. If left unconnected, this pin has a default value of HIGH.

TDO

Test Data Output *Output*

The test data output path from the PCnet-32 controller. TDO is floated when the JTAG port is inactive.

TMS

Test Mode Select *Input*

A serial input bit stream is used to define the specific boundary scan test to be executed. If left unconnected, this pin has a default value of HIGH.

Power Supply Pins

AVDD

Analog Power (4 Pins) ***Power***

There are four analog +5 Volt supply pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to Appendix B and the *PCnet Family Technical Manual (PID# 18216A)* for details.

AVSS

Analog Ground (2 Pins) ***Power***

There are two analog ground pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to Appendix B and the PCnet Family Technical Manual for details.

DVDD

Digital Power (3 Pins) ***Power***

There are 3 digital power supply pins (DVDD1, DVDD 2, and DVDD3) used by the internal digital circuitry.

DVDDCLK

Digital Power Clock (1 Pin) ***Power***

This pin is used to supply power to the clock buffering circuitry.

DVDDO

I/O Buffer Digital Power (5 Pins) ***Power***

There are 5 digital power supply pins (DVDDO1–DVDDO5) used by Input/Output buffer drivers.

DVSS

Digital Ground (4 Pins) ***Ground***

There are 4 digital ground pins (DVSS1–DVSS4) used by the internal digital circuitry.

DVSSCLK

Digital Ground Clock (1 Pin) ***Ground***

This pin is used to supply a ground to the clock buffering circuitry.

DVSSN

I/O Buffer Digital Ground (15 Pins) ***Ground***

These 15 ground pins (DVSSN1–DVSSN15) are used by the Input/Output buffer drivers.

DVSSPAD

Digital Ground Pad (1 Pin) ***Ground***

This pin is used by the Input/Output logic circuits.

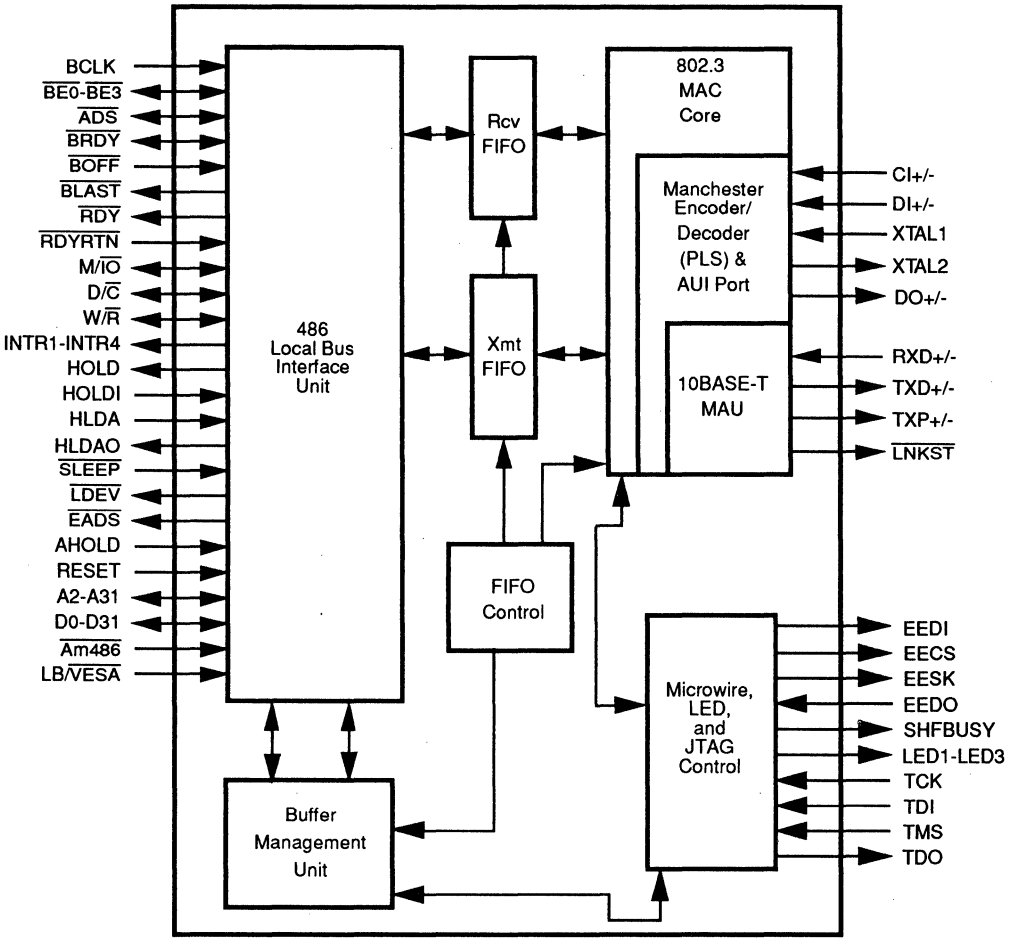
VESA VL-BUS / LOCAL BUS PIN CROSS-REFERENCE
Listed by Pin Number

Pin #	VESA VL-Bus Mode Pin Name	Local Bus Mode Pin Name	Pin #	VESA VL-Bus Mode Pin Name	Local Bus Mode Pin Name
1	NC	NC	41	NC	NC
2	LED2/SRDCLK	LED2/SRDCLK	42	RDYRTN	RDYRTN
3	DVSSN1	DVSSN1	43	LRDY	RDY
4	ADR17	A17	44	W/R	W/R
5	ADR16	A16	45	ADS	ADS
6	ADR15	A15	46	M/IO	M/IO
7	ADR14	A14	47	BE3	BE3
8	DVDDO1	DVDDO1	48	D/C	D/C
9	ADR13	A13	49	RESET	RESET
10	ADR12	A12	50	BE2	BE2
11	DVSSN2	DVSSN2	51	DVSSN5	DVSSN5
12	ADR11	A11	52	BE1	BE1
13	ADR10	A10	53	BE0	BE0
14	ADR9	A9	54	WBACK	BOFF
15	ADR8	A8	55	DAT31	D31
16	ADR7	A7	56	DAT30	D30
17	DVSSN3	DVSSN3	57	DAT29	D29
18	ADR6	A6	58	DAT28	D28
19	ADR5	A5	59	DVSSN6	DVSSN6
20	DVSSPAD	DVSSPAD	60	DAT27	D27
21	ADR4	A4	61	DAT26	D26
22	ADR3	A3	62	DAT25	D25
23	DVSSN4	DVSSN4	63	DAT24	D24
24	ADR2	A2	64	DAT23	D23
25	LBS16	AHOLD	65	DVDDO3	DVDDO3
26	DVDD1	DVDD1	66	DAT22	D22
27	LEADS	EADS	67	DVSSN7	DVSSN7
28	DVSSCLK	DVSSCLK	68	DAT21	D21
29	LCLK	BCLK	69	DVSS2	DVSS2
30	DVDDCLK	DVDDCLK	70	DAT20	D20
31	VLBEN	Am486	71	DAT19	D19
32	BLAST	BLAST	72	DAT18	D18
33	LGNT	HLDA	73	DAT17	D17
34	BRDY	BRDY	74	DVSSN8	DVSSN8
35	DVDDO2	DVDDO2	75	DAT16	D16
36	LREQ	HOLD	76	DAT15	D15
37	DVSS1	DVSS1	77	DAT14	D14
38	LDEV	LDEV	78	DAT13	D13
39	NC	NC	79	NC	NC
40	NC	NC	80	NC	NC

VESA VL-BUS/LOCAL BUS PIN CROSS-REFERENCE (continued)
Listed by Pin Number

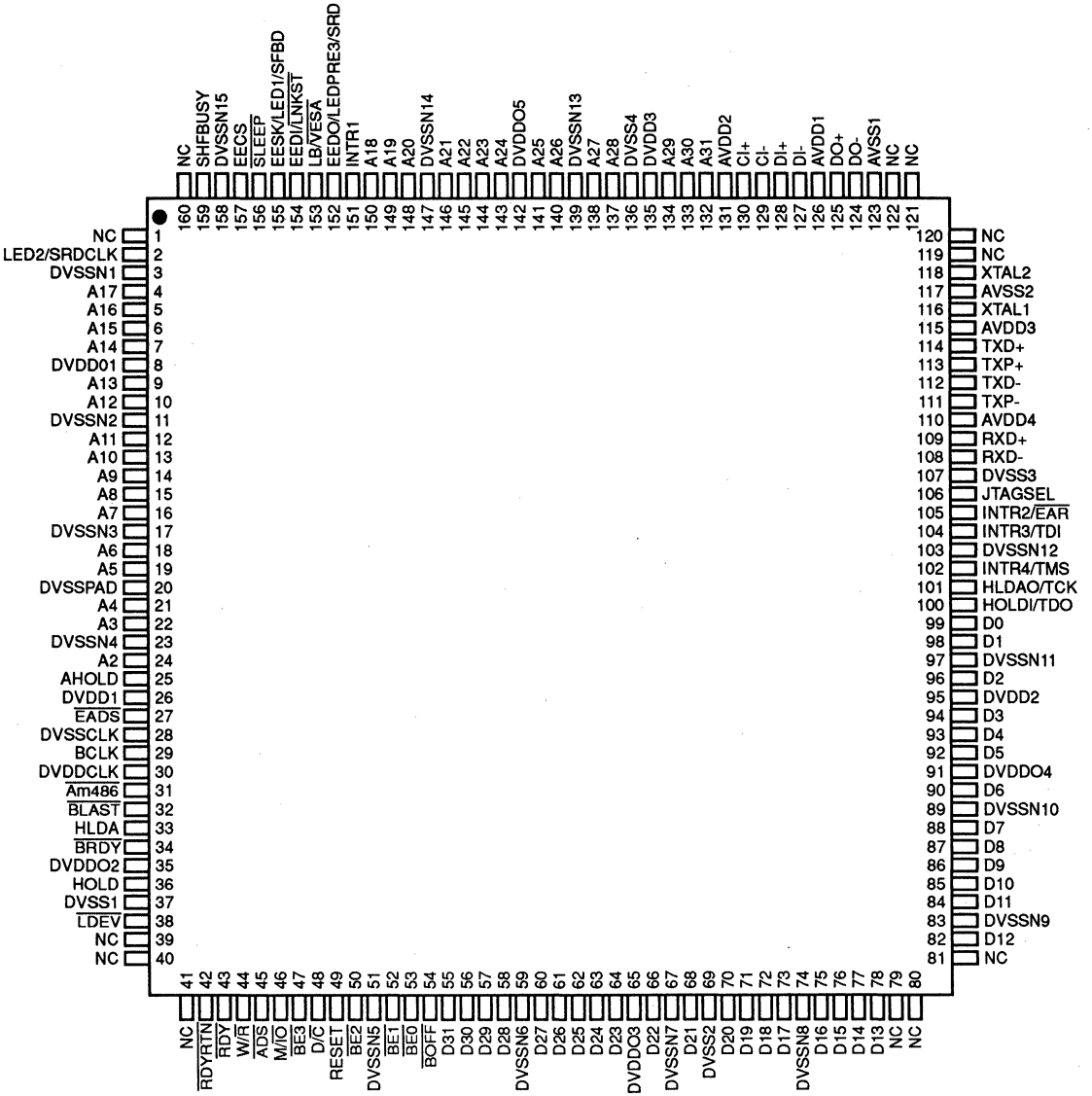
Pin #	VESA VL-Bus Mode Pin Name	Local Bus Mode Pin Name	Pin #	VESA VL-Bus Mode Pin Name	Local Bus Mode Pin Name
81	NC	NC	121	NC	NC
82	DAT12	D12	122	NC	NC
83	DVSSN9	DVSSN9	123	AVSS1	AVSS1
84	DAT11	D11	124	DO-	DO-
85	DAT10	D10	125	DO+	DO+
86	DAT9	D9	126	AVDD1	AVDD1
87	DAT8	D8	127	DI-	DI-
88	DAT7	D7	128	DI+	DI+
89	DVSSN10	DVSSN10	129	CI-	CI-
90	DAT6	D6	130	CI+	CI+
91	DVDDO4	DVDDO4	131	AVDD2	AVDD2
92	DAT5	D5	132	ADR31	A31
93	DAT4	D4	133	ADR30	A30
94	DAT3	D3	134	ADR29	A29
95	DVDD2	DVDD2	135	DVDD3	DVDD3
96	DAT2	D2	136	DVSS4	DVSS4
97	DVSSN11	DVSSN11	137	ADR28	A28
98	DAT1	D1	138	ADR27	A27
99	DAT0	D0	139	DVSSN13	DVSSN13
100	$\overline{\text{LREQI}}$/TDO	HOLDI/TDO	140	ADR26	A26
101	$\overline{\text{LGNTO}}$/TCK	$\overline{\text{HLDAO}}$/TCK	141	ADR25	A25
102	INTR4/TMS	INTR4/TMS	142	DVDDO5	DVDDO5
103	DVSSN12	DVSSN12	143	ADR24	A24
104	INTR3/TDI	INTR3/TDI	144	ADR23	A23
105	INTR2/$\overline{\text{EAR}}$	INTR2/$\overline{\text{EAR}}$	145	ADR22	A22
106	JTAGSEL	JTAGSEL	146	ADR21	A21
107	DVSS3	DVSS3	147	DVSSN14	DVSSN14
108	RXD-	RXD-	148	ADR20	A20
109	RXD+	RXD+	149	ADR19	A19
110	AVDD4	AVDD4	150	ADR18	A18
111	TXP-	TXP-	151	INTR1	INTR1
112	TXD-	TXD-	152	EEDO/LEDPRE3/SRD	EEDO/LEDPRE3/SRD
113	TXP+	TXP+	153	LB/$\overline{\text{VESA}}$	LB/$\overline{\text{VESA}}$
114	TXD+	TXD+	154	EEDI/$\overline{\text{LNKST}}$	EEDI/$\overline{\text{LNKST}}$
115	AVDD3	AVDD3	155	EESK/LED1/SFBD	EESK/LED1/SFBD
116	XTAL1	XTAL1	156	$\overline{\text{SLEEP}}$	$\overline{\text{SLEEP}}$
117	AVSS2	AVSS2	157	EECS	EECS
118	XTAL2	XTAL2	158	DVSSN15	DVSSN15
119	NC	NC	159	SHFBUSY	SHFBUSY
120	NC	NC	160	NC	NC

BLOCK DIAGRAM: 486 LOCAL BUS MODE



18219B-3

CONNECTION DIAGRAM: 486 LOCAL BUS MODE



18219B-4

PIN DESIGNATIONS: 486 LOCAL BUS MODE
Listed by Pin Number

Pin #	Name	Pin #	Name	Pin #	Name	Pin #	Name
1	NC	41	NC	81	NC	121	NC
2	LED2/SRDCLK	42	$\overline{\text{RDYRTN}}$	82	D12	122	NC
3	DVSSN1	43	$\overline{\text{RDY}}$	83	DVSSN9	123	AVSS1
4	A17	44	W/$\overline{\text{R}}$	84	D11	124	DO-
5	A16	45	$\overline{\text{ADS}}$	85	D10	125	DO+
6	A15	46	M/$\overline{\text{IO}}$	86	D9	126	AVDD1
7	A14	47	$\overline{\text{BE3}}$	87	D8	127	DI-
8	DVDDO1	48	D/$\overline{\text{C}}$	88	D7	128	DI+
9	A13	49	RESET	89	DVSSN10	129	CI-
10	A12	50	$\overline{\text{BE2}}$	90	D6	130	CI+
11	DVSSN2	51	DVSSN5	91	DVDDO4	131	AVDD2
12	A11	52	$\overline{\text{BE1}}$	92	D5	132	A31
13	A10	53	$\overline{\text{BE0}}$	93	D4	133	A30
14	A9	54	$\overline{\text{BOFF}}$	94	D3	134	A29
15	A8	55	D31	95	DVDD2	135	DVDD3
16	A7	56	D30	96	D2	136	DVSS4
17	DVSSN3	57	D29	97	DVSSN11	137	A28
18	A6	58	D28	98	D1	138	A27
19	A5	59	DVSSN6	99	D0	139	DVSSN13
20	DVSSPAD	60	D27	100	HOLDI/TDO	140	A26
21	A4	61	D26	101	HLDAO/TCK	141	A25
22	A3	62	D25	102	INTR4/TMS	142	DVDDO5
23	DVSSN4	63	D24	103	DVSSN12	143	A24
24	A2	64	D23	104	INTR3/TDI	144	A23
25	AHOLD	65	DVDDO3	105	INTR2/$\overline{\text{EAR}}$	145	A22
26	DVDD1	66	D22	106	JTAGSEL	146	A21
27	$\overline{\text{EADS}}$	67	DVSSN7	107	DVSS3	147	DVSSN14
28	DVSSCLK	68	D21	108	RXD-	148	A20
29	BCLK	69	DVSS2	109	RXD+	149	A19
30	DVDDCLK	70	D20	110	AVDD4	150	A18
31	$\overline{\text{Am486}}$	71	D19	111	TXP-	151	INTR1
32	$\overline{\text{BLAST}}$	72	D18	112	TXD-	152	EEDO/LEDPRE3/SRD
33	HLDA	73	D17	113	TXP+	153	LB/$\overline{\text{VESA}}$
34	$\overline{\text{BRDY}}$	74	DVSSN8	114	TXD+	154	EEDI/$\overline{\text{LNKST}}$
35	DVDDO2	75	D16	115	AVDD3	155	EESK/LED1/SFBD
36	HOLD	76	D15	116	XTAL1	156	$\overline{\text{SLEEP}}$
37	DVSS1	77	D14	117	AVSS2	157	EECS
38	$\overline{\text{LDEV}}$	78	D13	118	XTAL2	158	DVSSN15
39	NC	79	NC	119	NC	159	SHFBUSY
40	NC	80	NC	120	NC	160	NC

PIN DESIGNATIONS: 486 LOCAL BUS MODE
Listed by Pin Name

Name	Pin #	Name	Pin #	Name	Pin #	Name	Pin #
A2	24	$\overline{BE0}$	53	D30	56	EESK/LED1/SFBD	155
A3	22	$\overline{BE1}$	52	D31	55	HLDA	33
A4	21	$\overline{BE2}$	50	DI+	128	HLDAO/TCK	101
A5	19	$\overline{BE3}$	47	DI-	127	HOLD	36
A6	18	$\overline{BLAST}$	32	DO+	125	HOLDI/TDO	100
A7	16	$\overline{BOFF}$	54	DO-	124	INTR1	151
A8	15	$\overline{BRDY}$	34	DVDD1	26	INTR2/$\overline{EAR}$	105
A9	14	CI+	130	DVDD2	95	INTR3/TDI	104
A10	13	CI-	129	DVDD3	135	INTR4/TMS	102
A11	12	D/$\overline{C}$	48	DVDDCLK	30	JTAGSEL	106
A12	10	D0	99	DVDDO1	8	LB/$\overline{VESA}$	153
A13	9	D1	98	DVDDO2	35	LDEV	38
A14	7	D2	96	DVDDO3	65	LED2/SRDCLK	2
A15	6	D3	94	DVDDO4	91	M/$\overline{IO}$	46
A16	5	D4	93	DVDDO5	142	NC	1
A17	4	D5	92	DVSS1	37	NC	39
A18	150	D6	90	DVSS2	69	NC	40
A19	149	D7	88	DVSS3	107	NC	41
A20	148	D8	87	DVSS4	136	NC	79
A21	146	D9	86	DVSSCLK	28	NC	80
A22	145	D10	85	DVSSN1	3	NC	81
A23	144	D11	84	DVSSN2	11	NC	119
A24	143	D12	82	DVSSN3	17	NC	120
A25	141	D13	78	DVSSN4	23	NC	121
A26	140	D14	77	DVSSN5	51	NC	122
A27	138	D15	76	DVSSN6	59	NC	160
A28	137	D16	75	DVSSN7	67	$\overline{RDY}$	43
A29	134	D17	73	DVSSN8	74	$\overline{RDYRTN}$	42
A30	133	D18	72	DVSSN9	83	RESET	49
A31	132	D19	71	DVSSN10	89	RXD+	109
ADS	45	D20	70	DVSSN11	97	RXD-	108
AHOLD	25	D21	68	DVSSN12	103	SHFBUSY	159
$\overline{Am486}$	31	D22	66	DVSSN13	139	$\overline{SLEEP}$	156
AVDD1	126	D23	64	DVSSN14	147	TXD+	114
AVDD2	131	D24	63	DVSSN15	158	TXD-	112
AVDD3	115	D25	62	DVSSPAD	20	TXP+	113
AVDD4	110	D26	61	$\overline{EADS}$	27	TXP-	111
AVSS1	123	D27	60	EECS	157	W/$\overline{R}$	44
AVSS2	117	D28	58	EEDI/$\overline{LNKST}$	154	XTAL1	116
BCLK	29	D29	57	EEDO/LEDPRE3/SRD	152	XTAL2	118

PIN DESIGNATIONS: 486 LOCAL BUS MODE
Listed by Group

Pin Name	Pin Function	Type	Driver	No. of Pins
486/386DX Local Bus Interface				
A2–A31	Address Bus	IO	TS	30
$\overline{ADS}$	Address Status	I/O	TS	1
AHOLD	Address Hold	I		1
$\overline{Am486}$	Am486 Mode Select	I		1
BCLK	Bus Clock	I		1
$\overline{BE0}$–$\overline{BE3}$	Byte Enable	I/O	TS	4
$\overline{BLAST}$	Burst Last	O	TS	1
$\overline{BOFF}$	Backoff	I		1
$\overline{BRDY}$	Burst Ready	I/O	TS	1
D/$\overline{C}$	Data/Control Select	I/O	TS	1
D0–D31	Data Bus	I/O	TS	32
$\overline{EADS}$	External Address Strobe	O	TS	1
HLDA	Hold Acknowledge	I		1
HLDAO	Hold Acknowledge Out	I/O	TS	1
HOLD	Hold Request	O	O8	1
HOLDI	Hold Request In	I/O	TS	1
INTR1	Interrupt Number 1	O	TS	1
INTR2	Interrupt Number 2	I/O	TS	1
INTR3	Interrupt Number 3	I/O	TS	1
INTR4	Interrupt Number 4	I/O	TS	1
JTAGSEL	JTAG Select	I		1
LB/$\overline{VESA}$	Local Bus/VESA VL-Bus Select pin	I		1
$\overline{LDEV}$	Local Device	O	O4	1
M/$\overline{IO}$	Memory/I/O Select	I/O	TS	1
$\overline{RDY}$	Ready	O	TS	1
$\overline{RDYRTN}$	Ready Return	I		1
W/$\overline{R}$	Write/Read Select	I/O	TS	1
Board Interface				
EECS	Microwire Serial PROM Chip Select	O	O8	1
EEDI/$\overline{LNKST}$	Microwire Serial EEPROM Data In/Link Status	O	LED	1
EEDO/LEDPRE3	Microwire Address PROM Data Out/LED3 predriver	I/O	LED	1
EESK/LED1	Microwire Serial PROM Clock/LED1	O	LED	1
LED2	LED output number 2	O	LED	1
RESET	Reset	I		1
SHFBUSY	Shift Busy (for external EEPROM-programmable logic)	O	O8	1
$\overline{SLEEP}$	Sleep Mode	I		1
XTAL1	Crystal Input	I		1
XTAL2	Crystal Output	O		1

PIN DESIGNATIONS: 486 LOCAL BUS MODE
Listed by Group

Pin Name	Pin Function	Type	Driver	No. of Pins
Attachment Unit Interface (AUI)				
CI+/CI-	AUI Collision Differential Pair	I		2
DI+/DI-	AUI Data In Differential Pair	I		2
DO+/DO-	AUI Data Out Differential Pair	O	DO	2
Twisted Pair Transceiver Interface (10BASE-T)				
RXD+/RXD-	Receive Differential Pair	I		2
TXD+/TXD-	Transmit Differential Pair	O	TDO	2
TXP+/TXP-	Transmit Pre-distortion Differential Pair	O	TPO	2
$\overline{\text{LNKST}}$/EEDI	Link Status/microwire Serial EEPROM Data In	O	LED	1
IEEE 1149.1 Test Access Port Interface (JTAG)				
TCK	Test Clock	I/O	TS	1
TDI	Test Data In	I/O	TS	1
TDO	Test Data Out	I/O	TS	1
TMS	Test Mode Select	I/O	TS	1
External Address Detection Interface (EADI)				
$\overline{\text{EAR}}$	External Address Reject Low	I/O	TS	1
SRD	Serial Receive Data	I/O	TS	1
SRDCLK	Serial Receive Data Clock	I/O	TS	1
SFBD	Start Frame - Byte Delimiter	O	LED	1
Power Supplies				
AVDD	Analog Power	P		4
AVSS	Analog Ground	P		2
DVDD	Digital Power	P		3
DVDDCLK	Digital Power Clock	P		1
DVDDO	I/O Buffer Digital Power	P		5
DVSS	Digital Ground	P		4
DVSSCLK	Digital Ground Clock	P		1
DVSSN	I/O Buffer Digital Ground	P		15
DVSSPAD	Digital Ground Pad	P		1

PIN DESIGNATIONS: 486 LOCAL BUS MODE

Driver Type

Table 9. Output Driver Types

Name	Type	I_{OL} (mA)	I_{OH} (mA)	pF
TS	Tri-State	8	–0.4	50
O8	Totem Pole	8	–0.4	50
O4	Totem Pole	4	–0.4	50
OD	Open Drain	8		50
LED	LED	12	–0.4	50

Table 10. Pins with Pull-Ups

Signal	Pull-up
TDI	≥ 10 KΩ
TMS	≥ 10 KΩ
TCK	≥ 10 KΩ

PIN DESCRIPTION: 486 LOCAL BUS MODE

Configuration Pins

JTAGSEL

JTAG Function Select ***Input***

The value of this pin will asynchronously select between JTAG Mode and Multi-Interrupt Mode.

The value of this pin will asynchronously affect the function of the JTAG–INTR–Daisy chain arbitration pins, regardless of the state of the RESET pin and regardless of the state of the BCLK pin. If the value is a "1", then the PCnet-32 controller will be programmed for JTAG mode. If the value is a "0", then the PCnet-32 controller will be programmed for Multi-Interrupt Mode.

When programmed for JTAG mode, four pins of the PCnet-32 controller will be configured as a JTAG (IEEE 1149.1) Test Access Port. When programmed for Multi-Interrupt Mode, two of the JTAG pins will become interrupts and two JTAG pins will be used for daisy chain arbitration support. Table 11 below outlines the pin changes that will occur by programming the JTAGSEL pin.

Table 11. JTAG Pin Changes

Pin	JTAGSEL=1 JTAG mode	JTAGSEL=0 Multi-Interrupt Mode
HLDAO/TCK	TCK	HLDAO
HOLDI/TDO	TDO	HOLDI
INTR3/TDI	TDI	INTR3
INTR4/TMS	TMS	INTR4

The JTAGSEL pin may be tied directly to V_{DD} or V_{SS}. A series resistor may be used but is not necessary.

LB/$\overline{VESA}$

Local Bus/VESA VL-Bus Select ***Input***

The value of this pin will asynchronously determine the operating mode of the PCnet-32 controller, regardless of the state of RESET and regardless of the state of the BCLK pin. If the LB/$\overline{VESA}$ pin is tied to V_{DD}, then the PCnet-32 controller will be programmed for Local Bus Mode. If the LB/$\overline{VESA}$ pin is tied to V_{SS}, then the PCnet-32 controller will be programmed for VESA VL-Bus Mode.

Note that the setting of LB/$\overline{VESA}$ determines the functionality of the following pins (names in parentheses are pins in the VESA VL-Bus Mode): $\overline{Am486}$ (VLBEN), RESET ($\overline{RESET}$), AHOLD ($\overline{LBS16}$), HOLD ($\overline{LREQ}$), HLDA($\overline{LGNT}$), HOLDI ($\overline{LREQI}$), and HLDAO ($\overline{LGNTO}$).

$\overline{Am486}$

Am486 Mode Select ***Input***

The $\overline{Am486}$ pin should be tied directly to V_{SS}. A series resistor may be used but is not necessary.

Note: *This pin is used to enable bursing in the VESA VL-Bus mode when the LB/$\overline{VESA}$ pin has been tied to V_{SS}. See the pin description for the VLBEN pin in the VESA VL-Bus Mode section.*

Configuration Pin Settings Summary

Table 12 shows the possible pin configurations that may be invoked with the PCnet-32 controller configuration pins.

Table 12. Configuration Pin Settings

LB/$\overline{VESA}$	$\overline{Am486}$/Am386	JTAGSEL	Mode Selected
0	X*	0	VL Bus mode with 4 interrupts and daisy chain arbitration
0	X*	1	VL Bus mode with 2 interrupts and JTAG
1	0	0	Am486 mode with 4 interrupts and daisy chain arbitration
1	0	1	Am486 mode with 2 interrupts and JTAG
1	1	X	Reserved

**X = Don't care*

Table 13. Pin Connections to Power/Ground

Pin Name	Pin No	Supply Strapping	Resistive Connection to Supply	Recommended Resistor Size
LED2/SRDCLK	2	Required	Required	324 Ω in series with LED, or 10 KΩ without LED
AHOLD	25	Optional	Required	10 KΩ
$\overline{Am486}$	31	Required	Optional	NA
$\overline{BOFF}$	54	Optional	Required	10 KΩ
HOLDI/TDO	100	Optional	Required	10 KΩ
JTAGSEL	106	Required	Optional	NA
EEDO/LEDPRE3/SRD	152	Optional	Required	10 KΩ
LB/$\overline{VESA}$	153	Required	Optional	NA
EEDI/$\overline{LNKST}$	154	Optional	Required	324 Ω in series with LED, or 10 KΩ without LED
EESK/LED1/SFBD	155	Required	Required	324 Ω in series with LED, or 10 KΩ without LED
$\overline{SLEEP}$	156	Optional	Required	10 KΩ
All Other Pins	—	Optional	Required	10 KΩ

Pin Connections to V_{DD} or V_{SS}

Several pins may be connected to V_{DD} or V_{SS} for various application options. Some pins are required to be connected to V_{DD} or V_{SS} in order to set the controller into a particular mode of operation, while other pins might be connected to V_{DD} or V_{SS} if that pin's function is not implemented in a specific application. Table 13 shows which pins *require* a connection to V_{DD} or V_{SS}, and which pins may *optionally* be connected to V_{DD} or V_{SS} because the application does not support that pin's function. The table also shows whether or not the connections need to be resistive.

Local Bus Interface

A2–A31

Address Bus ***Input/Output***

Address information which is stable during a bus operation, regardless of the source. When the PCnet-32 controller is Current Master, A1–A31 will be driven. When the PCnet-32 controller is not Current Master, the A2–A31 lines are continuously monitored to determine if an address match exists for I/O slave transfers.

Some portion of the Address Bus will be floated at the time of an address hold operation, which is signaled with the AHOLD pin. The number of Address Bus pins to be floated will be determined by the value of the Cache Line Length register (BCR18, bits 15-11).

$\overline{ADS}$

Address Status ***Input/Output***

When driven LOW, this signal indicates that a valid bus cycle definition and address are available on the M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$ and A2–A31 pins of the local bus interface. At that time, the PCnet-32 controller will examine the combination of M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$, and the A2–A31 pins to determine if the current access is directed toward the PCnet-32 controller.

$\overline{ADS}$ will be driven LOW when the PCnet-32 controller performs a bus master access on the local bus.

AHOLD

Address Hold ***Input***

This pin is always an input. The PCnet-32 controller will put some portion of the address bus into a high impedance state whenever this signal is asserted. AHOLD may be asserted by an external cache controller when a cache invalidation cycle is being performed. AHOLD may be asserted at any time, including times when the PCnet-32 controller is the active bus master. Note that this pin is multiplexed with a VESA VL function: $\overline{LBS16}$.

Some portion of the Address Bus will be floated at the time of an address hold operation, which is signaled with the AHOLD pin. The number of Address Bus pins to be floated will be determined by the value of the Cache Line Length (CLL) register (BCR18, bits 15-11) as shown in Table 14.

Table 14. CLL Value and Floating Address Pins

CLL Value	Floated Portion of Address Bus During AHOLD
00000	None
00001	A31–A2
00010	A31–A3
00011	Reserved CLL Value
00100	A31–A4
00101–00111	Reserved CLL Values
01000	A31–A5
01001–01111	Reserved CLL Values
10000	A31–A6
10001–11111	Reserved CLL Values

BCLK

Bus Clock *Input*

Clock input that provides timing edges for all interface signals. This clock is used to drive the system bus interface and the internal buffer management unit. This clock is NOT used to drive the network functions.

$\overline{BE0}$–$\overline{BE3}$

Byte Enable *Input/Output*

These signals indicate which bytes on the data bus are active during read and write cycles. When $\overline{BE3}$ is active, the byte on D31–D24 is valid. $\overline{BE2}$–$\overline{BE0}$ active indicate valid data on pins D23–D16, D15–D8, D7–D0, respectively. The byte enable signals are outputs for bus master and inputs for bus slave operations.

$\overline{BLAST}$

Burst Last *Output*

When the $\overline{BLAST}$ signal is asserted, then the next time that $\overline{BRDY}$ or $\overline{RDYRTN}$ is asserted, the burst cycle is complete.

$\overline{BOFF}$

Backoff *Input*

$\overline{BOFF}$ is monitored as an input during Bus Master accesses. When PCnet-32 controller is current local bus master, it will float all appropriate bus mastering signals within 1 clock period of the assertion of $\overline{BOFF}$. When $\overline{BOFF}$ is deasserted, PCnet-32 controller will restart any accesses that were suspended due to the assertion of $\overline{BOFF}$ and then will proceed with other scheduled accesses, if any. Register access cannot be performed to the PCnet-32 device while $\overline{BOFF}$ is asserted.

$\overline{BRDY}$

Burst Ready *Input/Output*

$\overline{BRDY}$ functions as an input to the PCnet-32 controller during bus master cycles. When $\overline{BRDY}$ is asserted during a master cycle, it indicates to the PCnet-32 controller that the target device is accepting burst transfers. It also serves the same function as $\overline{RDYRTN}$ does for non-burst accesses. That is, it indicates that the target device has accepted the data on a master write cycle, or that the target device has presented valid data onto the bus during master read cycles.

If $\overline{BRDY}$ and $\overline{RDYRTN}$ are sampled active in the same cycle, then $\overline{RDYRTN}$ takes precedence, causing the next transfer cycle to begin with a T1 cycle.

$\overline{BRDY}$ functions as an output during PCnet-32 controller slave cycles and is always driven inactive (HIGH).

$\overline{BRDY}$ is floated if the PCnet-32 controller is not being accessed as the current slave device on the local bus.

D/$\overline{C}$

Data/Control Select *Input/Output*

During slave accesses to the PCnet-32 controller, the D/$\overline{C}$ pin, along with M/$\overline{IO}$ and W/$\overline{R}$, indicates the type of cycle that is being performed. PCnet-32 controller will only respond to local bus accesses in which D/$\overline{C}$ is driven HIGH by the local bus master.

During PCnet-32 controller bus master accesses, the D/$\overline{C}$ pin is an output and will always be driven HIGH.

D/$\overline{C}$ is floated if the PCnet-32 controller is not the current master on the local bus.

D0–D31

Data Bus *Input/Output*

Used to transfer data to and from the PCnet-32 controller to system resources via the local bus. D31–D0 are driven by the PCnet-32 controller when performing bus master writes and slave read operations. Data on D31–D0 is latched by the PCnet-32 controller when performing bus master reads and slave write operations.

The PCnet-32 controller will always follow Am386DX byte lane conventions. This means that for word and byte accesses in which PCnet-32 controller drives the data bus (i.e. master write operations and slave read operations) the PCnet-32 controller will produce duplicates of the active bytes on the unused half of the 32-bit data bus. Table 15 illustrates the cases in which duplicate bytes are created.

Table 15. Byte Duplication on Data Bus

$\overline{BE3}$–$\overline{BE0}$	DAT [31:24]	DAT [23:16]	DAT [15:8]	DAT [7:0]
1110	Undef	Undef	Undef	A
1101	Undef	Undef	A	Undef
1011	Undef	A	Undef	Copy A
0111	A	Undef	Copy A	Undef
1100	Undef	Undef	B	A
1001	Undef	C	B	Undef
0011	D	C	Copy D	Copy C
1000	Undef	C	B	A
0001	D	C	B	Undef
0000	D	C	B	A

$\overline{EADS}$

External Address Strobe *Output*

During master write accesses in which Generate Cache Invalidation Cycles mode has been selected, the $\overline{EADS}$ pin will be asserted as part of the PCnet-32 controller cache invalidation cycle. Cache invalidation cycles will occur as often as a new cache line is reached. The cache line size can be set with the cache line length bits of BCR18 (bits [15:11]).

HLDA

Bus Hold Acknowledge *Input*

PCnet-32 controller examines the HLDA signal to determine when it has been granted ownership of the bus. HLDA is active HIGH.

When HLDA is asserted and HOLD is being asserted by the PCnet-32 controller, the PCnet-32 controller assumes ownership of the local bus. However, if the PCnet-32 controller is asserting HOLD because HOLDI is asserted (as in a daisy chain arbitration), then PCnet-32 controller will assert HLDAO and will not assume ownership of the local bus.

Note that it changes polarity when the VL mode is selected (see pin description of $\overline{LGNT}$ in VESA VL-Bus Interface section).

HLDAO

Bus Hold Acknowledge Out *Output*

This signal is multiplexed with the TCK pin, and is available only when the Multi-Interrupt mode has been selected with the JTAGSEL pin.

An additional local bus master may daisy-chain its HLDA signal through the PCnet-32 controller HLDAO pin. The PCnet-32 controller will deliver a HLDAO signal to the additional local bus master whenever the PCnet-32 controller receives a HLDA from the CPU, but is not simultaneously requesting the bus internally. The second local bus master must connect its HOLD output to the HOLDI input of the PCnet-32 controller in order to complete the local bus daisy-chain arbitration control.

When $\overline{SLEEP}$ is not asserted, daisy chain arbitration signals that pass through the PCnet-32 controller will experience a one-clock delay from input to output (i.e. HOLDI to HOLD and HLDA to HLDAO).

While $\overline{SLEEP}$ is asserted (either in **snooze** mode or **coma** mode), if the PCnet-32 controller is configured for a daisy chain (HOLDI and HLDAO signals have been selected with the JTAGSEL pin), then the system arbitration signal HLDA will be passed directly to the daisy-chain signal HLDAO without experiencing a one-clock delay. However, some combinatorial delay will be introduced in this path.

Note that this pin changes polarity when VL mode has been selected (see pin description of $\overline{LGNTO}$ in VESA VL-Bus Interface section).

HOLD

Bus Hold Request *Output*

PCnet-32 controller asserts the HOLD pin as a signal that it wishes to become the local bus master. HOLD is active high. Once asserted, HOLD remains active until HLDA has become active, independent of subsequent assertion of $\overline{SLEEP}$ or setting of the STOP bit or access to the S_RESET port (offset 14h).

Note that this pin changes polarity when the VL mode is selected (see pin description of $\overline{LREQ}$ in VESA VL-Bus Interface section).

HOLDI

Bus Hold Request In *Input*

This signal is multiplexed with the TDO pin, and is available only when the Multi-Interrupt mode has been selected with the JTAGSEL pin.

An additional local bus master may daisy-chain its bus hold request signal through the PCnet-32 controller HOLDI pin. The PCnet-32 controller will convey the HOLDI request to the CPU via the PCnet-32 controller HOLD output. The second local bus master must connect its HLDA input to the HLDAO output of the PCnet-32 controller in order to complete the local bus daisy-chain arbitration control.

When $\overline{SLEEP}$ is not asserted, daisy chain arbitration signals that pass through the PCnet-32 controller will experience a one-clock delay from input to output (i.e. HOLDI to HOLD and HLDA to HLDAO).

While $\overline{SLEEP}$ is asserted (either in **snooze** mode or **coma** mode), if the PCnet-32 controller is configured for a daisy chain (HOLDI and HLDAO signals have been selected with the JTAGSEL pin), then the daisy-chain signal HOLDI will be passed directly to the system arbitration signal HOLD without experiencing a one-clock delay. However, some combinatorial delay will be introduced in this path.

If Multi-Interrupt mode has been selected and the daisy-chain arbitration feature is not used, then the HOLDI input should be tied to VSS through a resistor.

Note that this pin changes polarity when VL mode has been selected (see pin description of $\overline{\text{LREQI}}$ in VESA VL-Bus Interface section).

INTR1–INTR4

Interrupt Request *Output*

An attention signal which indicates that one or more of the following status flags is set: BABL, MISS, MERR, RINT, IDON, MFCO, RCVCCO, TXSTRT, or JAB. Each of these status flags has a mask bit which allows for suppression of INTR assertion. These flags have the meaning shown in Table 16.

Table 16. Status Flags

BABL	Babble (CSR0, bit 14)
MISS	Missed Frame (CSR0, bit 12)
MERR	Memory Error (CSR0, bit 11)
RINT	Receive Interrupt (CSR0, bit 10)
IDON	Initialization Done (CSR0, bit 8)
MFCO	Missed Packet Count Overflow (CSR4, bit 9)
RCVCCO	Receive Collision Count Overflow (CSR4, bit 5)
TXSTRT	Transmit Start (CSR4, bit 3)
JAB	Jabber (CSR4, bit 1)

Note that there are four possible interrupt pins, depending upon the mode that has been selected with the JTAGSEL pin. Only one interrupt pin may be used at one time. The active interrupt pin is selected by programming the interrupt select register (BCR21). The default setting of BCR121will select interrupt INTR1 as the active interrupt. Note that BCR21 is EEPROM-programmable. Inactive interrupt pins are floated.

The polarity of the interrupt signal is determined by the INTLEVEL bit of BCR2. The interrupt pins may be programmed for level-sensitive or edge-sensitive operation.

PCnet-32 controller interrupt pins will be floated at H_RESET and will remain floated until either the EEPROM has been successfully read, or, following an EEPROM read failure, a Software Relocatable Mode sequence has been successfully executed.

$\overline{\text{LDEV}}$

Local Device *Output*

$\overline{\text{LDEV}}$ is driven by the PCnet-32 controller when it recognizes an access to PCnet-32 controller I/O space. Such recognition is dependent upon a valid sampled $\overline{\text{ADS}}$ strobe plus valid M/$\overline{\text{IO}}$, D/$\overline{\text{C}}$ and A31–A5 values.

M/$\overline{\text{IO}}$

Memory I/O Select *Input/Output*

During slave accesses to the PCnet-32 controller, the M/$\overline{\text{IO}}$ pin, along with D/$\overline{\text{C}}$ and W/$\overline{\text{R}}$, indicates the type of cycle that is being performed. PCnet-32 controller will only respond to local bus accesses in which M/$\overline{\text{IO}}$ is sampled as a zero by the PCnet-32 controller.

During PCnet-32 controller bus master accesses, the M/$\overline{\text{IO}}$ pin is an output and will always be driven high.

M/$\overline{\text{IO}}$ is floated if the PCnet-32 controller is not the current master on the local bus.

RESET

System Reset *Input*

When RESET is asserted high and the LB/$\overline{\text{VESA}}$ pin has been tied to VDD, then the PCnet-32 controller performs an internal system reset of the type H_RESET (HARDWARE_RESET). The RESET pin must be held for a minimum of 30 BCLK periods. While in the H_RESET state, the PCnet-32 controller will float or deassert all outputs.

Note that this pin changes polarity when VL mode has been selected (see pin description of $\overline{\text{RESET}}$ in *VESA VL-Bus Interface* section).

$\overline{\text{RDY}}$

Ready *Output*

$\overline{\text{RDY}}$ functions as an output from the PCnet-32 controller during PCnet-32 controller slave cycles. During PCnet-32 controller slave read cycles, $\overline{\text{RDY}}$ is asserted to indicate that valid data has been presented on the data bus. During PCnet-32 controller slave write cycles, $\overline{\text{RDY}}$ is asserted to indicate that the data on the data bus has been internally latched. $\overline{\text{RDY}}$ is asserted for one BCLK period. $\overline{\text{RDY}}$ is then driven high for one-half of one clock period before being released.

$\overline{\text{RDY}}$ is floated if the PCnet-32 controller is not the current slave on the local bus.

In systems where both $\overline{\text{RDY}}$ and $\overline{\text{RDYRTN}}$ (or equivalent) signals are provided, $\overline{\text{RDY}}$ must NOT be tied to $\overline{\text{RDYRTN}}$. Most systems now provide for a local device ready input to the memory controller that is separate from the CPU $\overline{\text{READY}}$ signal. This second $\overline{\text{READY}}$ signal is usually labeled as $\overline{\text{READYIN}}$. This signal should be connected to the PCnet-32 controller $\overline{\text{RDY}}$ signal. The CPU $\overline{\text{READY}}$ signal should be connected to the PCnet-32 controller $\overline{\text{RDYRTN}}$ pin.

In systems where only one $\overline{\text{READY}}$ signal is provided, then $\overline{\text{RDY}}$ may be tied to $\overline{\text{RDYRTN}}$.

$\overline{\text{RDYRTN}}$

Ready Return *Input*

$\overline{\text{RDYRTN}}$ functions as an input to the PCnet-32 controller. $\overline{\text{RDYRTN}}$ is used to terminate all master accesses

performed by the PCnet-32 controller, except that linear burst transfers may also be terminated with the $\overline{BRDY}$ signal. $\overline{RDYRTN}$ is used to terminate slave read accesses to PCnet-32 controller I/O space.

When asserted during slave read accesses to PCnet-32 controller I/O space, $\overline{RDYRTN}$ indicates that the bus mastering device has seen the $\overline{RDY}$ that was generated by the PCnet-32 controller and has accepted the PCnet-32 controller slave read data. Therefore, PCnet-32 controller will hold slave read data on the bus until it synchronously samples the $\overline{RDYRTN}$ input as active low. The PCnet-32 controller will *not* hold $\overline{RDY}$ valid asserted during this time. The duration of the $\overline{RDY}$ pulse generated by the PCnet-32 controller will always be a single BCLK cycle.

$\overline{RDYRTN}$ is ignored during slave write accesses to PCnet-32 controller I/O space. Slave write accesses to PCnet-32 controller I/O space are considered terminated by the PCnet-32 controller at the end of the cycle during which the PCnet-32 controller issues an active $\overline{RDY}$.

In systems where both a $\overline{RDY}$ and $\overline{RDYRTN}$ (or equivalent) signals are provided, then $\overline{RDY}$ must *not* be tied to $\overline{RDYRTN}$. Most systems now provide for a local device ready input to the memory controller that is separate from the CPU $\overline{READY}$ signal. This second $\overline{READY}$ signal is usually labeled as $\overline{READYIN}$. This signal should be connected to the PCnet-32 controller $\overline{RDY}$ signal. The CPU $\overline{READY}$ signal should be connected to the PCnet-32 controller $\overline{RDYRTN}$ pin.

In systems where only one $\overline{READY}$ signal is provided, then the PCnet-32 controller $\overline{RDY}$ output may be tied to the PCnet-32 controller $\overline{RDYRTN}$ input.

W/$\overline{R}$

Write/Read Select ***Input/Output***

During slave accesses to the PCnet-32 controller, the W/$\overline{R}$ pin, along with D/$\overline{C}$ and M/$\overline{IO}$, indicates the type of cycle that is being performed.

During PCnet-32 controller bus master accesses, the W/$\overline{R}$ pin is an output.

W/$\overline{R}$ is floated if the PCnet-32 controller is not the current master on the local bus.

Board Interface

LED1

LED1 ***Output***

This pin is shared with the EESK function. When operating as LED1, the function and polarity on this pin are programmable through BCR5. The LED1 output from the PCnet-32 controller is capable of sinking the necessary 12 mA of current to drive an LED directly.

The LED1 pin is also used during EEPROM Auto-detection to determine whether or not an EEPROM is present at the PCnet-32 controller microwire interface. At the trailing edge of RESET, this pin is sampled to determine the value of the EEDET bit in BCR19. A sampled HIGH value means that an EEPROM is present, and EEDET will be set to ONE. A sampled LOW value means that an EEPROM is not present, and EEDET will be set to ZERO. See the "EEPROM Auto-detection" section for more details.

If no LED circuit is to be attached to this pin, then a pull-up or pull-down resistor must be attached instead, in order to resolve the EEDET setting.

LED2

LED2 ***Output***

This pin is shared with the SRDCLK function. When operating as LED2, the function and polarity on this pin are programmable through BCR6. The LED2 output from the PCnet-32 controller is capable of sinking the necessary 12 mA of current to drive an LED directly.

This pin also selects address width for Software Relocatable Mode. When this pin is HIGH during Software Relocatable Mode, then the device will be programmed to use 32 bits of addressing while snooping accesses on the bus during Software Relocatable Mode. When this pin is LOW during Software Relocatable Mode, then the device will be programmed to use 24 bits of addressing while snooping accesses on the bus during Software Relocatable Mode. The upper 8 bits of address will be assumed to match during the snooping operation when LED2 is LOW. The 24-bit addressing mode is intended for use in systems that employ the GPSI signals. For more information on the GPSI function see section *General Purpose Serial Interface*.

If no LED circuit is to be attached to this pin, then a pull up or pull down resistor must be attached instead, in order to resolve the Software Relocatable Mode address setting.

LEDPRE3

LEDPRE3 ***Output***

This pin is shared with the EEDO function. When operating as LEDPRE3, the function and polarity on this pin are programmable through BCR7. This signal is labeled as LED "*PRE*"3 because of the multi-function nature of this pin. If an LED circuit were directly attached to this pin, it would create an I_{OL} requirement that could not be met by the serial EEPROM that would also be attached to this pin. Therefore, if this pin is to be used as an additional LED output while an EEPROM is used in the system, then buffering is required between the LEDPRE3 pin and the LED circuit. If no EEPROM is included in the system design, then the LEDPRE3 signal may be directly connected to an LED without buffering. The LEDPRE3 output from the PCnet-32 controller is capable of sinking the necessary 12 mA of current to drive an LED in this case. For more details regarding LED connection, see the section on LEDs.

LNKST

LINK Status *Output*

This pin provides 12 mA for driving an LED. It indicates an active link connection on the 10BASE-T interface. The signal is programmable through BCR4. Note that this pin is multiplexed with the EEDI function.

This pin remains active in snooze mode.

SHFBUSY

Shift Busy *Output*

The function of the SHFBUSY signal is to indicate when the last byte of the EEPROM contents has been shifted out of the EEPROM on the EEDO signal line. This information is useful for *external EEPROM-programmable registers* that do not use the microwire protocol, as is described herein: When the PCnet-32 controller is performing a serial read of the EEPROM through the microwire interface, the SHFBUSY signal will be driven HIGH. SHFBUSY can serve as a serial shift enable to allow the EEPROM data to be serially shifted into an external device or series of devices. The SHFBUSY signal will remain actively driven HIGH until the end of the EEPROM read operation. If the EEPROM checksum was verified, then the SHFBUSY signal will be driven LOW at the end of the EEPROM read operation. If the EEPROM checksum verification failed, then the SHFBUSY signal will remain HIGH. This function effectively demarcates the end of a successful EEPROM read operation and therefore is useful as a programmable-logic *low-active output enable* signal. For more details on external EEPROM-programmable registers, see the EEPROM. *Microwire Access section* under *Hardware Access.*

This pin can be controlled by the host system by writing to BCR19, bit 3 (EBUSY).

SLEEP

Sleep *Input*

When $\overline{\text{SLEEP}}$ input is asserted (active LOW), the PCnet-32 controller performs an internal system reset and then proceeds into a power savings mode. (The reset operation caused by $\overline{\text{SLEEP}}$ assertion will not affect BCR registers.) All outputs will be placed in their normal reset condition. During sleep mode, all PCnet-32 controller inputs will be ignored except for the $\overline{\text{SLEEP}}$ pin itself. De-assertion of $\overline{\text{SLEEP}}$ results in wake-up. The system must refrain from starting the network operations of the PCnet-32 controller for 0.5 seconds following the deassertion of the $\overline{\text{SLEEP}}$ signal in order to allow internal analog circuits to stabilize.

Both BCLK and XTAL1 inputs must have valid clock signals present in order for the $\overline{\text{SLEEP}}$ command to take effect.

If $\overline{\text{SLEEP}}$ is asserted while $\overline{\text{LREQ}}$/HOLD is asserted, then the PCnet-32 controller will perform an internal system reset and then wait for the assertion of $\overline{\text{LGNT}}$/HLDA. When $\overline{\text{LGNT}}$/HLDA is asserted, the $\overline{\text{LREQ}}$/HOLD signal will be deasserted and then the PCnet-32 controller will proceed to the power savings mode. Note that the internal system reset will not cause the HOLD/$\overline{\text{LREQ}}$ signal to be deasserted.

The $\overline{\text{SLEEP}}$ pin should not be asserted during power supply ramp-up. If it is desired that $\overline{\text{SLEEP}}$ be asserted at power up time, then the system must delay the assertion of $\overline{\text{SLEEP}}$ until three BCLK cycles after the completion of a valid pin $\overline{\text{RESET}}$ operation.

XTAL1–XTAL2

Crystal Oscillator Inputs *Input/Output*

The crystal frequency determines the network data rate. The PCnet-32 controller supports the use of quartz crystals to generate a 20 MHz frequency compatible with the ISO 8802-3 (IEEE/ANSI 802.3) network frequency tolerance and jitter specifications. See the section *External Crystal Characteristics* (in section Manchester Encoder/Decoder) for more detail.

The network data rate is one-half of the crystal frequency. XTAL1 may alternatively be driven using an external CMOS level source, in which case XTAL2 must be left unconnected. Note that when the PCnet-32 controller is in coma mode, there is an internal 22 KΩ resistor from XTAL1 to ground. If an external source drives XTAL1, some power will be consumed driving this resistor. If XTAL1 is driven LOW at this time power consumption will be minimized. In this case, XTAL1 must remain active for at least 30 cycles after the assertion of $\overline{\text{SLEEP}}$ and deassertion of HOLD.

Microwire EEPROM Interface

EESK

EEPROM Serial Clock *Output*

The EESK signal is used to access the external ISO 8802-3 (IEEE/ANSI 802.3) address PROM. This pin is designed to directly interface to a serial EEPROM that uses the microwire interface protocol. EESK is connected to the microwire EEPROM's Clock pin. It is controlled by either the PCnet-32 controller directly during a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 1.

EESK can be used during programming of *external EEPROM-programmable registers* that do not use the microwire protocol as follows:

When the PCnet-32 controller is performing a serial read of the IEEE Address EEPROM through the microwire interface, the SHFBUSY signal will serve as a serial shift enable to allow the EEPROM data to be serially shifted into an external device or series of devices. This same signal can be used to gate the *output* of the programmed logic to avoid the problem of releasing intermediate values to the rest of the system board logic. The EESK signal can serve as the clock, and EEDO will serve as the input data stream to the programmable shift register.

EEDO

EEPROM Data Out *Input*

The EEDO signal is used to access the external ISO 8802-3 (IEEE/ANSI 802.3) address PROM. This pin is designed to directly interface to a serial EEPROM that uses the microwire interface protocol. EEDO is connected to the microwire EEPROM's Data Output pin. It is controlled by the EEPROM during reads. It may be read by the host system by reading BCR19, bit 0.

EEDO can be used during programming of *external EEPROM-programmable registers* that do not use the microwire protocol as follows:

When the PCnet-32 controller is performing a serial read of the IEEE Address EEPROM through the microwire interface, the SHFBUSY signal will serve as a serial shift enable to allow the EEPROM data to be serially shifted into an external device or series of devices. This same signal can be used to gate the *output* of the programmed logic to avoid the problem of releasing intermediate values to the rest of the system board logic. The EESK signal can serve as the clock, and EEDO will serve as the input data stream to the programmable shift register.

EECS

EEPROM Chip Select *Output*

The function of the EECS signal is to indicate to the microwire EEPROM device that it is being accessed. The EECS signal is active high. It is controlled by either the PCnet-32 controller during a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 2.

EEDI

EEPROM Data In *Output*

The EEDI signal is used to access the external ISO 8802-3 (IEEE/ANSI 802.3) address PROM. EEDI functions as an output. This pin is designed to directly interface to a serial EEPROM that uses the microwire interface protocol. EEDI is connected to the microwire EEPROM's Data Input pin. It is controlled by either the PCnet-32 controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 0.

Attachment Unit Interface

CI±

Collision In *Input*

A differential input pair signaling the PCnet-32 controller that a collision has been detected on the network media, indicated by the CI± inputs being driven with a 10 MHz pattern of sufficient amplitude and pulse width to meet ISO 8802-3 (IEEE/ANSI 802.3) standards. Operates at pseudo ECL levels.

DI±

Data In *Input*

A differential input pair to the PCnet-32 controller carrying Manchester encoded data from the network. Operates at pseudo ECL levels.

DO±

Data Out *Output*

A differential output pair from the PCnet-32 controller for transmitting Manchester encoded data to the network. Operates at pseudo ECL levels.

Twisted Pair Interface

RXD±

10-BASE-T Receive Data *Input*

10BASE-T port differential receivers.

TXD±

10BASE-T Transmit Data *Output*

10BASE-T port differential drivers.

TXP±

10BASE-T Pre-distortion Control *Output*

These outputs provide transmit predistortion control in conjunction with the 10BASE-T port differential drivers.

External Address Detection Interface

The EADI interface is enabled through BCR2, bit 3 (EADISEL).

$\overline{\text{EAR}}$

External Address Reject Low *Input*

An EADI input signal. The incoming frame will be checked against the internally active address detection mechanisms and the result of this check will be OR'd with the value on the $\overline{\text{EAR}}$ pin. The $\overline{\text{EAR}}$ pin is defined as $\overline{\text{REJECT}}$.

See the EADI section for details regarding the function and timing of this signal.

Note that this pin is multiplexed with the INTR2 pin.

SFBD

Start Frame–Byte Delimiter *Output*

Start Frame–Byte Delimiter Enable. EADI output signal. An initial rising edge on this signal indicates that a start of frame delimiter has been detected. The serial bit stream will follow on the SRD signal, commencing with the destination address field. SFBD will go high for 4 bit times (400 ns) after detecting the second "1" in the SFD (Start of Frame Delimiter) of a received frame. SFBD will subsequently toggle every 400 ns (1.25 MHz frequency) with each rising edge indicating the first bit of each subsequent byte of the received serial bit stream. SFBD will be inactive during frame transmission.

Note that this pin is multiplexed with the LED1 pin.

SRD

Serial Receive Data *Output*

An EADI output signal. SRD is the decoded NRZ data from the network. This signal can be used for external address detection. Note that when the 10BASE-T port is selected, transitions on SRD will only occur during receive activity. When the AUI port is selected, transitions on SRD will occur during both transmit and receive activity.

Note that this pin is multiplexed with the LEDPRE3 pin.

SRDCLK

Serial Receive Data Clock *Output*

An EADI output signal. Serial Receive Data is synchronous with reference to SRDCLK. Note that when the 10BASE-T port is selected, transitions on SRDCLK will only occur during receive activity. When the AUI port is selected, transitions on SRDCLK will occur during both transmit and receive activity.

Note that this pin is multiplexed with the LED2 pin.

General Purpose Serial Interface

The GPSI interface is selected through the PORTSEL bits of the Mode register (CSR15) and enabled through the TSTSHDW[1] bit (BCR18) or the GPSIEN bit (CSR124).

Note that when GPSI test mode is invoked, slave address decoding must be restricted to the lower 24 bits of the address bus by setting the IOAW24 bit in BCR2 and by pulling LED2 LOW during Software Reloactable Mode. The upper 8 bits of the address bus will always be considered matched when examining incoming I/O addresses. During master accesses while in GPSI mode, the PCnet-32 controller will not drive the upper 8 bits of the address bus with address information. See the GPSI section for more detail.

TXDAT

Transmit Data *Input/Output*

TXDAT is an output, providing the serial bit stream for transmission, including preamble, SFD data and FCS field, if applicable.

Note that the TxDAT pin is multiplexed with the A31 pin.

TXEN

Transmit Enable *Input/Output*

TXEN is an output, providing an enable signal for transmission. Data on the TXDAT pin is not valid unless the TXEN signal is HIGH.

Note that the TXEN pin is multiplexed with the A30 pin.

STDCLK

Serial Transmit Data Clock *Input*

STDCLK is an input, providing a clock signal for MAC activity, both transmit and receive. Rising edges of the STDCLK can be used to validate TXDAT output data.

The STDCLK pin is multiplexed with the A29 pin.

Note that this signal must meet the frequency stability requirement of the ISO 8802-3 (IEEE/ANSI 802.3) specification for the crystal.

CLSN

Collision *Input/Output*

CLSN is an input, indicating to the core logic that a collision has occurred on the network.

Note that the CLSN pin is multiplexed with the A28 pin.

RXCRS

Receive Carrier Sense *Input/Output*

RXCRS is an input. When this signal is HIGH, it indicates to the core logic that the data on the RXDAT input pin is valid.

Note that the RXCRS pin is multiplexed with the A27 pin.

SRDCLK

Serial Receive Data Clock *Input/Output*

SRDCLK is an input. Rising edges of the SRDCLK signal are used to sample the data on the RXDAT input whenever the RXCRS input is HIGH.

Note that the SRDCLK pin is multiplexed with the A26 pin.

RXDAT

Receive Data *Input/Output*

RXDAT is an input. Rising edges of the SRDCLK signal are used to sample the data on the RXDAT input whenever the RXCRS input is HIGH.

Note that the RXDAT pin is multiplexed with the A25 pin.

IEEE 1149.1 Test Access Port Interface

TCK

Test Clock *Input*

The clock input for the boundary scan test mode operation. TCK can operate up to 10 MHz. If left unconnected this pin has a default value of HIGH.

TDI

Test Data Input *Input*

The test data input path to the PCnet-32 controller. If left unconnected, this pin has a default value of HIGH.

TDO

Test Data Output *Output*

The test data output path from the PCnet-32 controller. TDO is floated when the JTAG port is inactive.

TMS

Test Mode Select *Input*

A serial input bit stream is used to define the specific boundary scan test to be executed. If left unconnected, this pin has a default value of HIGH.

Power Supply Pins

AVDD

Analog Power (4 Pins) *Power*

There are four analog +5 V supply pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to Appendix B and the *PCnet Family Technical Manual (PID# 18216A)* for details.

AVSS

Analog Ground (2 Pins) *Power*

There are two analog ground pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to Appendix B and the PCnet Family Technical Manual for details.

DVDD

Digital Power (3 Pins) *Power*

There are 3 digital power supply pins. (DVDD1, DVDD2, and DVDD3) used by the internal circuitry.

DVDDCLK

Digital Power Clock (1 Pin) *Power*

This pin is used to supply power to the clock buffering circuitry.

DVDDO

I/O Buffer Digital Power (5 Pins) *Power*

There are 5 digital power supply pins (DVDD01–DVDD05) used by Input/Output buffer drivers.

DVSS

Digital Ground (4 Pins) *Ground*

There are 4 digital ground pins (DVSS1–DVSS4) used by the internal digital circuitry.

DVSSCLK

Digital Ground Clock (1 Pin) *Ground*

This pin is used to supply a ground to the clock buffering circuitry.

DVSSN

I/O Buffer Digital Ground (15 Pins) *Ground*

These 15 ground pins (DVSSN1–DVSSN15) are used by the Input/Output buffer drivers.

DVSSPAD

Digital Ground Pad (1 Pin) *Ground*

This pin is used by the Input/Output logic circuits.

BASIC FUNCTIONS

System Bus Interface Function

The PCnet-32 controller is designed to operate as a Bus Master during normal operations. Some slave accesses to the PCnet-32 controller are required in normal operations as well. Initialization of the PCnet-32 controller is achieved through a combination of Bus Slave accesses, Bus Master accesses and an optional read of a serial EEPROM that is performed by the PCnet-32 controller. The EEPROM read operation is performed through the microwire interface. The ISO 8802-3 (IEEE/ANSI 802.3) Ethernet Address may reside within the serial EEPROM. Some PCnet-32 controller configuration registers may also be programmed by the EEPROM read operation.

The address PROM, on-chip board-configuration registers, and the Ethernet controller registers occupy 32 bytes of I/O space which can be located by modifying the I/O base address register. The default base address can be written to the EEPROM. The PCnet-32 controller will automatically read the I/O Base Address from the EEPROM after H_RESET, or at any time that the software requests that the EEPROM should be read. When no EEPROM is attached to the serial microwire interface, the PCnet-32 controller detects the condition, and enters Software Relocatable Mode. While in Software Relocatable Mode, the PCnet-32 controller will not respond to any bus accesses, but will snoop the bus for accesses to I/O address 378h. When a successfully executed and uninterrupted sequence of write operations is seen at this location, the PCnet-32 controller will accept the next sequence of accesses as carrying I/O Base Address relocation and interrupt pin programming information. After this point, the PCnet-32 controller will begin to respond directly to accesses directed toward offsets from the newly loaded I/O Base Address. This scheme allows for jumperless relocatable I/O implementations.

Software Interface

The software interface to the PCnet-32 controller is divided into two parts. One part is the direct access to the I/O resources of the PCnet-32 controller. The PCnet-32 controller occupies 32 bytes of I/O space that must begin on a 32-byte block boundary. The I/O Base Address can be changed to any 32-bit quantity that begins on a 32-byte block boundary through the function of the Software Relocatable Mode. It can also be changed to any 32-bit value that begins on a 32-byte block boundary through the automatic EEPROM read operation that occurs immediately after the H_RESET function has completed. This read operation automatically alters the I/O Base Address of the PCnet-32 controller.

The 32-byte I/O space is used by the software to program the PCnet-32 controller operating mode, to enable and disable various features, to monitor operating status and to request particular functions to be executed by the PCnet-32 controller.

The other portion of the software interface is the descriptor and buffer areas that are shared between the software and the PCnet-32 controller during normal network operations. The descriptor area boundaries are set by the software and do not change during normal network operations. There is one descriptor area for receive activity and there is a separate area for transmit activity. The descriptor space contains relocatable pointers to the network packet data *and* it is used to transfer packet status from the PCnet-32 controller to the software. The buffer areas are locations that hold packet data for transmission or that accept packet data that has been received.

Network Interfaces

The PCnet-32 controller can be connected to an 802.3 network via one of two network interfaces. The Attachment Unit Interface (AUI) provides an ISO 8802-3 (IEEE/ANSI 802.3) compliant differential interface to a remote MAU or an on-board transceiver. The 10BASE-T interface provides a twisted-pair Ethernet port. While in auto-selection mode, the interface in use is determined by an auto-sensing mechanism which checks the link status on the 10BASE-T port. If there is no active link status, then the device assumes an AUI connection.

DETAILED FUNCTIONS

Bus Interface Unit

The bus interface unit is built of several state machines that run synchronously to BCLK. One bus interface unit state machine handles accesses where the PCnet-32 controller is the bus slave, and another handles accesses where the PCnet-32 controller is the bus master. All inputs are synchronously sampled *except* $\overline{ADS}$, M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$ and the A[31:5] bus when this bus is an input to the PCnet-32 controller. All outputs are synchronously generated on the rising edge of BCLK *with the following exceptions:*

$\overline{LDEV}$ is generated asynchronously.

RDY is *driven/floated* on falling edges of BCLK but will *change state* on rising edges of BCLK.

The following sections describe the various bus master and bus slave operations that will be performed by the PCnet-32 controller. The timing diagrams that are included in these sections (*Bus Acquisition* section through *Slave Timing* section) show the signals and timings of the Am486 32-bit mode of operation. The sections from *Bus Acquisition* through *Linear Burst DMA Transfers* show bus master operations. The *Slave Timing* section shows bus slave operations. Note that the PCnet-32 controller operation in Am486 32-bit mode represents a merger of the requirements of the VESA VL-Bus specification and Am486 bus specification,

whichever is more stringent. The concepts discussed in the following sections and the basic nature of the timings shown is applicable in a general sense to PCnet-32 controller operational modes. For specific differences in timing between modes and for examples of timing diagrams showing basic transfers in each of the modes, please consult the following sections:

Am486 32-bit mode:	*Bus Acquisition* section through *Slave Timing* section
VESA VL-Bus mode:	*VESA VL-Bus Mode Timing* section

For selection of each mode, consult the section on Configuration Pins in the *Pin Description* section.

Note that all timing diagrams in this document have been drawn showing reference to two groups of address pins: namely, A4–A31 and A2–A3, $\overline{BE0}$–$\overline{BE3}$. In the AHOLD timing diagrams, the two groups are shown separately, because the upper address pins become floated, while the lower address pins do not. The point of division between the two groups of address pins will depend upon the value of CLL in BCR18. In the case of Linear Burst operations, the upper address pins are shown separately because that group does not change its value through a single linear burst, while the lower address pins do change value. In this case, the point of division between the two groups of address pins will depend upon the value of LINBC in BCR18. In all other timing diagrams, the two groups are shown separately just to maintain consistency with the AHOLD and Linear Burst timing diagrams. For more details, see the AHOLD and Linear Burst Count sections.

Bus Acquisition

The PCnet-32 controller microcode (in the buffer management section) will determine when a DMA transfer should be initiated. The first step in any PCnet-32 controller bus master transfer is to acquire ownership of the bus. This task is handled by synchronous logic within the BIU. Bus ownership is requested with the HOLD signal and ownership is granted by the CPU (or an arbiter) through the HLDA signal. The PCnet-32 controller additionally supplies HOLDI and HLDAO signals to allow daisy chaining of devices through the PCnet-32 controller. Priority of the HOLDI input versus the PCnet-32 controller's own internal request for bus mastership can be set using the PRPCNET bit of BCR17. Simple bus arbitration (HOLD, HLDA only) is shown in Figure 1.

Note that assertion of the STOP bit will **not** cause a deassertion of the HOLD signal. Note also that a read of the S_RESET register (I/O resource at offset 14h from the PCnet-32 controller base I/O address) will **not** cause a deassertion of the HOLD signal. Either of these actions will cause the internal master state machine logic to cease operations, but the HOLD signal will remain active until the HLDA signal is synchronously sampled as asserted. Following either of the above actions, on the next clock cycle after the HLDA signal is synchronously sampled as asserted, the PCnet-32 controller will deassert the HOLD signal. No bus master accesses will have been performed during this brief bus ownership period.

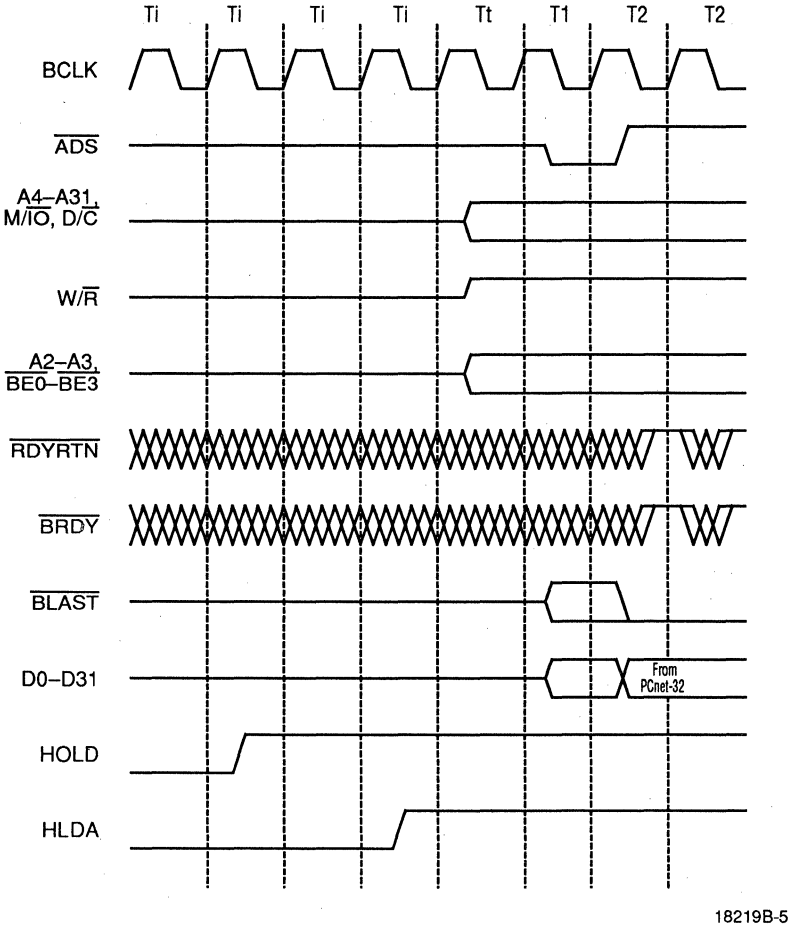


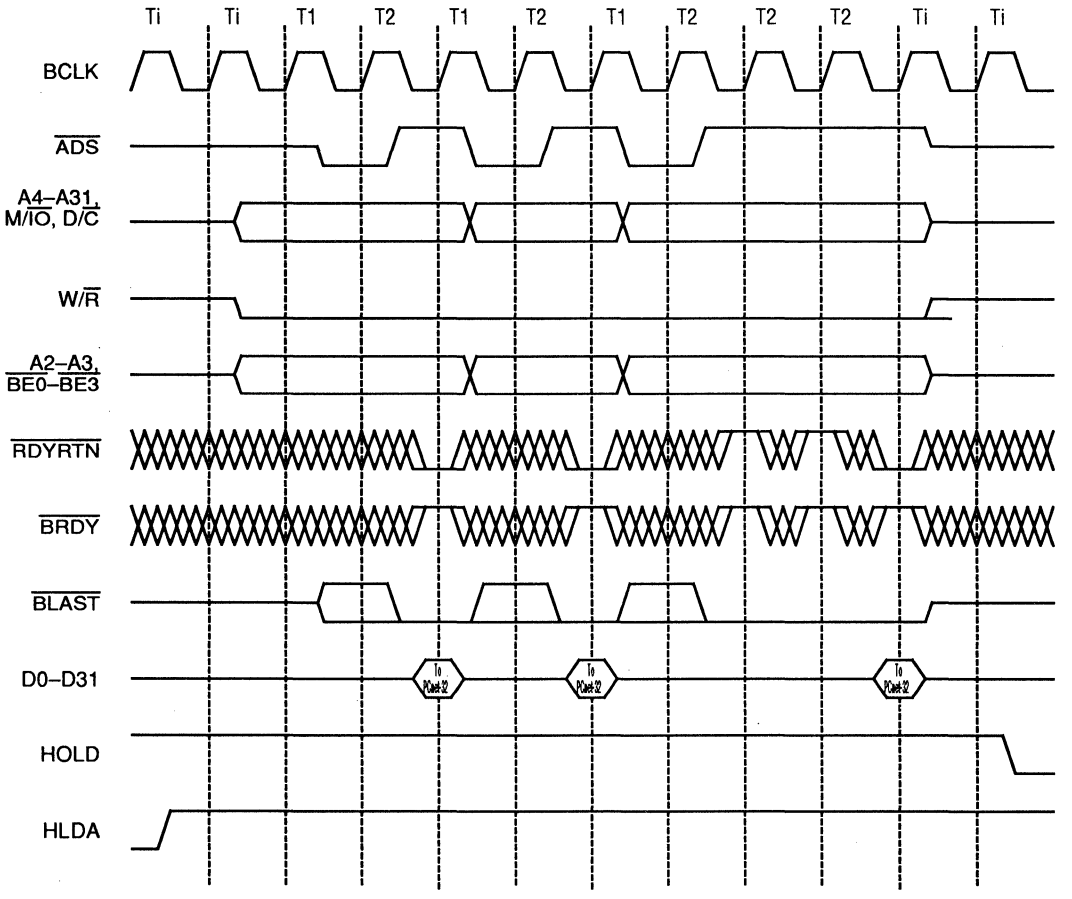
Figure 1. Bus Acquisition

Assertion of a minimum-width pulse on the RESET pin will cause the HOLD signal to deassert within six clock cycles following the assertion of the RESET pin. In this case, the PCnet-32 controller will not wait for the assertion of the HLDA signal before deasserting the HOLD signal.

This description of behavior is identical for the VESA VL-Bus mode of operation, except that the HOLD, HLDA and RESET signals are replaced by the inverted sense signals LREQ, LGNT, and RESET, respectively; the BCLK, BOFF, EADS, and RDY signals are replaced by LCLK, WBACK, LEADS, and LRDY, respectively.

Bus Master DMA Transfers

There are three primary types of DMA transfers. All DMA transfers will have wait states inserted until either RDYRTN or BRDY is asserted. Figure 2 and Figure 3 show the basic bus transfers of read and write without ready wait states and then with ready wait states.



18219B-6

Figure 2. Basic Read Cycles without and with Wait States

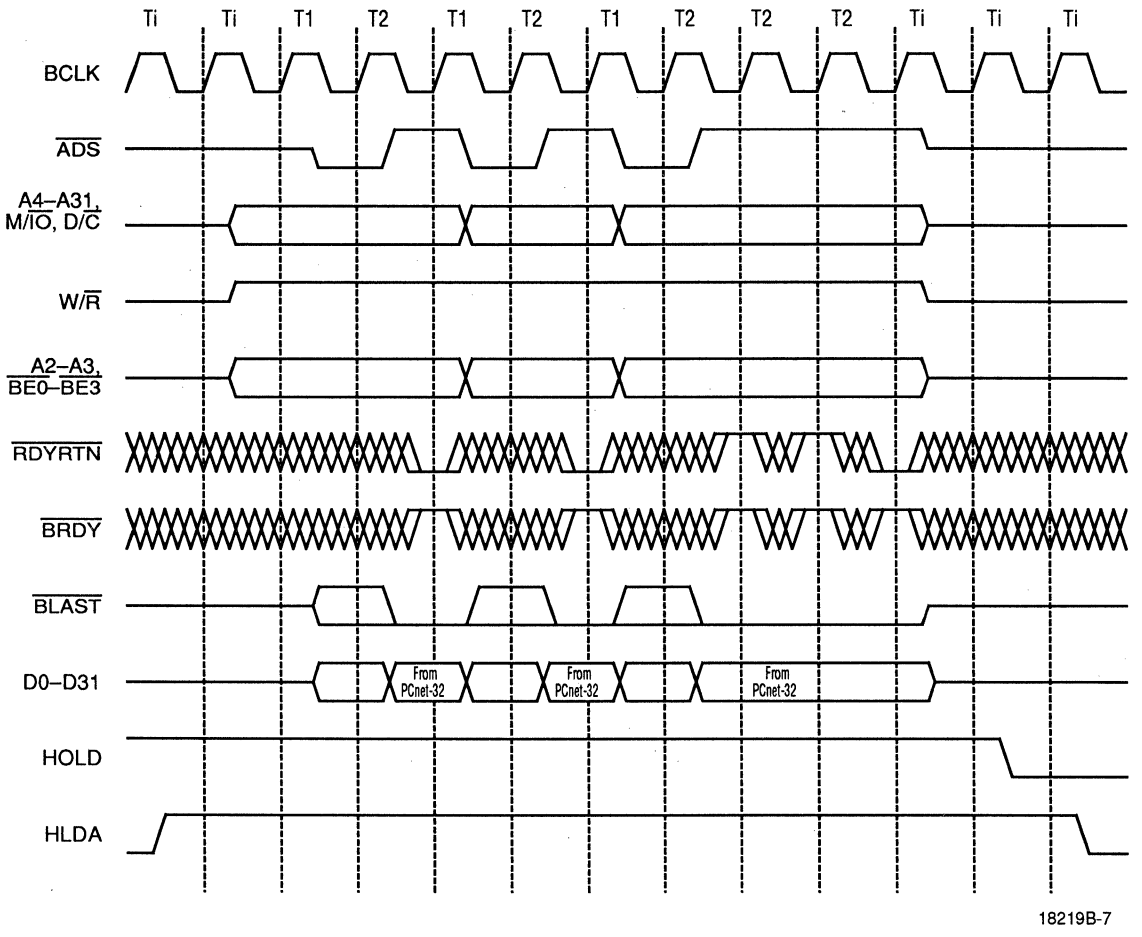


Figure 3. Basic Write Cycles without and with Wait States

Effect of BOFF

Assertion of BOFF during bus master transfers will cause the PCnet-32 controller to float all of its bus signals beginning at the next clock cycle. Any access which has been interrupted by BOFF will be restarted when the BOFF signal is deasserted. Simultaneous assertion of RDYRTN (or BRDY) and BOFF is resolved in favor of BOFF. The RDYRTN (or BRDY) is ignored for such a cycle, and when BOFF is deasserted, the cycle is restarted. See Figure 4 for detail.

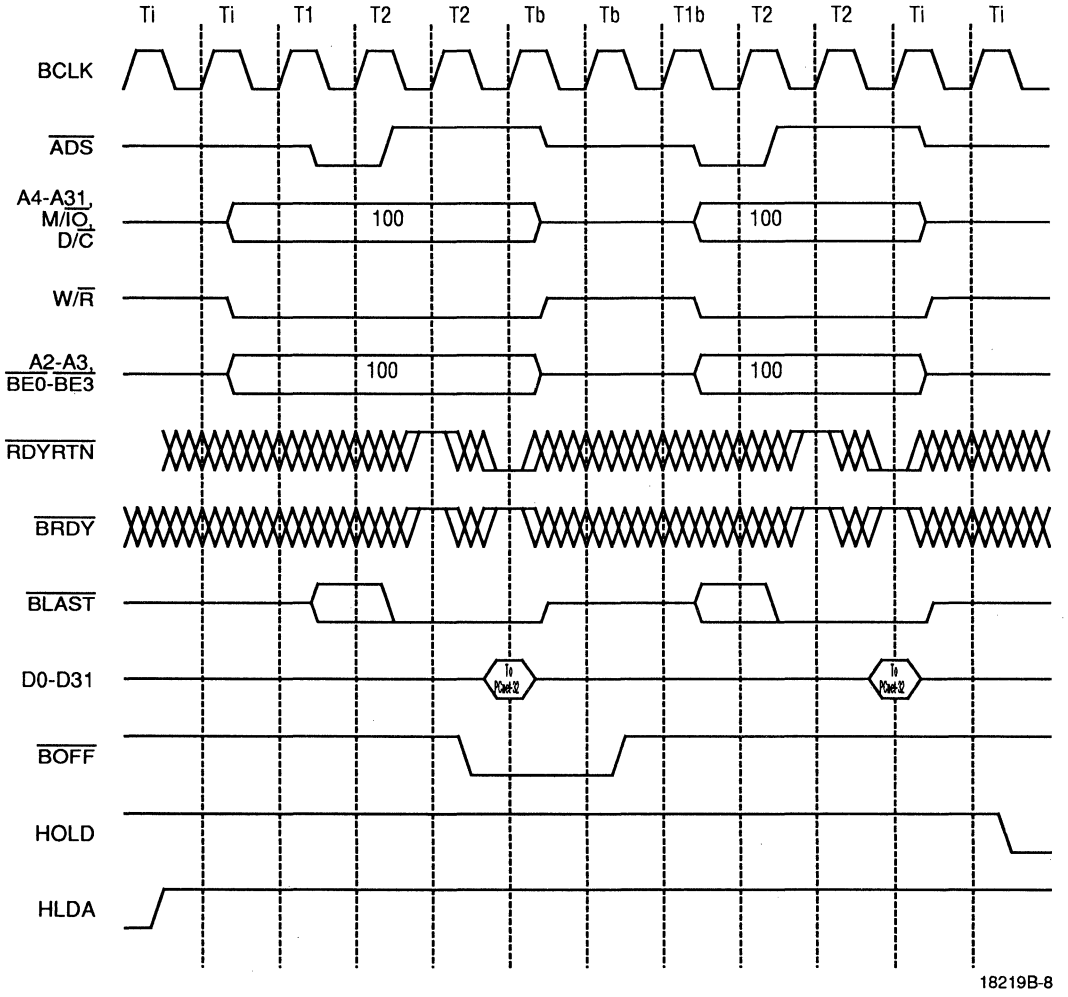


Figure 4. Restarted Read Cycle

Effect of AHOLD

Assertion of AHOLD during bus master transfers will cause the PCnet-32 controller to float some portion of the address bus beginning at the next clock cycle. If $\overline{\text{RDYRTN}}$ is returned while AHOLD is active, then the cycle completes, since the data bus may remain active during AHOLD. However, a new cycle will not be started while AHOLD is active.

The portion of the Address Bus that will be floated at the time of an address hold operation will be determined by the value of the Cache Line Length register (BCR18, bits 15-11). Table 17 lists all of the legal values of CLL showing the portion of the Address Bus that will become floated during an address hold operation.

If the $\overline{\text{RDYRTN}}$ signal is not returned while AHOLD is active, then the PCnet-32 controller will resume driving the same address onto the address bus when AHOLD is released. The PCnet-32 controller will not reissue the $\overline{\text{ADS}}$ signal at this time. See Figure 5 and Figure 6 for details.

Table 17. CLL Value and Floating Address Pins

CLL Value	Floated Portion of Address Bus During AHOLD
00000	None
00001	A31–A2
00010	A31–A3
00011	Reserved CLL Value
00100	A31–A4
00101–00111	Reserved CLL Values
01000	A31–A5
01001–01111	Reserved CLL Values
10000	A31–A6
10001–11111	Reserved CLL Values

Note: *The default value of CLL after H_RESET is 00100. All timing diagrams in this document are drawn with the assumption that this is the value of CLL.*

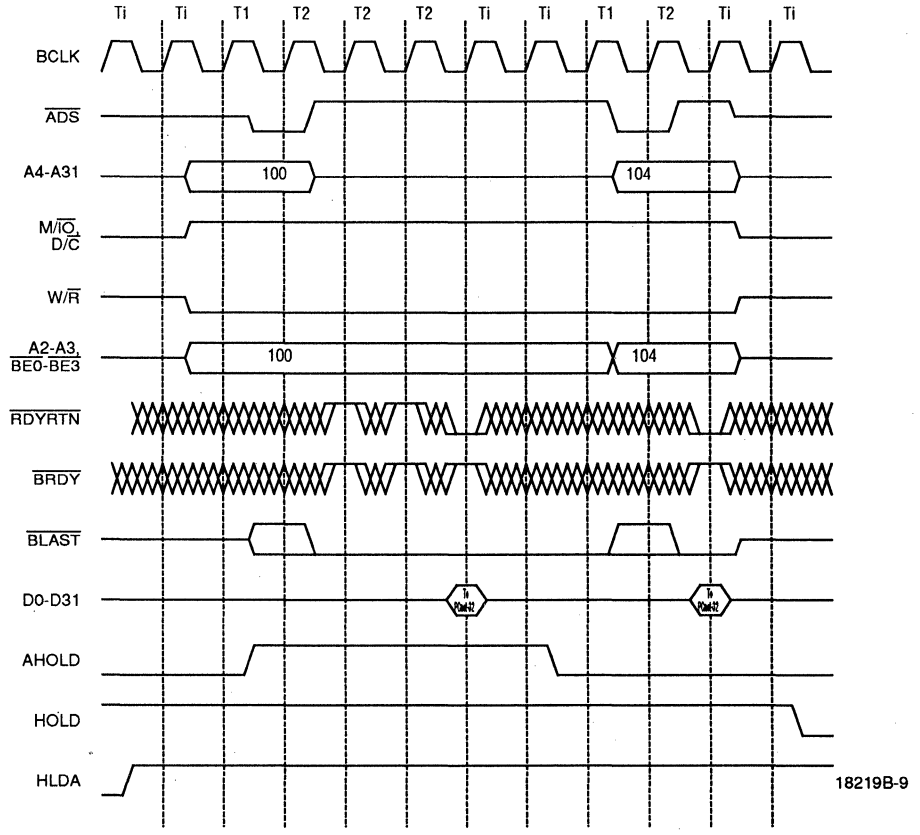


Figure 5. Read Cycle with AHOLD

Note that initial access is allowed to complete in spite of AHOLD, but next access is prevented from beginning until AHOLD is deasserted.

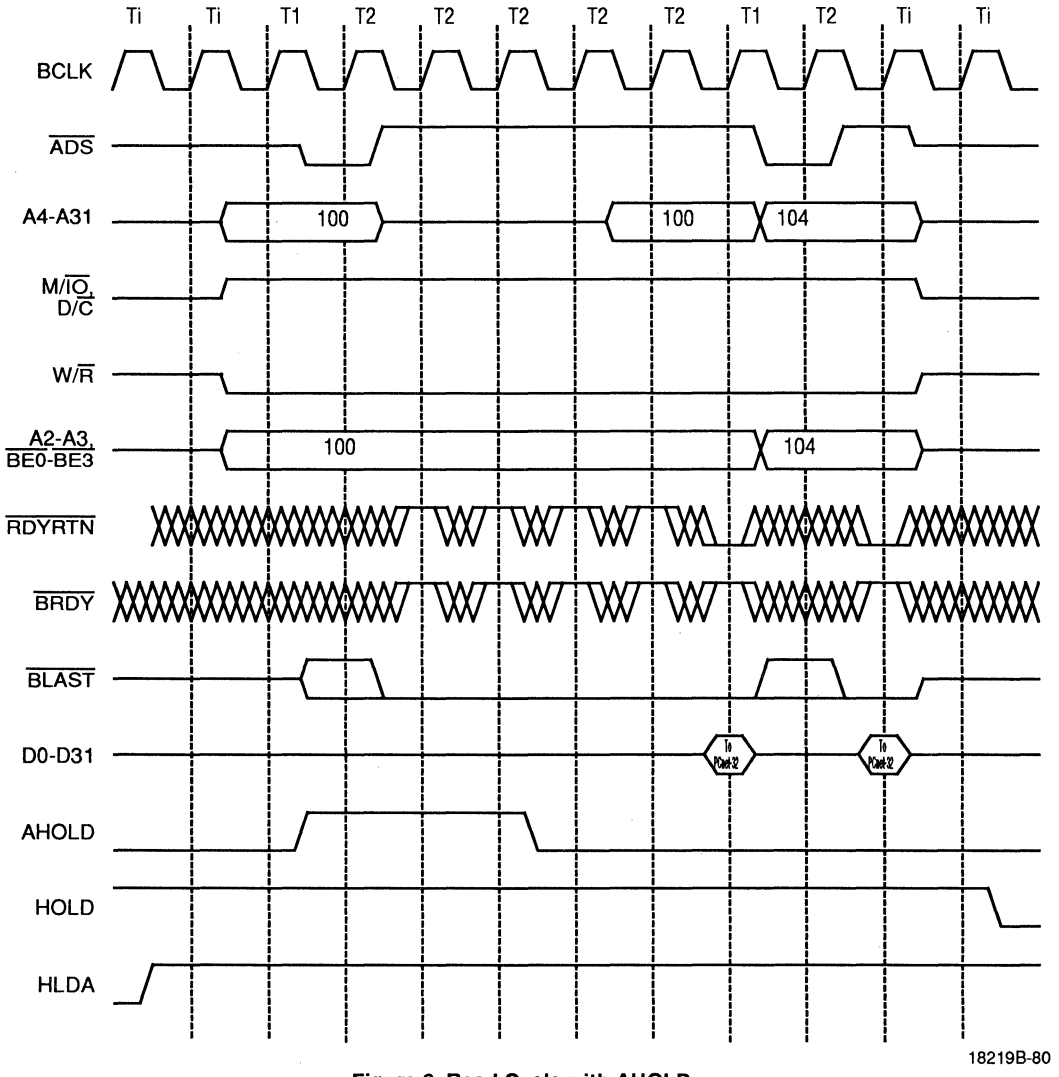


Figure 6. Read Cycle with AHOLD
Address is re-driven when AHOLD is deasserted, since RDYRTN had not yet arrived.

Effect of Bus Preemption

If a bus preemption event occurs during a basic transfer cycle, then the behavior of the PCnet-32 controller will depend upon which specific type of access is being performed. The general response of the PCnet-32 controller is that the current operation will complete before the PCnet-32 controller relinquishes the bus in response to the preemption. "Current operation" in this sense refers to the general PCnet-32 controller operation, such as "descriptor access". Note that a "descriptor access" consists of one or two basic transfers. Therefore, both transfers of a descriptor access must be completed before the bus will be released in response to a preemption. See each of the sections for Initialization Block DMA transfers, Descriptor DMA transfers, FIFO DMA transfers and Linear Burst Transfers for more specific information.

Effect of $\overline{LBS16}$ (VL-Bus mode only)

Dynamic bus sizing is recognized by the PCnet-32 controller while operating in the VL-Bus mode. The $\overline{LBS16}$ signal is used to indicate to the PCnet-32 controller whether the VL-Bus target is a 16-bit or 32-bit peripheral. When the target device indicates that it is 16 bits in width by asserting the $\overline{LBS16}$ signal at least one LCLK period before asserting the $\overline{RDYRTN}$ signal, then the PCnet-32 controller will dynamically respond to the size constraints of the peripheral by performing additional accesses. Table 18 shows the sequence of accesses that will be performed by the PCnet-32 controller in response to the assertion of $\overline{LBS16}$.

Figure 7 shows an example of an exchange between a 16-bit VL-Bus peripheral and the PCnet-32 controller. Note that the $\overline{LBS16}$ signal is asserted during the LCLK that precedes the assertion of $\overline{RDYRTN}$. In this particular case, in order to maintain zero-wait state accesses, the 16-bit target must generate $\overline{LBS16}$ in a very short time in order to meet the required setup time of $\overline{LBS16}$ into the PCnet-32 controller. If the peripheral were incapable of meeting the required setup time, then a wait state would be needed in order to insure that $\overline{LBS16}$ is asserted at least one LCLK prior to the assertion of the $\overline{RDYRTN}$ signal.

When the assertion of $\overline{LBS16}$ during a PCnet-32 controller master access has created the need for a second access as specified in the table above, **and** the $\overline{WBACK}$ signal becomes active during the second access, then, when $\overline{WBACK}$ is deasserted, the PCnet-32 controller will repeat both accesses of the pair.

Table 18. Data Transfer Sequence from 32-Bit Wide to 16-Bit Wide

Current Access				Next with $\overline{LBS16}$			
$\overline{BE3}$	$\overline{BE2}$	$\overline{BE1}$	$\overline{BE0}$	$\overline{BE3}$	$\overline{BE2}$	$\overline{BE1}$	$\overline{BE0}$
1	1	1	0	NR*			
1	1	0	0	NR*			
1	0	0	0	1	0	1	1
0	0	0	0	0	0	1	1
1	1	0	1	NR*			
1	0	0	1	1	0	1	1
0	0	0	1	0	0	1	1
1	0	1	1	NR*			
0	0	1	1	NR*			
0	1	1	1	NR*			

**NR = No second access Required for these cases*

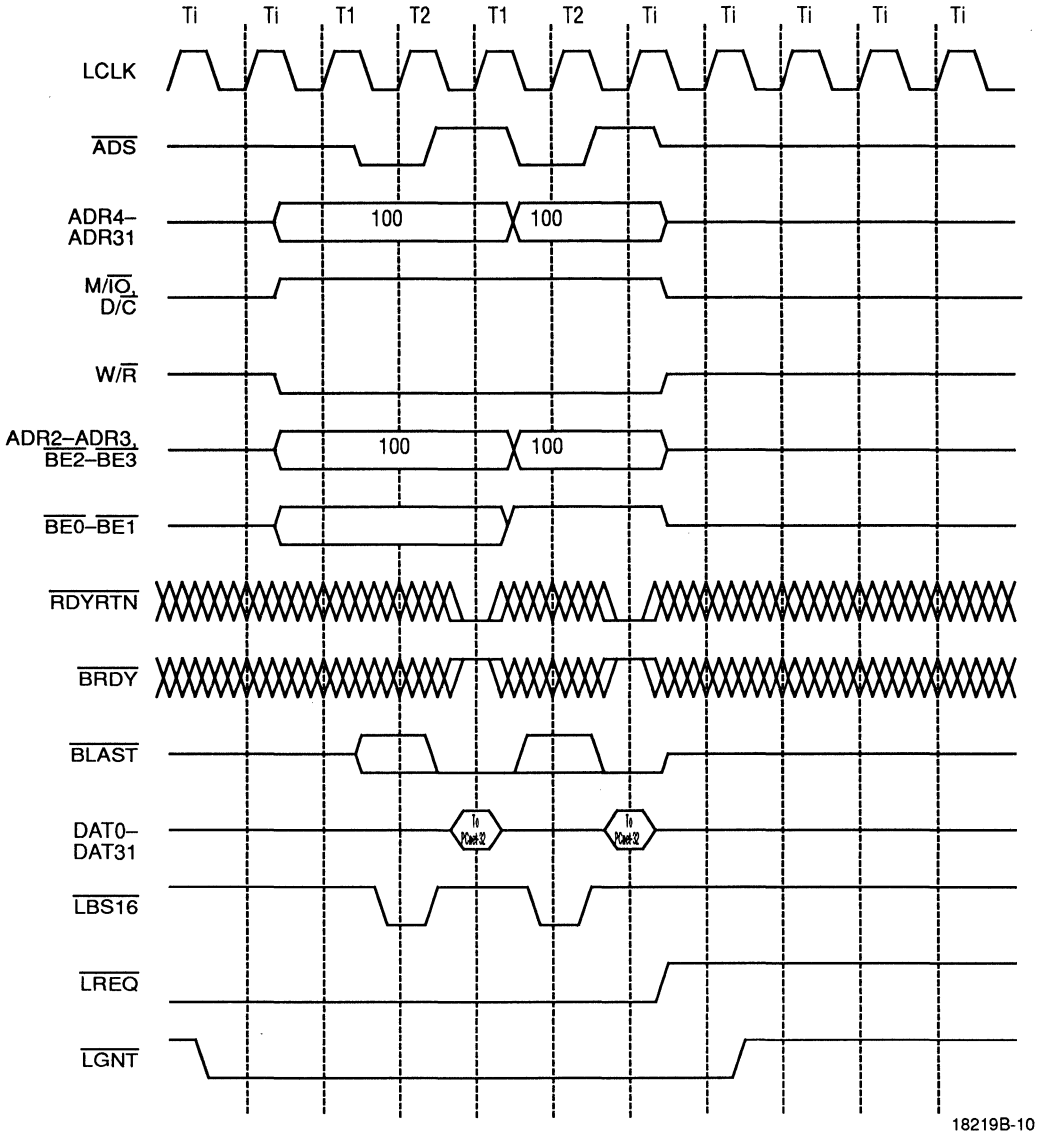


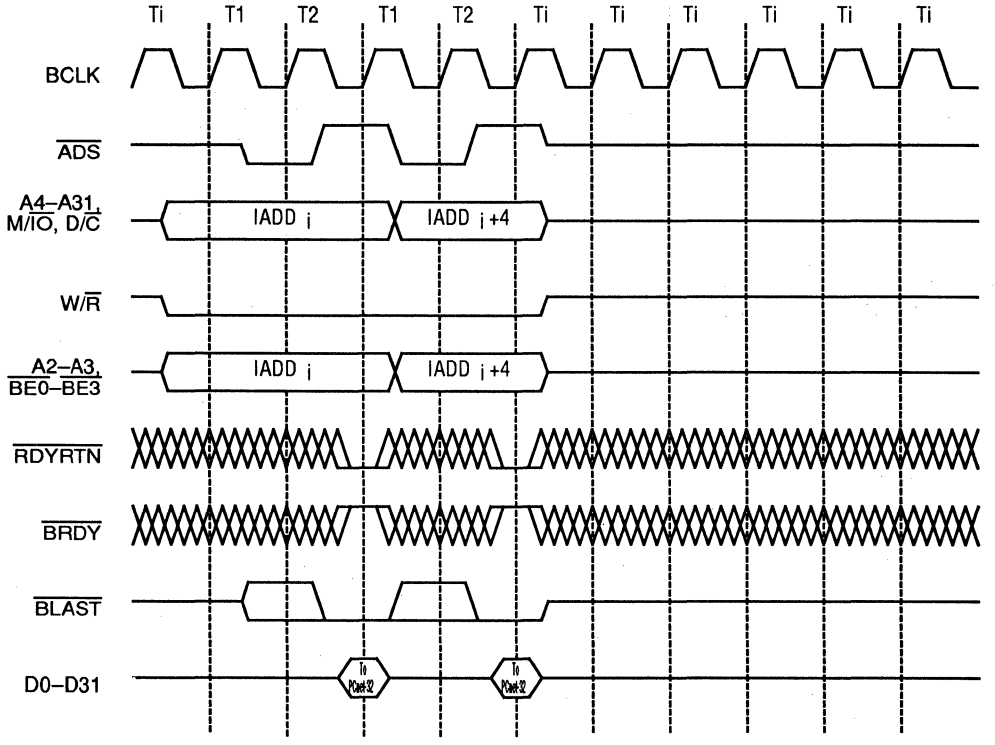
Figure 7. VL-Bus Ready Cycle with $\overline{\text{LBS16}}$ Asserted
Four-byte single cycle access is converted to two 2-byte accesses.

Initialization Block DMA Transfers

During execution of the PCnet-32 controller bus master initialization procedure, the PCnet-32 controller microcode will repeatedly request DMA transfers from the BIU. During each of these initialization block DMA transfers, the BIU will perform two data transfer cycles (eight bytes) and then it will relinquish the bus (see Figure 8). The two transfers within the mastership period will always be read cycles to ascending contiguous addresses. The two transfers in each initialization block DMA transfer will never be executed using linear burst mode. In 32-bit software mode, the number of bus mastership periods needed to complete the initialization procedure is 4. There are 7 doublewords to transfer during the bus master initialization procedure, so four bus mastership periods are needed in order to complete the initialization sequence. Note that the last doubleword transfer of the last bus mastership period of the initialization sequence accesses an unneeded location. Data from this transfer is discarded internally.

If a bus preemption event occurs during an initialization block DMA transfer, then the PCnet-32 controller will complete both of the two data transfer cycles of the initialization block DMA transfer before releasing the HOLD signal and relinquishing the bus.

When SSIZE32 = 0 (CSR58[8]/BCR20[8]), then the number of bus mastership periods needed to complete the initialization procedure is 3 or 4.



18219B-11

Figure 8. Initialization DMA Transfer

Descriptor DMA Transfers

PCnet-32 controller will determine when a descriptor access is required. A descriptor DMA *read* will consist of *two* doubleword transfers. A descriptor DMA *write* will consist of *one* **or** *two* double word transfer. The transfers within a descriptor DMA transfer mastership period will always be of the same type (either all read or all write). The transfers will be to addresses in the order as specified in Table 19 and Table 20 (note that *MD* indicates *TMD* or *RMD*).

If buffer chaining is used (see Transmit and Receive Descriptor Table Entry sections), *writes* to the descriptors that do **not** contain the End of Packet bit will consist of only one doubleword. This write will be to the same location as the second of the two writes performed when the End of Frame has been processed (i.e. to the location that contains the descriptor OWNership bit, MD1[31]).

Descriptor DMA transfers will never be executed using linear burst mode. During read accesses, the byte enable signals will indicate that all byte lanes are active. Should some of the bytes not be needed, then the PCnet-32 controller will internally discard the extraneous information that was gathered during such a read. During write accesses, only the bytes which *need* to be written are enabled, by activating the corresponding byte enable pins. See Figure 9 and Figure 10.

If a *bus preemption* event occurs during a descriptor DMA transfer, then the PCnet-32 controller will complete both of the two data transfer cycles of the descriptor DMA transfer, before releasing the HOLD signal and relinquishing the bus.

The only significant differences between descriptor DMA transfers and initialization DMA transfers are that the addresses of the accesses follow different ordering.

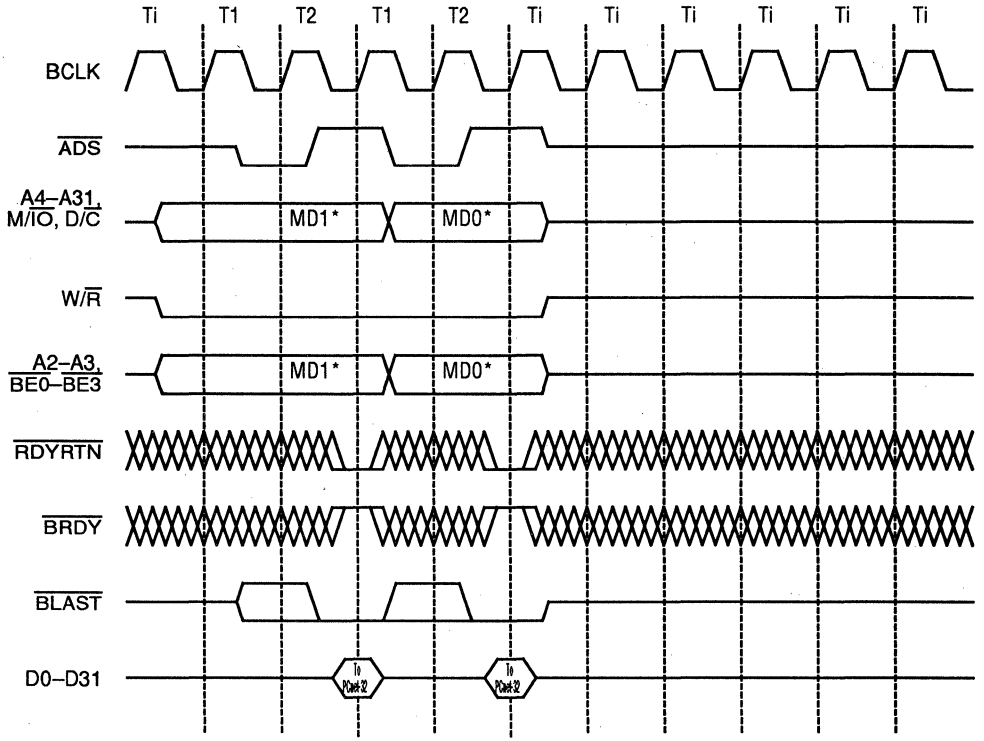
Table 19. Bus Master Reads of Descriptors

16-Bit Software Mode			32-Bit Software Mode		
Address Sequence A[7:0]	**LANCE Item Accessed**	**PCnet-32 Item Accessed**	**Address Sequence A[7:0]***	**LANCE Item Accessed**	**PCnet-32 Item Accessed**
00	MD1[15:0], MD0[15:0]	MD1[31:24], MD0[23:0]	04	MD1[15:8], MD2[15:0]	MD1[31:0]
04	MD3[15:0], MD2[15:0]	MD2[15:0], MD1[15:0]	00	MD1[7:0], MD0[15:0]	MD0[31:0]
Bus Break			Bus Break		

Table 20. Bus Master Writes of Descriptors

16-Bit Software Mode			32-Bit Software Mode		
Address Sequence A[7:0]	**LANCE Item Accessed**	**PCnet-32 Item Accessed**	**Address Sequence A[7:0]***	**LANCE Item Accessed**	**PCnet-32 Item Accessed**
04	MD3[15:0], MD2[15:0]	MD2[15:0], MD1[15:0]	08	MD3[15:0], MD2[15:0]	MD2[31:0]
00	MD1[15:0], MD0[15:0]	MD1[31:24], MD0[23:0]	04	MD1[15:8], MD2[15:0]	MD1[31:0]
Bus Break			Bus Break		

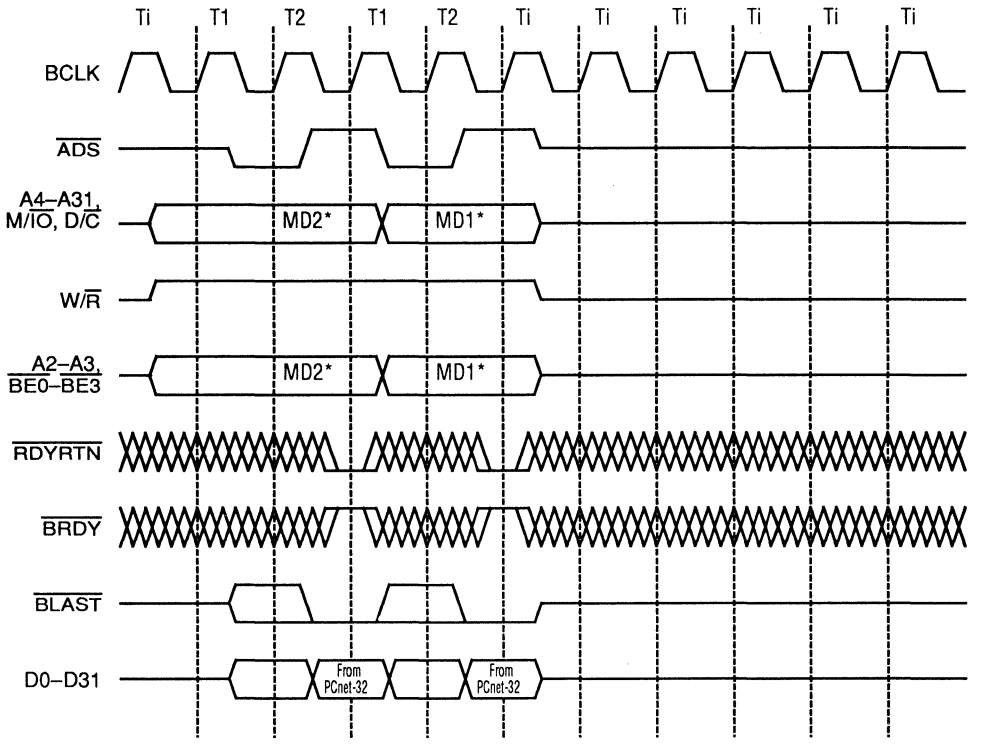
**Address values for A[31:8] are constant throughout any single descriptor DMA transfer. Note that even though bits A[1:0] do not physically exist in the system, these bits must be set to ZERO in the descriptor base address.*



Note that Message Descriptor addresses 1 and 0 are in descending order.

18219B-12

Figure 9. Descriptor DMA Read



*Note that Message Descriptor addresses 2 and 1 are in descending order.

18219B-13

Figure 10. Descriptor DMA Write

FIFO DMA Transfers

PCnet-32 controller microcode will determine when a FIFO DMA transfer is required. This transfer mode will be used for transfers of data to and from the PCnet-32 controller FIFOs. Once the PCnet-32 controller BIU has been granted bus mastership, it will perform a series of consecutive transfer cycles before relinquishing the bus. Each transfer will be performed sequentially, with the issue of an address, and the transfer of the corresponding data with appropriate output signals to indicate selection of the active data bytes during the transfer. All transfers within the master cycle will be either read or write cycles, and all transfers will be to contiguous, ascending addresses. The number of data transfer cycles contained within a single bus cycle is in general, dependent on the programming of the DMAPLUS option (CSR4, bit 14). Several other factors will also affect the length of the bus cycle period. The possibilities are as follows:

If DMAPLUS = 0, a maximum of 16 transfers will be performed by default. This default value may be changed by writing to the burst register (CSR80). Note that DMAPLUS = 0 merely sets a *maximum* value. The minimum number of transfers in the bus cycle will be determined by all of the following variables: the settings of the FIFO watermarks, the particular conditions existing within the FIFOs, receive and transmit status conditions, the value of the DMA Burst Cycle (CSR80), the value of the DMA Bus Activity Timer (CSR82), and the timing of any occurrence of preemption that takes place during the FIFO DMA transfer.

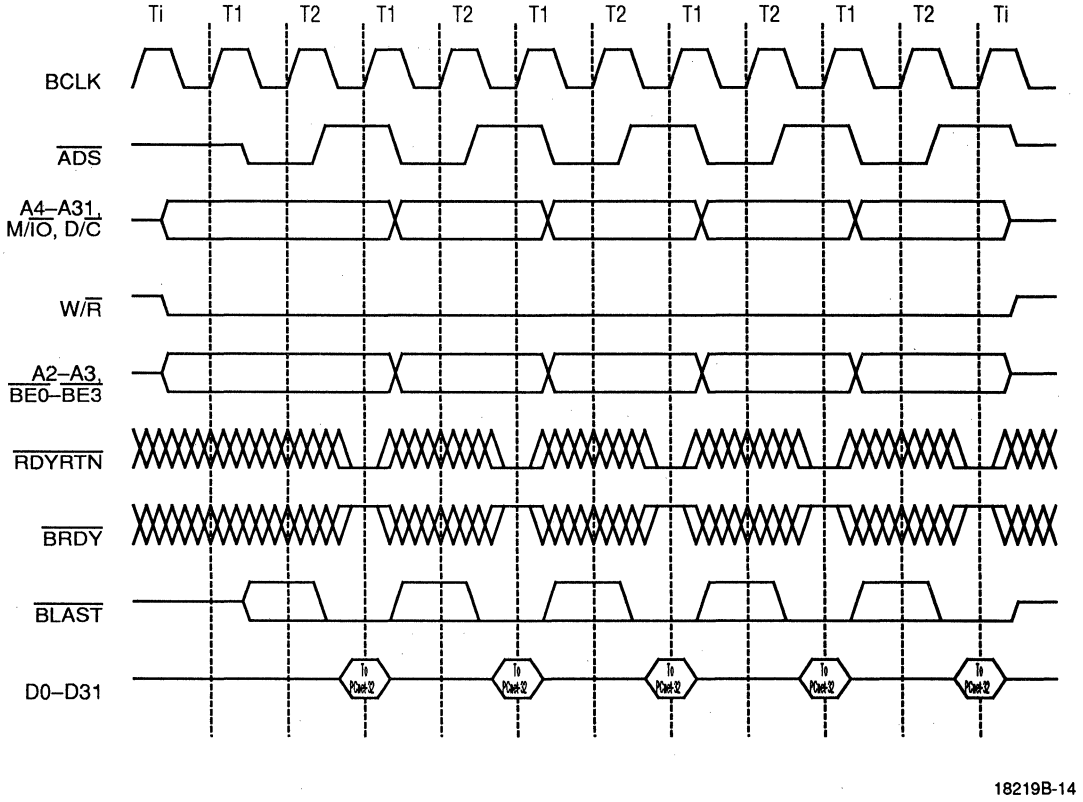
If DMAPLUS = 1, the bus cycle will continue until the transmit FIFO is filled to its high threshold (read transfers) or the receive FIFO is emptied to its low threshold (write transfers), or until the DMA Bus Activity Timer value (CSR82) has expired. Other variables may also affect the end point of the burst in this mode. Among those variables are: the particular conditions existing within the FIFOs, receive and transmit status conditions, and bus preemption events.

The FIFO thresholds are programmable (see description of CSR80), as are the Burst Cycle and Bus Activity Timer values. The exact number of transfer cycles in the case of DMAPLUS = 1 will be dependent on the latency

of the system bus to the PCnet-32 controller's mastership request and the speed of bus operation, but will be limited by the value in the Bus Activity Timer register, the FIFO condition, receive and transmit status, and by preemption events, if any. Barring a time-out by either of these registers, or a bus preemption by another mastering device, or exceptional receive and transmit events, or an end of packet signal from the FIFO, the FIFO watermark settings and the extent of Bus Acknowledge latency will be the major factors determining the number of accesses performed during any given arbitration cycle when DMAPLUS = 1.

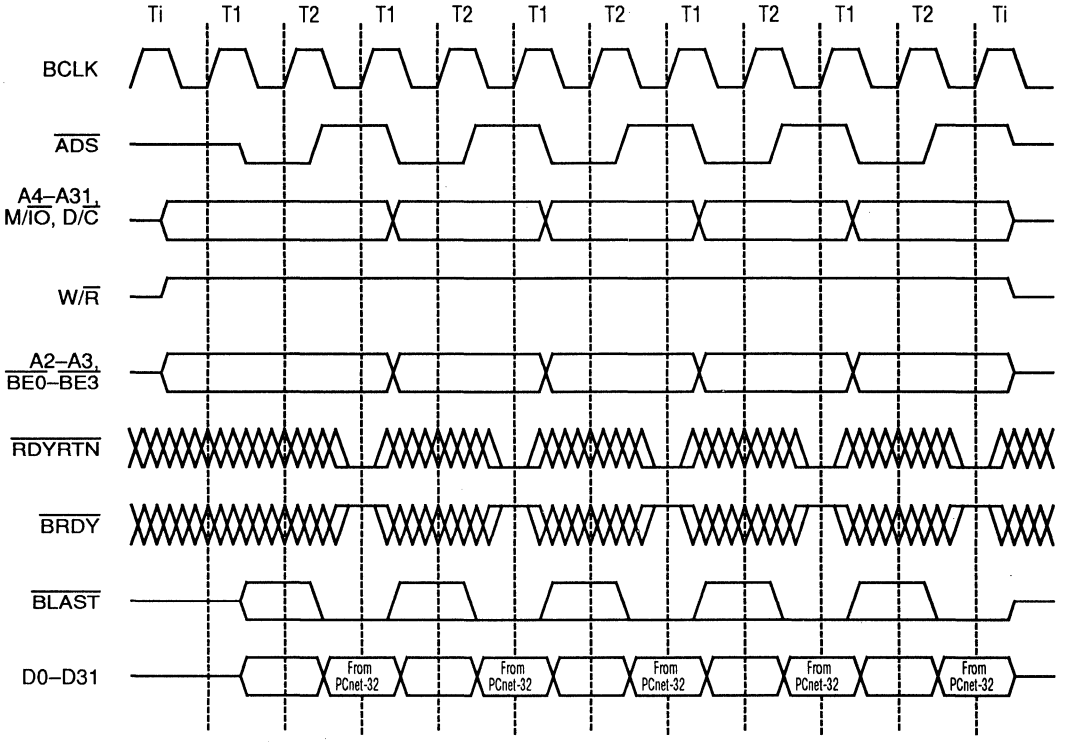
The READY response of the memory device will also affect the number of transfers when DMAPLUS = 1, since the speed of the accesses will affect the state of the FIFO. (During accesses, the FIFO may be filling or emptying on the network end. A slower memory response will allow additional data to accumulate inside of the FIFO (during write transfers from the receive FIFO). If the accesses are slow enough, a complete double word may become available before the end of the arbitration cycle and thereby increase the number of transfers in that cycle.) The general rule is that the longer the bus grant latency or the slower the bus transfer operations (or clock speed) or the higher the transmit watermark or the lower the receive watermark or any combination thereof the longer will be the average burst length.

If a *bus preemption* event occurs during a FIFO DMA transfer, then the PCnet-32 controller will complete the current transfer **and** it will complete a maximum of four additional data transfer cycles before releasing the HOLD signal and relinquishing the bus.



18219B-14

Figure 11. FIFO DMA Read



18219B-15

Figure 12. FIFO DMA Write

Note that A[1:0] do not exist in a 32-bit system, but both of these bits do exist in the buffer pointers that are passed to the PCnet-32 controller in the descriptor. A[1:0] values will be decoded and presented on the bus as byte enable ($\overline{BE0}$-$\overline{BE3}$) values during FIFO DMA transfers.

Linear Burst DMA Transfers

Once the PCnet-32 controller has been granted bus mastership, the PCnet-32 controller may request to perform linear burst cycles by de-asserting the $\overline{BLAST}$ signal. If the device being accessed wishes to support linear bursting, then it must assert $\overline{BRDY}$ and de-assert $\overline{RDYRTN}$, with the same timing that $\overline{RDYRTN}$ would normally be provided. Linear bursting is only performed by the PCnet-32 controller if the BREADE and/or BWRITE bits of BCR18 are set. These bits individually enable/disable the ability of the PCnet-32 controller to perform linear burst accesses during master read operations and master write operations, respectively. Only FIFO data transfers will make use of the linear burst mode.

The first transfer in the linear burst will consist of both an address and a data cycle, but subsequent transfers will contain data only, until the LINBC upper limit of transfers have been executed. LINBC is a value from the BCR18 register. The linear burst "upper limit" is created by taking the BCR18 LINBC[2:0] value and multiplying by 4. The result is the number of transfers that will be performed within a single linear burst sequence.

The entire address bus will still be driven with appropriate values during the data cycles. When the LINBC upper limit of data transfers have been performed, a new $\overline{ADS}$ may be asserted (if there is more data to be transferred), with a new address on the A2–A31 pins. Following the new $\overline{ADS}$ cycle, the linear bursting of data will resume. Ownership of the bus will be maintained until other variables cause the PCnet-32 controller to relinquish the bus. These variables have been discussed in the FIFO DMA transfers section above. They will be reviewed again within this section of the document.

Transfers within a linear burst cycle will either be all read or all write cycles, and will always be to contiguous ascending addresses. Linear Bursting of Read and Write operations can be individually enabled or disabled through the BREADE and BWRITE bits of BCR18 (bits 6 and 5).

Linear Burst DMA transfers should be considered as a superset of the FIFO DMA transfers. Linear burst DMA transfers will only be used for data transfers to and from the PCnet-32 controller FIFOs and they will only be allowed when the burst enable bits of BCR18 have been set. Linear Read bursting and Linear Write bursting have individual enable bits in BCR18. Any combination of linear burst enable bit settings is permissible.

Linear bursting is not allowed in systems that have BCLK frequencies above 33 MHz. Linear bursting is *automatically* disabled in VL-Bus systems that operate above this frequency by connecting the VLBEN pin to either ID(3) (for VL-Bus version 1.0 systems) or ID(4) AND ID(3) AND ID(1) AND $\overline{ID(0)}$ (for VL-Bus version 1.1 or 2.0 systems). In Am486-style systems that have BCLK frequencies above 33 MHz, disabling the linear burst capability is ideally carried out through EEPROM bit programming, since the EEPROM programming can be setup for a particular machine's architecture.

All byte lanes are always considered to be active during all linear burst transfers. The $\overline{BE3}$–$\overline{BE0}$ signals will reflect this fact.

Linear Burst DMA Starting Address Restrictions

A PCnet-32 controller linear burst will begin only when the address of the current transfer meets the following condition:

A[31:0] MOD (LINBC x 16) = 0,

The following table illustrates all possible starting address values for all legal LINBC values. Note that A[31:6] are don't care values for all addresses. Also note that while A[1:0] do not physically exist within a 32 bit system, they are valid bits within the buffer pointer field of descriptor word 0. Thus, where A[1:0] are listed, they refer to the lowest two bits of the descriptor's buffer pointer field. These bits will have an affect on determining when a PCnet-32 controller linear burst operation may legally begin and they will affect the output values of the $\overline{BE3}$–$\overline{BE0}$ pins, therefore they have been included in Table 21 as A[1:0].

It is not necessary for the software to insure that the buffer address pointer contained in descriptor word 0 matches the address restrictions given in the table. *If the buffer pointer does not meet the conditions set forth in the table, then the PCnet-32 controller will simply postpone the start of linear bursting until enough ordinary FIFO DMA transfers have been performed to bring the current working buffer pointer value to a valid linear burst starting address. This operation is referred to as* "aligning" the buffer address to a valid linear burst starting address. Once this has been done, the PCnet-32 controller will recognize that the address for the current access is a valid linear burst starting address, and it will automatically begin to perform linear burst accesses at that time, provided of course that the software has enabled the linear burst mode.

Note that if the software *would* provide only valid linear burst starting addresses in the buffer pointer, then the PCnet-32 controller could avoid performing the alignment operation. It would begin linear burst accesses on the very first of the buffer transfers thereby allowing a slight gain in bus bandwidth efficiency.

Because of the linear burst starting address restrictions given in the table above, the PCnet-32 controller linear burst mode is completely compatible with the Am486-style burst cycle when the LINBC[2:0] bits have been programmed with the value of 001.

Table 21. Linear Burst Addresses

LINBC[2:0]	LBS = Linear Burst Size (No. of Transfers)	Linear Burst Addresses Beginning A[5:0] = (A[31:6] = Don't Care)
0	0 (no linear bursting)	Not Applicable
1	4	00, 10, 20, 30
2	8	00, 20
4	16	00
3,5,6,7	Reserved	Not Applicable

Linear Burst DMA Timing Diagram Explanatory Note

Note that in all of the following timing diagrams for linear burst operations, a LINBC[2:0] value of 001 has been assumed. This translates to a linear burst length of four transfers. When the linear burst size is four transfers, then A[31:4] are stable within a single linear burst sequence, while A[3:2] and $\overline{BE3}$-$\overline{BE0}$ will change to reflect the address of the current transfer. Note that for larger values of LINBC[2:0] which correspond to longer linear burst lengths, the range of address pins that is stable during each burst sequence is smaller. For example, if LINBC[2:0] has the value of 010, then the linear burst length is eight double word transfers or 32 bytes of data. With this value of LINBC, it takes five address bits to track the changing addresses through the burst. This means that only A[31:5] are stable during each linear burst sequence, while A[4:2] and $\overline{BE3}$-$\overline{BE0}$ will change to reflect the address of the current transfer. For LINBC[2:0] = 100, only A[31:6] are stable during each linear burst sequence, while A[5:2] and $\overline{BE3}$-$\overline{BE0}$ will change to reflect the address of the current transfer, and so on. Table 22 summarizes this information.

Table 22. Stable Address Lines During Linear Burst

LINBC Value	Portion of Address Bus Stable During Linear Burst
000	Linear Bursting Disabled
001	A[31:4]
010	A[31:5]
100	A[31:6]

Values of LINBC not shown in the table are not allowed. See the LINBC section of BCR18 for more details.

Since all of the timing diagrams assume a LINBC[2:0] value of 001, then A[31:4] are shown to be stable within each linear burst sequence, and A[3:2] and $\overline{BE3}$–$\overline{BE0}$ are shown as changing in order to reflect the address of the current transfer.

Linear Burst DMA Address Alignment

Linear bursting may begin during a bus mastership period which was initially performing only ordinary DMA operations. (I.e. the value of $\overline{BLAST}$ is not restricted to ZERO for an entire bus mastership period if ZERO was the value of $\overline{BLAST}$ on the first access of a bus mastership period.) A change from non-linear bursting to linear bursting will normally occur during linear burst DMA address alignment operations.

If the PCnet-32 controller is programmed for LINEAR burst mode (i.e. BREADE and/or BWRITE bits of BCR18 are set to ONE), and the PCnet-32 controller requests the bus, but the starting address of the first transaction does not meet the conditions as specified in the table above, then the PCnet-32 controller will perform burst-cycle accesses (i.e.. it will provide an $\overline{ADS}$ for each transfer) until it arrives at an address that does meet the conditions described in the table. At that time, and without releasing the bus, the PCnet-32 controller will invoke the linear burst mode. The simple external manifestation of this event is that the value of the $\overline{BLAST}$ signal will change to deasserted (driven high) on the next T2 cycle, thereby indicating a willingness of the PCnet-32 controller to perform linear bursting. (Note that burst-cycle accesses are performed with $\overline{BLAST}$ = 0.)

Figure 13 shows an example of a linear burst DMA alignment operation being performed:

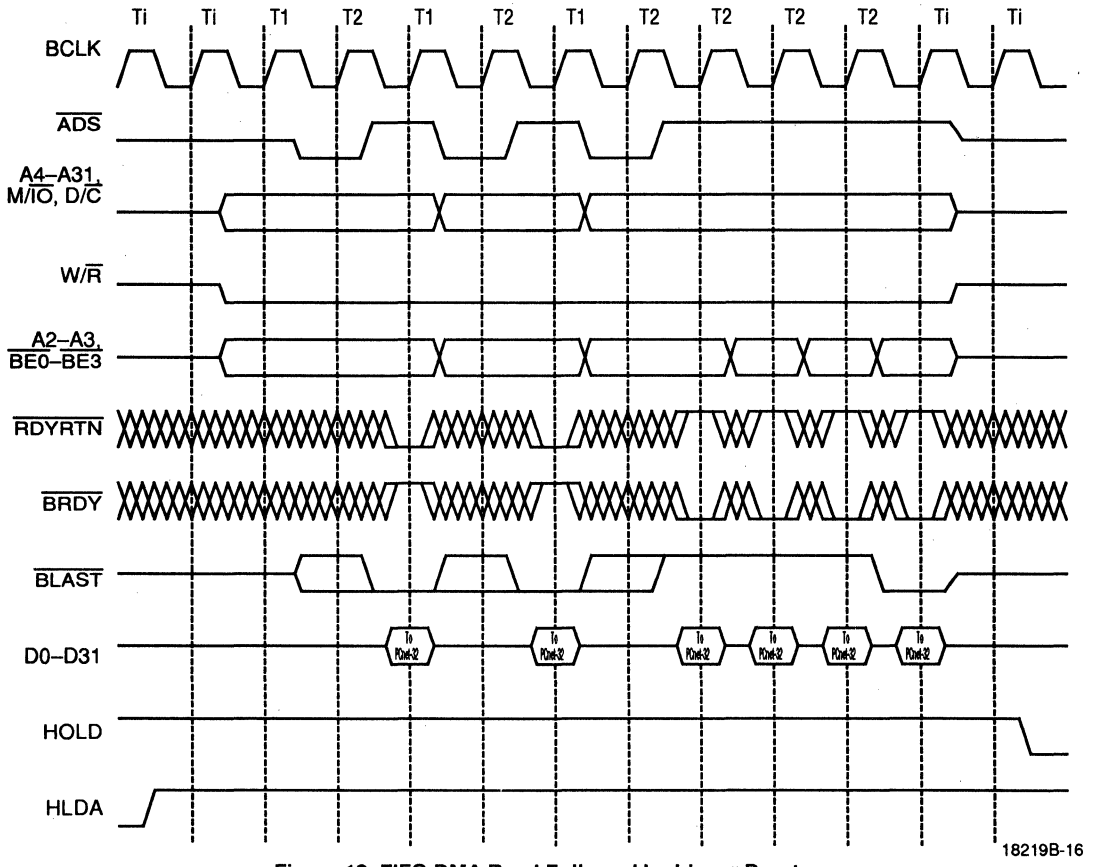


Figure 13. FIFO DMA Read Followed by Linear Burst Read During a Single Bus Mastership Period

Linear Burst DMA BLAST Signal Timing

Linear burst cycles are requested by the PCnet-32 controller by deasserting the BLAST signal (i.e. BLAST = 1). When BLAST is deasserted by the PCnet-32 controller, the slave device is under no obligation to provide BRDY. Instead, it may provide RDYRTN in response to each of the PCnet-32 controller transfers. If RDYRTN is asserted during accesses in which the PCnet-32 controller has deasserted BLAST, then the current transfer reverts to ordinary burst-cycle (see FIFO DMA Transfer section).

When BLAST is asserted, it signals the end of the current linear burst sequence.

In a cycle in which BLAST is asserted and following the assertion of either RDYRTN and/or BRDY by the slave device, the PCnet-32 controller may either relinquish the bus or it may initiate a new sequence of linear burst transfers without relinquishing the bus. If the PCnet-32 controller continues with a new sequence of linear burst transfers, the address asserted during the next T1 cycle will always be the next address in sequence from the previous T2 cycle. In other words, the PCnet-32 controller will never execute cycles within a single bus mastership period that are not both ascending and contiguous, with the exception of the descriptor DMA accesses described above.

The decision to continue bus ownership will depend upon several variables, including the state of the receive or transmit FIFO. All factors related to this decision are discussed later in this section.

Figure 14 illustrates a typical case of multiple linear burst sequences being performed during a single bus mastership period.

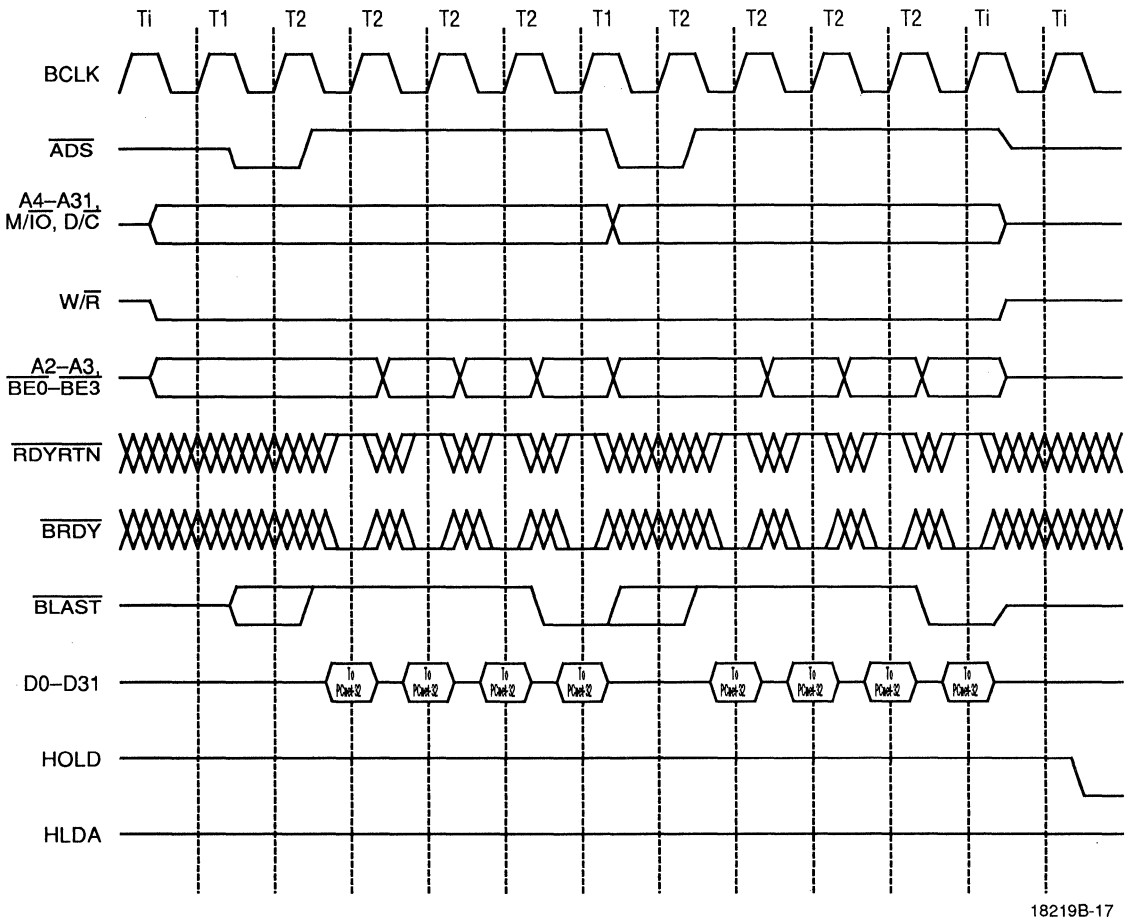


Figure 14. Linear Burst Read

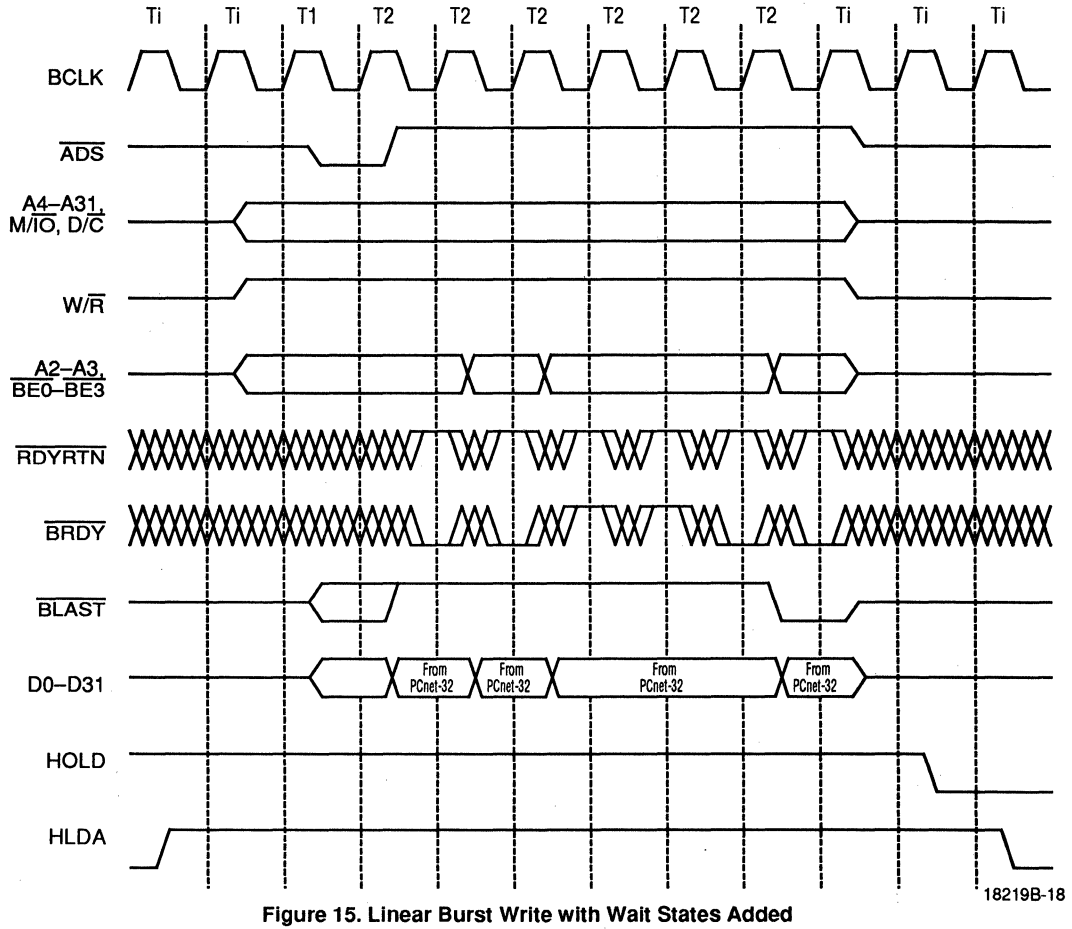


Figure 15. Linear Burst Write with Wait States Added by the Slave Device on the Third Transfer

Linear Burst DMA Ready Wait States

The PCnet-32 controller will insert wait states into linear burst DMA cycles if neither RDYRTN nor BRDY are sampled asserted at the end of each T2 cycle.

Interrupted Linear Burst DMA Cycles

The assertion of RDYRTN in the place of BRDY within a linear burst cycle will cause the linear burst to be "interrupted."

In that case, the PCnet-32 controller will revert to ordinary two-cycle transfers that contain both a T1 and a T2 cycle, except that BLAST will remain deasserted to show that linear bursting is still being requested by the PCnet-32 controller. This situation is defined as an interrupted linear burst cycle. If BRDY is sampled asserted (without also sampling RDYRTN asserted during the same access) during an interrupted linear burst cycle in which BLAST is deasserted, then linear bursting will resume. (Note that BLAST will become asserted during an interrupted linear burst cycle during the transfer that would have been the last transfer of the linear burst sequence, had the sequence not been interrupted.)

When an interrupted linear burst cycle is resumed, then the next assertion of ADS will depend upon the initial starting point of the linear burst, rather than on the resumption point.

For example, if the linear burst length = 4 (LINBC = 1), and BRDY is asserted during the first transfer, but RDYRTN is asserted on the second, then the PCnet-32 controller will revert to ordinary DMA transfers on the third transfer. If the responding device again asserts BRDY on the third access (while RDYRTN is deasserted), the PCnet-32 controller will resume linear bursting from the current address. It will produce 1 more data cycle before asserting the next ADS, i.e. PCnet-32 controller will keep track of the initial linear burst end point and will continue with the original linear burst after the RDYRTN interruption has occurred.

Figure 16 illustrates an example of an interrupted linear burst. Note that BLAST is asserted in the fourth transfer after the initial linear burst began, even though the linear burst was interrupted with the assertion of RDYRTN and a new ADS was driven for the third transfer. The external effect of the RDYRTN interruption is completely manifested in the insertion of the T1 cycle that contains the asserted ADS. The linear burst cycle is not affected in any other way.

Partial Linear Burst

Certain factors may cause the PCnet-32 controller to linearly burst fewer than the LINBC limit during a single linear burst sequence. BLAST will be asserted during the last data transfer. A linear burst that is terminated by BLAST before the LINBC limit is reached is called a partial linear burst.

A partial linear burst is recognizable in that the BLAST signal is asserted while fewer than the LINBC limit of transfers has been performed in the current linear burst sequence.

Factors that could generate a partial linear burst include:

- No more data available for transfers from the current transmit buffer
- No more data available for transfer from the receive FIFO *for this packet*
- No more space available for transfers to the current receive buffer

If any of these conditions occurs, then the PCnet-32 controller will end the Linear Burst by asserting BLAST. Typically, during the case of a master *read* operation (for transmit buffer transfers), the last transfer in the linear burst sequence will be the last transfer executed before the PCnet-32 controller releases the bus. This is true of both partial and completed linear burst sequences.

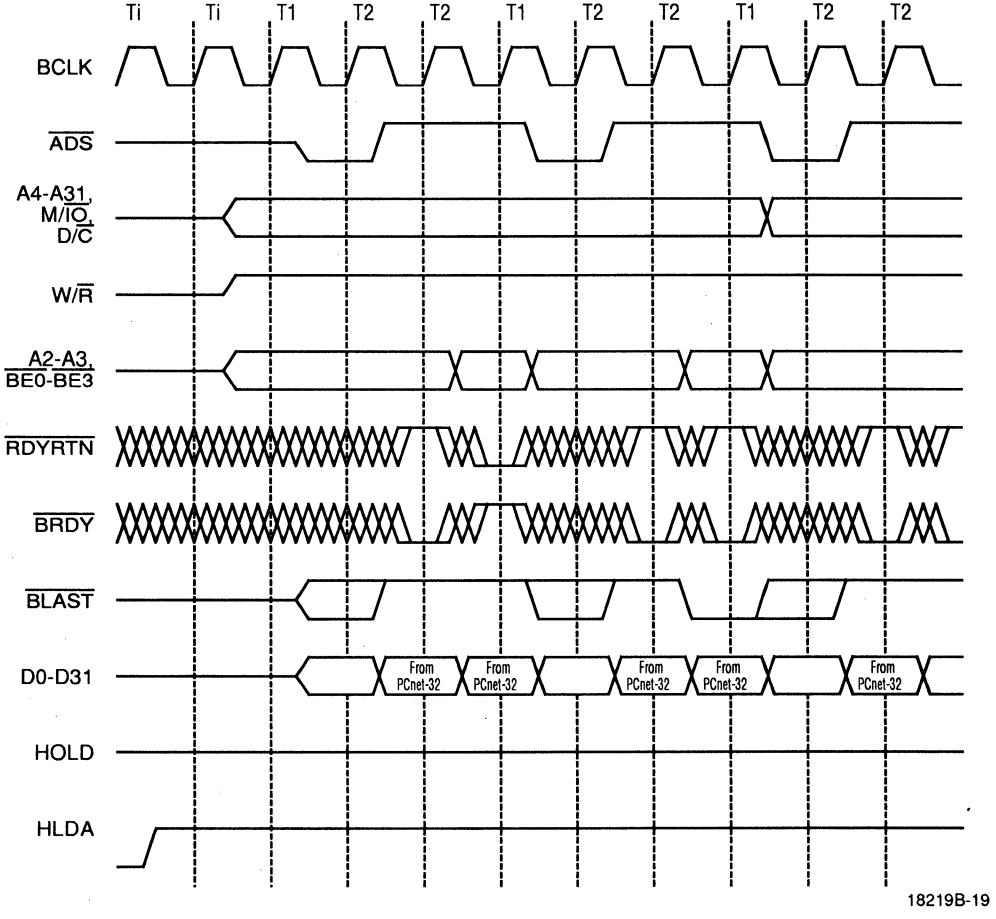


Figure 16. "Interrupted" Linear Burst Write

Typically, during the case of a master *write* operation (for receive buffer transfers) when receive packet data has ended, the last transfer in the linear burst sequence will be the last transfer executed before the PCnet-32 controller releases the bus. This is true of both partial and completed linear burst sequences.

However, if the *next transfer* that the PCnet-32 controller is scheduled to execute will be to the last available location of a receive or transmit buffer, then the PCnet-32 controller may assert $\overline{\text{BLAST}}$ on the current transfer and then use an ordinary cycle to make the last transfer to the buffer. This event occurs because of the restrictions placed upon the byte enable signals during the linear burst operation. As mentioned in the initial description of linear burst accesses (section Linear Burst DMA Transfers), all byte lanes of the data bus are always enabled during linear burst operations. Note, however, that in the case of the last buffer location, the PCnet-32 controller may own only a portion of the double word location. In such cases, it is necessary to discontinue linear burst accesses on the second from last buffer location so that an basic transfer with some byte lanes disabled can be used for the final transfer.

Figure 17 shows a partial linear burst that occurred while approaching the transfer of the last bytes of data to a receive buffer. The linear burst begins when 10 bytes of space still remain in the receive buffer. (The number of spaces remaining for the figure as drawn could be anywhere from 9 to 12 spaces. The value of 10 spaces has been chosen just for purposes of illustration.) After the first linear burst transfer, the PCnet-32 controller sees that between 6 bytes of space remain, and knowing that the second transfer will use another 4 bytes of space, the PCnet-32 controller is able to predict that the third transfer will be the last. Therefore, it asserts $\overline{\text{BLAST}}$ on the second transfer to terminate the linear burst operation. However, the PCnet-32 controller retains ownership of the bus so that it may, immediately thereon, make an basic transfer to the last two spaces in the buffer.

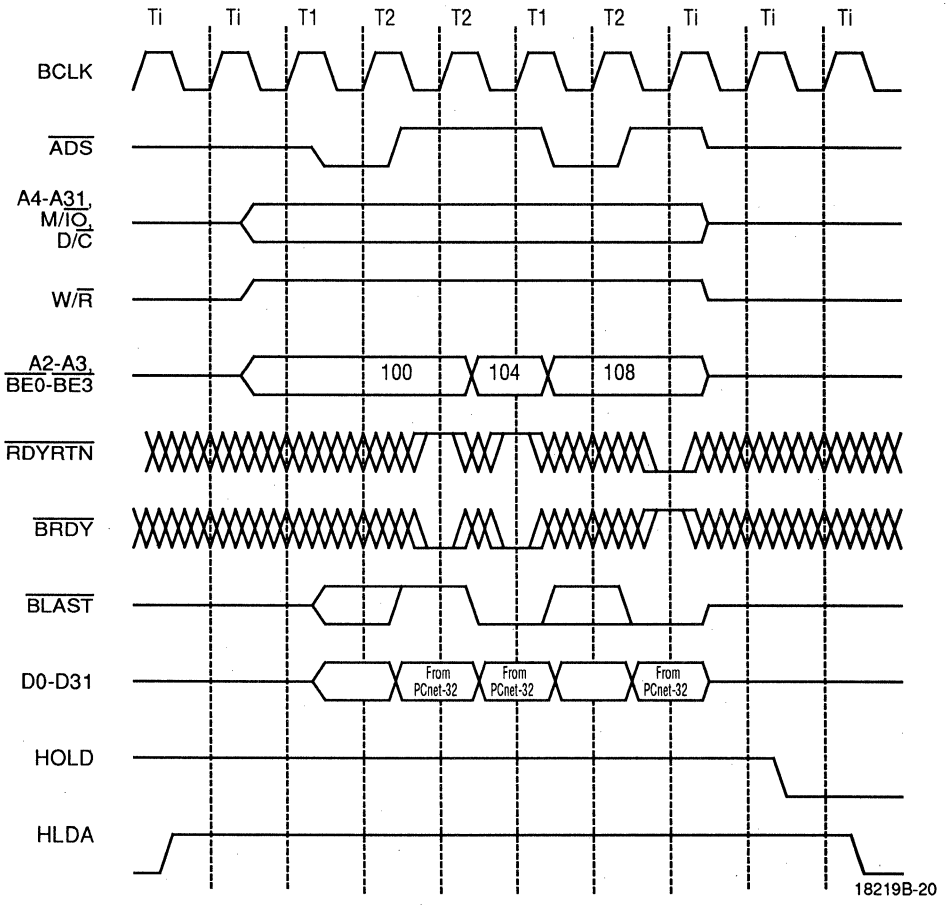


Figure 17. Typical Partial Linear Burst Write, Followed by a Basic Transfer During the Same Bus Mastership Period

Length of Bus Mastership Period

The number of data transfer cycles within the total bus mastership period is dependent on the programming of the DMAPLUS option (CSR4, bit 14). The possibilities are as follows:

If DMAPLUS = 0, a maximum of 16 transfers will be performed by default. This default value may be changed by writing to the burst register (CSR80). Note that DMAPLUS = 0 merely sets a maximum value. The minimum number of transfers in the burst will be determined by all of the following variables: the settings of the FIFO watermarks and the conditions of the FIFOs, the value of the DMA Burst Cycle (CSR80), the value of the DMA Bus Activity Timer (CSR82), and any occurrence of preemption that takes place during the burst.

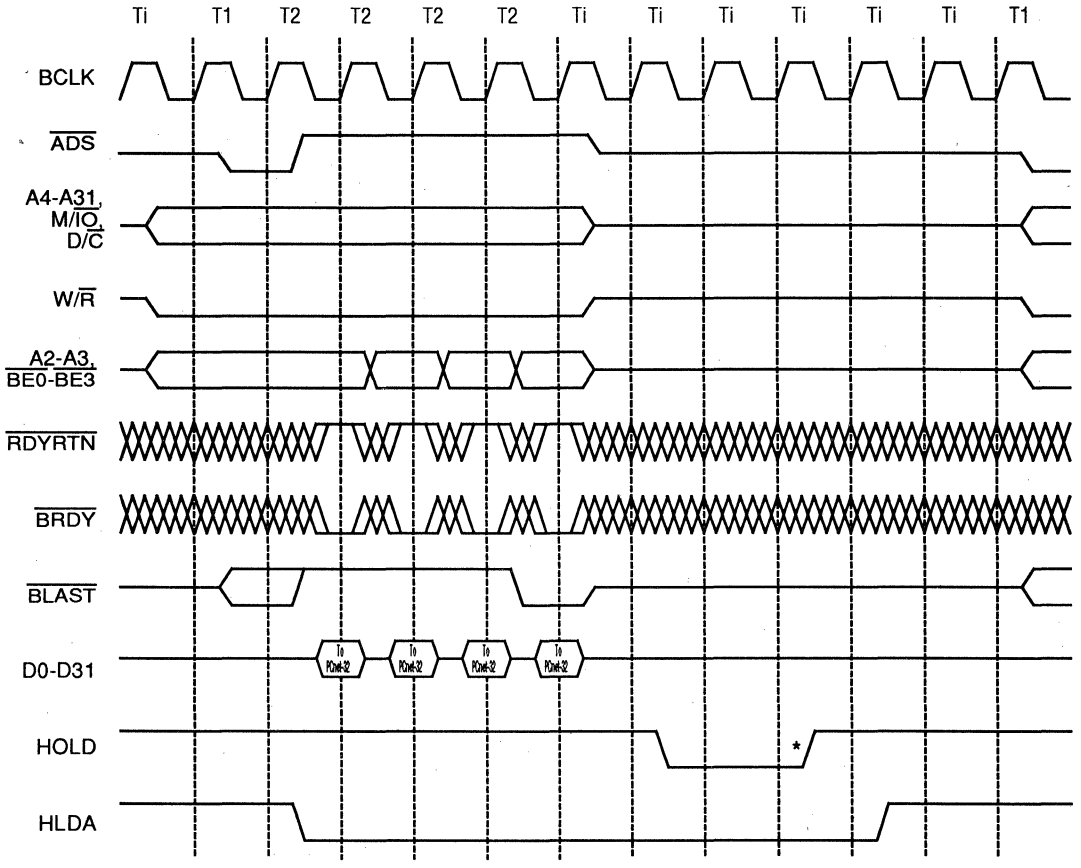
If DMAPLUS = 1, linear bursting will continue until the transmit FIFO is filled to its high threshold or the receive FIFO is emptied to its low threshold, or until the DMA Bus Activity Timer value (CSR82) has expired. A bus preemption event is another cause of termination of cycles. The FIFO thresholds are programmable (see description of CSR80), as are the Burst Cycle and Bus Activity Timer values. The exact number of total transfer cycles in the case of DMAPLUS = 1 will be dependent on the latency of the system bus to the PCnet-32 controller's mastership request and the speed of bus operation, but will be limited by the value in the Bus Activity Timer Register, the FIFO condition and by preemption occurrences, if any.

The exact response of the PCnet-32 controller to any of the conditions mentioned above can be complicated. For detail of the response to any particular stimulus, see each of the sections that describes PCnet-32 controller response.

Note that the number of transfer cycles between each $\overline{ADS}$ assertion will always only be controlled by LINBC, $\overline{RDYRTN}$, $\overline{BOFF}$, HLDA and FIFO conditions. The number of transfer cycles separating $\overline{ADS}$ assertions will not be affected by DMAPLUS or by the values in the Burst Cycle and Bus Activity Timer Register. However, these factors can influence the number of transfers that is performed during any given arbitration cycle.

Barring a time-out by the Burst Cycle or the Bus Activity Timer Register, or a bus preemption by another mastering device, the FIFO watermark settings and the extent of Bus Acknowledge latency will be the major factors in determining the number of accesses performed during any given arbitration cycle. The $\overline{BRDY}$ response time of the memory device will also affect the number of transfers, since the speed of the accesses will affect the state of the FIFO. (During accesses, the FIFO may be filling or emptying on the network end. For example, on a Receive operation, a slower device will allow additional data to accumulate inside of the FIFO. If the accesses are slow enough, a complete double word may become available before the end of the arbitration cycle and thereby increase the number of transfers in that cycle.) The general rule is that the longer the bus grant latency or the slower the bus transfer operations or the slower the clock speed or the higher the transmit watermark or the lower the receive watermark or any combination thereof, will produce longer total burst lengths.

If a bus preemption event occurs after the execution of the first T2 cycle of the fourth from the last transfer cycle within a linear burst DMA sequence, then the PCnet-32 controller will complete the current linear burst sequence and will execute a new linear burst sequence before releasing the HOLD signal and relinquishing the bus. If a bus preemption event occurs before or concurrent with the execution of the first T2 cycle of the fourth from the last transfer cycle within a linear burst DMA sequence, then the PCnet-32 controller will complete the current linear burst sequence and then will release the HOLD signal and will relinquish the bus. Within the context of this explanation, a single transfer cycle refers to the execution of a data transfer, regardless of the number of clock cycles taken, i.e. wait states are included in this definition of a transfer cycle. See Figure 18.



**Note*: HOLD will always go inactive for 2 BCLK cycles before being reasserted.

18219B-21

Figure 18. Linear Burst Read with Preemption During T2

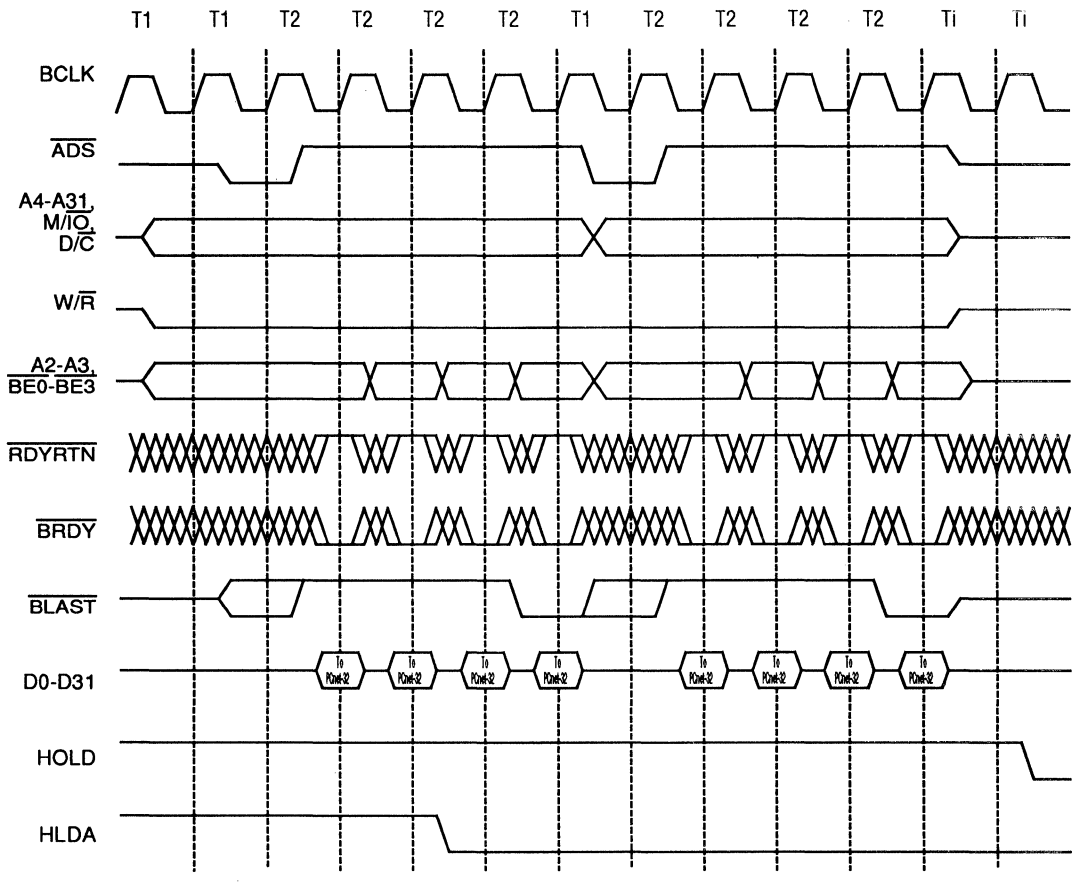
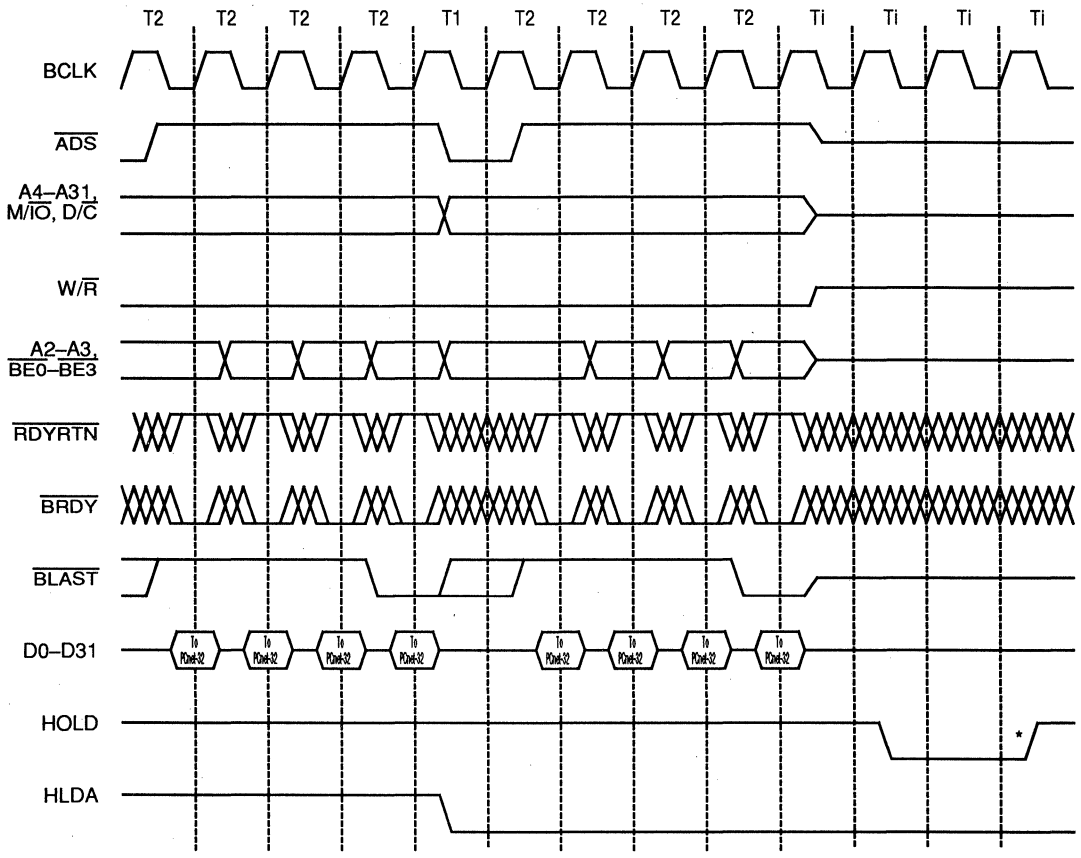


Figure 19. Linear Burst Read with Preemption During One of the Last Three T2 Cycles of the Sequence

If a bus preemption event occurs on a T1 cycle (specifically, a T1 cycle in which the $\overline{ADS}$ signal for a new linear burst sequence is asserted), then the next linear burst sequence will be executed before the PCnet-32 controller releases the HOLD signal and relinquishes the bus. (If the T1 cycle in which the preemption occurred was to begin an basic transfer, then the basic transfer plus as many as two additional basic transfers will be executed before relinquishing the bus.)



** **Note:** HOLD is always held inactive for 2 BCLK cycles before being reasserted.*

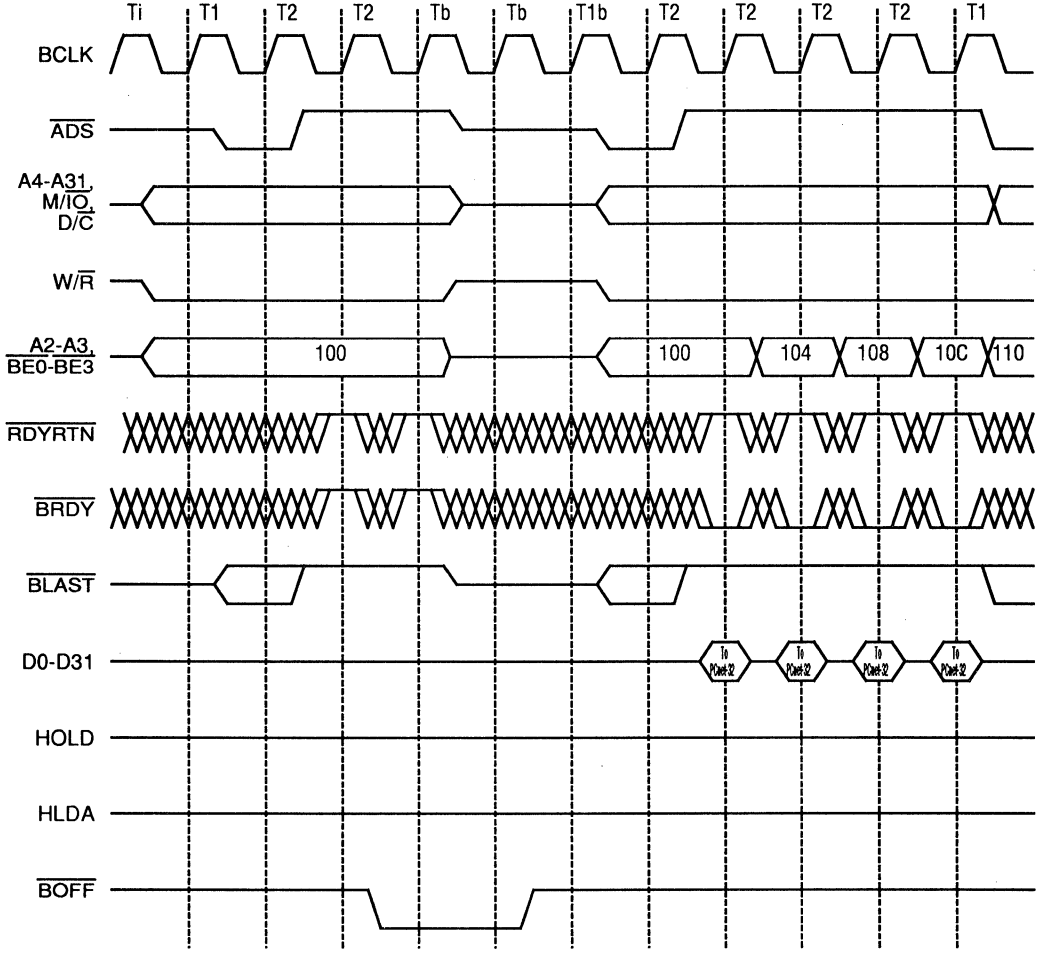
18219B-23

Figure 20. Linear Burst Read with Preemption that Occurs During the T1 Cycle

Effect of BOFF

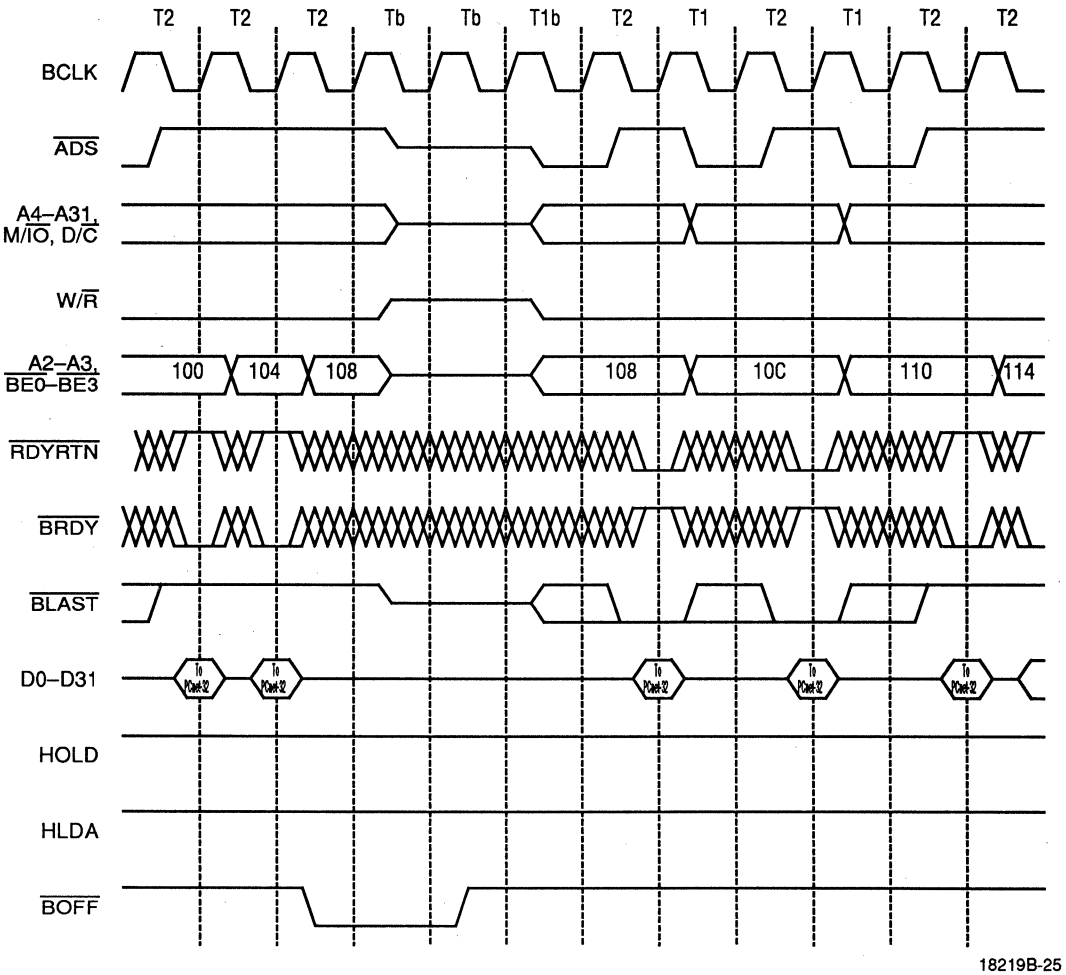
Assertion of BOFF during a linear burst has two possible outcomes. In the case of BOFF asserted ***before*** the first BRDY (or RDYRTN) has been sampled, the PCnet-32 controller will restart the linear burst after the BOFF event has ended. A new ADS will be asserted with the original starting address for the halted linear burst. See Figure 21.

However, if the BOFF signal is asserted ***after*** the first BRDY (or RDYRTN) has been sampled, the PCnet-32 controller will revert to ordinary burst-cycle accesses following the BOFF event. In this case, linear bursting will *next* occur when the memory address being accessed *next* meets the linear burst starting address requirements. If BOFF is sampled active on the same clock edge that BRDY or RDYRTN is sampled active, then the BOFF takes priority.



18219B-24

Figure 21. Restarted Linear Burst Read in which BOFF was Asserted Before BRDY of First Transfer in Linear Burst Sequency, Hence Linear Burst Sequence is Restarted When BOFF is Deasserted



18219B-25

Figure 22. Restarted Linear Burst Read in which BOFF was Asserted After BRDY of First Transfer in Linear Burst Sequency. Hence, Linear Burst Reverts to Ordinary Cycles Until Next Legal Linear Burst Starting Address is Reached

In general, if the linear burst is suspended by another bus master (either because of BOFF or PCnet-32 controller releasing HOLD) then any partially completed linear burst access will not resume when the PCnet-32 controller regains bus ownership. But if the PCnet-32 controller linear burst is interrupted by the receipt of RDYRTN in place of BRDY, then the PCnet-32 controller will resume the linear burst operation as indicated by the BLAST signal.

Register accesses cannot be performed to the PCnet-32 device while BOFF is asserted.

Effect of AHOLD

Assertion of AHOLD during Linear Burst transfers will cause the PCnet-32 controller to float some portion of the address bus beginning at the next clock cycle. If $\overline{BRDY}$ is returned while AHOLD is active, then the linear burst sequence will continue until the current burst would otherwise normally terminate, since the data bus and the lower portion of the address bus may remain active during AHOLD. However, a new linear burst sequence, requiring a new $\overline{ADS}$ assertion, will not be started while AHOLD is active.

When AHOLD is asserted during T1 of a linear burst, then the linear burst operation will be suspended until AHOLD is deasserted. Once AHOLD is deasserted, then the PCnet-32 controller will start the suspended linear burst with the intended address. See Figure 23. (Note that the intended T1 of the linear burst sequence has been labeled Ta in the figure, since a T1 was never executed due to the suspension of the address bus required by the assertion of AHOLD.)

When AHOLD is asserted in the middle of a linear burst, the linear burst may proceed without stalling or halting. AHOLD requires that PCnet-32 controller float a portion of its address bus, but linear burst data cycles will still proceed, since the AHOLD signal only affects a portion of the address bus, and that portion of the address bus is not being used for the middle accesses of a linear burst.

However, if AHOLD is asserted in the middle of a linear burst operation, and the AHOLD signal is held long enough that a new linear burst sequence will start (a new $\overline{ADS}$ is to be issued by the PCnet-32 controller) then at the end of the current linear sequence, the PCnet-32 controller must wait for the AHOLD signal to become inactive before beginning the next linear sequence, since the AHOLD signal would now interfere with the PCnet-32 controller's wish to assert $\overline{ADS}$ and a new address on the entire address bus. During the time that the PCnet-32 controller is waiting for the release of the AHOLD signal, the PCnet-32 controller will continue to drive the command signals, but $\overline{ADS}$ will be driven inactive.

Note that if $\overline{RDYRTN}$ is asserted during a linear burst sequence while AHOLD is active, then no more access will be performed until AHOLD is deasserted. This is because the assertion of $\overline{RDYRTN}$ will cause the PCnet-32 controller to insert a new T1 cycle into the linear burst. A T1 cycle requires assertion of $\overline{ADS}$, but $\overline{ADS}$ assertion is not allowed as long as AHOLD is still asserted. Therefore, the T1 cycle is delayed until the AHOLD is deasserted.

The portion of the Address Bus that will be floated at the time of an address hold operation will be determined by the value of the Cache Line Length register (BCR18, bits 15–11). Table 23 lists all of the legal values of CLL showing the portion of the Address Bus that will become floated during an address hold operation.

Table 23. CLL Value of Floated Address in AHOLD

CLL Value	Floated Portion of Address Bus During AHOLD
00000	None
00001	A31–A2
00010	A31–A3
00011	Reserved CLL Value
00100	A31–A4
00101–00111	Reserved CLL Values
01000	A31–A5
01001–01111	Reserved CLL Values
10000	A31–A6
10001–11111	Reserved CLL Values

Note that the default value of CLL after H_RESET is 00100. All timing diagrams in this document are drawn with the assumption that this is the value of CLL.

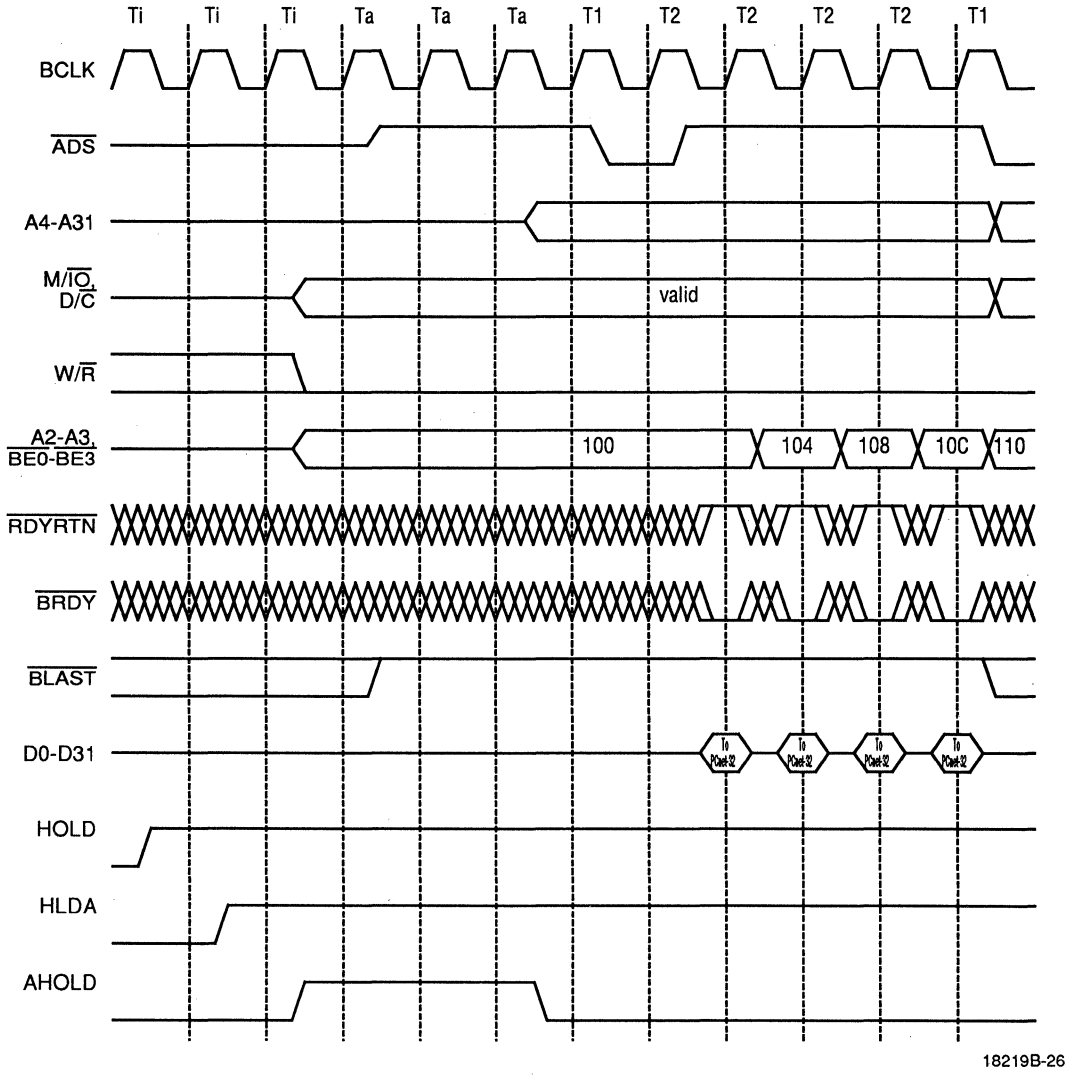


Figure 23. Linear Burst Read in which AHOLD was Asserted During T1 in the Linear Burst Sequence; Linear Burst Sequence is Started when AHOLD is Deasserted

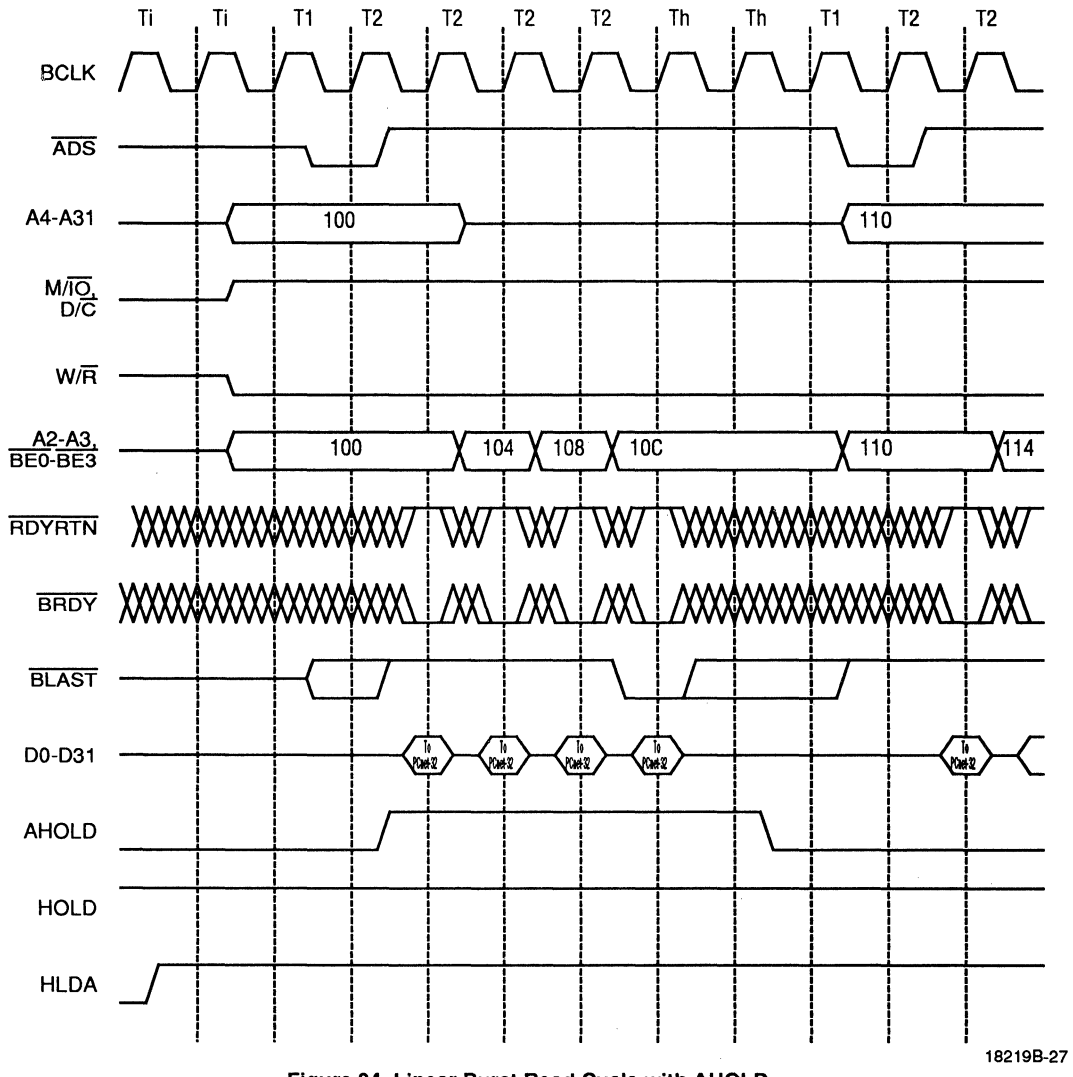


Figure 24. Linear Burst Read Cycle with AHOLD

Note that Linear Burst sequence is allowed to complete in spite of AHOLD, but next linear Burst sequence is prevented from beginning until AHOLD is deasserted.

Note that if the $\overline{BRDY}$ (or $\overline{RDYRTN}$) signal is not returned while AHOLD is active, then the PCnet-32 controller will resume driving the same address onto the address bus when AHOLD is released. The PCnet-32 controller will not reissue the $\overline{ADS}$ signal at this time.

Bus Activity Timer Register Time Out During Linear Burst

When the Bus Activity Timer Register (CSR82 bits [15:0]) times out before or concurrent with the execution of the first T2 cycle of the fourth from the last transfer cycle within a linear burst DMA sequence, then the linear burst will continue until a legal starting address is reached, and then the PCnet-32 controller will relinquish the bus. If the Bus Activity Timer Register times out after the execution of the first T2 cycle of the fourth from the last transfer cycle within a linear burst DMA sequence, then the PCnet-32 controller will complete the current linear burst sequence and will execute a new linear burst sequence before releasing the HOLD signal and relinquishing the bus. (Effectively, the Bus Activity Timer time-out is treated in a manner identical to the occurrence of a preemption event.) Therefore, when pro-

grammed for Linear Burst mode, the PCnet-32 controller bus mastership time may exceed the Bus Activity Timer limit.

This is done because an immediate abort of the linear burst due to timer expiration would leave the current buffer pointer at an unaligned location. This would cause an address alignment of several ordinary cycles to be executed during the next FIFO DMA operation. Repeated occurrences of this nature would compromise the usefulness of the linear burst mode, since this would increase the number of non-linear burst cycles that are performed. This in turn would increase the bus bandwidth requirement of the PCnet-32 controller. Therefore, because the PCnet-32 controller Linear Burst operation does not strictly obey the Burst Timer, the user should program the Burst Timer value in such a manner as to include the expected linear burst release time. If the user has enabled the Linear Burst function, and wishes the PCnet-32 controller to limit bus activity to MAX_TIME µs, then the Burst Timer should be programmed to a value of:

MAX_TIME- [((3+lbs) x w + 10 + lbs) x (BCLK period)]

This is because the PCnet-32 controller may use as much as one "linear burst size" plus three transfers in order to complete the linear burst before releasing the bus.

As an example, if the linear burst size is 4 transfers, and the number of wait states for the system memory is 2, and the BCLK period is 30 ns and the MAX time allowed on the bus is 3 µs, then the Burst Timer should be programmed for:

MAX_TIME- [((3+lbs) x w + 10 + lbs)
x (BCLK period)];

3 µs – [(3 + 4) x 2 +10 + 4) x (30 ns)]
= 3 µs – (28 x 30 ns) = 3 – 0.84 µs = 2.16 µs.

Then, if the PCnet-32 controller's Burst Timer times out after 2.16 µs when the PCnet-32 controller has completed all but the last three transfers of a linear burst, then the PCnet-32 controller *may* take *as much as* 0.84 µs to complete the bursts and release the bus. The bus release will occur at 2.16 + 0.84 = 3 µs.

Burst Cycle Time Out During Linear Burst

When the Burst Cycle (CSR80 bits [7:0]) times out in the middle of a linear burst, the linear burst will continue until a legal starting address is reached, and then the PCnet-32 controller will relinquish the bus.

The discussion for the Burst Cycle is identical to the discussion for the Bus Activity Timer Register, except that the quantities are in terms of transfers instead of in terms of time.

The equation for the proper burst register setting is:

Burst count setting = (desired_max DIV (length of linear burst in transfers)) x length of linear burst in transfers,

where DIV is the operation that yields the INTEGER portion of the ÷ operation.

Illegal Combinations of Watermark and LINBC

Certain combinations of watermark programming and LINBC programming may create situations where no linear bursting is possible, or where the FIFO may be excessively read or excessively written. Such combinations are declared as illegal.

Combinations of watermark settings and LINBC settings must obey the following relationship:

watermark (in bytes) ≥ LINBC (in bytes)

Combinations of watermark and LINBC settings that violate this rule may cause unexpected behavior.

Slave Timing

Slave timing in the PCnet-32 controller is designed to perform to both Am486 32-bit timing requirements and VESA VL-Bus timing requirements at the same time. Since the VESA VL-Bus is based upon Am486 bus timing, there is really little difference evident, except for hold-off requirements on the part of the slave driving the $\overline{RDY}$, $\overline{BRDY}$, and data signals when the high speed write signal is false. VESA VL-Bus requires that none of these signals are driven by the slave until the second T2 cycle when the high speed write signal is false. Since the PCnet-32 controller does not examine the high-speed write bit, it assumes that this signal is never true, and therefore always obeys the more stringent requirement of not being allowed to drive $\overline{RDY}$, $\overline{BRDY}$ and the data bus until the second T2. In addition, the PCnet-32 controller will drive $\overline{RDY}$ and $\overline{BRDY}$ inactive for one half BCLK cycle at the end of the slave access, immediately following the BCLK cycle in which the PCnet-32 controller asserted $\overline{RDY}$. Again, this behavior is required by the VESA VL-Bus specification, but it is not required for operation within an Am486 system. The PCnet-32 controller performs in this manner, regardless of the PCnet-32 controller mode setting.

Slave timing can generally be inferred from the bus master timing diagrams, with the exception of the following information:

PCnet-32 controller never responds with $\overline{BRDY}$ active during slave accesses. All PCnet-32 controller slave responses use only the $\overline{RDY}$ signal. $\overline{BRDY}$ will always be deasserted during all PCnet-32 controller slave accesses. $\overline{RDY}$ is a PCnet-32 controller output signal. It is used during PCnet-32 controller slave accesses. $\overline{RDYRTN}$ is a PCnet-32 controller input signal. It is used during all PCnet-32 controller bus master accesses, as well as during PCnet-32 controller slave read accesses.

The typical number of wait states added to a slave access on the part of the PCnet-32 controller is 6 or 7 BCLK cycles, depending upon the relative phases of the internal Buffer Management Unit clock and the BCLK

signal, since the internal Buffer Management Unit clock is a divide-by-two version of the BCLK signal.

The PCnet-32 controller $\overline{RDY}$ and $\overline{RDYRTN}$ signals may be wired together. This allows the PCnet-32 controller to operate within a system that has a single $\overline{READY}$ signal.

The $\overline{LDEV}$ signal is generated in response to a valid PCnet-32 controller I/O address on the bus together with a valid $\overline{ADS}$ signal. $\overline{LDEV}$ is generated in an asynchronous manner by the PCnet-32 controller. See the parameter listings for delay values of the $\overline{LDEV}$ signal.

$\overline{RDY}$, $\overline{BRDY}$ and D[31:0] are never driven until the second T2 state of a slave access. Before that time, it is expected that a system pull-up device is holding the $\overline{RDY}$ and $\overline{BRDY}$ signals in a deasserted state.

The $\overline{RDY}$ and $\overline{BRDY}$ signals are always driven high for one half BCLK cycle immediately following the BCLK period during which $\overline{RDY}$ was driven asserted. Then the $\overline{RDY}$ and $\overline{BRDY}$ signals are floated. This behavior is performed regardless of the PCnet-32 controller mode setting. See Figure 25.

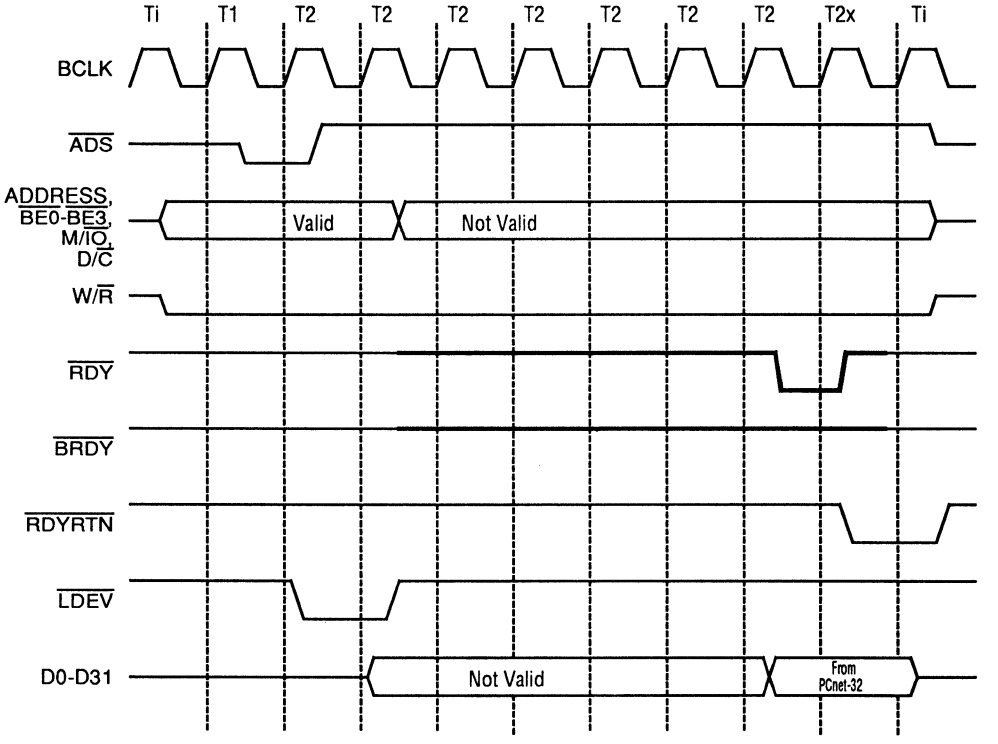


Figure 25. Slave $\overline{RDY}$ Timing

VESA VL-Bus Mode Timing

VESA VL-Bus mode functional timing is essentially identical to the timing of the Am486 32-bit mode, except that the bus request and bus acknowledge signals have inverted senses from those shown in the previous timing diagrams and the AHOLD signal does not exist while the PCnet-32 controller is programmed for VL-Bus mode. In addition, dynamic bus sizing is supported in VESA VL-Bus mode, through the use of the $\overline{LBS16}$ signal. The following section describes possible $\overline{LBS16}$ interactions while programmed for the VESA VL-Bus mode of operation. Other differences exist between VL-Bus mode and Am486 mode, but these other differences are not directly related to the master or slave cycle timings.

Effect of $\overline{LBS16}$ (VL-Bus mode only)

Dynamic bus sizing is recognized by the PCnet-32 controller while operating in the VL-Bus mode. The $\overline{LBS16}$ signal is used to indicate to the PCnet-32 controller whether the VL-Bus target is a 16-bit or 32-bit peripheral. When the target device indicates that it is 16 bits in width by asserting the $\overline{LBS16}$ signal at least one LCLK

period before asserting the $\overline{BRDY}$ signal, then the PCnet-32 controller will dynamically respond to the size constraints of the peripheral by performing additional accesses. Table 24 indicates the sequence of accesses that will be performed by the PCnet-32 controller in response to the assertion of $\overline{LBS16}$.

Table 24. Data Transfer Sequence from 32-Bit Wide to 16-Bit Wide

Current Access				Next with $\overline{LBS16}$			
$\overline{BE3}$	$\overline{BE2}$	$\overline{BE1}$	$\overline{BE0}$	$\overline{BE3}$	$\overline{BE2}$	$\overline{BE1}$	$\overline{BE0}$
1	1	1	0	NR*			
1	1	0	0	NR*			
1	0	0	0	1	0	1	1
0	0	0	0	0	0	1	1
1	1	0	1	NR*			
1	0	0	1	1	0	1	1
0	0	0	1	0	0	1	1
1	0	1	1	NR*			
0	0	1	1	NR*			
0	1	1	1	NR*			

**NR = No second access Required for these cases*

Figure 26 shows an example of an exchange between a 16-bit VL-Bus peripheral and the PCnet-32 controller during ordinary read cycles while programmed for VL-Bus mode of operation. Note that the $\overline{LBS16}$ signal is asserted during the LCLK that precedes the assertion of $\overline{RDYRTN}$. In this particular case, in order to maintain zero-wait state accesses, the 16-bit target must generate $\overline{LBS16}$ in a very short time in order to meet the required setup time of $\overline{LBS16}$ into the PCnet-32 controller. If the peripheral were incapable of meeting the required setup time, then a wait state would be needed in order to insure that $\overline{LBS16}$ is asserted at least one LCLK prior to the assertion of the $\overline{RDYRTN}$ signal. This situation is illustrated in the second double word access of the diagram. The wait state only needs to be inserted on the first access of the sequence, since from that point on, $\overline{LBS16}$ could be held active low until the entire double word transfer had completed, thus adequately satisfying the $\overline{LBS16}$ setup requirement for the second $\overline{RDYRTN}$ assertion. Note that only two bytes of data are transferred during each T2 cycle so that the total number of bytes transferred during each access is four.

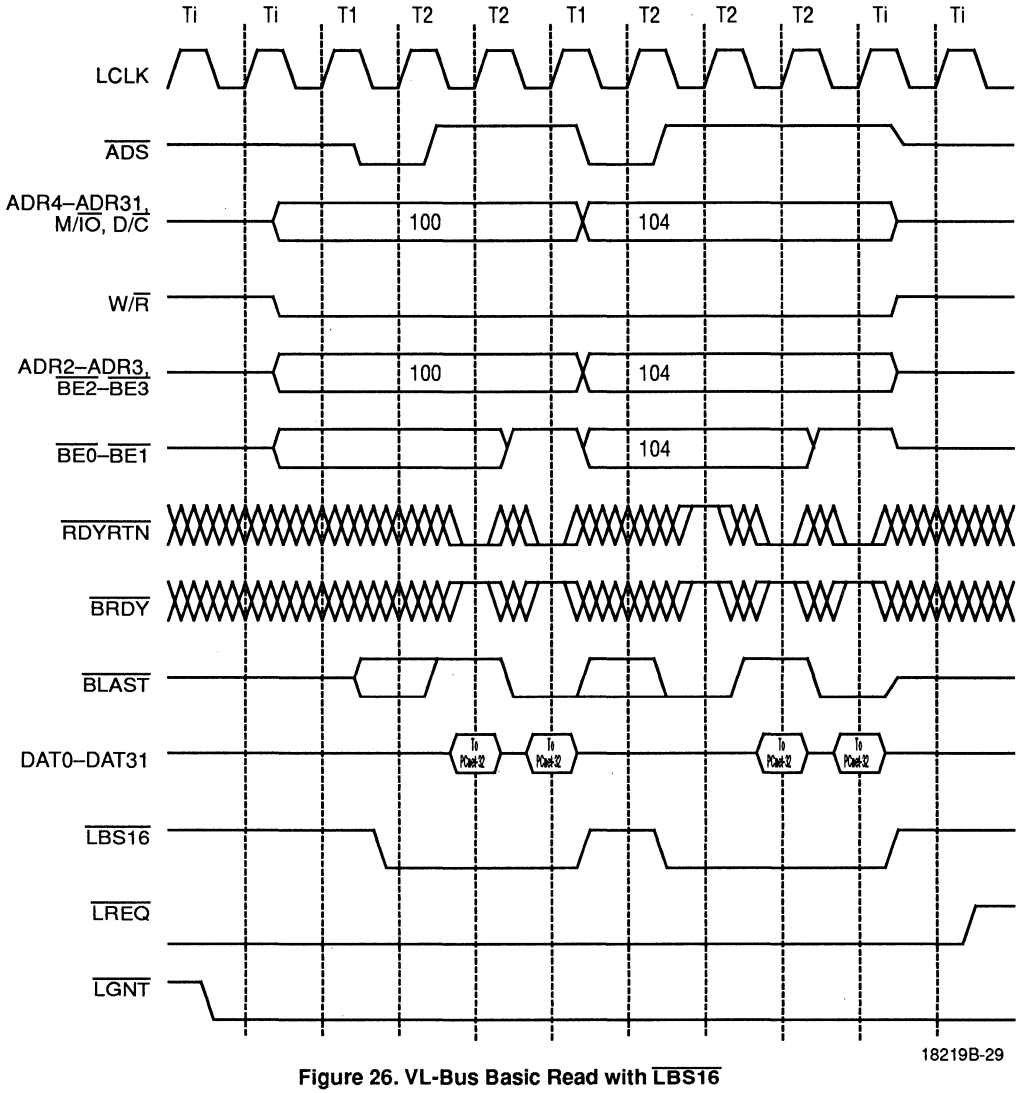


Figure 26. VL-Bus Basic Read with $\overline{LBS16}$

Figure 27 shows an example of an exchange between a 16-bit VL-Bus peripheral and the PCnet-32 controller during linear burst mode while programmed for VL-Bus mode of operation. Note that the $\overline{LBS16}$ signal is asserted during the LCLK that precedes the assertion of $\overline{BRDY}$. In this particular case, in order to maintain zero-wait state accesses, the 16-bit target must generate $\overline{LBS16}$ in a very short time in order to meet the required setup time of $\overline{LBS16}$ into the PCnet-32 controller. If the peripheral were incapable of meeting the required setup time, then a wait state would be needed in order to insure that $\overline{LBS16}$ is asserted at least one LCLK prior to the assertion of the $\overline{BRDY}$ signal. For linear burst sequences, this wait state would only need to be inserted on the first access of the sequence, since from that point on, $\overline{LBS16}$ could be held active low until the entire sequence had completed, thus adequately satisfying the $\overline{LBS16}$ setup requirement for each subsequent $\overline{BRDY}$ assertion. Note that only 2 bytes of data are transferred during each T2 cycle so that the total number of bytes transferred during the linear burst sequence is 16, even though 8 T2 cycles are executed.

When the assertion of $\overline{LBS16}$ during a PCnet-32 controller master access has created the need for a second access as specified in the table above, **and** the $\overline{WBACK}$ signal becomes active during the second access, then when $\overline{WBACK}$ is deasserted, the PCnet-32 controller will repeat both accesses of the pair.

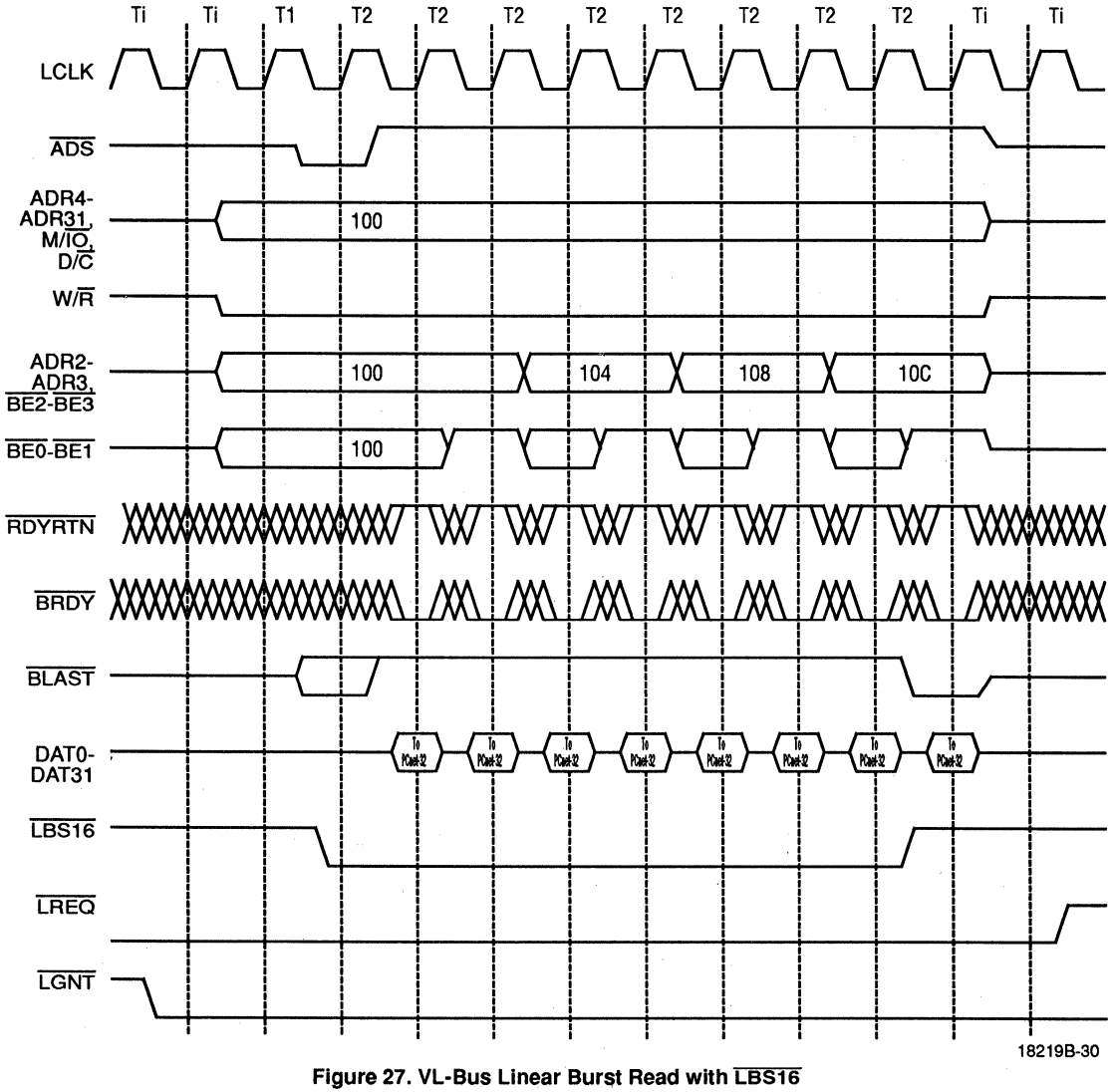


Figure 27. VL-Bus Linear Burst Read with $\overline{LBS16}$

Bus Master and Bus Slave Data Byte Placement

The general rule of data placement is that the active data byte lanes are indicated by the byte enable signals for all transfers. Note that during all master read operations, the PCnet-32 controller will always activate all byte enables, even though some byte lanes may not contain "valid" data as indicated by a buffer pointer value. In such instances, the PCnet-32 controller will internally discard unneeded bytes.

Note that in all 32-bit environments, regardless of the mode settings, the placement of data bytes on the data bus during all PCnet-32 controller bus operations (master and slave) will proceed in accordance with the data byte duplication rules of the Am386DX. The Am386DX requirement is for duplication of active data bytes in corresponding lower-half byte lanes when the access is a byte or word access that utilizes the upper half of the data bus. PCnet-32 controller performs data byte duplication in this manner. The Am386DX does not indicate byte duplication when the active data bytes of a byte or word access are exclusively contained in the lower half of the data bus, therefore, the PCnet-32 controller will not perform data byte duplication in this case. Byte duplication for bus master writes and bus slave reads will follow Table 25.

A[1:0] in the table refer to software pointers, since A[1:0] pins do not physically exist in the system. (Software pointers include I/O address software pointers in the driver code for I/O accesses to the PCnet-32 controller, or software pointers for the initialization block, descriptor areas or buffer areas that are used by the PCnet-32 controller during master accesses.)

For master read operations, the PCnet-32 controller expects data according to the byte enable signaling only. Byte lanes with inactive byte enables are expected to carry invalid data.

For slave write operations, the PCnet-32 controller expects data according to the byte enable signaling only. Byte lanes with inactive byte enables are expected to carry invalid data.

For master write operations, the PCnet-32 controller will produce data as indicated in Table 25.

For slave read operations, the PCnet-32 controller will produce data as indicated for the BSWP = 0 cases in Table 25, regardless of the actual setting of the BSWP bit.

Table 25. Master and Slave Byte Placement

Case No.	A[1:0]	BSWP	$\overline{BE3}$–$\overline{BE0}$	D[31:24]	D[23:16]	D[15:8]	D[7:0]
1a	00	0	0000	Byte3	Byte2	Byte1	Byte0
1b	00	1	0000	Byte0	Byte1	Byte2	Byte3
2a	01	0	0001	Byte2	Byte1	Byte0	Undef
2b	01	1	1000	Undef	Byte0	Byte1	Byte2
3a	10	0	0011	Byte1	Byte0	Copy1	Copy0
3b	10	1	1100	Undef	Undef	Byte0	Byte1
4a	11	0	0111	Byte0	Undef	Copy0	Undef
4b	11	1	1110	Undef	Undef	Undef	Byte0
5a	00	0	1000	Undef	Byte2	Byte1	Byte0
5b	00	1	0001	Byte0	Byte1	Byte2	Undef
6a	01	0	0001	Byte2	Byte1	Byte0	Undef
6b	01	1	1000	Undef	Byte0	Byte1	Byte2
7a	10	0	0011	Byte1	Byte0	Copy1	Copy0
7b	10	1	1100	Undef	Undef	Byte0	Byte1
8a	11	0	0111	Byte0	Undef	Copy0	Undef
8b	11	1	1110	Undef	Undef	Undef	Byte0
9a	00	0	1100	Undef	Undef	Byte1	Byte0
9b	00	1	0011	Byte0	Byte1	Copy0	Copy1
10a	01	0	1001	Undef	Byte1	Byte0	Undef
10b	01	1	1001	Undef	Byte0	Byte1	Undef
11a	10	0	0011	Byte1	Byte0	Copy1	Copy0
11b	10	1	1100	Undef	Undef	Byte0	Byte1
12a	11	0	0111	Byte0	Undef	Copy0	Undef
12b	11	1	1110	Undef	Undef	Undef	Byte0
13a	00	0	1110	Undef	Undef	Undef	Byte0
13b	00	1	0111	Byte0	Undef	Copy0	Undef
14a	01	0	1101	Undef	Undef	Byte0	Undef
14b	01	1	1011	Undef	Byte0	Undef	Copy0
15a	10	0	1011	Undef	Byte0	Undef	Copy0
15b	10	1	1101	Undef	Undef	Byte0	Undef
16a	11	0	0111	Byte0	Undef	Copy0	Undef
16b	11	1	1110	Undef	Undef	Undef	Byte0

Note that cases 10, 12, and 15 will not normally be produced during PCnet-32 controller bus master operations. These cases will only occur during bus master operations if the software programs an extremely short buffer size into the descriptor BCNT field, where extremely short means exactly 4, 3, 2 or 1 bytes in length.

BSWP = 0 corresponds to little Endian byte ordering.
BSWP = 1 corresponds to big Endian byte ordering.

Buffer Management Unit

The buffer management unit is a micro-coded state machine which implements the initialization procedure and manages the descriptors and buffers. The buffer management unit operates at a speed of BCLK ÷2.

Initialization

PCnet-32 controller initialization includes the reading of the initialization block in memory to obtain the operating parameters. The initialization block must be located on a double word (4-byte) address boundary, regardless of the setting of the SSIZE32, (CSR58[8]/BCR20[8]) bit. The initialization block is read when the INIT bit in CSR0 is set. The INIT bit should be set before or concurrent with the STRT bit to insure correct operation. Two doublewords are read during each period of bus mastership. When SSIZE32 = 1 (CSR58[8]/ BCR20[8]) , this results in a total of 4 arbitration cycles (3 arbitration cycles if SSIZE32 = 0). Once the initialization block has been completely read in and internal registers have been updated, IDON will be set in CSR0, and an interrupt generated (if IENA is set). At this point, the BMU knows where the receive and transmit descriptor rings and hence, normal network operations will begin.

The Initialization Block is vectored by the contents of CSR1 (least significant 16 bits of address) and CSR2 (most significant 16 bits of address). The block contains the user defined conditions for PCnet-32 controller operation, together with the base addresses and length information of the transmit and receive descriptor rings.

There is an alternative method to initialize the PCnet-32 controller. Instead of initialization via the initialization block in memory, data can be written directly into the appropriate registers. Either method may be used at the discretion of the programmer. If the registers are written to directly, the INIT bit must not be set, or the initialization block will be read in, thus overwriting the previously written information. Please refer to Appendix C for details on this alternative method.

Re-Initialization

The transmitter and receiver sections of the PCnet-32 controller can be turned on via the initialization block (MODE Register DTX, DRX bits; CSR15[1:0]). The states of the transmitter and receiver are monitored by the host through CSR0 (RXON, TXON bits). The PCnet-32 controller should be reinitialized if the transmitter and/or the receiver were not turned on during the original initialization, and it was subsequently required to activate them or if either section was shut off due to the detection of an error condition (MER, UFLO, TX BUFF error).

Re-initialization may be done via the initialization block or by setting the STOP bit in CSR0, followed by writing to CSR15, and then setting the START bit in CSR0. Note that this form of restart will not perform the same in the PCnet-32 controller as in the LANCE. In particular, upon restart, the PCnet-32 controller reloads the transmit and receive descriptor pointers with their respective base addresses. This means that the software must clear the descriptor own bits and reset its descriptor ring pointers before the restart of the PCnet-32 controller. The reload of descriptor base addresses is performed in the LANCE only after initialization, so a restart of the LANCE without initialization leaves the LANCE pointing at the same descriptor locations as before the restart.

Buffer Management

Buffer management is accomplished through message descriptor entries organized as ring structures in memory. There are two rings, a receive ring and a transmit ring. The size of a message descriptor entry is 4 doublewords, or 16 bytes, when SSIZE32 = 1. The size of a message descriptor entry is 4 words, or 8 bytes, when SSIZE32 = 0 (CSR58[8]/BCR20[8]).

Descriptor Rings

Each descriptor ring must be organized in a contiguous area of memory. At initialization time (setting the INIT bit in CSR0), the PCnet-32 controller reads the user-defined base address for the transmit and receive descriptor rings, as well as the number of entries contained in the descriptor rings. Descriptor ring base addresses must be on a 16-byte boundary when SSIZE32 = 1, or on an 8-byte boundary when SSIZE32 = 0. A maximum of 128 (or 512, depending upon the value of SSIZE32) ring entries is allowed when the ring length is set through the TLEN and RLEN fields of the initialization block. However, the ring lengths can be set beyond this range (up to 65535) by writing the transmit and receive ring length registers (CSR76, CSR78) directly.

Each ring entry contains the following information:

1. The address of the actual message data buffer in user or host memory
2. The length of the message buffer
3. Status information indicating the condition of the buffer

To permit the queuing and de-queuing of message buffers, ownership of each buffer is allocated to either the PCnet-32 controller or the host. The OWN bit within the descriptor status information, either TMD or RMD (see section on TMD or RMD), is used for this purpose. OWN = "1" signifies that the PCnet-32 controller currently has ownership of this ring descriptor and its associated buffer. Only the owner is permitted to relinquish ownership or to write to any field in the descriptor entry. A device that is not the current owner of a descriptor entry cannot assume ownership or change any field in the entry. A device may, however, read from a descriptor that it does not currently own. Software should always read descriptor entries in sequential order. When software finds that the current descriptor is owned by the PCnet-32 controller, then the software must not read "ahead" to the next descriptor. The software should wait at the unOWNed descriptor until ownership has been granted to the software (when SPRINTEN = 1 (CSR3,

bit5), then this rule is modified. See the SPRINTEN description). Strict adherence to these rules insures that "Deadly Embrace" conditions are avoided.

Descriptor Ring Access Mechanism

At initialization, the PCnet-32 controller reads the base address of both the transmit and receive descriptor rings into CSRs for use by the PCnet-32 controller during subsequent operations.

As the final step in the self-initialization process, the base address of each ring is loaded into each of the current descriptor address registers and the address of the next descriptor entry in the transmit and receive rings is computed and loaded into each of the next descriptor address registers.

When SSIZE32 = 0, software data structures are 16 bits wide. Figure 28 illustrates the relationship between the Initialization Base Address, the Initialization Block, the Receive and Transmit Descriptor Ring Base Addresses, the Receive and Transmit Descriptors and the Receive and Transmit Data Buffers, for the case of SSIZE32 = 0.

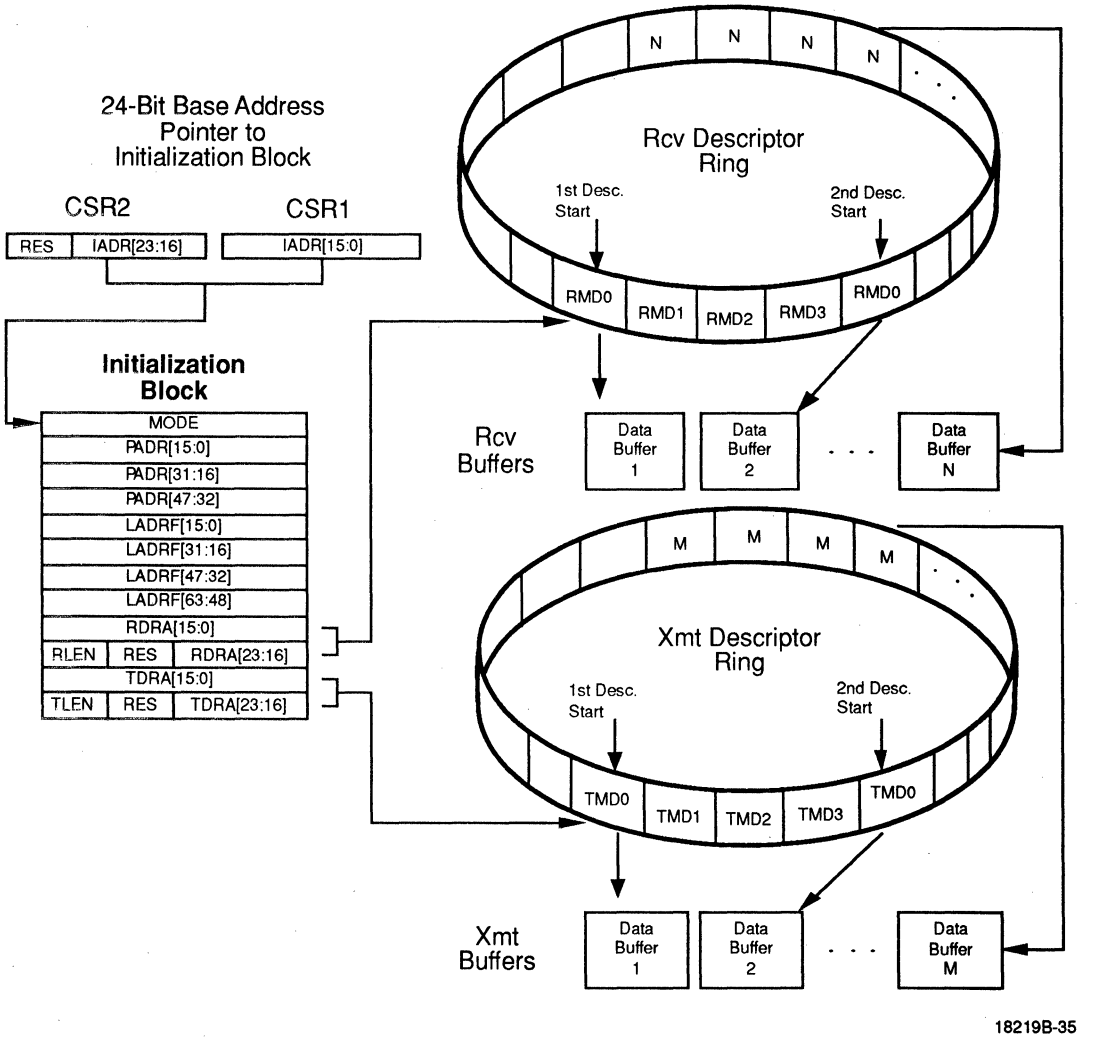


Figure 28. 16-Bit Initialization Block and Descriptor Rings

When SSIZE32 = 1, software data structures are 32 bits wide. Figure 29 illustrates the relationship between the Initialization Base Address, the Initialization Block, the Receive and Transmit Descriptor Ring Base Addresses (TDRA/RDRA), the Receive and Transmit Descriptors and the Receive and Transmit Data Buffers, for the case of SSIZE32 = 1.

Polling

If there is no network channel activity and there is no pre- or post-receive or pre- or post-transmit activity being performed by the PCnet-32 controller, then the PCnet-32 controller will periodically poll the current receive and transmit descriptor entries in order to ascertain their ownership. If the DPOLL bit in CSR4 is set, then transmit polling function is disabled.

A typical polling operation consists of the following: The PCnet-32 controller will use the current receive descriptor address stored internally to vector to the appropriate Receive Descriptor Table Entry (RDTE). It will then use the current transmit descriptor address (stored internally) to vector to the appropriate Transmit Descriptor Table Entry (TDTE). The accesses will be made in the following order: RMD1, then RMD0 of the current RDTE during one bus arbitration, and after that, TMD1, then TMD0 of the current TDTE during a second bus arbitration. All information collected during polling activity will be stored internally in the appropriate CSRs (i.e. CSR18, CSR19, CSR20, CSR21, CSR40, CSR42, CSR50, CSR52). UnOWNed descriptor status will be internally ignored.

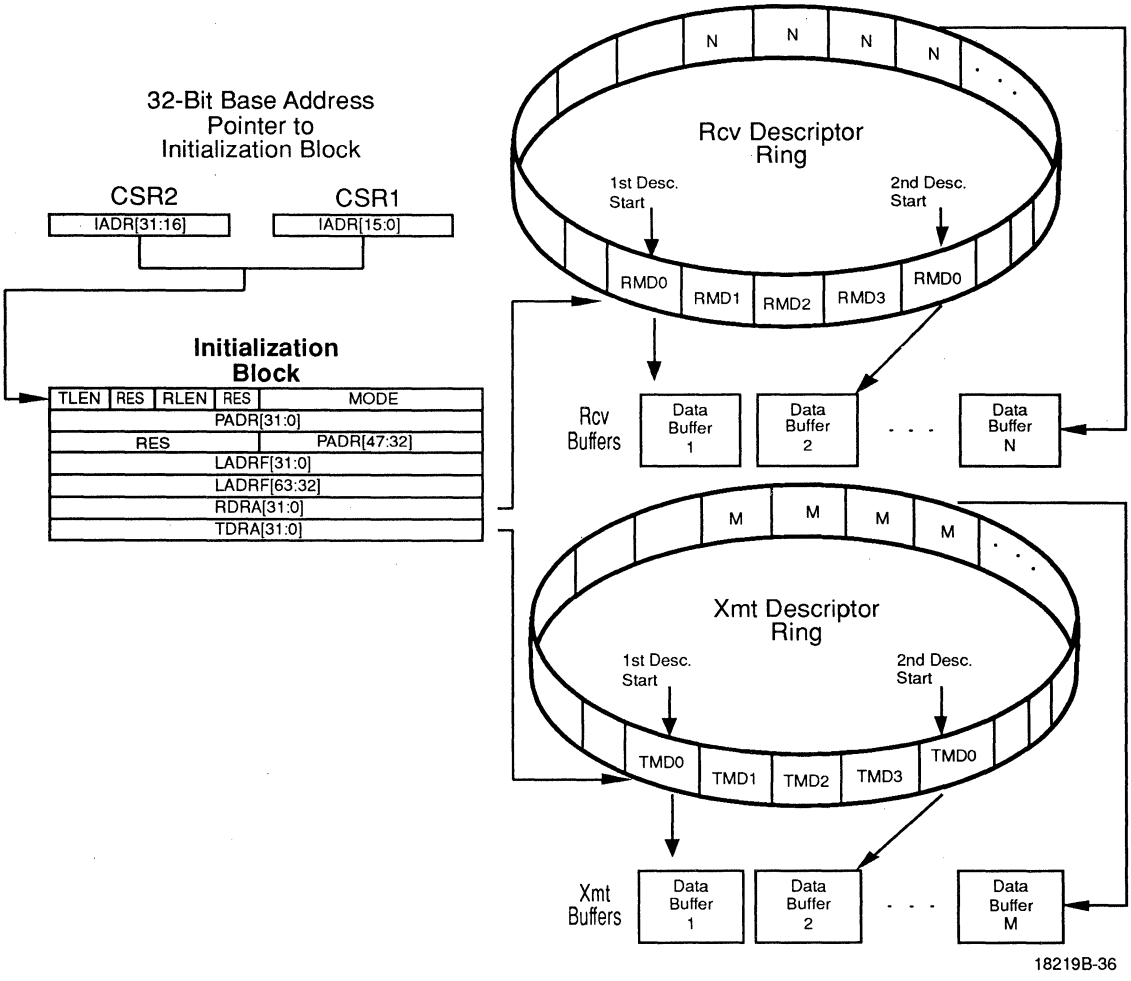


Figure 29. 32-Bit Initialization Block and Descriptor Rings

A typical receive poll is the product of the following conditions:

1. PCnet-32 controller does not possess ownership of the current RDTE and the poll time has elapsed and RXON = 1, or
2. PCnet-32 controller does not possess ownership of the next RDTE the poll time has elapsed and RXON = 1.

If RXON = 0 the PCnet-32 controller will never poll RDTE locations.

The ideal system should always have at least one RDTE available for the possibility of an unpredictable receive event. (This condition is not a requirement. If this condition is not met, it simply means that frames will be missed by the system because there was no buffer space available.) But the typical system usually has at least one or two RDTEs available for the possibility of an unpredictable receive event. Given that this condition is satisfied, the current and next RDTE polls are rarely seen and hence, the typical poll operation simply consists of a check of the status of the current TDTE. When there is only one RDTE (because the RLEN was set to zero), then there is no "next RDTE" and ownership of "next RDTE" cannot be checked. If there is at least one RDTE, the RDTE poll will rarely be seen and the typical poll operation simply consists of a check of the current TDTE.

A typical transmit poll is the product of the following conditions:

1. PCnet-32 controller does not possess ownership of the current TDTE and

   DPOLL = 0 and

   TXON = 1 and

   the poll time has elapsed, or
2. PCnet-32 controller does not possess ownership of the current TDTE and

   DPOLL = 0 and

   TXON = 1 and

   a frame has just been received, or
3. PCnet-32 controller does not possess ownership of the current TDTE and

   DPOLL = 0 and

   TXON = 1 and

   a frame has just been transmitted.

Setting the TDMD bit of CSR0 will cause the microcode controller to exit the poll counting code and immediately perform a polling operation. If RDTE ownership has not been previously established, then an RDTE poll will be performed ahead of the TDTE poll. If the microcode is not executing the poll counting code when the TDMD bit is set, then the demanded poll of the TDTE will be delayed until the microcode returns to the poll counting code.

The user may change the poll time value from the default of 65,536 BCLK periods by modifying the value in the Polling Interval register (CSR47). Note that if a non-default value is desired, then a strict sequence of setting the INIT bit in CSR0, waiting for IDON (CSR0[8]), then writing to CSR47, and then setting STRT in CSR0 must be observed, otherwise the default value will not be overwritten. See the CSR47 section for details.

Transmit Descriptor Table Entry (TDTE)

If, after a TDTE access, the PCnet-32 controller finds that the OWN bit of that TDTE is not set, then the PCnet-32 controller resumes the poll time count and reexamines the same TDTE at the next expiration of the poll time count.

If the OWN bit of the TDTE is set, but STP = 0, the PCnet-32 controller will immediately request the bus in order to reset the OWN bit of this descriptor. (This condition would normally be found following a LCOL or RETRY error that occurred in the middle of a transmit frame chain of buffers.) After resetting the OWN bit of this descriptor, the PCnet-32 controller will again immediately request the bus in order to access the next TDTE location in the ring.

If the OWN bit is set and the buffer length is 0, the OWN bit will be reset. In the LANCE the buffer length of 0 is interpreted as a 4096-byte buffer. It is acceptable to have a 0 length buffer on transmit with STP = 1 or STP = 1 *and* ENP = 1. It is not acceptable to have 0 length buffer with STP = 0 and ENP =1.

If the OWN bit is set and the start of packet (STP) bit is set, then microcode control proceeds to a routine that will enable transmit data transfers to the FIFO. The PCnet-32 controller will look ahead to the next transmit descriptor after it has performed at least one transmit data transfer from the first buffer. (More than one transmit data transfer may possibly take place, depending upon the state of the transmitter.) The contents of TMD0 and TMD1 will be stored in Next Xmt Buffer Address (CSR64 and CSR65), Next Xmt Byte Count (CSR66) and Next Xmt Status (CSR67) regardless of the state of the OWN bit. This transmit descriptor look-ahead operation is performed only once.

If the PCnet-32 controller does not own the next TDTE (i.e. the second TDTE for this frame), then it will complete transmission of the current buffer and then update the status of the current (first) TDTE with the BUFF and UFLO bits being set. This will cause the transmitter to be disabled (CSR0, TXON=0). The PCnet-32 controller will have to be re-initialized to restore the transmit function. The situation that matches this description implies that the system has not been able to stay ahead of the PCnet-32 controller in the transmit descriptor ring and therefore, the condition is treated as a fatal error. (To avoid this situation, the system should always set the transmit chain descriptor own bits in reverse order.)

If the PCnet-32 controller does own the second TDTE in a chain, it will gradually empty the contents of the first

buffer (as the bytes are needed by the transmit operation), perform a single-cycle DMA transfer to update the status of the first descriptor (reset the OWN bit in TMD1), and then it may perform one data DMA access on the second buffer in the chain before executing another look-ahead operation (i.e. a look-ahead to the third descriptor.)

The PCnet-32 controller can queue up to two frames in the transmit FIFO. Call them frame "X" and frame "Y", where "Y" is after "X". Assume that frame "X" is currently being transmitted. Because the PCnet-32 controller can perform look-ahead data transfer past the ENP of frame "X", it is possible for the PCnet-32 controller to completely transfer the data from a buffer belonging to frame "Y" into the FIFO even though frame "X" has not yet been completely transmitted. At the end of this "Y" buffer data transfer, the PCnet-32 controller will write intermediate status (change the OWN bit to a zero) for the "Y" frame buffer, if frame "Y" uses data chaining. The last TDTE for the "X" frame (containing ENP) has not yet been written, since the "X" frame has not yet been completely transmitted. Note that the PCnet-32 controller has, in this instance, returned ownership of a TDTE to the host out of a "normal" sequence. For this reason, it becomes imperative that the host system should never read the Transmit DTE ownership bits out of order.

There should be no problems for software which processes buffers in sequence, waiting for ownership before proceeding.

If an error occurs in the transmission before all of the bytes of the current buffer have been transferred, then TMD2 and TMD1 of the current buffer will be written. In such a case, data transfers from the next buffer will not commence. Instead, following the TMD2/TMD1 update, the PCnet-32 controller will go to the next transmit frame, if any, skipping over the rest of the frame which experienced an error, including chained buffers. This is done by returning to the polling microcode where PCnet-32 controller will immediately access the next descriptor and find the condition OWN=1 and STP=0 as described earlier. As described for that case, the PCnet-32 controller will reset the own bit for this descriptor and continue in like manner until a descriptor with OWN=0 (no more transmit frames in the ring) or OWN=1 and STP=1 (the first buffer of a new frame) is reached.

At the end of any transmit operation, whether successful or with errors, immediately following the completion of the descriptor updates, the PCnet-32 controller will always perform another poll operation. As described earlier, this poll operation will begin with a check of the current RDTE, unless the PCnet-32 controller already owns that descriptor. Then the PCnet-32 controller will proceed to polling the next TDTE. If the transmit descriptor OWN bit has a zero value, then the PCnet-32 controller will resume poll time count incrementing. If the transmit descriptor OWN bit has a value of ONE, then the PCnet-32 controller will begin filling the FIFO with transmit data and initiate a transmission. This end-of-operation poll coupled with the TDTE look-ahead operation allows the PCnet-32 controller to avoid inserting poll time counts between successive transmit frames.

Whenever the PCnet-32 controller completes a transmit frame (either with or without error) and writes the status information to the current descriptor, then the TINT bit of CSR0 is set to indicate the completion of a transmission. This causes an interrupt signal if the IENA bit of CSR0 has been set and the TINTM bit of CSR3 is reset.

Receive Descriptor Table Entry (RDTE)

If the PCnet-32 controller does not own both the current and the next Receive Descriptor Table Entry then the PCnet-32 controller will continue to poll according to the polling sequence described above. If the receive descriptor ring length is 1, then there is no next descriptor to be polled.

If a poll operation has revealed that the current and the next RDTE belong to the PCnet-32 controller then additional poll accesses are not necessary. Future poll operations will not include RDTE accesses as long as the PCnet-32 controller retains ownership of the current and the next RDTE.

When receive activity is present on the channel, the PCnet-32 controller waits for the complete address of the message to arrive. It then decides whether to accept or reject the frame based on all active addressing schemes. If the frame is accepted the PCnet-32 controller checks the current receive buffer status register CRST (CSR41) to determine the ownership of the current buffer.

If ownership is lacking, then the PCnet-32 controller will immediately perform a (last ditch) poll of the current RDTE. If ownership is still denied, then the PCnet-32 controller has no buffer in which to store the incoming message. The MISS bit will be set in CSR0 and an interrupt will be generated if INEA=1 (CSR0) and MISSM=0 (CSR3). Another poll of the current RDTE will not occur until the frame has finished.

If the PCnet-32 controller sees that the last poll (either a normal poll, or the last-ditch effort described in the above paragraph) of the current RDTE shows valid ownership, then it proceeds to a poll of the next RDTE. Following this poll, and regardless of the outcome of this poll, transfers of receive data from the FIFO may begin.

Regardless of ownership of the second receive descriptor, the PCnet-32 controller will continue to perform receive data DMA transfers to the first buffer, using burst-cycle DMA transfers. If the frame length exceeds the length of the first buffer, and the PCnet-32 controller does not own the second buffer, ownership of the current descriptor will be passed back to the system by writing a zero to the OWN bit of RMD1 and status will be written indicating buffer (BUFF=1) and possibly overflow (OFLO=1) errors.

If the frame length exceeds the length of the first (current) buffer, and the PCnet-32 controller does own the second (next) buffer, ownership will be passed back to the system by writing a zero to the OWN bit of RMD1 when the first buffer is full. Receive data transfers to the second buffer may occur before the PCnet-32 controller proceeds to look ahead to the ownership of the third buffer. Such action will depend upon the state of the FIFO when the status has been updated on the first descriptor. In any case, look-ahead will be performed to the third buffer and the information gathered will be stored in the chip, regardless of the state of the ownership bit. As in the transmit flow, look-ahead operations are performed only once.

This activity continues until the PCnet-32 controller recognizes the completion of the frame (the last byte of this receive message has been removed from the FIFO). The PCnet-32 controller will subsequently update the current RDTE status with the end of frame (ENP) indication set, write the message byte count (MCNT) of the complete frame into RMD2 and overwrite the "current" entries in the CSRs with the "next" entries.

Media Access Control

The Media Access Control engine incorporates the essential protocol requirements for operation of a compliant Ethernet/802.3 node, and provides the interface between the FIFO sub-system and the Manchester Encoder/Decoder (MENDEC).

The MAC engine is fully compliant to Section 4 of ISO/IEC 8802-3 (ANSI/IEEE Standard 1990 Second edition) and ANSI/IEEE 802.3 (1985).

The MAC engine provides programmable enhanced features designed to minimize host supervision, bus utilization, and pre- or post-message processing. These include the ability to disable retries after a collision, dynamic FCS generation on a frame-by-frame basis, and automatic pad field insertion and deletion to enforce minimum frame size attributes and reduces bus bandwidth use.

The two primary attributes of the MAC engine are:

- Transmit and receive message data encapsulation.
  - — Framing (frame boundary delimitation, frame synchronization)
  - — Addressing (source and destination address handling)
  - — Error detection (physical medium transmission errors)
- Media access management.
  - — Medium allocation (collision avoidance)
  - — Contention resolution (collision handling)

Transmit and Receive Message Data Encapsulation

The MAC engine provides minimum frame size enforcement for transmit and receive frames. When APAD_XMT = 1 (CSR4[11]), transmit messages will be padded with sufficient bytes (containing 00h) to ensure that the receiving station will observe an information field (destination address, source address, length/type, data and FCS) of 64 bytes. When ASTRP_RCV = 1 (CSR4[10]), the receiver will automatically strip pad bytes from the received message by observing the value in the length field, and stripping excess bytes if this value is below the minimum data size (46 bytes). Both features can be independently over-ridden to allow illegally short (less than 64 bytes of frame data) messages to be transmitted and/or received. Use of this feature decreases bus usage because the pad bytes are not transferred into or out of host memory.

Framing (Frame Boundary Delimitation, Frame Synchronization)

The MAC engine will autonomously handle the construction of the transmit frame. Once the Transmit FIFO has been filled to the predetermined threshold (set by XMTSP in CSR80), and providing access to the channel is currently permitted, the MAC engine will commence the 7 byte preamble sequence (10101010b, where first bit transmitted is a 1). The MAC engine will subsequently append the Start Frame Delimiter (SFD) byte (10101011b) followed by the serialized data from the Transmit FIFO. Once the data has been completed, the MAC engine will append the FCS (most significant bit first) which was computed on the message (destination address, source address, length field, data field, and pad (if applicable)).

Note that the user is responsible for the correct ordering and content in each of the fields in the frame, including the destination address, source address, length/type and frame data.

The receive section of the MAC engine will detect an incoming preamble sequence and lock to the encoded clock. The internal MENDEC will decode the serial bit stream and present this to the MAC engine. The MAC will discard the first 8-bits of information before searching for the SFD sequence. Once the SFD is detected, all subsequent bits are treated as part of the frame. The MAC engine will inspect the length field to ensure minimum frame size, strip unnecessary pad characters (if enabled), and pass the remaining bytes through the Receive FIFO to the host. If pad stripping is performed, the MAC engine will also strip the received FCS bytes, although the normal FCS computation and checking will occur. Note that apart from pad stripping, the frame will

be passed unmodified to the host. If the length field has a value of 46 or greater, the MAC engine will not attempt to validate the length against the number of bytes contained in the message.

If the frame terminates or suffers a collision before 64 bytes of information (after SFD) have been received, the MAC engine will automatically delete the frame from the Receive FIFO, without host intervention.

Addressing (Source and Destination Address Handling)

The first 6 bytes of information after SFD will be interpreted as the destination address field. The MAC engine provides facilities for physical, logical (multicast) and broadcast address reception. In addition, multiple physical addresses can be constructed (perfect address filtering) using external logic in conjunction with the EADI interface.

Error Detection (Physical Medium Transmission Errors)

The MAC engine provides several facilities which report and recover from errors on the medium. In addition, the network is protected from gross errors due to inability of the host to keep pace with the MAC engine activity.

On completion of transmission, the following transmit status is available in the appropriate TMD and CSR areas:

- The number of transmission retry attempts (ONE, MORE or RTRY).
- Whether the MAC engine had to Defer (DEF) due to channel activity.
- Excessive deferral (EXDEF), indicating that the transmitter has experienced Excessive Deferral on this transmit frame, where Excessive Deferral is defined in ISO 8802-3 (IEEE/ANSI 802.3).
- Loss of Carrier (LCAR) indicates that there was an interruption in the ability of the MAC engine to monitor its own transmission. Repeated LCAR errors indicate a potentially faulty transceiver or network connection.
- Late Collision (LCOL) indicates that the transmission suffered a collision after the slot time. This is indicative of a badly configured network. Late collisions should not occur in a normal operating network.
- Collision Error (CER) indicates that the transceiver did not respond with an SQE Test message within the predetermined time after a transmission completed. This may be due to a failed transceiver, disconnected or faulty transceiver drop cable, or the fact the transceiver does not support this feature (or it is disabled).

In addition to the reporting of network errors, the MAC engine will also attempt to prevent the creation of any network error due to the inability of the host to service the MAC engine. During transmission, if the host fails to keep the Transmit FIFO filled sufficiently, causing an underflow, the MAC engine will guarantee the message is either sent as a runt packet (which will be deleted by the receiving station) or has an invalid FCS (which will also cause the receiver to reject the message).

The status of each receive message is available in the appropriate RMD and CSR areas. FCS and Framing errors (FRAM) are reported, although the received frame is still passed to the host. The error will only be reported if an FCS error is detected and there are a non integral number of bytes in the message. The MAC engine will ignore up to 7 additional bits at the end of a message (dribbling bits), which can occur under normal network operating conditions. The reception of 8 additional bits will cause the MAC engine to de-serialize the entire byte, and will result in the received message and FCS being modified.

The PCnet-32 controller can handle up to 7 dribbling bits when a received frame terminates. During the reception, the FCS is generated on every serial bit (including the dribbling bits) coming from the cable, although the internally saved FCS value is only updated on the eighth bit (on each byte boundary). The framing error is reported to the user as follows:

- If the number of dribbling bits are 1 to 7 and there is no CRC (FCS) error, then there is no Framing error (FRAM = 0).
- If the number of dribbling bits are 1 to 7 and there is a CRC (FCS) error, then there is also a Framing error (FRAM = 1).
- If the number of dribbling bits = 0, then there is no Framing error. There may or may not be a CRC (FCS) error.

Counters are provided to report the Receive Collision Count and Runt Packet Count for network statistics and utilization calculations.

Note that if the MAC engine detects a received frame which has a 00b pattern in the preamble (after the first 8-bits which are ignored), the entire frame will be ignored. The MAC engine will wait for the network to go inactive before attempting to receive additional frames.

Media Access Management

The basic requirement for all stations on the network is to provide fairness of channel allocation. The 802.3/Ethernet protocols define a media access mechanism which permits all stations to access the channel with equality. Any node can attempt to contend for the channel by waiting for a predetermined time (Inter Packet Gap internal) after the last activity, before transmitting on the media. The channel is a multidrop communications media (with various topological configurations permitted) which allows a single station to transmit and all other stations to receive. If two nodes simultaneously contend for the channel, their signals will interact causing loss of data, defined as a collision. It is the responsibility of the MAC to attempt to avoid and recover from a

collision, to guarantee data integrity for the end-to-end transmission to the receiving station.

Medium Allocation

The IEEE/ANSI 802.3 Standard (ISO/IEC 8802-3 1990) requires that the CSMA/CD MAC monitor the medium for traffic by watching for carrier activity. When carrier is detected, the media is considered busy, and the MAC should defer to the existing message.

The ISO 8802-3 (IEEE/ANSI 802.3) Standard also allows optional two part deferral after a receive message.

See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.1:

> ***"Note:*** *It is possible for the PLS carrier sense indication to fail to be asserted during a collision on the media. If the deference process simply times the inter packet gap based on this indication it is possible for a short inter packet gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness the following optional measures, as specified in 4.2.8, are recommended when InterFrameSpacing Part1 is other than zero:*
>
> 1. *Upon completing a transmission, start timing the interpacket gap, as soon as transmitting and carrier sense are both false.*
> 2. *When timing an inter packet gap following reception, reset the inter packet gap timing if carrier sense becomes true during the first 2/3 of the inter packet gap timing interval. During the final 1/3 of the interval the timer shall not be reset to ensure fair access to the medium. An initial period shorter than 2/3 of the interval is permissible including zero."*

The MAC engine implements the optional receive two part deferral algorithm, with a first part inter-frame-spacing time of 6.0 µs. The second part of the inter-frame-spacing interval is therefore 3.6 µs.

The PCnet-32 controller will perform the two part deferral algorithm as specified in Section 4.2.8 (Process Deference). The Inter Packet Gap (IPG) timer will start timing the 9.6 µs InterFrameSpacing after the receive carrier is de-asserted. During the first part deferral (InterFrameSpacingPart1 - IFS1) the PCnet-32 controller will defer any pending transmit frame and respond to the receive message. The IPG counter will be reset to zero continuously until the carrier de-asserts, at which point the IPG counter will resume the 9.6 µs count once again. Once the IFS1 period of 6.0 µs has elapsed, the PCnet-32 controller will begin timing the second part deferral (InterFrameSpacingPart2 - IFS2) of 3.6 µs. Once IFS1 has completed, and IFS2 has commenced, the PCnet-32 controller will not defer to a receive frame if a transmit frame is pending. This means that the PCnet-32 controller will not attempt to receive the receive frame, since it will start to transmit, and generate a collision at 9.6 µs. The PCnet-32 controller will guarantee to complete the preamble (64-bit) and jam (32-bit) sequence before ceasing transmission and invoking the random backoff algorithm.

This transmit two part deferral algorithm is implemented as an option which can be disabled using the DXMT2PD bit in CSR3. Two part deferral after transmission is useful for ensuring that severe IPG shrinkage cannot occur in specific circumstances, causing a transmit message to follow a receive message so closely as to make them indistinguishable.

During the time period immediately after a transmission has been completed, the external transceiver (in the case of a standard AUI connected device), should generate the SQE Test message (a nominal 10 MHz burst of 5-15 Bit Times duration) on the CI± pair (within 0.6 - 1.6 µs after the transmission ceases). During the time period in which the SQE Test message is expected the PCnet-32 controller will not respond to receive carrier sense.

See ANSI/IEEE Std 802.3-1990 Edition, 7.2.4.6 (1)):

> *"At the conclusion of the output function, the DTE opens a time window during which it expects to see the signal_quality_error signal asserted on the Control In circuit. The time window begins when the CARRIER_ STATUS becomes CARRIER_OFF. If execution of the output function does not cause CARRIER_ON to occur, no SQE test occurs in the DTE. The duration of the window shall be at least 4.0 µs but no more than 8.0 µs. During the time window the Carrier Sense Function is inhibited."*

The PCnet-32 controller implements a carrier sense "blinding" period within 0 µs–4.0 µs from de-assertion of carrier sense after transmission. This effectively means that when transmit two part deferral is enabled (DXMT2PD is cleared) the IFS1 time is from 4 µs to 6 µs after a transmission. However, since IPG shrinkage below 4 µs will rarely be encountered on a correctly configured networks, and since the fragment size will be larger than the 4 µs blinding window, then the IPG counter will be reset by a worst case IPG shrinkage/fragment scenario and the PCnet-32 controller will defer its transmission. In addition, the PCnet-32 controller will not restart the "blinding" period if carrier is detected within the 4.0 µs – 6.0 µs IFS1 period, but will commence timing of the entire IFS1 period.

Contention Resolution (Collision Handling)

Collision detection is performed and reported to the MAC engine by the integrated Manchester Encoder/Decoder (MENDEC).

If a collision is detected before the complete preamble/SFD sequence has been transmitted, the MAC Engine will complete the preamble/SFD before appending the jam sequence. If a collision is detected after the preamble/SFD has been completed, but prior to 512 bits being transmitted, the MAC Engine will abort the transmission, and append the jam sequence immediately. The jam sequence is a 32-bit all zeroes pattern.

The MAC Engine will attempt to transmit a frame a total of 16 times (initial attempt plus 15 retries) due to normal collisions (those within the slot time). Detection of

collision will cause the transmission to be re-scheduled, dependent on the backoff time that the MAC Engine computes. If a single retry was required, the ONE bit will be set in the Transmit Frame Status. If more than one retry was required, the MORE bit will be set. If all 16 attempts experienced collisions, the RTRY bit will be set (ONE and MORE will be clear), and the transmit message will be flushed from the FIFO. If retries have been disabled by setting the DRTY bit in CSR15, the MAC Engine will abandon transmission of the frame on detection of the first collision. In this case, only the RTRY bit will be set and the transmit message will be flushed from the FIFO.

If a collision is detected after 512 bit times have been transmitted, the collision is termed a late collision. The MAC Engine will abort the transmission, append the jam sequence and set the LCOL bit. No retry attempt will be scheduled on detection of a late collision, and the transmit message will be flushed from the FIFO.

The ISO 8802-3 (IEEE/ANSI 802.3) Standard requires use of a "truncated binary exponential backoff" algorithm which provides a controlled pseudo random mechanism to enforce the collision backoff interval, before re-transmission is attempted.

See ANSI/IEEE Std 802.3-1990 Edition, 4.2.3.2.5:

> *"At the end of enforcing a collision (jamming), the CSMA/CD sublayer delays before attempting to re-transmit the frame. The delay is an integer multiple of slotTime. The number of slot times to delay before the nth re-transmission attempt is chosen as a uniformly distributed random integer r in the range:*
>
> $0 \le r < 2k-1$
>
> *where* $k = min(n,10)$."

The PCnet-32 controller provides an alternative algorithm, which suspends the counting of the slot time/IPG during the time that receive carrier sense is detected. This aids in networks where large numbers of nodes are present, and numerous nodes can be in collision. It effectively accelerates the increase in the backoff time in busy networks, and allows nodes not involved in the collision to access the channel whilst the colliding nodes await a reduction in channel activity. Once channel activity is reduced, the nodes resolving the collision time out their slot time counters as normal.

Manchester Encoder/Decoder (MENDEC)

The integrated Manchester Encoder/Decoder provides the PLS (Physical Layer Signaling) functions required for a fully compliant ISO 8802-3 (IEEE/ANSI 802.3) station. The MENDEC provides the encoding function for data to be transmitted on the network using the high accuracy on-board oscillator, driven by either the crystal oscillator or an external CMOS level compatible clock. The MENDEC also provides the decoding function from data received from the network. The MENDEC contains a Power On Reset (POR) circuit, which ensures that all analog portions of the PCnet-32 controller are forced into their correct state during power up, and prevents erroneous data transmission and/or reception during this time.

External Crystal Characteristics

When using a crystal to drive the oscillator, the crystal specification shown in Table 26 may be used to ensure less than ±0.5 ns jitter at the transmit outputs.

Table 26. External Crystal Specification

Parameter	Min	Nom	Max	Unit
1. Parallel Resonant Frequency		20		MHz
2. Resonant Frequency Error	–50		+50	PPM
3. Change in Resonant Frequency With Respect To Temperature (0°C–70°C)*	–40		+40	PPM
4. Crystal Load Capacitance	20		50	pF
5. Motional Crystal Capacitance (C1)		0.022		pF
6. Series Resistance			25	ohm
7. Shunt Capacitance			7	pF
8. Drive Level			TBD	mW

**Requires trimming spec.; no trim is 50 ppm total.*

External Clock Drive Characteristics

When driving the oscillator from an external clock source, XTAL2 must be left floating (unconnected). An external clock having the following characteristics must be used to ensure less than ±0.5 ns jitter at the transmit outputs. See Table 27.

Table 27. External Clock Drive Characteristics

Clock Frequency:	20 MHz ±0.01%
Rise/Fall Time (t_R/t_F):	< 6 ns from 0.5 V to V_{DD}–0.5
XTAL1 HIGH/LOW Time (t_{HIGH}/t_{LOW}):	20 ns min
XTAL1 Falling Edge to Falling Edge Jitter:	< ±0.2 ns at 2.5 V input ($V_{DD}/2$)

MENDEC Transmit Path

The transmit section encodes separate clock and NRZ data input signals into a standard Manchester encoded serial bit stream. The transmit outputs (DO±/TXD±) are designed to operate into terminated transmission lines. When operating into a 78 Ω terminated transmission line, the transmit signaling meets the required output levels and skew for Cheapernet, Ethernet and IEEE-802.3.

Transmitter Timing and Operation

A 20 MHz fundamental mode crystal oscillator provides the basic timing reference for the MENDEC portion of the PCnet-32 controller. The crystal is divided by two, to create the internal transmit clock reference. Both clocks are fed into the MENDEC's Manchester Encoder to generate the transitions in the encoded data stream. The

internal transmit clock is used by the MENDEC to internally synchronize the Internal Transmit Data (ITXDAT) from the controller and Internal Transmit Enable (ITXEN). The internal transmit clock is also used as a stable bit rate clock by the receive section of the MENDEC and controller.

The oscillator requires an external ±0.01% timing reference. The accuracy requirements, if an external crystal is used are tighter because allowance for the on-board parasitics must be made to deliver a final accuracy of 0.01%.

Transmission is enabled by the controller. As long as the ITXEN request remains active, the serial output of the controller will be Manchester encoded and appear at DO±/TXD±. When the internal request is dropped by the controller, the differential transmit outputs go to one of two idle states, dependent on TSEL in the Mode Register (CSR15, bit 9):

TSEL LOW:	The idle state of DO±/TXD± yields "zero" differential to operate transformer-coupled loads.
TSEL HIGH:	In this idle state, DO+/TXD+ is positive with respect to DO–/TXD– (logical HIGH).

Receiver Path

The principal functions of the Receiver are to signal the PCnet-32 controller that there is information on the receive pair, and separate the incoming Manchester encoded data stream into clock and NRZ data.

The Receiver section (see Figure 30) consists of two parallel paths. The receive data path is a zero threshold, wide bandwidth line receiver. The carrier path is an offset threshold bandpass detecting line receiver. Both receivers share common bias networks to allow operation over a wide input common mode range.

Input Signal Conditioning

Transient noise pulses at the input data stream are rejected by the Noise Rejection Filter. Pulse width rejection is proportional to transmit data rate.

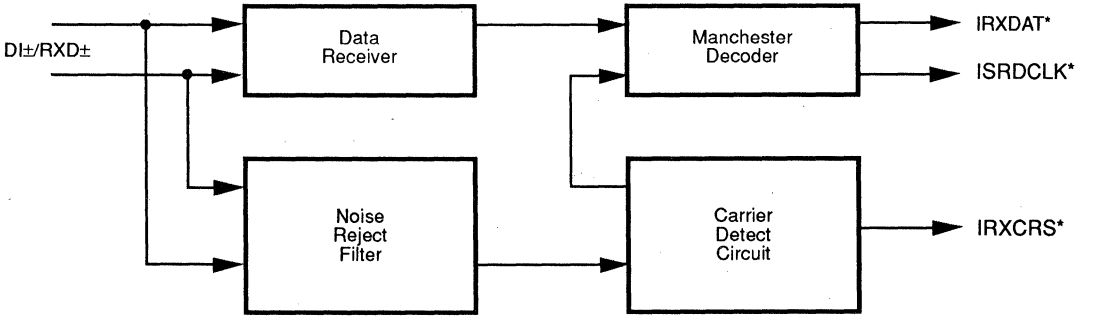
The Carrier Detection circuitry detects the presence of an incoming data frame by discerning and rejecting noise from expected Manchester data, and controls the stop and start of the phase-lock loop during clock acquisition. Clock acquisition requires a valid Manchester bit pattern of 1010b to lock onto the incoming message.

When input amplitude and pulse width conditions are met at DI±/RXD±, the internal enable signal from the MENDEC to controller (IRXCRS) is asserted and a clock acquisition cycle is initiated.

Clock Acquisition

When there is no activity at DI± (receiver is idle), the receive oscillator is phase locked to the internal transmit clock. The first negative clock transition (bit cell center of first valid Manchester "0") after IRXCRS is asserted interrupts the receive oscillator. The oscillator is then restarted at the second Manchester "0" (bit time 4) and is phase locked to it. As a result, the MENDEC acquires the clock from the incoming Manchester bit pattern in 4 bit times with a "1010" Manchester bit pattern.

ISRDCLK and IRXDAT are enabled 1/4 bit time after clock acquisition in bit cell 5. IRXDAT is at a HIGH state when the receiver is idle (no ISRDCLK). IRXDAT however, is undefined when clock is acquired and may remain HIGH or change to LOW state whenever ISRDCLK is enabled. At 1/4 bit time through bit cell 5, the controller portion of the PCnet-32 controller sees the first ISRDCLK transition. This also strobes in the incoming fifth bit to the MENDEC as Manchester "1". IRXDAT may make a transition after the ISRDCLK rising edge in bit cell 5, but its state is still undefined. The Manchester "1" at bit 5 is clocked to IRXDAT output at 1/4 bit time in bit cell 6.



*Internal signal

18219B-37

Figure 30. Receiver Block Diagram

PLL Tracking

After clock acquisition, the phase-locked clock is compared to the incoming transition at the bit cell center (BCC) and the resulting phase error is applied to a correction circuit. This circuit ensures that the phase-locked clock remains locked on the received signal. Individual bit cell phase corrections of the Voltage Controlled Oscillator (VCO) are limited to 10% of the phase difference between BCC and phase-locked clock. Hence, input data jitter is reduced in ISRDCLK by 10 to 1.

Carrier Tracking and End of Message

The carrier detection circuit monitors the DI± inputs after IRXCRS is asserted for an end of message. IRXCRS de-asserts 1 to 2 bit times after the last positive transition on the incoming message. This initiates the end of reception cycle. The time delay from the last rising edge of the message to IRXCRS de-assert allows the last bit to be strobed by ISRDCLK and transferred to the controller section, but prevents any extra bit(s) at the end of message.

Data Decoding

The data receiver is a comparator with clocked output to minimize noise sensitivity to the DI±/RXD± inputs. Input error is less than ± 35 mV to minimize sensitivity to input rise and fall time. ISRDCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit, and clocks the data out on IRXDAT on the following ISRDCLK. The data receiver also generates the signal used for phase detector comparison to the internal MENDEC voltage controlled oscillator (VCO).

Jitter Tolerance Definition

The MENDEC utilizes a clock capture circuit to align its internal data strobe with an incoming bit stream. The clock acquisition circuitry requires four valid bits with the values 1010b. Clock is phase-locked to the negative transition at the bit cell center of the second "0" in the pattern.

Since data is strobed at 1/4 bit time, Manchester transitions which shift from their nominal placement through 1/4 bit time will result in improperly decoded data. With this as the criteria for an error, a definition of "Jitter Handling" is:

> *The peak deviation approaching or crossing 1/4 bit cell position from nominal input transition, for which the MENDEC section will properly decode data.*

Attachment Unit Interface(AUI)

The AUI is the PLS (Physical Layer Signaling) to PMA (Physical Medium Attachment) interface which effectively connects the DTE to a MAU. The differential interface provided by the PCnet-32 controller is fully compliant to Section 7 of ISO 8802-3 (ANSI/IEEE 802.3).

After the PCnet-32 controller initiates a transmission it will expect to see data "looped-back" on the DI± pair (when the AUI port is selected). This will internally generate a "carrier sense", indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted within TBD bit times after the first transmitted bit on DO± (when using the AUI port). If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in the Transmit Descriptor Ring (TMD2, bit27) after the frame has been transmitted.

Differential Input Terminations

The differential input for the Manchester data (DI±) is externally terminated by two 40.2 Ω ±1% resistors and one optional common-mode bypass capacitor, as shown in the Differential Input Termination diagram below. The differential input impedance, ZIDF, and the common-mode input impedance, ZICM, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. If SIP devices are used, 39 Ω is also a suitable value. The CI± differential inputs are terminated in exactly the same way as the DI± pair.

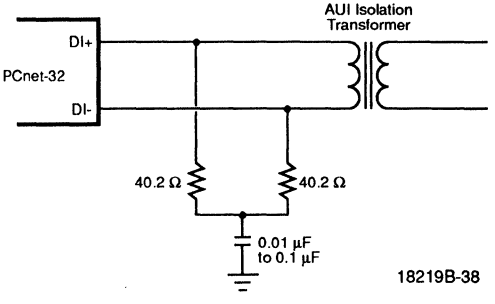


Figure 31. AUI Differential Input Termination

Collision Detection

A MAU detects the collision condition on the network and generates a differential signal at the CI± inputs. This collision signal passes through an input stage which detects signal levels and pulse duration. When the signal is detected by the MENDEC it sets the ICLSN line HIGH. The condition continues for approximately 1.5 bit times after the last LOW-to-HIGH transition on CI±.

Twisted-Pair Transceiver (T-MAU)

The T-MAU implements the Medium Attachment Unit (MAU) functions for the Twisted Pair Medium, as specified by the supplement to ISO 8802-3 (IEEE/ANSI 802.3) standard (Type 10BASE-T). The T-MAU provides twisted pair driver and receiver circuits, including on-board transmit digital predistortion and receiver squelch and a number of additional features including Link Status indication, Automatic Twisted Pair Receive Polarity Detection/Correction and Indication, Receive

Carrier Sense, Transmit Active and Collision Present indication.

T-MAU gets reset during power-up by H_RESET, by S_RESET when reset port is read, or by asserting the RESET pin. T-MAU is not reset by STOP.

Twisted Pair Transmit Function

The differential driver circuitry in the TXD± and TXP± pins provides the necessary electrical driving capability and the pre-distortion control for transmitting signals over maximum length Twisted Pair cable, as specified by the 10BASE-T supplement to the ISO 8802-3 (IEEE/ANSI 802.3) Standard. The transmit function for data output meets the propagation delays and jitter specified by the standard.

Twisted Pair Receive Function

The receiver complies with the receiver specifications of the ISO 8802-3 (IEEE/ANSI 802.3) 10BASE-T Standard, including noise immunity and received signal rejection criteria ('Smart Squelch'). Signals meeting this criteria appearing at the RXD± differential input pair are routed to the MENDEC. The receiver function meets the propagation delays and jitter requirements specified by the standard. The receiver squelch level drops to half its threshold value after unsquelch to allow reception of minimum amplitude signals and to offset carrier fade in the event of worst case signal attenuation and crosstalk noise conditions.

Note that the 10BASE-T Standard defines the receive input amplitude at the external Media Dependent Interface (MDI). Filter and transformer loss are not specified. The T-MAU receiver squelch levels are defined to account for a 1 dB insertion loss at 10 MHz, which is typical for the type of receive filters/transformers employed.

Normal 10BASE-T compatible receive thresholds are employed when the LRT (CSR15[9]) bit is LOW. When the LRT bit is set (HIGH), the Low Receive Threshold option is invoked, and the sensitivity of the T-MAU receiver is increased. This allows longer line lengths to be employed, exceeding the 100 m target distance of normal 10BASE-T (assuming typical 24 AWG cable). The increased receiver sensitivity compensates for the increased signal attenuation caused by the additional cable distance.

However, making the receiver more sensitive means that it is also more susceptible to extraneous noise, primarily caused by coupling from co-resident services (crosstalk). For this reason, it is recommended that when using the Low Receive Threshold option that the service should be installed on 4-pair cable only. Multipair cables within the same outer sheath have lower crosstalk attenuation, and may allow noise emitted from adjacent pairs to couple into the receive pair, and be of sufficient amplitude to falsely unsquelch the T-MAU.

Link Test Function

The link test function is implemented as specified by 10BASE-T standard. During periods of transmit pair inactivity, 'Link beat pulses' will be periodically sent over the twisted pair medium to constantly monitor medium integrity.

When the link test function is enabled (DLNKTST bit in CSR15 is cleared), the absence of link beat pulses and receive data on the RXD± pair will cause the T-MAU to go into a link fail state. In the link fail state, data transmission, data reception, data loopback and the collision detection functions are disabled, and remain disabled until valid data or >5 consecutive link pulses appear on the RXD± pair. During link fail, the Link Status signal is inactive. When the link is identified as functional, the Link Status signal is asserted. The $\overline{\text{LNKST}}$ pin displays the Link Status signal by default.

Transmission attempts during Link Fail state will produce no network activity and will produce LCAR and CERR error indications.

In order to inter-operate with systems which do not implement Link Test, this function can be disabled by setting the DLNKTST bit in CSR15. With link test disabled, the data driver, receiver and loopback functions as well as collision detection remain enabled irespective of the presence or absence of data or link pulses on the RXD± pair. Link Test pulses continue to be sent regardless of the state of the DLNKTST bit.

Polarity Detection and Reversal

The T-MAU receive function includes the ability to invert the polarity of the signals appearing at the RXD± pair if the polarity of the received signal is reversed (such as in the case of a wiring error). This feature allows data frames received from a reverse wired RXD± input pair to be corrected in the T-MAU prior to transfer to the MENDEC. The polarity detection function is activated following H_RESET or Link Fail, and will reverse the receive polarity based on both the polarity of any previous link beat pulses and the polarity of subsequent frames with a valid End Transmit Delimiter (ETD).

When in the Link Fail state, the T-MAU will recognize link beat pulses of either positive or negative polarity. Exit from the Link Fail state is made due to the reception of 5–6 consecutive link beat pulses of identical polarity. On entry to the Link Pass state, the polarity of the last 5 link beat pulses is used to determine the initial receive polarity configuration and the receiver is reconfigured to subsequently recognize only link beat pulses of the previously recognized polarity.

Positive link beat pulses are defined as received signal with a positive amplitude greater than 585 mV (LRT = HIGH) with a pulse width of 60 ns–200 ns. This positive excursion may be followed by a negative excursion. This definition is consistent with the expected received signal at a correctly wired receiver, when a link beat pulse which fits the template of Figure 14-12 of the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

Negative link beat pulses are defined as received signals with a negative amplitude greater than 585 mV with a pulse width of 60 ns–200 ns. This negative excursion

may be followed by a positive excursion. This definition is consistent with the expected received signal at a reverse wired receiver, when a link beat pulse which fits the template of Figure 14-12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

The polarity detection/correction algorithm will remain "armed" until two consecutive frames with valid ETD of identical polarity are detected. When "armed", the receiver is capable of changing the initial or previous polarity configuration based on the ETD polarity.

On receipt of the first frame with valid ETD following H_RESET or link fail, the T-MAU will utilize the inferred polarity information to configure its RXD± input, regardless of its previous state. On receipt of a second frame with a valid ETD with correct polarity, the detection/correction algorithm will "lock-in" the received polarity. If the second (or subsequent) frame is not detected as confirming the previous polarity decision, the most recently detected ETD polarity will be used as the default. Note that frames with invalid ETD have no effect on updating the previous polarity decision. Once two consecutive frames with valid ETD have been received, the T-MAU will disable the detection/correction algorithm until either a Link Fail condition occurs or H_RESET is activated.

During polarity reversal, an internal POL signal will be active. During normal polarity conditions, this internal POL signal is inactive. The state of this signal can be read by software and/or displayed by LED when enabled by the LED control bits in the Bus Configuration Registers (BCR4–BCR7).

Twisted Pair Interface Status

Three signals (XMT, RCV and COL) indicate whether the T-MAU is transmitting, receiving, or in a collision state with both functions active simultaneously. These signals are internal signals and the behavior of the LED outputs depends on how the LED Output circuiting is programmed.

The T-MAU will power up in the Link Fail state and the normal algorithm will apply to allow it to enter the Link Pass state. In the Link Pass state, transmit or receive activity will be indicated by assertion of RCV signal going active. If T-MAU is selected using the PORTSEL bits in CSR15, then when moving from AUI to T-MAU selection the T-MAU will be forced into the Link Fail state.

In the Link Fail state, XMT, RCV and COL are inactive.

Collision Detect Function

Activity on both twisted pair signals RXD± and TXD± constitutes a collision, thereby causing the COL signal to be activated. (COL is used by the LED control circuits) COL will remain active until one of the two colliding signals changes from active to idle. However, transmission attempt in Link Fail state results in LCAR and CERR indication. COL stays active for 2-bit times at the end of a collision.

Signal Quality Error (SQE) Test (Heartbeat) Function

The SQE function is disabled when the 10BASE-T port is selected.

Jabber Function

The Jabber function inhibits the twisted pair transmit function of the T-MAU if the TXD± circuit is active for an excessive period (20 ms–150 ms). This prevents any one node from disrupting the network due to a 'stuck-on' or faulty transmitter. If this maximum transmit time is exceeded, the T-MAU transmitter circuitry is disabled, the JAB bit is set (CSR4, bit1), the COL signal is asserted. Once the transmit data stream to the T-MAU is removed, an "unjab" time of 250 ms–750 ms will elapse before the T-MAU COL and re-enables the transmit circuitry.

Power Down

The T-MAU circuitry can be made to go into power savings mode. This feature is useful in battery powered or low duty cycle systems. The T-MAU will go into power down mode when H_RESET is active, coma mode is active, or the T-MAU is not selected. Refer to the Power Savings Modes section for descriptions of the various power down modes.

Any of the three conditions listed above resets the internal logic of the T-MAU and places the device into power down mode. In this mode, the Twisted Pair driver pins (TXD±,TXP±) are driven LOW, and the internal T-MAU status signals (LNKST, RCVPOL, XMT, RCV and COL) signals are inactive.

Once H_RESET ends, coma mode is disabled, and the T-MAU is selected. The T-MAU will remain in the reset state for up to 10 µs. Immediately after the reset condition is removed, the T-MAU will be forced into the Link Fail state. The T-MAU will move to the Link Pass state only after 5 – 6 link beat pulses and/or a single received message is detected on the RD± pair.

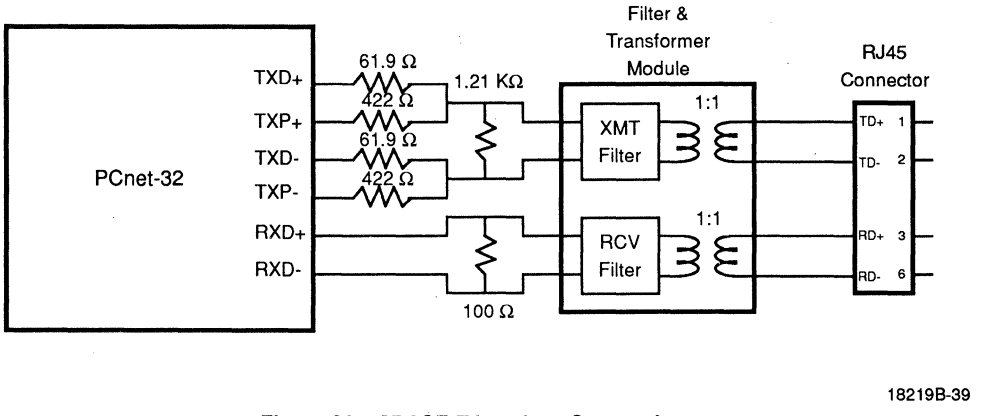
In snooze mode, the T-MAU receive circuitry will remain enabled even while the $\overline{\text{SLEEP}}$ pin is driven LOW.

The T-MAU circuitry will always go into power down mode if H_RESET is asserted, coma mode is enabled, or the T-MAU is not selected.

10BASE-T Interface Connection

Figure 32 shows the proper 10BASE-T network interface design. Refer to the *PCnet Family Technical Manual* (PID # 18216A) for more design details, and refer to Appendix A for a list of compatible 10BASE-T filter/transformer modules.

Note that the recommended resistor values and filter and transformer modules are the same as those used by the IMR (Am79C980) and the IMR+ (Am79C981).



18219B-39

Figure 32. 10BASE-T Interface Connection

IEEE 1149.1 Test Access Port Interface

An IEEE 1149.1 compatible boundary scan Test Access Port is provided for board level continuity test and diagnostics. All digital input, output and input/output pins are tested. Analog pins, including the AUI differential driver (DO±) and receivers (DI±, CI±), and the crystal input (XTAL1/XTAL2) pins, are tested. The T-MAU drivers TXD±, TXP± and receiver RXD± are also tested.

The following is a brief summary of the IEEE 1149.1 compatible test functions implemented in the PCnet-32 controller.

Boundary Scan Circuit

The boundary scan test circuit requires four pins (TCK, TMS, TDI and TDO), defined as the Test Access Port (TAP). It includes a finite state machine (FSM), an instruction register, a data register array, and a power-on reset circuit. Internal pull-up resistors are provided for the TDI, TCK, and TMS pins. The TCK pin must not be left unconnected. The boundary scan circuit remains active during Sleep.

TAP FSM

The TAP engine is a 16-state FSM, driven by the Test Clock (TCK) and the Test Mode Select (TMS) pins. This FSM is in its reset state at power-up or after H_RESET. An independent power-on reset circuit is provided to ensure the FSM is in the TEST_LOGIC_RESET state at power-up.

Supported Instructions

In addition to the minimum IEEE 1149.1 requirements (BYPASS, EXTEST, and SAMPLE instructions), three additional instructions (IDCODE, TRIBYP and SETBYP) are provided to further ease board-level testing. All unused instruction codes are reserved. See Table 28 for a summary of supported instructions.

Instruction Register and Decoding Logic

After H_RESET or S_RESET or STOP, the IDCODE instruction is always invoked. The decoding logic gives signals to control the data flow in the DATA registers according to the current instruction.

Boundary Scan Register (BSR)

Each BSR cell has two stages. A flip-flop and a latch are used for the SERIAL SHIFT STAGE and the PARALLEL OUTPUT STAGE, respectively.

Table 28. IEEE 1149.1 Supported Instruction Summary

Instruction Name	Description	Selected Data Reg	Mode	Instruction Code
EXTEST	External Test	BSR	Test	0000
IDCODE	ID Code Inspection	ID REG	Normal	0001
SAMPLE	Sample Boundary	BSR	Normal	0010
TRIBYP	Force Tristate	Bypass	Normal	0011
SETBYP	Control Boundary To 1/0	Bypass	Test	0100
BYPASS	Bypass Scan	Bypass	Normal	1111

There are four possible operation modes in the BSR cell:

1	Capture
2	Shift
3	Update
4	System Function

Other Data Register

1) BYPASS REG (1 Bit)

2) DEV ID REG (32 bits)

Bits 31–28:	Version
Bits 27–12:	Part number (0010 0100 0011 0000)
Bits 11–1:	Manufacturer ID. The 11 bit manufacturer ID code for AMD is 00000000001 according to JEDEC Publication 106-A.
Bit 0:	Always a logic 1

3) INSCAN0

This is an internal scan path for AMD internal testing use.

EADI (External Address Detection Interface)

This interface is provided to allow external address filtering. It is selected by setting the EADISEL bit in BCR2 to a ONE. This feature is typically utilized for terminal servers, bridges and/or router products. The EADI interface can be used in conjunction with external logic to capture the packet destination address from the serial bit stream as it arrives at the PCnet-32 controller, compare the captured address with a table of stored addresses or identifiers, and then determine whether or not the PCnet-32 controller should accept the packet.

The EADI interface outputs are delivered directly from the NRZ decoded data and clock recovered by the Manchester decoder or input into the GPSI port. This allows the external address detection to be performed in parallel with frame reception and address comparison in the MAC Station Address Detection (SAD) block of the PCnet-32 controller.

SRDCLK is provided to allow clocking of the receive bit stream into the external address detection logic. SRDCLK runs only during frame reception activity. Once a received frame commences and data and clock are available from the decoder, the EADI logic will monitor the alternating ("1,0") preamble pattern until the two ones of the Start Frame Delimiter ("1,0,1,0,1,0,1,1") are detected, at which point the SF/BD output will be driven HIGH.

The SF/BD signal will initially be LOW. The assertion of SF/BD is a signal to the external address detection logic that the SFD has been detected and that subsequent SRDCLK cycles will deliver packet data to the external logic. Therefore, when SF/BD is asserted, the external address matching logic should begin de-serialization of the SRD data and send the resulting destination address to a content addressable memory (CAM) or other address detection device.

In order to reduce the amount of logic external to the PCnet-32 controller for multiple address decoding systems, the SF/BD signal will toggle at each new byte boundary within the packet, subsequent to the SFD. This eliminates the need for externally supplying byte framing logic.

The $\overline{EAR}$ pin should be driven LOW by the external address comparison logic to reject a frame.

If an address match is detected by internal address comparison with either the Physical or Logical or broadcast Address contained within the PCnet-32 controller, then the frame will be accepted regardless of the condition of $\overline{EAR}$. Internal address match is disabled when PROM (CSR15[15]) = 0, DRCVBC (CSR15[14]) = 1, DRCVPA (CSR15[13]) = 1 and Logical Address Filter (CSR8–CSR11) = 0.

When the EADISEL bit of BCR2 is set to a ONE and internal address match is disabled, then all incoming frames will be accepted by the PCnet-32 controller, unless the $\overline{EAR}$ pin becomes active during the first 64 bytes of the frame (excluding preamble and SFD). This allows external address lookup logic approximately 58 byte times after the last destination address bit is available to generate the $\overline{EAR}$ signal, assuming that the PCnet-32 controller is not configured to accept runt packets. $\overline{EAR}$ will be ignored after 64 byte times after the SFD. The frame will be accepted if $\overline{EAR}$ has not been asserted before this time. If Runt Packet Accept is enabled, then the $\overline{EAR}$ signal must be generated prior to the receive message completion, if packet rejection is to be guaranteed. Runt packet sizes could be as short as 12 byte times (assuming 6 bytes for source address, 2 bytes for length, no data, 4 bytes for FCS) after the last bit of the destination address is available. $\overline{EAR}$ must have a pulse width of at least 150 ns.

When the EADISEL bit of BCR2 is set to a ONE and the PROM bit of the Mode Register is set to a ONE, then all incoming frames will be accepted by the PCnet-32 controller, regardless of any activity on the $\overline{EAR}$ pin.

The EADI outputs continue to provide data throughout the reception of a packet. This allows the external logic to capture packet header information to determine protocol type, inter-networking information, and other useful data.

The EADI interface will operate as long as the STRT bit in CSR0 is set, even if the receiver and/or transmitter are disabled by software (DTX and DRX bits in CSR15 are set). This configuration is useful as a semi-power-down mode in that the PCnet-32 controller will not perform any power-consuming DMA operations. However,

external circuitry can still respond to "control" frames on the network to facilitate remote node control.

Table 29 summarizes the operation of the EADI interface.

Table 29. EADI Operations

PROM	$\overline{EAR}$	Required Timing	Received Messages
1	X	No timing requirements	All Received Frames
0	1	No timing requirements	All Received Frames
0	0	Low for 150 ns within 512 bits after SFD	PCnet-32 Controller Internal Physical and Logical Address Matches

General Purpose Serial Interface (GPSI)

The PCnet-32 controller is capable of providing a purely digital network interface in the form of the General Purpose Serial Interface. This interface matches the functionality provided by earlier AMD network controller products, such as the Am7990 LANCE, Am79C90 C-LANCE, Am79C900 ILACC controller, Am79C940 MACE controller, and Am79C960 PCnet-ISA controller.

The GPSI interface is selected through the PORTSEL bits of the Mode register (CSR15) and enabled through the TSTSHDW bits of BCR18 (bits 3 and 4) or enabled through the GPSIEN bit in CSR124 (bit 4). The possible settings to invoke the GPSI mode are shown in Table 30.

Table 30. GPSI Mode Selection

TSTSHDW Value (BCR18 [4:3])	PVALID (BCR19 [15])	GPSIEN (CSR124[4])	Operating Mode
00	1	0	Normal Operating Mode
10	1	X	GPSI Mode
01	1	0	Reserved
11	1	X	Reserved
XX	0	0	Normal Operating Mode
XX	0	1	GPSI Mode

Note that the TSTSHDW bit values are only active when PVALID is TRUE.

The GPSI interface is multiplexed through 7 of the upper 8 address bits of the system bus interface. Therefore, applications that require the use of the GPSI interface will be limited to the use of only 24 address bits (A[23] through A[2] plus the byte enables, which decode into two effective address bits).

In order to prevent the PCnet-32 controller from interpreting the GPSI signals as address bits during the Software Relocatable Mode and during slave accesses, the 24-bit Software Relocatable Mode Address 24 mode *and* the I/O Address Width 24 mode of the PCnet-32 controller must be invoked. Note that if Software Relocatable Mode is invoked, then PVALID must have been set to ZERO, and, therefore, the GPSI mode is not active and therefore, the Software Relocatable Mode might assume that all 30 address bits are visible. But in a system that uses GPSI mode, the GPSI signals would likely all be hardwired to the address pins, and, therefore, even though the device never made it into GPSI mode, it will still not be able to see the upper address bits. Therefore, it is always recommended that SRMA24 mode be invoked as described in the next paragraph:

Software Relocatable Mode Address 24 mode is invoked by connecting the LED2/SRDCLK pin to a LOW level during H_RESET and during the execution of the Software Relocation operation. When the LED2/SRDCLK pin is LOW during H_RESET and Software Relocatable Mode, then the device will be programmed to use 24 bits of addressing while snooping accesses on the bus during Software Relocatable Mode; In this case, the PCnet-32 controller will assume that bits A[31] through A[24] are matched at all times, regardless of the actual values on these pins.

I/O Address Width 24 mode is invoked by writing a ONE to the IOAW24 bit of BCR21 (bit 8 of BCR21). This can be accomplished *safely* in either of two ways:

1) A Software Relocation operation can write a ONE to BCR21, bit 8
2) A read of the EEPROM contents can write a ONE to BCR21, bit 8, if the EEPROM contents are correctly programmed

These two methods do NOT require a slave access to the PCnet-32 controller, and therefore may be performed in a system in which the GPSI signals are permanently connected to the A[31] through A[23] pins (assuming SRMA24 mode is invoked via the LED2/SRDCLK pin to use method 1).

The PCnet-32 controller upper address pins are reconfigured during GPSI mode to the functions listed in Table 31. Note that pin number 143 (A24) has no equivalent GPSI function and should be left unconnected when GPSI mode is enabled.

GPSI signal functions are described in the pin description section under the GPSI subheading.

At the time that GPSI mode is entered, the internal MAC clock is switched from a XTAL1-derived clock to a clock derived from the STDCLK input. The STDCLK input determines the network data rate and therefore must meet frequency and stability specifications.

Table 31. GPSI Pin Configurations

GPSI Function	GPSI I/O Type	LANCE GPSI Pin	ILACC GPSI Pin	PCnet-32/ PCnet-ISA GPSI Pin	PCnet-32 Pin Number	PCnet-32 Normal Pin Function
Transmit Data	O	TX	TXD	TXDAT	132	A31
Transmit Enable	O	TENA	RTS	TXEN	133	A30
Transmit Clock	I	TCLK	TXC	STDCLK	134	A29
Collision	I	CLSN	CDT	CLSN	137	A28
Receive Carrier Sense	I	RENA	CRS	RXCRS	138	A27
Receive Clock	I	RCLK	RXC	SRDCLK	140	A26
Receive Data	I	RX	RXD	RXDAT	141	A25

Note that the XTAL1 input must always be driven with a clock source, even if GPSI mode is to be used. It is not necessary for the XTAL1 clock to meet the normal frequency and stability requirements *in this case.* Any frequency between 8 MHz and 20 MHz is acceptable. However, voltage drive requirements do not change. When GPSI mode is used, XTAL1 must be driven for several reasons:

1) The default pin RESET configuration for the PCnet-32 controller is "AUI port selected," and until GPSI mode is selected, the XTAL1 clock is needed for some internal operations (namely, RESET).
2) The XTAL1 clock drives the EEPROM read operation, regardless of the network mode selected.
3) The XTAL1 clock determines the length of a Reset Register read operation, regardless of the network mode selected (due to the internal RESET caused by the read).

Note that if a clock slower than 20 MHz is provided at the XTAL1 input, the time needed for EEPROM read and Reset Register Read operations will increase.

Power Savings Modes

The PCnet-32 controller supports two hardware power savings modes. Both are entered by driving the $\overline{\text{SLEEP}}$ pin LOW.

In **coma** mode, the PCnet-32 controller will go into a deep sleep with no means to use the network to automatically wake itself up. Coma mode is enabled when the AWAKE bit in BCR2 is reset. Coma mode is the default power down mode. When coma mode is invoked, the T-MAU circuitry will go into power down mode. The system bus interface will be floated and inactive during coma mode. $\overline{\text{LDEV}}$ and the selected interrupt pin will be driven to inactive levels. While in coma mode, if the PCnet-32 controller is configured for a daisy chain (HOLDI and HLDAO or $\overline{\text{LREQI}}$ and $\overline{\text{LGNTO}}$ signals have been selected with the JTAGSEL pin), then the daisy chain signal $\overline{\text{LREQI}}$/HOLDI will be passed directly to $\overline{\text{LREQ}}$/HOLD and the system arbitration signal $\overline{\text{LGNT}}$/HLDA will be passed directly to the daisy-chain signal $\overline{\text{LGNTO}}$/HLDAO.

In **snooze** mode, enabled by setting the AWAKE bit in BCR2 and driving the $\overline{\text{SLEEP}}$ pin LOW, the T-MAU receive circuitry will remain enabled even while the $\overline{\text{SLEEP}}$ pin is driven LOW. The $\overline{\text{LNKST}}$ output will also continue to function, indicating a good 10BASE-T link if there are link beat pulses or valid frames present. This $\overline{\text{LNKST}}$ pin can be used to drive an LED and/or external hardware that directly controls the $\overline{\text{SLEEP}}$ pin of the PCnet-32 controller. This configuration effectively wakes the system when there is any activity on the 10BASE-T link. Auto wake mode can be used only if the T-MAU is the selected network port. Link beat pulses are not transmitted during Auto-wake mode.

The system bus interface will be floated and inactive during snooze mode. $\overline{\text{LDEV}}$ and the selected interrupt pin will be driven to inactive levels. While in snooze mode, if the PCnet-32 controller is configured for a daisy chain (HOLDI and HLDAO or $\overline{\text{LREQI}}$ and $\overline{\text{LGNTO}}$ signals have been selected with the JTAGSEL pin), then the daisy chain signal $\overline{\text{LREQI}}$/HOLDI will be passed directly to $\overline{\text{LREQ}}$/HOLD and the system arbitration signal $\overline{\text{LGNT}}$/HLDA will be passed directly to the daisy-chain signal $\overline{\text{LGNTO}}$/HLDAO.

If the HOLD output is active when the $\overline{\text{SLEEP}}$ pin is asserted, then the PCnet-32 controller will wait until the HLDA input is asserted. Then the PCnet-32 controller will deassert the HOLD pin and finally, it will internally enter either the coma or snooze sleep mode.

Before the sleep mode is invoked, the PCnet-32 controller will perform an internal S_RESET. This S_RESET operation will not affect the values of the BCR registers.

The $\overline{\text{SLEEP}}$ pin should not be asserted during power supply ramp-up. If it is desired that $\overline{\text{SLEEP}}$ be asserted at power up time, then the system must delay the assertion of $\overline{\text{SLEEP}}$ until three BCLK cycles after the completion of a valid pin $\overline{\text{RESET}}$ operation.

Software Access

I/O Resources

PCnet-32 Controller I/O Resource Mapping

The PCnet-32 controller has several I/O resources. These resources use 32 bytes of I/O space that begin at the PCnet-32 controller I/O Base Address.

The PCnet-32 controller allows two modes of slave access. Word I/O mode treats all PCnet-32 controller I/O Resources as two-byte entities spaced at two-byte address intervals. Double Word I/O mode treats all PCnet-32 controller I/O Resources as four-byte entities spaced at four-byte address intervals. The selection of WIO or DWIO mode is accomplished by one of two ways:

1) H_RESET function.

   The PCnet-32 controller I/O mode setting will default to WIO after H_RESET (i.e. DWIO =0).

2) Automatic determination of DWIO mode due to doubleword I/O write access to offset 10h.

DWIO is automatically programmed as active when the system attempts a double word write access to offset 10h of the PCnet-32 controller I/O space. Note that this space corresponds to RDP, regardless of whether DWIO or WIO mode has been programmed. The power up H_RESET value of DWIO will be ZERO, and this value will be maintained until a double word access is performed to PCnet-32 controller I/O space.

Therefore, if DWIO mode is desired, it is imperative that the first access to the PCnet-32 controller be a double word write access to offset 10h.

Alternatively, if DWIO mode is not desired, then it is imperative that the software never executes a double word write access to offset 10h of the PCnet-32 controller I/O space.

Once the DWIO bit has been set to a ONE, only a H_RESET can reset it to a ZERO.

The DWIO mode setting is unaffected by S_RESET or the STOP bit.

WIO I/O Resource Map

When the PCnet-32 controller I/O space is mapped as Word I/O, then the resources that are allotted to the PCnet-32 controller occur on word boundaries that are offset from the PCnet-32 controller I/O Base Address as shown in Table 32.

Table 32. Word I/O Mapping

Offset (Hex)	No. of Bytes	Register
0	2	Address PROM
2	2	Address PROM
4	2	Address PROM
6	2	Address PROM
8	2	Address PROM
A	2	Address PROM
C	2	Address PROM
E	2	Address PROM
10	2	RDP
12	2	RAP (shared by RDP and BDP)
14	2	Reset Register
16	2	BDP
18	2	Vendor Specific Word
1A	2	Reserved
1C	2	Reserved
1E	2	Reserved

When PCnet-32 controller I/O space is Word mapped, all I/O resources fall on word boundaries and all I/O resources are word quantities. However, while in Word I/O mode, address PROM accesses may also be accessed as individual bytes on byte addresses.

Attempts to write to any PCnet-32 controller I/O resources (except to offset 10h, RDP) as 32 bit quantities while in Word I/O mode are illegal and may cause unexpected reprogramming of the PCnet-32 controller control registers. Attempts to read from any PCnet-32 controller I/O resources as 32 bit quantities while in Word I/O mode are illegal and will yield undefined values.

An attempt to *write* to offset 10H (RDP) as a 32 bit quantity while in Word I/O mode will cause the PCnet-32 controller to exit WIO mode and immediately thereafter, to enter DWIO mode.

Accesses to non-word address boundaries are not allowed while in WIO mode, with the exception of the

APROM locations. The PCnet-32 controller may or may not produce an $\overline{LDEV}$ and a $\overline{RDY}$ signal in response to such accesses, but data will be undefined.

Accesses of non-word quantities to any I/O resource are not allowed while in WIO mode, *with the exception of byte reads from the APROM locations.* PCnet-32 controller may or may not produce an $\overline{LDEV}$ and will not produce a $\overline{RDY}$ signal in response to such accesses, but data will be undefined.

The Vendor Specific Word (VSW) is not implemented by the PCnet-32 controller. This particular I/O address is reserved for customer use and will not be used by future AMD Ethernet controller products. If more than one Vendor Specific Word is needed, it is suggested that the VSW location should be divided into a VSW Register Address Pointer (VSWRAP) at one location (e.g. VSWRAP at byte location 18h) and a VSW Data Port (VSWDP) at the other location (e.g. VSWDP at byte location 19h). Alternatively, the system may capture RAP data accesses in parallel with the PCnet-32 controller and therefore share the PCnet-32 controller RAP to allow expanded VSW space. PCnet-32 controller will not respond to access to the VSW I/O address.

DWIO I/O Resource Map

When the PCnet-32 controller I/O space is mapped as Double Word I/O, then all of the resources that are allotted to the PCnet-32 controller occur on double word boundaries that are offset from the PCnet-32 controller I/O Base Address as shown in Table 33.

Table 33. Double Word I/O Mapping

Offset (Hex)	No. of Bytes	Register
0	4	Address PROM
4	4	Address PROM
8	4	Address PROM
C	4	Address PROM
10	4	RDP
14	4	RAP (shared by RDP and BDP)
18	4	Reset Register
1C	4	BDP

When PCnet-32 controller I/O space is Double Word mapped, all I/O resources fall on double word boundaries. Address PROM resources are double word quantities in DWIO mode. RDP, RAP and BDP contain only two bytes of valid data. The other two bytes of these resources are **reserved** for future use. The reserved bits must be written as zeros, and when read, are considered *undefined.*

Accesses to non-double word address boundaries are not allowed while in DWIO mode. The PCnet-32 controller may or may not produce an $\overline{LDEV}$ and a $\overline{RDY}$ signal in response to such accesses, but data will be undefined.

Accesses of less than 4 bytes to any I/O resource are not allowed while in DWIO mode (i.e. PCnet-32 controller will not respond to such accesses. PCnet-32 controller will not produce an $\overline{LDEV}$ and a $\overline{RDY}$ signal in response to such accesses), but data will be undefined. A double word write access to the RDP offset of 10h will automatically program DWIO mode.

Note that in all cases when I/O resource width is defined as 32 bits, the upper 16 bits of the I/O resource is reserved and written as ZEROs and read as undefined, except for the APROM locations and CSR88.

DWIO mode is exited by asserting the RESET pin. Assertion of S_RESET or setting the STOP bit of CSR0 will have no effect on the DWIO mode setting.

I/O Space Comments

The following statements apply to both WIO and DWIO mapping:

The RAP is shared by the RDP and the BDP.

The PCnet-32 controller does not respond to any addresses outside of the offset range 0h-17h when DWIO = 0 or 0h-1F when DWIO = 1. I/O offsets 18h through 1Fh are not used by the PCnet-32 controller when programmed for DWIO = 0 mode. Locations 1Ah through 1Fh are reserved for future AMD use and therefore should not be implemented by the user if upward compatibility to future AMD devices is desired.

Note that Address PROM accesses do not directly access the EEPROM, but are redirected to a set of shadow registers on board the PCnet-32 controller that contain a copy of the EEPROM contents that was obtained during the automatic EEPROM read operation that follows the RESET operation.

PCnet-32 Controller I/O Base Address

The I/O Base Address Registers (BCR16 and BCR17) will reflect the current value of the base of the PCnet-32 controller I/O address space. BCR16 contains the lower 16 bits of the 32-bit I/O base address for the PCnet-32 controller. BCR17 contains the upper 16 bits of the 32-bit I/O base address for the PCnet-32 controller. This set of registers is both readable and writeable by the host. The value contained in these registers is affected through three means:

1. Immediately *following* the H_RESET operation, the I/O Base Address will be determined by the EEPROM read operation. During this operation, the I/O Base Address register will become programmed with the value of the I/O Base Address field of the EEPROM.
2. If no EEPROM exists, or if an error is detected in the EEPROM data, then the PCnet-32 controller will enter **Software Relocatable Mode**. While in this mode, the PCnet-32 controller will not respond to any I/O accesses directly. However, the PCnet-32

controller will snoop accesses on the system bus. When the PCnet-32 controller detects a specific sequence of four write accesses to I/O address 378h, then the PCnet-32 controller will assume that the software is attempting to relocate the PCnet-32 controller. On eight subsequent write accesses to I/O address 378h, the PCnet-32 controller will accept the data on the bus as a new I/O Base Address and other programming information, and it will leave Software Relocatable Mode. At this point, the PCnet-32 controller will begin responding to I/O accesses directly.

While the PCnet-32 controller is in software relocatable mode, if the LED2 pin is pulled LOW, then the SRMA24 mode is entered and only the lower 24 bits of address are matched to 378h.

3. The I/O Base Address Registers may be directly written to, provided that the PCnet-32 controller is not currently in the Software Relocatable Mode.

Software Relocation of I/O Resources

In order to allow for jumperless Ethernet implementations, the I/O Base Address register value will be automatically altered by the PCnet-32 controller during an EEPROM read operation. In this case, the value of the I/O Base Address for the PCnet-32 controller will be directly dependent upon the value of the BCR16 and BCR17 fields that are stored in the EEPROM. If no EEPROM exists and an EEPROM read is attempted, then the PCnet-32 controller will enter Software Relocatable Mode.

Software Relocatable Mode

While in Software Relocatable Mode, the PCnet-32 controller will not respond to any access on the system bus. However, the PCnet-32 controller will snoop any I/O write accesses that may be present. The PCnet-32 controller will watch for a specific sequence of accesses in order to determine a value for the I/O Base Address Registers. Specifically, the PCnet-32 controller will wait for a sequence of 12 uninterrupted byte-write accesses to I/O address 378h.

The 12 byte-write accesses must occur without intervening I/O accesses to other locations, and they must contain the data in the order shown in Table 34.

$\overline{BE0}$ is required to be active during all Software Relocatable Mode snoop accesses. $\overline{BE3}$–$\overline{BE1}$ may have any value during Software Relocatable Mode snoop accesses.

Table 34. I/O Base Address Write Sequence in SRM

Access No.	I/O Address (Hex)	D[7:0]* (Hex)	ASCII Inter–pretation
1	0000 0378	41	A
2	0000 0378	4D	M
3	0000 0378	44	D
4	0000 0378	01	NA
5	0000 0378	IOBASEL[7:0]	NA
6	0000 0378	IOBASEL[15:8]	NA
7	0000 0378	IOBASEU[7:0]	NA
8	0000 0378	IOBASEU[15:8]	NA
9	0000 0378	BCR2[7:0]	NA
10	0000 0378	BCR2[15:8]	NA
11	0000 0378	BCR21[7:0]	NA
12	0000 0378	BCR21[15:8]	NA

**Note that D[31:8] are don't care, since the accesses are required to one byte in width.*

Immediately following the 12th write access in the sequence, the PCnet-32 controller will leave Software Relocatable Mode, and it will then respond to I/O accesses to the 32 bytes of I/O space that begins at the I/O Base Address location.

Note that in Figure 33 (Software Relocatable Mode Snoop Accesses), the data that is accepted by the PCnet-32 controller will always be the data that is presented during the first T2 cycle, regardless of the state of the $\overline{RDYRTN}$ input.

Since the PCnet-32 controller will not respond to the Software Relocatable Mode snoop accesses, some other device must drive the $\overline{RDYRTN}$ signal during these accesses. In a typical PC environment, these I/O accesses will be directed toward the data port of a parallel port. Therefore, the $\overline{RDYRTN}$ will typically be generated by the parallel port controller. In systems in which the parallel port device does not exist, or is at a different location, the 378h accesses will go unclaimed by any device on the local bus or on the expansion bus. In this case, the chipset will typically terminate the access by providing the $\overline{RDYRTN}$ signal after some access-time-out counter has elapsed.

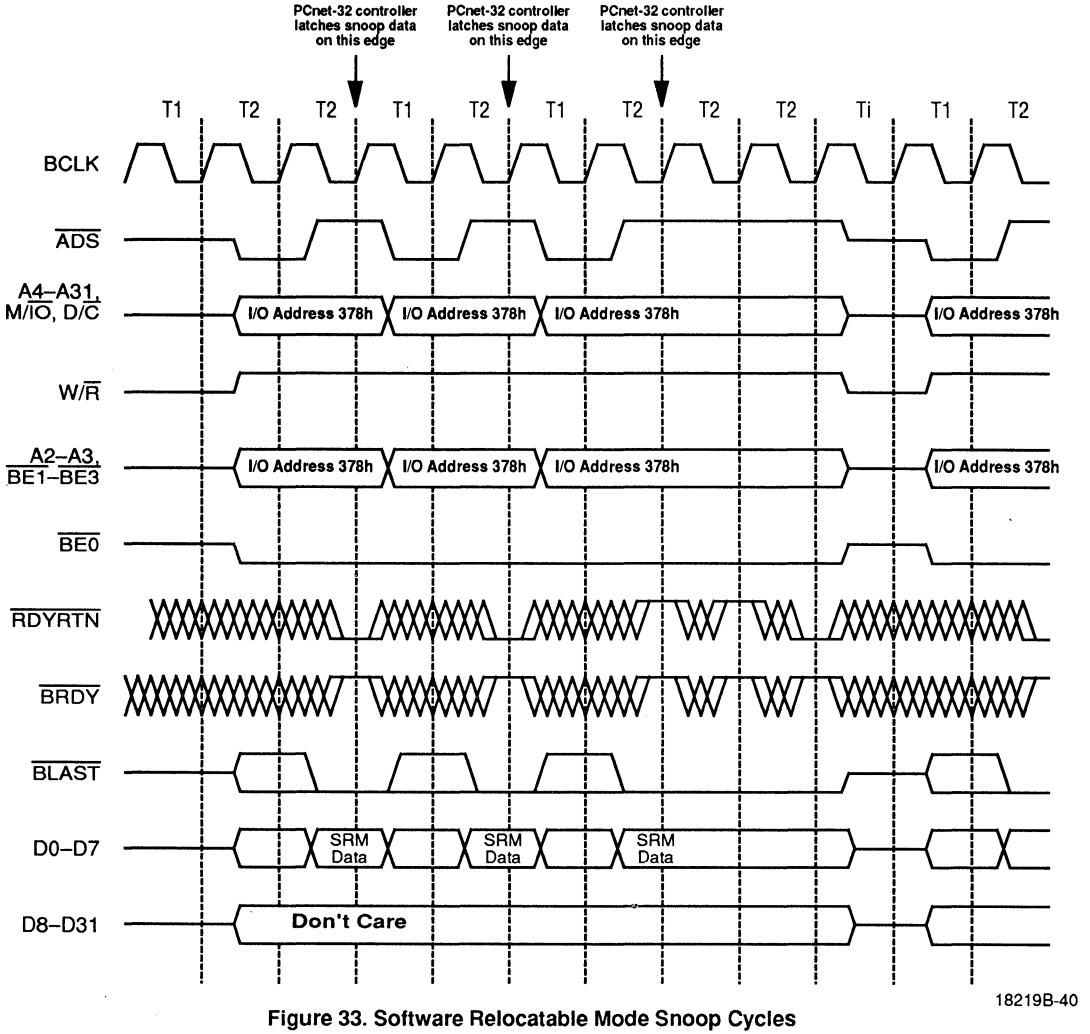


Figure 33. Software Relocatable Mode Snoop Cycles

The I/O Base Address register value may be altered by a direct write to BCR16 and BCR17 by the host. Slave accesses to the PCnet-32 controller should not be performed until both BCR16 and BCR17 have been written with new values. An intermediate I/O Base address will be created when only one of the two registers has been written. Therefore, this method of I/O Base address reassignment is not recommended.

I/O Register Access

All I/O resources are accessed with similar I/O bus cycles.

I/O accesses to the PCnet-32 controller begin with a valid $\overline{ADS}$ strobe and an address on the A[31:2] lines that falls within the I/O space of the PCnet-32 controller. The PCnet-32 controller I/O space will be determined by the I/O Base Address Registers. EEPROM read operations will determine the value of the I/O Base Address Registers.

The PCnet-32 controller will respond to an access to its I/O space by asserting the $\overline{LDEV}$ signal and eventually, by asserting the $\overline{RDY}$ signal.

Typical I/O access times are 6 or 7 BCLK periods. The exact number of clock cycles for any particular access will depend upon the relative phases of the signal on the BCLK pin and the internal clock that is used to drive the PCnet-32 controller buffer management unit. Since the PCnet-32 controller buffer management unit operates on a clock that is a ÷2 version of the interface clock, it is possible for the buffer management unit to introduce one or more BCLK wait delays to allow for

synchronization of the two state machines before proceeding with the access. The PCnet-32 buffer management unit uses BCLK ÷ 2, so the maximum number of BCLK cycles needed to synchronize the BIU and buffer management units is one BCLK period.

APROM Access

The APROM space is a convenient place to store the value of the IEEE station address. This space is automatically loaded from the serial EEPROM, if an EEPROM is present. Its contents have no effect on the operation of the controller. The software must copy the station address from the APROM space to the Initialization Block or to CSR12–CSR14 in order for the receiver to accept unicast frames directed to this station.

When programmed for WIO mode, any byte or word address from an offset of 0h to an offset of Fh may be read. An appropriate byte or word of APROM contents will be delivered by the PCnet-32 controller in response to accesses that fall within the APROM range of 0h to Fh.

When programmed for DWIO mode, only double word addresses from an offset of 0h to an offset of Fh may be read. An appropriate double word of APROM contents will be delivered in response to accesses that fall within the APROM range of 0h to Fh.

Reads of non-double word *quantities* are not allowed in DWIO mode, even though such an access may be properly aligned to a double word address boundary.

Write access to any of the APROM locations is allowed, but only 4 bytes on doubleword boundaries in DWIO mode or 2 bytes on word boundaries in WIO mode. The IESRWE bit (see BCR2) must be set in order to enable such a write. Only the PCnet-32 controller on-board IEEE Shadow registers are modified by writes to APROM locations. The EEPROM is unaffected by writes to APROM locations.

Note that the APROM locations occupy 16 bytes of space, yet the IEEE station address requirement is for 6 bytes. The 6 bytes of IEEE station address occupy the first 6 locations of the APROM space. The next six bytes are reserved. Bytes 12 and 13 should match the value of the checksum of bytes 1 through 11 and 14 and 15. Bytes 14 and 15 should each be ASCII "W" (57 h) if compatibility to AMD driver software is desired.

RDP Access (CSR Register Space)

RDP = Register Data Port. The RDP is used with the RAP to gain access to any of the PCnet-32 controller CSR locations.

Access to any of the CSR locations of the PCnet-32 controller is performed through the PCnet-32 controller's Register Data Port (RDP). In order to access a particular CSR location, the Register Address Port (RAP) should first be written with the appropriate CSR address. The RDP now points to the selected CSR. A read of the RDP will yield the selected CSR's data. A write to the RDP will write to the selected CSR.

When programmed for WIO mode, the RDP has a width of 16 bits, hence, all CSR locations have 16 bits of width. Note that when accessing RDP, the upper two bytes of the data bus will be undefined since the byte masks will not be active for those bytes.

If DWIO mode has been invoked, then the RDP has a width of 32 bits, hence, all CSR locations have 32 bits of width and the upper two bytes of the data bus will be active, as indicated by the byte mask. In this case, note that the upper 16 bits of all CSR locations (except CSR88) are reserved and written as zeros and read as undefined values. Therefore, during RDP write operations in DWIO mode, the upper 16 bits of all CSR locations should be written as ZEROs.

RAP Access

RAP = Register Address Port. The RAP is used with the RDP and with the BDP to gain access to any of the CSR and BCR register locations, respectively. The RAP contains the address pointer that will be used by an access to either the RDP or BDP. Therefore, it is necessary to set the RAP value before accessing a specific CSR or BCR location. Once the RAP has been written with a value, the RAP value remains unchanged until another RAP write occurs, or until an H_RESET or S_RESET occurs. RAP is set to all zeros when an H_RESET or S_RESET occurs. RAP is unaffected by the STOP bit.

When programmed for WIO mode, the RAP has a width of 16 bits. Note that when accessing RAP, the lower two bytes of the data bus will be undefined since the byte masks will not be active for those bytes

When programmed for DWIO mode, the RAP has a width of 32 bits. In DWIO mode, the upper 16 bits of the RAP are reserved and written as zeros and read as undefined. These bits should be written as zeros.

BDP Access (BCR Register Space)

BDP = Bus Configuration Register Data Port. The BDP is used with the RAP to gain access to any of the PCnet-32 controller BCR locations.

Access to any of the BCR locations of the PCnet-32 controller is performed through the PCnet-32 controller's BCR Data Port (BDP). In order to access a particular BCR location, the Register Address Port (RAP) should first be written with the appropriate BCR address. The BDP now points to the selected BCR. A read of the BDP will yield the selected BCR 's data. A write to the BDP will write to the selected BCR.

When programmed for WIO mode, the BDP has a width of 16 bits, hence, all BCR locations have 16 bits of width in WIO mode. Note that when operating in WIO mode, the upper two bytes of the data bus will be undefined since the byte mask will not be active for those bytes.

If DWIO mode has been invoked, then the BDP has a width of 32 bits, hence, all BCR locations have 32 bits of width and the upper two bytes of the data bus will be active, as indicated by the byte mask. In this case, note

that the upper 16 bits of all BCR locations are reserved and written as zeros and read as undefined. Therefore, during BDP write operations in DWIO mode, the upper 16 bits of all BCR locations should be written as zeros.

Reset Register (S_RESET)

A read of the reset register creates an internal S_RESET pulse in the PCnet-32 controller. This read access cycle must be 16 bits wide in WIO mode and 32 bits wide in DWIO mode. The internal S_RESET pulse that is generated by this access is different from both the assertion of the hardware RESET pin (H_RESET) and from the assertion of the software STOP bit. Specifically, the Reset register's S_RESET will be the equivalent of the assertion of the RESET pin (H_RESET) assertion for all CSR locations, but S_RESET will have no effect at all on the BCR locations, and S_RESET will not cause a deassertion of the HOLD pin.

The NE2100 LANCE based family of Ethernet cards requires that a write access to the reset register follows each read access to the reset register. The PCnet-32 controller does not have a similar requirement. The write access is not required but it does not have any harmful effects.

Write accesses to the reset register will have no effect on the PCnet-32 controller.

Note that a read of the Reset register will take longer than the normal I/O access time of the PCnet-32 controller. This is because an internal S_RESET pulse will be generated due to this access, and the access will not be allowed to complete on the system bus until the internal S_RESET operation has been completed. This is to avoid the problem of allowing a new I/O access to proceed while the S_RESET operation has not yet completed, which would result in erroneous data being returned by (or written into) the PCnet-32 controller. The length of a read of the Reset register can be as long as 128 BCLK cycles when Am386 mode has been selected and 64 BCLK cycles when Am486 mode or VESA VL-Bus mode has been selected.

Note that a read of the Reset register will **not** cause a deassertion of the HOLD signal, if it happens to be active at the time of the read to the reset register. The HOLD signal will remain active until the HLDA signal is synchronously sampled as asserted. Following the read of the RESET register, on the next clock cycle after the HLDA signal is synchronously sampled as asserted, the PCnet-32 controller will deassert the HOLD signal). No bus master accesses will have been performed during this brief bus ownership period.

Note that this behavior differs from that which occurs following the assertion of a minimum-width pulse on the RESET pin (H_RESET). A RESET pin assertion will cause the HOLD signal to deassert within six clock cycles following the assertion. In the RESET pin case, the PCnet-32 controller will not wait for the assertion of the HLDA signal before deasserting the HOLD signal.

Vendor Specific Word

This I/O offset is reserved for use by the system designer. The PCnet-32 controller will not respond to accesses directed toward this offset.

Reserved I/O Space

These locations are reserved for future use by AMD. The PCnet-32 controller does not respond to accesses directed toward these locations, but future AMD products that are intended to be upward compatible with the PCnet-32 controller may decode accesses to these locations. Therefore, the system designer may not utilize these I/O locations.

Hardware Access

PCnet-32 Controller Master Accesses

The particular signals involved in a PCnet-32 controller bus master transfer depends upon the bus mode that has been selected. There are two bus modes to choose from. They are:

- Am486 32-bit mode
- VESA VL-Bus mode

Complete descriptions of the signals involved in bus master transactions for each mode may be found in the pin description section of this document. Timing diagrams for master accesses may be found in the block description section for the Bus Interface Unit. This section simply lists the types of master accesses that will be performed by the PCnet-32 controller with respect to data size and address information.

The PCnet-32 controller will support master accesses only to 32-bit peripherals in Am486 environments. The PCnet-32 controller does not support master accesses to 16-bit peripherals in the 486 local bus mode.

The PCnet-32 controller will support master accesses to either 32-bit or 16-bit peripherals in VESA VL-Bus mode. Support of master accesses to 16-bit peripherals in VESA VL-Bus mode is provided through the $\overline{LBS16}$ input pin.

The PCnet-32 controller is not compatible with 8-bit systems, since there is no mode that supports PCnet-32 controller accesses to 8-bit peripherals.

Table 35 describes all possible bus master accesses that the PCnet-32 controller will perform. The rightmost column lists all operations that may execute the given access.

Table 35. Master Accesses

Access	R/W	$\overline{BE3}$–$\overline{BE0}$	Possible Instance
4-Byte Read	RD	0000	Descriptor Read or Initialization Block Read or Transmit Data Buffer Read
4-Byte Write	WR	0000	Descriptor Write or Receive Data Buffer Write
3-Byte Write	WR	1000	Receive Data Buffer Write
3-Byte Write	WR	0001	Receive Data Buffer Write
2-Byte Write	WR	1100	Receive Data Buffer Write
2-Byte Write	WR	1001*	Receive Data Buffer Write
2-Byte Write	WR	0011	Receive Data Buffer Write
1-Byte Write	WR	1110	Receive Data Buffer Write
1-Byte Write	WR	1101*	Receive Data Buffer Write
1-Byte Write	WR	1011*	Receive Data Buffer Write
1-Byte Write	WR	0111	Descriptor Write or Receive Data Buffer Write

**Cases marked with an asterisk represent extreme boundary conditions that are the result of programming one- and two-byte buffer sizes, and therefore will not be seen under normal circumstances.*

Note that all PCnet-32 controller master read operations will always activate all byte enables. (Note the exception, when $\overline{LBS16}$ has been asserted in VL-Bus mode, requiring a second access. In all $\overline{LBS16}$ cases, the second access (if required) will have $\overline{BE1}$ and $\overline{BE0}$ disabled.). Therefore, no one-, two- or three-byte read operations are indicated in the table.

In the instance where a transmit buffer pointer address begins on a non-doubleword boundary, the pointer will be truncated to the next double word boundary address that lies below the given pointer address and the first read access from the transmit buffer will be indicated on the byte enable signals as a four-byte read from this address. Any data from byte lanes that lie outside of the boundary indicated by the buffer pointer will be discarded inside of the PCnet-32 controller. Similarly, if the end of a transmit buffer occurs on a non-doubleword boundary, then all byte lanes will be indicated as active by the byte enable signals, and any data from byte lanes that lie outside of the boundary indicated by the buffer pointer will be discarded inside of the PCnet-32 controller.

Slave Access to I/O Resources

The PCnet-32 controller is a 32-bit peripheral device on the system bus. However, the width of individual software resources on board the PCnet-32 controller may be either 16-bits or 32-bits. The PCnet-32 controller I/O resource widths are determined by the value of the DWIO bit (BCR 18, bit 7) as indicated in Table 36.

Note that when I/O resource width is defined as 32 bits (DWIO mode), the upper 16 bits of the I/O resource is reserved and written as ZEROs and read as undefined, *except for the APROM locations*. The APROM locations are the only I/O resources for which all 32 bits will have defined values. However, this is true only when the PCnet-32 controller is in DWIO mode.

Configuring the PCnet-32 controller for DWIO mode is accomplished whenever there is any attempt to perform a 32-bit write access to the RDP location (offset 10h). See the DWIO section for more details.

Table 37 describes all possible bus slave accesses that may be directed toward the PCnet-32 controller. (I.e. the PCnet-32 controller is the slave device during the transfer.) The four byte columns (D31–D24, D23–D16, D15–D8, D7–D0) indicate the value on the data bus during the access.

Table 36. I/O Resource Access

DWIO Setting	PCnet-32 Controller I/O Resource Width	Example Application
DWIO = 0	16-bit	Existing PCnet-ISA driver that assumes 16-bit I/O mapping and 16-bit resource widths
DWIO = 1	32-bit	New drivers written specifically for the PCnet-32 controller

Table 37. Slave Accesses

R/W	$\overline{BE3}$–$\overline{BE0}$	D31–D24	D23–D16	D15–D8	D7–D0	Comments
RD	0000	Data	Data	Data	Data	Double word access to double word address, e.g. 300, 30C, 310 (DWIO mode only)
RD	1100	Undef	Undef	Data	Data	*WIO Mode Only:* Word access to even word address, e.g. 300, 30C, 310
RD	0011	Data	Data	Copy	Copy	*WIO Mode Only:* Word access to odd word address, e.g. 302, 30E, 312
RD	1110	Undef	Undef	Undef	Data	*WIO Mode APROM Read Only:* Byte access to lower byte of even word address, e.g. 300, 304
RD	1101	Undef	Undef	Data	Undef	*WIO Mode APROM Read Only:* Byte access to upper byte of even word address, e.g. 301, 305
RD	1011	Undef	Data	Undef	Copy	*WIO Mode APROM Read Only:* Byte access to lower byte of odd word address, e.g. 302, 306
RD	0111	Data	Undef	Copy	Undef	*WIO Mode APROM Read Only:* Byte access to upper byte of odd word address, e.g. 303, 307
WR	0000	Data	Data	Data	Data	Double word access to double word address, e.g. 300, 30C, 310 (DWIO mode only)
WR	1100	Undef	Undef	Data	Data	*WIO Mode Only:* Word access to even word address, e.g. 300, 30C, 310
WR	0011	Data	Data	Undef	Undef	*WIO Mode Only:* Word access to odd word address, e.g. 302, 30E, 312

***Data** = indicates the position of the active bytes.*

***Copy** = indicates the positions of copies of the active bytes.*

***Undef** = indicates byte locations that are undefined during the transfer.*

EEPROM Microwire Access

The PCnet-32 controller contains a built-in capability for reading and writing to an external EEPROM. This built-in capability consists of a microwire interface for direct connection to a microwire compatible EEPROM, an automatic EEPROM read feature, and a user-programmable register that allows direct access to the microwire interface pins.

Automatic EEPROM Read Operation

Shortly after the deassertion of the RESET pin, the PCnet-32 controller will read the contents of the EEPROM that is attached to the microwire interface. The automatic EEPROM read begins with EECS being asserted approximately 2.5 EESK periods ($\frac{2.5}{f_{42}}$) following the deassertion of the $\overline{RESET}$ pin. Because of this automatic-read capability of the PCnet-32 controller, an EEPROM can be used to program many of the features of the PCnet-32 controller at power-up, allowing system-dependent configuration information to be stored in the hardware, instead of inside of operating code. PCnet-32 controller interrupt pins will be floated during H_RESET and will remain floated until either the EEPROM has been successfully read, or, following an EEPROM read failure, a Software Relocatable Mode sequence has been successfully executed.

If an EEPROM exists on the microwire interface, the PCnet-32 controller will read the EEPROM contents at the end of the H_RESET operation. The EEPROM contents will be serially shifted into a temporary register and then sent to various register locations on board the PCnet-32 controller. System bus interaction will not occur during the EEPROM read operation after H_RESET, since H_RESET will put the PCnet-32 controller into a state that will not recognize any I/O accesses and the PCnet-32 controller will not yet be at an operating point that requires it to request the bus for bus mastering.

Thirty-four bytes of the EEPROM are set aside for PCnet-32 configuration programming and 2 bytes of the EEPROM are set aside for user programming of logic that is external to the PCnet-32 controller and may or may not be pertinent to the operation of the PCnet-32 controller within the system. The user may gain access to the EEPROM data by snooping the PCnet-32

controller automatic read operation. Logic may be attached to the EEPROM interface to snoop the entire EEPROM read operation using the microwire signals directly, or a simpler scheme may be invoked by taking advantage of an additional signal provided by the PCnet-32 controller (i.e. the SHFBUSY signal).

A checksum verification is performed on the data that is read from the EEPROM. If the checksum verification of the EEPROM data fails, then at the end of the EEPROM read sequence, the PCnet-32 controller will force all EEPROM-programmable register locations back to their H_RESET default values and then the PCnet-32 controller will enter Software Relocatable Mode. The 8-bit checksum for the entire 36 bytes of the EEPROM should be FFh. In the event of a checksum failure, Software Relocatable Mode is entered (PCnet-32 controller begins snooping for a 12-byte sequence) within 1 EESK period following the deassertion of EECS ($\frac{1}{t_{42}}$).

If the absence of an EEPROM has been signaled by the EESK/LED1/SFBD pin at the time of the automatic read operation, then the PCnet-32 controller will recognize this condition and will abort the automatic read operation and reset both the PREAD and PVALID bits in BCR19. At this point, the PCnet-32 controller will enter the Software Relocatable Mode, and the EEPROM-programmable registers will be assigned their H_RESET default values. Software Relocatable Mode is entered (PCnet-32 controller begins snooping for 12-byte sequence) within 2.5 EESK periods ($\frac{2.5}{t_{42}}$) following the deassertion of the RESET pin when absence of an EEPROM is signalled by the EESK/LEDI SFBD pin.

If the user wishes to modify any of the configuration bits that are contained in the EEPROM, then the 7 command, data and status bits of BCR19 can be used to write to the EEPROM. After writing to the EEPROM, the host should set the PREAD bit of BCR19. This action forces a PCnet-32 controller re-read of the EEPROM so that the new EEPROM contents will be loaded into the EEPROM-programmable registers on board the PCnet-32 controller. (The EEPROM-programmable registers may also be reprogrammed directly, but only information that is stored in the EEPROM will be preserved at system power-down.) When the PREAD bit of BCR19 is set, it will cause the PCnet-32 controller to ignore further accesses on the system interface bus until the completion of the EEPROM read operation.

EEPROM Auto-Detection

The PCnet-32 controller uses the EESK/LED1/SFBD pin to determine if an EEPROM is present in the system. At all rising BCLK edges during the assertion of the RESET pin, the PCnet-32 controller will sample the value of the EESK/LED1/SFBD pin. If the sampled value is a ONE, then the PCnet-32 controller assumes that an EEPROM is present, and the EEPROM read operation begins shortly after the RESET pin is deasserted. If the sampled value of EESK/LED1/SFBD is a ZERO, then the PCnet-32 controller assumes that an external pulldown device is holding the EESK/LED1/SFBD pin low, and therefore, there is no EEPROM in the system. In this case, the PCnet-32 controller will enter Software Relocatable Mode. Note that if the designer creates a system that contains an LED circuit on the EESK/LED1/SFBD pin but has no EEPROM present, then the EEPROM auto-detection function will incorrectly conclude that an EEPROM is present in the system. However, this will not pose a problem for the PCnet-32 controller, since it will recognize the lack of an EEPROM at the end of the read operation, when the checksum verification fails. At this point, the PCnet-32 controller will enter Software Relocatable Mode.

The real intention of the EEPROM auto-detection feature is to allow a user to preempt a "good EEPROM" by temporarily resistively shorting the EESK/LED1/SFBD pin to ground. This may need to be done if an add-in card containing the PCnet-32 controller and its EEPROM has been programmed in one system and then later moved to a different system without also moving configuration information that indicates the I/O Base address of the card. The card would power up in the second system with an unknown I/O Base address if the configuration information were not carried with the card to the new system. By allowing the EESK/LED1/SFBD pin to be temporarily resistively shorted to ground, the PCnet-32 controller is fooled into believing that the EEPROM does not exist, and it will enter Software Relocatable Mode. This allows the new system to reconfigure the I/O Base address of the PCnet-32 controller to a location that is compatible to the parameters of the new system. This information will then be written into the EEPROM by a configuration utility through the EEPROM access port (BCR19), in spite of the fact that the PCnet-32 controller believes that there is no EEPROM. The resistive short to ground may now be removed, and the next power-up of the system will place the PCnet-32 controller into a I/O location that is known by this system. When the PREAD bit of BCR19 is set, an EEPROM read operation will be performed, regardless of the value of the EESK/LED1/SFBD pin. Note that the H_RESET-generated EEPROM read operation always obeys the EESK/LED1/SFBD indication.

Table 38 indicates the possible combinations of EEDET and the existence of an EEPROM and the resulting operations that are possible on the EEPROM microwire interface. *Note that the EEDET value (BCR19, bit 3) is determined from EESK/LED1/SFBD pin setting, and it may be set even though there is no EEPROM present.*

Table 38. Effect of EEDET on EEPROM Operation

EEDET Value (BCR19[3])	EEPROM Connected?	Result if PREAD is set to ONE	Result of Automatic EEPROM Read Operation Following H_RESET
0	No	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to ZERO.	First TWO EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to ZERO.
0	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.	First TWO EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to ZERO.
1	No	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to ZERO.	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to ZERO.
1	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.

Systems Without an EEPROM

Some systems may be able to save the cost of an EEPROM by storing the ISO 8802-3 (IEEE/ANSI 802.3) station address and other configuration information somewhere else in the system. For such a system, a two step process is required. The first step will get the PCnet-32 controller into its normal operating mode within the system. The second step will load the IEEE station address. The designer has several choices:

1) *If the LED1 and SFBD functions are not needed in the system*, then the system designer may connect the EESK/LED1/SFBD pin to a resistive pulldown device. This will indicate to the EEPROM auto-detection function that no EEPROM is present, and the PCnet-32 controller will use Software Relocatable Mode to acquire its I/O Base Address and other configuration information (See the section on Software Relocatable Mode).

2) *If either of the LED1 or SFBD functions is needed in the system*, then the system designer will connect the EESK/LED1/SFBD pin to a resistive pullup device and the EEPROM auto-detection function will incorrectly conclude that an EEPROM is present in the system. However, this will not pose a problem for the PCnet-32 controller, since it will recognize the lack of an EEPROM at the end of the read operation, when the checksum verification fails. At this point, the PCnet-32 controller into the Software Relocatable Mode to acquire its I/O Base Address and other configuration information (See the section on Software Relocatable Mode).

In either case, following the execution of Software Relocatable Mode, additional information, including the ISO 8802-3 (IEEE/ANSI 802.3) station address, may be loaded into the PCnet-32 controller. Note that the IESRWE bit (bit 8 of BCR2) must be set before the PCnet-32 controller will accept writes to the APROM offsets within the PCnet-32 controller I/O resources map. Startup code in the system BIOS can perform the Software Relocatable Mode accesses, the IESRWE bit write, and the APROM writes.

If compatibility to existing driver code is desired, then it is not recommended that the ISO 8802-3 (IEEE/ANSI 802.3) station address be loaded into the Initialization Block structure in memory instead of the APROM locations, since existing code typically expects to find the ISO 8802-3 (IEEE/ANSI 802.3) station address at the APROM offsets from the PCnet-32 controller I/O Base Address.

Direct Access to the Microwire Interface

The user may directly access the microwire port through the EEPROM register, BCR19. This register contains bits that can be used to control the microwire interface pins. By performing an appropriate sequence of I/O accesses to BCR19, the user can effectively write to and read from the EEPROM. This feature may be used by a system configuration utility to program hardware configuration information into the EEPROM.

EEPROM-Programmable Registers

The following registers contain configuration information that will be programmed automatically during the EEPROM read operation:

1) I/O offsets 0h–Fh	Address PROM locations
2) BCR2	Miscellaneous Configuration register
3) BCR16	I/O Base Address Lower
4) BCR17	I/O Base Address Upper
5) BCR18	Burst Size and Bus Control Register
6) BCR21	Interrupt Control Register

If the PREAD bit (BCR19) is reset to ZERO *and* the PVALID bit (BCR19) is reset to ZERO, then the EEPROM read has experienced a failure and the contents of the EEPROM programmable register will be set to default H_RESET values. At this point, the PCnet-32 controller will enter Software Relocatable Mode.

Note that accesses to the Address PROM I/O locations do not directly access the Address EEPROM itself. Instead, these accesses are routed to a set of "shadow" registers on board the PCnet-32 controller that are loaded with a copy of the EEPROM contents during the automatic read operation that immediately follows the H_RESET operation.

EEPROM MAP

The automatic EEPROM read operation will access 18 words (i.e. 36 bytes) of the EEPROM. The format of the EEPROM contents is shown in Table 39, beginning with the byte that resides at the lowest EEPROM address.

Table 39. EEPROM Content

EEPROM Word Addr	EEPROM Contents			
	Byte Addr	MSB (Most Significant Byte)	Byte Addr	LSB (Lease Significant Byte)
0 (Lowest Address)	1	2nd byte of the ISO 8802-3 (IEEE/ANSI 802.3) station physical address for this node	0	First byte of the ISO 8802-3 (IEEE/ANSI 802.3) station physical address for this node, where first byte refers to the first byte to appear on the 802.3 medium
1	3	4th byte of the node address	2	3rd byte of the node address
2	5	6th byte of the node address	4	5th byte of the node address
3	7	Reserved location: *must be 00h*	6	Reserved location: *must be 00h*
4	9	Hardware ID: must be 10h if compatibility to AMD drivers is desired	8	Driver IRQ: Must be programmed to the system IRQ channel number being used if AMD drivers are used.
5	B	User programmable space	A	User programmable space
6	D	MSB of two-byte checksum, which is the sum of bytes 0-B and bytes E and F	C	LSB of two-byte checksum, which is the sum of bytes 0-B and bytes E and F
7	F	Must be ASCII "W" (57h) if compatibility to AMD driver software is desired	E	Must be ASCII "W" (57h) if compatibility to AMD driver software is desired
8	11	BCR16[15:8] (I/O Base Address Lower)	10	BCR16[7:0] (I/O Base Address Lower)
9	13	BCR17[15:8] (I/O Base Address Upper)	12	BCR17[7:0] (I/O Base Address Upper)
A	15	BCR18[15:8] (Burst Size and Bus Control)	14	BCR18[7:0] (Burst Size and Bus Control)
B	17	BCR2[15:8] (Miscellaneous configuration)	16	BCR2[7:0] (Miscellaneous configuration)
C	19	BCR21[15:8] (Interrupt Control)	18	BCR21[7:0] (Interrupt Control)
D	1B	Reserved location: *must be 00h*	1A	Reserved location: *must be 00h*
E	1D	Reserved location: *must be 00h*	1C	Reserved location: *must be 00h*
F	1F	checksum *adjust* byte for the first 36 bytes of the EEPROM contents; checksum of the first 36 bytes of the EEPROM should total to FFh	1E	Reserved location: *must be 00h*
10	21	Reserved location: *must be 00h*	20	Reserved location: *must be 00h*
11	23	User programmable byte locations	22	User programmable byte locations

Note that the first bit out of any WORD location in the EEPROM is treated as the MSB of the register that is being programmed. For example, the first bit out of EEPROM WORD location 08h will be written into BCR16[15], the second bit out of EEPROM WORD location 08h will be written into BCR16[14], etc.

There are two checksum locations within the EEPROM. The first is required for the EEPROM address. This checksum will be used by AMD driver software to verify that the ISO 8802-3 (IEEE/ANSI 802.3) station address has not been corrupted. The value of bytes C and D should match the 16-bit sum of bytes 0 through B and E and F. The second checksum location "byte 1F" is not a checksum total, but is, instead, a checksum *adjustment*. The value of this byte should be such that the total 8-bit checksum for the entire 36 bytes of EEPROM data equals the value FFh. The checksum adjust byte is needed by the PCnet-32 controller in order to verify that the EEPROM contents have not been corrupted.

Byte address 8h of the EEPROM map contains the driver IRQ field. The content of this field is used by AMD drivers to program the interrupt channel being used by the PCnet-32. Note that the PCnet-32 interrupt pin selection is NOT effected by this field. The interrupt pin selection is controlled only by the appropriate bits in BCR21. The system interrupt channel associated with each of the PCnet-32 INTR pins is application-dependent. AMD drivers utilize byte location 8h of the EEPROM to resolve this dependency.

EEPROM-Programming of System Logic

When the user has shareable hardware resources in the system and wishes to have these resources programmed at power up, the user may desire to take advantage of the extra space in the EEPROM that is used to configure the PCnet-32 controller in order to store the additional configuration information. The PCnet-32 controller provides a convenient means of access for the user who wishes to utilize this space. The schematic in Figure 34 illustrates an example of logic that is used to generate static control signals for some programmable features of the system, where the programming information is stored on board the PCnet-32 controller's EEPROM and the logic is to be automatically programmed after RESET.

Note that the EECS signal pulses low during the EEPROM read operation and is therefore unsuitable for use as a gate signal for the programmable logic outputs. PCnet-32 controller provides an additional signal, SHFBUSY, which will remain active HIGH during the entire EEPROM read operation. This signal will therefore be suitable for use as the gate of the programmable logic outputs as shown in the diagram. Note that since most of the EEPROM microwire interface signals are multiplexed with other PCnet-32 controller functions, it is necessary for the SHFBUSY pin to enable the shift path of the programmable logic, otherwise the shift path would become active when the EESK and EEDO functions were operating as their alternate functions.

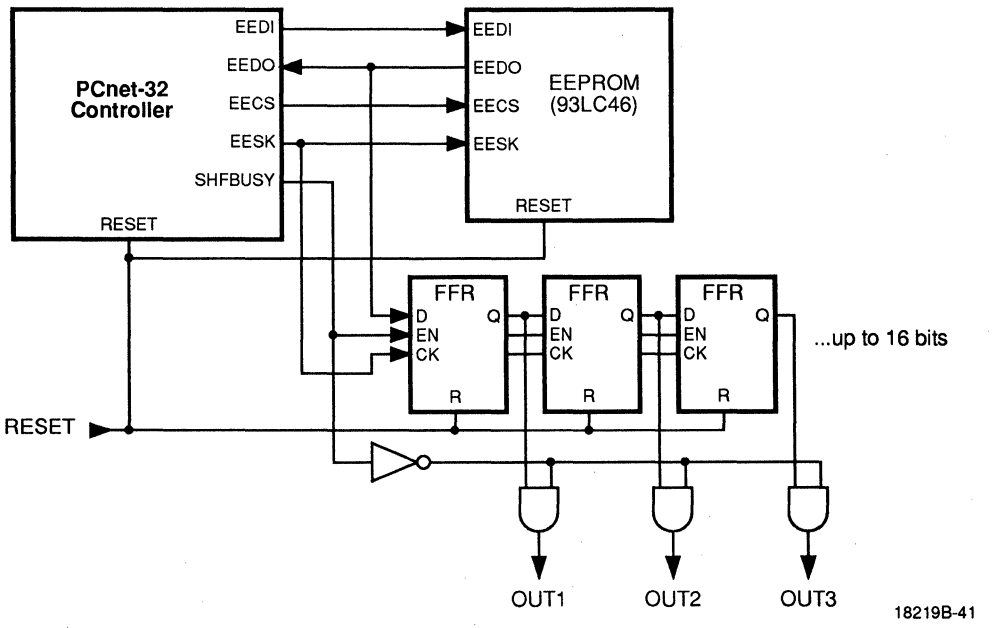


Figure 34. Programming System Logic Through the PCnet-32 EEPROM Read Operation

SHFBUSY will be HIGH for the entire EEPROM read operation, and therefore all EEPROM contents will have been shifted through the external logic before it settles on its final, programmed value. If the EEPROM checksum verification fails, then the EEPROM contents are assumed to be invalid, and the SHFBUSY signal will remain HIGH after the completion of the EEPROM read operation. This action will prevent incorrect system logic values from being driven into the system. If the EEPROM checksum verification passes, then the EEPROM contents are assumed to be valid, and the SHFBUSY signal will return to a LOW state after the completion of the EEPROM read operation.

Transmit Operation

The transmit operation and features of the PCnet-32 controller are controlled by programmable options.

Transmit Function Programming

Automatic transmit features such as retry on collision, FCS generation/transmission, and pad field insertion can all be programmed to provide flexibility in the (re-)transmission of messages.

Disable retry on collision (DRTY) is controlled by the DRTY bit of the Mode register (CSR15) in the initialization block.

Automatic pad field insertion is controlled by the APAD_XMT bit in CSR4. If APAD_XMT is set, automatic pad field insertion is enabled, the DXMTFCS feature is over-ridden, and the 4 byte FCS will be added to the transmitted frame unconditionally. If APAD_XMT is clear, no pad field insertion will take place and runt packet transmission is possible.

The disable FCS generation/transmission feature can be programmed dynamically on a frame by frame basis. See the ADD_FCS description of TMD1.

Transmit FIFO Watermark (XMTFW in CSR80) sets the point at which the BMU requests more data from the transmit buffers for the FIFO. A minimum of "XMTFW" empty spaces must be available in the transmit FIFO before the BMU will request the system bus in order to transfer transmit packet data into the transmit FIFO.

Transmit Start Point (XMTSP in CSR80) sets the point when the transmitter actually attempts to transmit a frame onto the media. A minimum of "XMTSP" bytes must be written to the transmit FIFO for the current frame before transmission of the current frame will begin. (When automatically padded packets are being sent, it is conceivable that the XMTSP is not reached when all of the data has been transferred to the FIFO. In this case, the transmission will begin when all of the packet data has been placed into the transmit FIFO.)

When the entire frame is in the FIFO, attempts at transmission of preamble will commence regardless of the value in XMTSP. The default value of XMTSP is 10b, meaning 64 bytes full.

Automatic Pad Generation

Transmit frames can be automatically padded to extend them to 64 data bytes (excluding preamble). This allows the minimum frame size of 64 bytes (512 bits) for 802.3/Ethernet to be guaranteed with no software intervention from the host/controlling process. Setting the APAD_XMT bit in CSR4 enables the automatic padding feature. The pad is placed between the LLC data field and FCS field in the 802.3 frame (see Figure 35). FCS is always added if the frame is padded, regardless of the state of DXMTFCS. The transmit frame will be padded by bytes with the value of 00h. The default value of APAD_XMT is 0; this will disable auto pad generation after H_RESET.

It is the responsibility of upper layer software to correctly define the actual length field contained in the message to correspond to the total number of LLC Data bytes encapsulated in the packet (length field as defined in the ISO 8802-3 (IEEE/ANSI 802.3) standard). The length value contained in the message is not used by the PCnet-32 controller to compute the actual number of pad bytes to be inserted. The PCnet-32 controller will append pad bytes dependent on the actual number of bits transmitted onto the network. Once the last data byte of the frame has completed, prior to appending the FCS, the PCnet-32 controller will check to ensure that 544 bits have been transmitted. If not, pad bytes are added to extend the frame size to this value, and the FCS is then added.

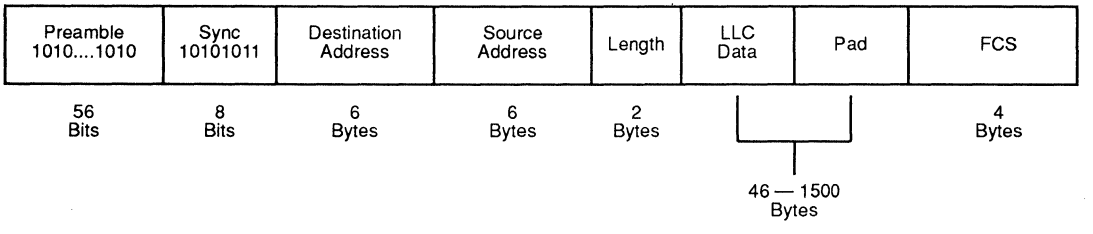


Figure 35. ISO 8802-3 (IEEE/ANSI 802.3) Data Frame

The 544 bit count is derived from the following:

Minimum frame size (excluding preamble, including FCS)	64 bytes	512 bits
Preamble/SFD size	8 bytes	64 bits
FCS size	4 bytes	32 bits

To be classed as a minimum size frame at the receiver, the transmitted frame must contain:

Preamble + (Min Frame Size + FCS) bits

At the point that FCS is to be appended, the transmitted frame should contain:

Preamble + (Min Frame Size–FCS) bits

64 + (512–32) bits

A minimum length transmit frame from the PCnet-32 controller will, therefore, be 576 bits, after the FCS is appended.

The Ethernet specification assumes that minimum length messages will be at least 64 bytes in length.

Transmit FCS Generation

Automatic generation and transmission of FCS for a transmit frame depends on the value of DXMTFCS bit in CSR15. When DXMTFCS = 0 the transmitter will generate and append the FCS to the transmitted frame. If the automatic padding feature is invoked (APAD_XMT is set in CSR4), the FCS will be appended by the PCnet-32 controller regardless of the state of DXMTFCS. Note that the calculated FCS is transmitted most significant bit first. The default value of DXMTFCS is 0 after H_RESET.

Transmit Exception Conditions

Exception conditions for frame transmission fall into two distinct categories. Those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-32 controller include collisions within the slot time with automatic retry. The PCnet-32 controller will ensure that collisions which occur within 512 bit times from the start of transmission (including preamble) will be automatically retired with no host intervention. The transmit FIFO ensures this by guaranteeing that data contained within the FIFO will not be overwritten until at least 64 bytes (512 bits) of preamble plus address, length and data fields have been transmitted onto the network without encountering a collision.

If 16 total attempts (initial attempt plus 15 retries) fail, the PCnet-32 controller sets the RTRY bit in the current transmit TDTE in host memory (TMD2), gives up ownership (resets the OWN bit to zero) for this frame, and processes the next frame in the transmit ring for transmission.

Abnormal network conditions include:

- Loss of carrier.
- Late collision.
- SQE Test Error (does not apply to 10BASE-T port)

None of the abnormal network conditions should not occur on a correctly configured 802.3 network, and will be reported if they do.

When an error occurs in the middle of a multi-buffer frame transmission, the error status will be written in the current descriptor. The OWN bit(s) in the subsequent descriptor(s) will be reset until the STP (the next frame) is found.

Loss of Carrier

A loss of carrier condition will be reported if the PCnet-32 controller cannot observe receive activity while it is transmitting on the AUI port. After the PCnet-32 controller initiates a transmission it will expect to see data "looped-back" on the DI± pair. This will internally generate a "carrier sense," indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted about 6 bit times before the last transmitted bit on DO±. If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in TMD2 after the frame has been transmitted. The frame will not be retried on the basis of an LCAR error.

When the 10BASE-T port is selected, LCAR will be reported for every frame transmitted during the Link fail condition.

Late Collision

A late collision will be reported if a collision condition occurs after one slot time (512 bit times) after the transmit process was initiated (first bit of preamble commenced). The PCnet-32 controller will abandon the transmit process for the particular frame, set Late Collision (LCOL) in the associated TMD2, and process the next transmit frame in the ring. Frames experiencing a late collision will not be re-tried. Recovery from this condition must be performed by upper layer software.

SQE Test Error

During the inter packet gap time following the completion of a transmitted message, the AUI CI± pair is asserted by some transceivers as a self-test. The integral Manchester Encoder/Decoder will expect the SQE Test Message (nominal 10 MHz sequence) to be returned via the CI± pair, within a 40 network bit time period after DI± goes inactive (this does not apply if the 10BASE-T port is selected). If the CI± input is not asserted within the 40 network bit time period following the completion of transmission, then the PCnet-32 controller will set the

CERR bit in CSR0. CERR will be asserted in 10BASE-T mode after transmit if T-MAU is in Link Fail state. CERR will never cause INTR to be activated. It will, however, set the ERR bit in CSR0.

Receive Operation

The receive operation and features of the PCnet-32 controller are controlled by programmable options.

Receive Function Programming

Automatic pad field stripping is enabled by setting the ASTRP_RCV bit in CSR4. this can provide flexibility in the reception of messages using the 802.3 frame format.

All receive frames can be accepted by setting the PROM bit in CSR15. When PROM is set, the PCnet-32 controller will attempt to receive all messages, subject to minimum frame enforcement. Promiscuous mode over rides the effect of the Disable Receive Broadcast bit on receiving broadcast frames.

The point at which the BMU will start to transfer data from the receive FIFO to buffer memory is controlled by the RCVFW bits in CSR80. The default established during H_RESET is 10b which sets the threshold flag at 64 bytes empty.

Automatic Pad Stripping

During reception of an 802.3 frame the pad field can be stripped automatically.

ASTRP_RCV (CSR4, bit 10) = 1 enables the automatic pad stripping feature. The pad field will be stripped before the frame is passed to the FIFO, thus preserving FIFO space for additional frames. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has had the pad characters stripped.

The number of bytes to be stripped is calculated from the embedded length field, as defined in the ISO 8802-3 (IEEE/ANSI 802.3) definition, contained in the frame. The length indicates the actual number of LLC data bytes contained in the message. Any received frame which contains a length field less than 46 bytes will have the pad field stripped (if ASTRP_RCV is set). Receive frames which have a length field of 46 bytes or greater will be passed to the host unmodified.

Note that for some network protocols, the value passed in the Ethernet Type and/or 802.3 Length field is not compliant with either standard and may cause problems.

Figure 36 shows the byte/bit ordering of the received length field for an 802.3 compatible frame format.

Receive FCS Checking

Reception and checking of the received FCS is performed automatically by the PCnet-32 controller. Note that if the Automatic Pad Stripping feature is enabled, the FCS for padded frames will be verified against the value computed for the incoming bit stream including pad characters, but the FCS value for a padded frame will not be passed to the host. If an FCS error is detected in any frame, the error will be reported in the CRC bit in RMD1.

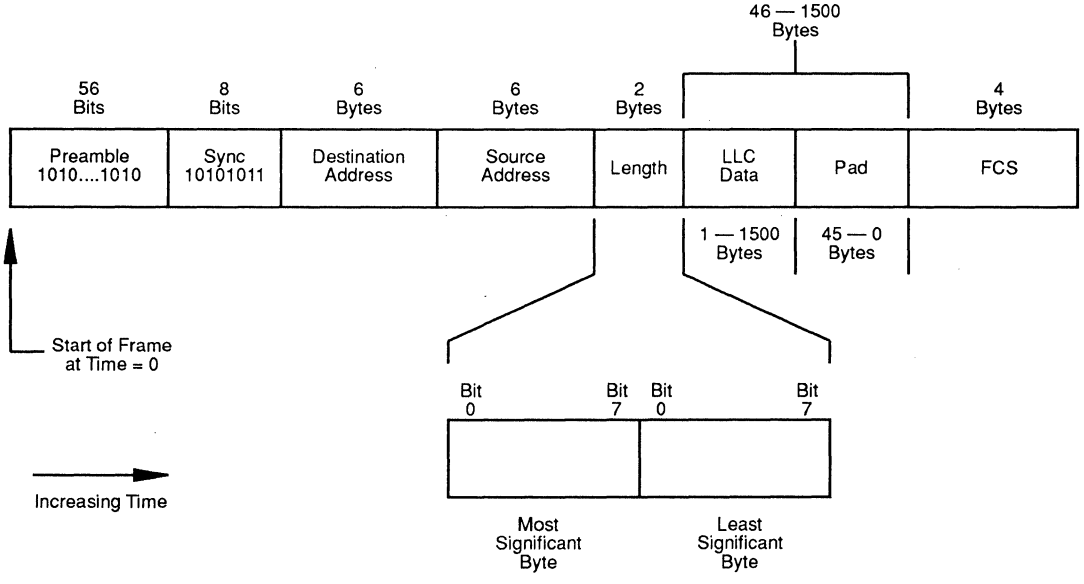


Figure 36. ISO 8802-3 (IEEE/ANSI 802.3) Data Frame

Receive Exception Conditions

Exception conditions for frame reception fall into two distinct categories: Those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-32 controller are basically collisions within the slot time and automatic runt packet rejection. The PCnet-32 controller will ensure that collisions which occur within 512 bit times from the start of reception (excluding preamble) will be automatically deleted from the receive FIFO with no host intervention. The receive FIFO will delete any frame which is composed of fewer than 64 bytes provided that the Runt Packet Accept (RPA bit in CSR124) feature has not been enabled. This criterion will be met regardless of whether the receive frame was the first (or only) frame in the FIFO or if the receive frame was queued behind a previously received message.

Abnormal network conditions include:

- FCS error
- Late collision

Host related receive exception conditions include MISS, BUFF, and OFLO. These are described in the BMU section.

Loopback Operation

Loopback is a mode of operation intended for system testing. In this mode the transmitter and receiver are both operating at the same time so that the controller receives its own transmissions. The controller provides two types of internal loopback and one type of external loopback. In internal loopback mode the transmitter data can be looped back to the receiver at one of two places inside the controller without actually transmitting any data to the external network. The receiver will move the received data to the external network. The receiver will move the received data to the next receive buffer, where it can be examined by software. Alternatively, in external loopback mode, data can be transmitted to and received from the external network.

There are restrictions on loopback operation. The PCnet-32 controller has only one FCS generator circuit. The FCS generator can be used by the transmitter to generate the FCS that is appended to the frame, or it can be used by the receiver to verify the FCS of the received frame. It can not be used by the receiver and transmitter at the same time.

If the FCS generator is connected to the receiver, the transmitter will not append an FCS to the frame, but the receiver will check for one. The user can, however, calculate the FCS value for a frame and include this four-byte number in the transmit buffer.

If the FCS generator is connected to the transmitter, the transmitter will append an FCS to the frame, but the receiver will not check it. However, the user can verify the FCS by software.

During loopback the FCS logic can be allocated to the receiver by setting DXMTFCS = 1 in CSR15.

If DXMTFCS=0, the MAC Engine will calculate and append the FCS to the transmitted message. The receive message passed to the host will therefore contain an additional 4 bytes of FCS. In this loopback configuration, the receive circuitry cannot detect FCS errors if they occur.

If DXMTFCS=1, the last four bytes of the transmit message must contain the (software generated) FCS computed for the transmit data preceding it. The MAC Engine will transmit the data without addition of an FCS field, and the FCS will be calculated and verified at the receiver.

The loopback facilities of the MAC Engine allow full operation to be verified without disturbance to the network. Loopback operation is also affected by the state of the Loopback Control bits (LOOP, MENDECL, and INTL) in CSR15. This affects whether the internal MENDEC is considered part of the internal or external loopback path.

The multicase address detection logic uses the FCS generator. Therefore, when in the loopback mode(s), the multicast address detection feature of the MAC Engine, programmed by the contents of the Logical Address Filter (LADRF [63:0] in CSRs 8-11) can only be tested when DXMTFCS=1, allocating the FCS generator to the receiver. All other features operate identically in loopback as in normal operation, such as automatic transmit padding and receive pad stripping.

When performing an internal loopback, no frame will be transmitted to the network. However, when the PCnet-32 controller is configured for internal loopback the receiver will not be able to detect network traffic. External loopback tests will transmit frames onto the network when the AUI port is selected. Runt Packet Accept is automatically enabled when any loopback mode is invoked.

Loopback mode can be performed with any frame size. Runt Packet Accept is internally enabled (RPA bit in CSR 124 is not affected) when any loopback mode is invoked. This is to be backwards compatible to the LANCE (Am7990) software.

When external loopback is performed while the 10BASE-T MAU is selected, collision detection is disabled. This is necessary, because a collision in a 10BASE-T system is defined as activity on the transmitter outputs and receiver inputs at the same time, which is exactly what happens during external loopback.

Since a 10BASE-T hub does not normally feed the station's transmitter outputs back into the station's receiver inputs, the use of external loopback in a 10BASE-T system usually requires some sort of external hardware that connects the outputs of the 10BASE-T MAU to its inputs.

LED Support

The PCnet-32 controller can support up to 4 LEDs.

LED outputs $\overline{\text{LNKST}}$, LED1 and LED2 allow for direct connection of an LED and its supporting pull-up device. LED output LEDPRE3 may require an additional buffer between the PCnet-32 controller output pin and the LED and its supporting pull-up device.

Because the LEDPRE3 output is multiplexed with other PCnet-32 controller functions, it may not always be possible to connect an LED circuit directly to the LEDPRE3 pin. For example, when an LED circuit is directly connected to the EEDO/LEDPRE3/SRD pin, then it is not possible for most serial EEPROM devices to sink enough IOL to maintain a valid low level on the EEDO input to the PCnet-32 controller. Therefore, in applications that require both an EEPROM and a fourth LED, then it is necessary to buffer the LEDPRE3 circuit from the EEPROM-PCnet-32 controller connection. The LED registers in the BCR resource space allow each LED output to be programmed for either active high or active low operation, so that both inverting and non-inverting buffering choices are possible.

In applications where an EEPROM is not needed, the LEDPRE3 pin may be directly connected to an LED circuit. The PCnet-32 controller LEDPRE3 pin driver will be able to sink enough current to properly drive the LED circuit.

By default, after H_RESET, the 4 LED outputs are configured as shown in Table 40.

Table 40. LED Configuration

LED Output	Default Interpretation	Default Drive Enable	Default Output Polarity
$\overline{\text{LNKST}}$	Link Status	Enabled	Active LOW
LED1	Receive	Enabled	Active LOW
LED2	Receive Polarity	Enabled	Active LOW
LEDPRE3	Transmit	Enabled	Active LOW

For each LED register, each of the status signals is ANDed with its enable signal, and these signals are all OR'd together to form a combined status signal. Each LED pin's combined status signal runs to a pulse stretcher, which consists of a 3-bit shift register clocked at 38 Hz (26 ms). The data input of each shift register is normally at logic 0. The OR gate output for each LED register asynchronously sets all three bits of its shift register when the output becomes asserted. The inverted output of each shift register is used to control an LED pin. Thus the pulse stretcher provides 2–3 clocks of stretched LED output, or 52 ms to 78 ms.

Figure 37 shows the LED signal circuit that exists for each LED pin.

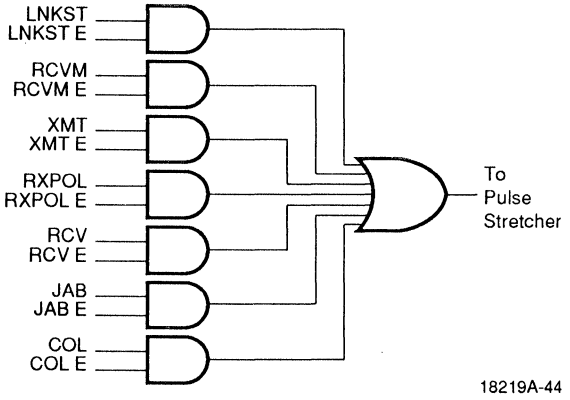


Figure 37. On-Chip LED Control Logic

H_RESET, S_RESET and STOP

There are three different types of RESET operations that may be performed on the PCnet-32 device, H_RESET, S_RESET and STOP. These names have been used throughout the document. The following is a description of each type of RESET operation:

H_RESET

H_RESET= HARDWARE_RESET is a PCnet-32 RESET operation that has been created by the proper assertion of the RESET pin of the PCnet-32 device. When the minimum pulse width timing as specified in the RESET pin description has been satisfied, then an internal RESET operation will be performed.

H_RESET will RESET all of or some portions of CSR0, 3, 4, 15, 58, 80, 82, 100, 112, 114, 122, 124 and 126 to default values; H_RESET will RESET all of or some portions of BCR 2, 4, 5, 6, 7, 18, 19, 20, 21 to default values. H_RESET will cause the microcode program to jump to its RESET state. Following the end of the H_RESET operation, the PCnet-32 controller will attempt to read the EEPROM device through the EEPROM microwire interface. The H_RESET operation will unconditionally cause all INTR pins to become inactive. (Note that there may be either 2 or 4 INTR pins, depending upon the JTAGSEL pin setting.) The H_RESET operation will unconditionally cause the HOLD signal to become deasserted. H_RESET will reset T-MAU to Link Fail state.

H_RESET is generated by proper assertion of either the RESET or $\overline{\text{RESET}}$ pin, depending upon the mode that has been selected through the LB/$\overline{\text{VESA}}$ pin.

S_RESET

S_RESET = SOFTWARE_RESET is a PCnet-32 RESET operation that has been created by a *read* access to the RESET REGISTER which is located at offset 14h from the PCnet-32 controller I/O base address.

S_RESET will RESET all of or some portions of CSR0, 3, 4, 15, 80, 100 and 124 to default values. S_RESET will RESET *NONE* of the BCR locations to default values. S_RESET will cause the microcode program to jump to its RESET state. Following the end of the S_RESET operation, the PCnet-32 controller will NOT attempt to read the EEPROM device. See also the subsection on RESET Register in the I/O Register Access section under *Software Access*. The S_RESET operation will not cause INTR pins to become inactive. S_RESET will set the T-MAU into Link Fail state.

Note that S_RESET will **not** cause a deassertion of the HOLD signal, if it happens to be active at the time of the read to the reset register. The HOLD signal will remain active until the HLDA signal is synchronously sampled as asserted. Following the read of the RESET register, on the next clock cycle after the HLDA signal is synchronously sampled as asserted, (except in Am386 mode, when two cycles are needed) the PCnet-32 controller will deassert the HOLD signal; No bus master accesses will have been performed during this brief bus ownership period.

STOP

STOP is a PCnet-32 RESET operation that has been created by the ASSERTION of the STOP bit in CSR0. That is, a STOP RESET is generated by writing a ONE to the STOP bit of CSR0 *when the STOP bit currently has a value of ZERO*. If the STOP bit value is currently a ONE, and a ONE is rewritten to the STOP bit, then NO STOP RESET will be generated. The STOP operation will *not* cause INTR pins to become inactive.

STOP will RESET all or some portions of CSR0, 3, and 4 to default values; STOP will RESET *NONE* of the BCR locations to default values. STOP will cause the microcode program to jump to its RESET state. Following the end of the STOP operation, the PCnet-32 controller will not attempt to read the EEPROM device. For the identity of individual CSRs and bit locations that are affected by STOP, see the individual CSR register descriptions. Setting the STOP bit does not affect the T-MAU.

USER ACCESSIBLE REGISTERS

The PCnet-32 controller implements all PCnet-ISA (Am79C960) registers, all LANCE (Am7990) registers, all ILACC (Am79C900) registers, plus a number of additional registers. The PCnet-32 controller registers are compatible with both the PCnet-ISA (Am79C960) registers and all of the LANCE (Am7990) registers upon power up. Compatibility to the ILACC set of registers requires one access to the Software Style register (BCR20, bits 7–0) to be performed. By setting an appropriate value of the Software Style register (BCR20, bits 7–0) the user can select a set of registers that are compatible with the ILACC set of registers.

Note that all register locations are defined to be 16 bits in width when WIO mode is selected. When DWIO mode is selected, all register locations are defined to be 32 bits in width. When performing register write operations in DWIO mode, the upper 16 bits should always be written as zeros, except APROM locations. When performing register read operations in DWIO mode, the upper 16 bits of I/O resources should always be written as ZEROs, except for APROM locations and CSR88. When performing register read operations in DWIO mode, the upper 16 bits of I/O resources should always be regarded as having undefined values, except for the APROM locations and CSR88.

PCnet-32 controller registers can be divided into three groups:

Setup registers: Registers that are intended to be initialized by the system initialization procedure (e.g. BIOS device initialization routine) or by the device driver to program the operation of various PCnet-32 controller features

Running registers: Registers that are intended to be used by the device driver software once the PCnet-32 controller is running to access status information and to pass control information

Test registers: Registers that are intended to be used only for testing and diagnostic purposes

Below is a list of the registers that fall into each of the first two categories. Those registers that are not included in either of these lists can be assumed to be intended for diagnostic purposes.

Setup Registers

The following is a list of those registers that would typically need to be programmed once during the setup of the PCnet-32 controller within a system. The control bits in each of these registers typically do not need to be modified once they have been written. However, there are no restrictions as to how many times these registers may actually be accessed. Note that if the default power up values of any of these registers is acceptable to the application, then such registers need never be accessed at all. Also note that some of these registers may be programmable through the EEPROM read operation, and therefore do not necessarily need to be written to by the system initialization procedure or by the driver software.

CRS1	Initialization Address[15:0]
CSR2	Initialization Address[31:16]
CSR3	Interrupt Masks and Deferral Control
CSR4	Test and Features Control
CSR8	Logical Address Filter [15:0]
CSR9	Logical Address Filter [31:16]
CSR10	Logical Address Filter [47:32]
CSR11	Logical Address Filter [63:48]
CSR12	Physical Address Filter [15:0]
CSR13	Physical Address Filter [31:16]
CSR14	Physical Address Filter [47:32]
CSR15	Mode Register
CSR24	Base Address of Receive Ring Lower
CSR25	Base Address of Receive Ring Upper
CSR30	Base Address of Transmit Ring Lower
CRS31	Base Address of Transmit Ring Upper
CSR47	Polling Interval
CSR58	Software Style
CSR76	Receive Ring Length
CSR78	Transmit Ring Length
CSR80	Cycle Register and FIFO Threshold Control
CSR82	Bus Activity Timer
CSR100	Memory Error Time-out Register
CSR122	Receiver Packet Alignment Control
BCR2	MAU configuration
BCR16	I/O Base Address Lower
BCR17	I/O Base Address Upper
BCR18	Bus Size and Burst Control Register
BCR19	EEPROM Control and Status Register
BCR20	Software Style
BCR21	Interrupt Control

Running Registers

The following is a list of those registers that would typically need to be periodically read and perhaps written during the normal running operation of the PCnet-32 controller within a system. Each of these registers contains control bits or status bits or both.

RAP	Register Address Port Register
CSR0	PCnet-32 Controller Status Register
CSR4	Test and Features Control
CSR112	Missed Frame Count
CSR114	Receive Collision Count

RAP Register

The RAP (Register Address Pointer) register is used to gain access to CSR and BCR registers on board the PCnet-32 controller. The value of the RAP indicates the address of a CSR or BCR whenever an RDP or BDP access is performed. That is to say, RAP serves as a pointer to CSR and BDP space.

As an example of RAP use, consider a read access to CSR4. In order to access this register, it is necessary to first load the value 0004 into the RAP by performing a write access to the RAP offset of 12h (12h when WIO mode has been selected, 14h when DWIO mode has been selected). The data for the RAP write would be 0004. Then a second access is performed on the PCnet-32 controller, this time to the RDP offset of 10h (for either WIO or DWIO mode). The RDP access is a read access, and since RAP has just been loaded with the value of 0004, the RDP read will yield the contents of CSR4. A read of the BDP at this time (offset of 16h when WIO mode has been selected, 1Ch when DWIO mode has been selected) will yield the contents of BCR4, since the RAP is used as the pointer into both BDP and RDP space.

RAP: Register Address Port

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	RES	Reserved locations. Read and written as zeros.
7-0	RAP	Register Address Port. The value of these 8 bits determines which CSR or BCR will be accessed when an I/O access to the RDP or BDP port, respectively, is performed. RAP is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.

Control and Status Registers

The CSR space is accessible by performing accesses to the RDP (Register Data Port). The particular CSR that is read or written during an RDP access will depend upon the current setting of the RAP. RAP serves as a pointer into the CSR space. RAP also serves as the pointer to BCR space, which is described in a later section.

CSR0: PCnet-32 Controller Status

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	ERR	Error is set by the ORing of BABL, CERR, MISS, and MERR. ERR remains set as long as any of the error flags are true. ERR is read only. Write operations are ignored.
14	BABL	Babble is a transmitter time-out error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum length frame. BABL will be set if 1519 bytes or greater are transmitted. When BABL is set, INTR is asserted if IENA = 1 and the mask bit BABLM in CSR3 is clear. BABL assertion will set the ERR bit. BABL is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. BABL is cleared by H_RESET or S_RESET or setting the STOP bit.
13	CERR	Collision Error indicates that the collision inputs to the AUI port failed to activate within 20 network bit times after chip terminated transmission (SQE Test). This feature is a transceiver test feature. In 10BASE-T mode CERR will be set if a transmission is attempted while the T-MAU is in Link Fail state. CERR assertion will not result in an interrupt being generated. CERR assertion will set the ERR bit. CERR is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. CERR is cleared by H_RESET or S_RESET or setting the STOP bit.
12	MISS	Missed Frame is set when PCnet-32 controller has lost an incoming receive frame resulting from a Receive Descriptor not being available. This bit is the only immediate indication that receive data has been lost since there is no current receive descriptor to write status to. When MISS is set, INTR is asserted if IENA = 1 and the mask bit MISSM in CSR3 is clear. MISS assertion will set the ERR bit. MISS is set by the Buffer Management Unit and cleared by writing a "1". Writing a "0" has no effect. MISS is cleared by H_RESET or S_RESET or setting the STOP bit.

11 MERR

Memory Error is set when PCnet-32 controller requests the use of the system interface bus by asserting HOLD and has not received HLDA assertion after a programmable length of time. The length of time in microseconds before MERR is asserted will depend upon the setting of the Bus Time-Out Register (CSR100). The default setting of CSR100 will give a MERR after 51.2 µs of bus latency.

When MERR is set, INTR is asserted if IENA = 1 **and** the mask bit MERRM in CSR3 is clear. MERR assertion will set the ERR bit, regardless of the settings of IENA and MERRM .

MERR is set by the Bus Interface Unit and cleared by writing a "1". Writing a "0" has no effect. MERR is cleared by H_RESET or S_RESET or by setting the STOP bit.

10 RINT

Receive interrupt. RINT is set by the Buffer Management Unit of the PCnet-32 controller after the last descriptor of a receive packet has been updated by writing a ZERO to the ownership bit. RINT may *also* be set when the first descriptor of a receive packet has been updated by writing a ZERO to the ownership bit *if* the SPRINTEN bit of CSR3 has been set to a ONE.

When RINT is set, INTR is asserted if IENA = 1 and the mask bit RINTM in CSR3 is clear.

RINT is cleared by the host by writing a "1". Writing a "0" has no effect. RINT is cleared by H_RESET or S_RESET or by setting the STOP bit.

9 TINT

Transmit interrupt is set after completion of a transmit frame and toggling of the OWN bit in the last buffer in the Transmit Descriptor Ring.

When TINT is set, INTR is asserted if IENA = 1 and the mask bit TINTM in CSR3 is clear.

TINT is set by the Buffer Management Unit after the last transmit buffer has been updated and cleared by writing a "1". Writing a "0" has no effect. TINT is cleared by H_RESET or S_RESET or setting the STOP bit.

8 IDON

Initialization Done indicates that the initialization sequence has completed. When IDON is set, PCnet-32 controller has read the Initialization block from memory.

When IDON is set, INTR is asserted if IENA = 1 and the mask bit IDONM in CSR3 is clear.

IDON is set by the Buffer Management Unit after the initialization block has been read from memory and cleared by writing a "1". Writing a "0" has no effect. IDON is cleared by H_RESET or S_RESET or setting the STOP bit.

7 INTR

interrupt Flag indicates that one or more following interrupt causing conditions has occurred: BABL, MISS, MERR, MFCO, RCVCCO, RINT, RPCO, TINT, IDON, JAB or TXSTRT. and its associated mask bit is clear. If IENA = 1 and INTR is set, INTR will be active.

INTR is read only. INTR is cleared by H_RESET or S_RESET or by setting the STOP bit or by clearing all of the active individual interrupt bits that have not been masked out.

6 IENA

interrupt Enable allows INTR to be active if the interrupt Flag is set. If IENA = "0" then INTR will be disabled regardless of the state of INTR.

IENA is set by writing a "1" and cleared by writing a "0". IENA is cleared by H_RESET or S_RESET or setting the STOP bit.

5 RXON

Receive On indicates that the Receive function is enabled. RXON is set if DRX (CSR15[0]) = "0" after the START bit is set. If INIT and START are set together, RXON will not be set until after the initialization block has been read in.

RXON is read only. RXON is cleared by H_RESET or S_RESET or setting the STOP bit.

4 TXON

Transmit On indicates that the Transmit function is enabled. TXON is set if DTX (CSR15[1]) = "0" after the START bit is set. If INIT and START are set together, TXON will not be set until

after the initialization block has been read in.

TXON is read only. TXON is cleared by H_RESET or S_RESET or setting the STOP bit.

3 TDMD — Transmit Demand, when set, causes the Buffer Management Unit to access the Transmit Descriptor Ring without waiting for the poll-time counter to elapse. If TXON is not enabled, TDMD bit will be reset and no Transmit Descriptor Ring access will occur.

TDMD is required to be set if the DPOLL bit in CSR4 is set. Setting TDMD while DPOLL = 0 merely hastens the PCnet-32 controller's response to a Transmit Descriptor Ring Entry.

TDMD is set by writing a "1". Writing a "0" has no effect. TDMD will be cleared by the Buffer Management Unit when it fetches a Transmit Descriptor. TDMD is cleared by H_RESET or S_RESET or setting the STOP bit.

2 STOP — STOP assertion disables the chip from all DMA activity. The chip remains inactive until either STRT or INIT are set. If STOP, STRT and INIT are all set together, STOP will override STRT and INIT.

STOP is set by writing a "1" or by H_RESET or S_RESET. Writing a "0" has no effect. STOP is cleared by setting either STRT or INIT.

1 STRT — STRT assertion enables PCnet-32 controller to send and receive frames, and perform buffer management operations. Setting STRT clears the STOP bit. If STRT and INIT are set together, PCnet-32 controller initialization will be performed first.

STRT is set by writing a "1". Writing a "0" has no effect. STRT is cleared by H_RESET or S_RESET or by setting the STOP bit.

0 INIT — INIT assertion enables PCnet-32 controller to begin the initialization procedure which reads in the initialization block from memory. Setting INIT clears the STOP bit. If STRT and INIT are set together, PCnet-32 controller initialization will be performed first. INIT is not cleared when the initialization sequence has completed.

INIT is set by writing a "1". Writing a "0" has no effect. INIT is cleared by H_RESET or S_RESET or by setting the STOP bit.

CSR1: IADR[15:0]

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	IADR[15:0]	Lower 16 bits of the address of the Initialization Block. Regardless of the value of SSIZE32(BCR20/CSR58, bit 8) IADR[1:0] must be zero. This register is aliased with CSR16. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by H_RESET or S_RESET or by setting the STOP bit.

CSR2: IADR[31:16]

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	IADR[31:24]	If SSIZE32 is set (BCR20[8]), then the IADR[31:24] bits will be used *strictly* as the upper 8 bits of the initialization block address. However, if SSIZE32 is reset, then the IADR[31:24] bits will be used to generate the upper 8 bits of all bus mastering addresses, as required for a 32 bit address bus. Note that the 16-bit software structures specified by the SSIZE32 = 0 setting will yield only 24 bits of address for PCnet-32 controller bus master accesses, while the 32-bit hardware for which the PCnet-32 controller is intended will require 32 bits of address. Therefore, whenever SSIZE32 = 0, the IADR[31:24] bits will be appended to the 24-bit initialization address, to each 24-bit descriptor base address and to each beginning 24-bit buffer address in order to form complete 32-bit addresses. The upper 8 bits that exist in the descriptor address registers and the buffer address

registers which are stored on board the PCnet-32 controller will be overwritten with the IADR[31:24] value, so that CSR accesses to these registers will show the 32 bit address that includes the appended field.

If SSIZE32 = 1, then software will provide 32-bit pointer values for all of the shared software structures - i.e. descriptor bases and buffer addresses, and therefore,

IADR[31:24] will not be written to the upper 8 bits of any of these resources, but it will be used as the upper 8 bits of the initialization address.

Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by H_RESET or S_RESET or by setting the STOP bit.

7-0 IADR[23:16] Bits 23 through 16 of the address of the Initialization Block. Whenever this register is written, CSR17 is updated with CSR2's contents.

Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by H_RESET or S_RESET or by setting the STOP bit.

CSR3: Interrupt Masks and Deferral Control

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	RES	Reserved location. Read and written as zero.
14	BABLM	Babble Mask. If BABLM is set, the BABL bit in CSR0 will be masked and unable to set INTR flag in CSR0. Read/Write accessible always. BABLM is cleared by H_RESET or S_RESET and is not affected by STOP.
13	RES	Reserved location. Read and written as zero.
12	MISSM	Missed Frame Mask. If MISSM is set, the MISS bit in CSR0 will be masked and unable to set INTR flag in CSR0. Read/Write accessible always. MISSM is cleared by H_RESET or S_RESET and is not affected by STOP.
11	MERRM	Memory Error Mask. If MERRM is set, the MERR bit in CSR0 will be masked and unable to set INTR flag in CSR0. Read/Write accessible always. MERRM is cleared by H_RESET or S_RESET and is not affected by STOP.
10	RINTM	Receive Interrupt Mask. If RINTM is set, the RINT bit in CSR0 will be masked and unable to set INTR flag in CSR0. Read/Write accessible always. RINTM is cleared by H_RESET or S_RESET and is not affected by STOP.
9	TINTM	Transmit interrupt Mask. If TINTM is set, the TINT bit in CSR0 will be masked and unable to set INTR flag in CSR0. Read/Write accessible always. TINTM is cleared by H_RESET or S_RESET and is not affected by STOP.
8	IDONM	Initialization Done Mask. If IDONM is set, the IDON bit in CSR0 will be masked and unable to set INTR flag in CSR0. Read/Write accessible always. IDONM is cleared by H_RESET or S_RESET and is not affected by STOP.
7	RES	Reserved location. Read and written as zeroes.
6	DXSUFLO	Disable Transmit Stop on Underflow error. When DXSUFLO (CSR3, bit 6) is set to ZERO, the transmitter is turned off when an UFLO error occurs (CSR0, TXON = 0). When DXSUFLO is set to ONE, the PCnet-32 controller gracefully recovers from an UFLO error. It scans the transmit descriptor ring until it finds the start of a new frame and starts a new transmission. Read/Write accessible always. DXSUFLO is cleared by H_RESET or S_RESET and is not affected by STOP.
5	LAPPEN	Look-Ahead Packet Processing Enable. When set to a ONE, the LAPPEN bit will cause the PCnet-32 controller to generate an interrupt following the descriptor write operation to the *first* buffer of a receive frame. This

interrupt will be generated *in addition* to the interrupt that is generated following the descriptor write operation to the last *buffer* of a receive frame. The interrupt will be signaled through the RINT bit of CSR0.

Setting LAPPEN to a ONE also enables the PCnet-32 controller to read the STP bit of receive descriptors. The PCnet-32 controller will use the STP information to determine where it should begin writing a receive frame's data. Note that while in this mode, the PCnet-32 controller can write intermediate frame data to buffers whose descriptors do not contain STP bits set to ONE. Following the write to the last descriptor used by a frame, the PCnet-32 controller will scan through the next descriptor entries to locate the next STP bit that is set to a ONE. The PCnet-32 controller will begin writing the next frame's data to the buffer pointed to by that descriptor.

Note that because several descriptors may be allocated by the host for each frame, and not all messages may need all of the descriptors that are allocated between descriptors that contain STP = ONE, then some descriptors/buffers may be skipped in the ring. While performing the search for the next STP bit that is set to ONE, the PCnet-32 controller will advance through the receive descriptor ring regardless of the state of ownership bits. If any of the entries that are examined during this search indicate PCnet-32 controller ownership of the descriptor but also indicate STP = "0", then the PCnet-32 controller will RESET the OWN bit to ZERO in these entries. If a scanned entry indicates host ownership with STP = "0" then the PCnet-32 controller will not alter the entry, but will advance to the next entry.

When the STP bit is found to be true, but the descriptor that contains this setting is not owned by the PCnet-32 controller, then the PCnet-32 controller will stop advancing through the ring entries and begin periodic polling of this entry. When the STP bit is found to be true, and the descriptor that contains this setting is owned by the PCnet-32 controller, then the PCnet-32 controller will stop advancing through the ring entries, store the descriptor information that it has just read, and wait for the next receive to arrive.

This behavior allows the host software to pre-assign buffer space in such a manner that the "header" portion of a receive frame will always be written to a particular memory area, and the "data" portion of a receive frame will always be written to a separate memory area. The interrupt is generated when the "header" bytes have been written to the "header" memory area.

Read/Write accessible always. The LAPPEN bit will be reset to ZERO by H_RESET or S_RESET and will be unaffected by STOP.

See Appendix D for more information on the LAPP concept.

4 DXMT2PD

Disable Transmit Two Part Deferral. If DXMT2PD is set, Transmit Two Part Deferral will be disabled.

Read/Write accessible always. DXMT2PD is cleared by H_RESET or S_RESET and is not affected by STOP.

3 EMBA

Enable Modified Back-off Algorithm. If EMBA is set, a modified back-off algorithm is implemented.

Read/Write accessible always. EMBA is cleared by H_RESET or S_RESET and is not affected by STOP.

2 BSWP

Byte Swap. This bit is used to choose between big and little Endian modes of operation. When BSWP is set to a ONE, big Endian mode is selected. When BSWP is set to ZERO, little Endian mode is selected.

When big Endian mode is selected, the PCnet-32 controller will swap the order of bytes on the data bus during FIFO transfers only. Specifically, D31-24 becomes Byte0, D23-16 becomes Byte1, D15-8 becomes Byte2 and D7-0 becomes Byte3 when big Endian mode is selected. When little Endian mode is selected, the order of bytes on the data bus is: D31-24 is Byte3,

		D23-16 is Byte2, D15-8 is Byte1 and D7-0 is Byte0. Byte swap only affects data transfers that involve the FIFOs. Initialization block transfers are not affected by the setting of the BSWP bit. Descriptor transfers are not affected by the setting of the BSWP bit. RDP, RAP and BDP accesses are not affected by the setting of the BSWP bit. APROM transfers are not affected by the setting of the BSWP bit. BSWP is write/readable regardless of the state of the STOP bit. BSWP is cleared by H_RESET or S_RESET and is not affected by STOP.
1-0	RES	Reserved locations. The default value is ZERO for both locations. Writing a ONE to these bits has no effect on device function; if a ONE is written to these bits, then a ONE will be read back. Existing drivers may write a ONE to these bits for compatibility, but new drivers should write a ZERO to these bits and should treat the read value as undefined.

CSR4: Test and Features Control

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	ENTST	Enable Test Mode operation. Setting ENTST to ONE enables internal test functions which are useful only for stand alone integrated circuit testing. In addition, the Runt Packet Accept (RPA) bit (CSR124, bit 3) may be changed only when ENTST is set to ONE. To enable RPA, the user must first write a ONE to the ENTST bit. Next, the user must first write a ONE to the RPA bit (CSR124, bit 3). Finally, the user must write a ZERO to the ENTST bit to take the device out of test mode operation. Once the RPA bit has been set to ONE, the device will remain in the Runt Packet Accept mode until the RPA bit is cleared to ZERO. Read/Write accessible. ENTST is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.
14	DMAPLUS	When DMAPLUS = "1", disables the burst transaction counter, CSR80. If DMAPLUS = "0", the burst transaction counter is enabled. Read and Write accessible. DMAPLUS is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.
13	TIMER	Timer Enable Register. If TIMER is set, the Bus Activity Timer Register, CSR82 is enabled. If TIMER is cleared, the Bus Activity Timer Register is disabled. Read/Write accessible. TIMER is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.
12	DPOLL	Disable Transmit Polling. If DPOLL is set, the Buffer Management Unit will disable transmit polling. Likewise, if DPOLL is cleared, automatic transmit polling is enabled. If DPOLL is set, TDMD bit in CSR0 must be periodically set in order to initiate a manual poll of a transmit descriptor. Transmit descriptor polling will not take place if TXON is reset. Read/Write accessible. DPOLL is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.
11	APAD_XMT	Auto Pad Transmit. When set, APAD_XMT enables the automatic padding feature. Transmit frames will be padded to extend them to 64 bytes including FCS. The FCS is calculated for the entire frame including pad, and appended after the pad field. APAD_XMT will override the programming of the DXMTFCS bit. Read and Write accessible. APAD_XMT is reset by H_RESET or S_RESET and is unaffected by the STOP bit.
10	ASTRP_RCV	Auto Strip Receive. When set, ASTRP_RCV enables the automatic pad stripping feature. The pad and FCS fields will be stripped from receive frames and not placed in the FIFO. Read and Write accessible. ASTRP_RCV is reset by H_RESET or S_RESET and is unaffected by the STOP bit.
9	MFCO	Missed Frame Counter Overflow interrupt. Indicates the MPC

(CSR112) wrapped around. Can be cleared by writing a 1 to this bit. Also cleared by H_RESET or S_RESET or setting the STOP bit. Writing a 0 has no effect.

When MFCO is set, INTR is asserted if IENA = 1 and the mask bit MFCOM is cleared.

When the SWSTYLE register (BCR20[7:0]) has been programmed to the ILACC compatibility mode, then this bit has no meaning and the PCnet-32 controller will never set the value of this bit to ONE.

8 MFCOM — Missed Frame Counter Overflow Mask. If MFCOM is set, MFCO will be unable to set INTR in CSR0. Set to a ONE by H_RESET or S_RESET, unaffected by the STOP bit.

When the SWSTYLE register (BCR20[7:0]) has been programmed to the ILACC compatibility mode, then this bit has no meaning and the PCnet-32 controller will set the value of this bit to a ZERO.

7 RES — Reserved location. Written as zero and read as zero.

6 RES — Reserved location. This bit may be written to as either a ONE or a ZERO, but will always be read as a ZERO. This bit has no effect on PCnet-32 controller operation.

5 RCVCCO — Receive Collision Counter Overflow. Indicates the Receive Collision Counter (CSR114) wrapped around. Can be cleared by writing a 1 to this bit. Also cleared by H_RESET or S_RESET or by setting the STOP bit. Writing a 0 has no effect.

When RCVCCO is set, INTR is asserted if IENA=1 and the mask bit RCVCCOM is cleared.

When the SWSTYLE register (BCR20[7:0]) has been programmed to the ILACC compatibility mode, then this bit has no meaning and PCnet-32 controller will not set the value of this bit to ONE.

4 RCVCCOM — Receive Collision Counter Overflow Mask. If RCVCCOM is set, RCVCCO will be unable to set INTR in CSR0. RCVCCOM is set to a ONE by H_RESET or S_RESET and is not affected by STOP.

When the SWSTYLE register (BCR20[7:0]) has been programmed to the ILACC compatibility mode, then this bit has no meaning and PCnet-32 controller will set the value of this bit to a ZERO.

3 TXSTRT — Transmit Start status is set whenever PCnet-32 controller begins transmission of a frame. When TXSTRT is set, INTR is asserted if IENA = 1 and the mask bit TXSTRTM (CSR4 bit 2) is cleared.

TXSTRT is set by the MAC Unit and cleared by writing a "1", by H_RESET or S_RESET, or setting the STOP bit. Writing a "0" has no effect.

2 TXSTRTM — Transmit Start Mask. If TXSTRTM is set, the TXSTRT bit in CSR4 will be masked and unable to set INTR flag in CSR0.

Read/Write accessible. TXSTRTM is set to a ONE by H_RESET or S_RESET and is not affected by the STOP bit.

1 JAB — Jabber Error is set when the PCnet-32 controller Twisted-pair MAU function exceeds an allowed transmission limit. Jabber is set by the T-MAU cell and can only be asserted in 10BASE-T mode.

When JAB is set, INTR is asserted if IENA = 1 and the mask bit JABM (CSR4[0]) is cleared.

JAB is set by the T-MAU circuit and cleared by writing a "1". Writing a "0" has no effect. JAB is also cleared by H_RESET or S_RESET or setting the STOP bit.

When the SWSTYLE register (BCR20[7:0]) has been programmed to the ILACC compatibility mode, then this bit has no meaning and PCnet-32 controller will never set the value of this bit to ONE.

0 JABM — Jabber Error Mask. If JABM is set, the JAB bit in CSR4 will be masked and unable to set INTR flag in CSR0.

Read/Write accessible. JABM is set to a ONE by H_RESET or S_RESET and is not affected by STOP.

When the SWSTYLE register (BCR20[7:0]) has been programmed to the ILACC compatibility mode, then this bit has no meaning and PCnet-32 controller will set the value of this bit to a ZERO.

CSR6: RX/TX Descriptor Table Length

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-12	TLEN	Contains a copy of the transmit encoded ring length (TLEN) field read from the initialization block during PCnet-32 controller initialization. This field is written during the PCnet-32 controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. TLEN is only defined after initialization. These bits are unaffected by H_RESET, S_RESET, or STOP.
11-8	RLEN	Contains a copy of the receive encoded ring length (RLEN) read from the initialization block during PCnet-32 controller initialization. This field is written during the PCnet-32 controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. RLEN is only defined after initialization. These bits are unaffected by H_RESET, S_RESET, or STOP.
7-0	RES	Reserved locations. Read as zero. Write operations should not be performed.

CSR8: Logical Address Filter, LADRF[15:0]

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	LADRF[15:0]	Logical Address Filter, LADRF[15:0]. Defined only after the initialization block has been successfully read or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR9: Logical Address Filter, LADRF[31:16]

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	LADRF[31:16]	Logical Address Filter, LADRF[31:16]. Defined only after the initialization block has been successfully read or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR10: Logical Address Filter, LADRF[47:32]

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	LADRF[47:32]	Logical Address Filter, LADRF[47:32]. Defined only after the initialization block has been successfully read or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR11: Logical Address Filter, LADRF[63:48]

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	LADRF[63:48]	Logical Address Filter, LADRF[63:48]. Defined only after the initialization block has been successfully read or a direct I/O write has been performed on this register Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR12: Physical Address Register, PADR[15:0]

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	PADR[15:0]	Physical Address Register, PADR[15:0]. Defined only after the initialization block has been successfully read or a direct I/O write has been performed on this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR13: Physical Address Register, PADR[31:16]

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	PADR[31:16]	Physical Address Register, PADR[31:16]. Defined only after the initialization block has been successfully read or a direct I/O write has been performed on this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR14: Physical Address Register, PADR[47:32]

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	PADR[47:32]	Physical Address Register, PADR[47:32]. Defined only after the initialization block has been successfully read or a direct I/O write has been performed on this register. The PADR bits are transmitted PADR[0] first and PADR[47] last. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR15: Mode Register

Bit	Name	Description
		This register's fields are loaded during the PCnet-32 controller initialization routine with the corresponding Initialization Block values or a direct I/O write has been performed to this register.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	PROM	Promiscuous Mode. When PROM = "1", all incoming receive frames are accepted. Read/write accessible only when STOP bit is set.
14	DRCVBC	Disable Receive Broadcast. When set, disables the PCnet-32 controller from receiving broadcast messages. Used for protocols that do not support broadcast addressing, except as a function of multicast. DRCVBC is cleared by H_RESET or S_RESET (broadcast messages will be received) and is unaffected by STOP. Read/write accessible only when STOP bit is set.
13	DRCVPA	Disable Receive Physical Address. When set, the physical address detection (Station or node ID) of the PCnet-32 controller will be disabled. Frames addressed to the nodes individual physical address will not be recognized (although the frame may be accepted by the EADI mechanism). Read/write accessible only when STOP bit is set.
12	DLNKTST	Disable Link Status. When DLNKTST = "1", monitoring of Link Pulses is disabled. When DLNKTST = "0", monitoring of Link Pulses is enabled. This bit only has meaning when the 10BASE-T network interface is selected. Read/write accessible only when STOP bit is set.
11	DAPC	Disable Automatic Polarity Correction. When DAPC = "1", the 10BASE-T receive polarity reversal algorithm is disabled. Like-

wise, when DAPC = "0", the polarity reversal algorithm is enabled.

This bit only has meaning when the 10BASE-T network interface is selected.

Read/write accessible only when STOP bit is set.

10 MENDECL — MENDEC Loopback Mode. See the description of the LOOP bit in CSR15.

Read/write accessible only when STOP bit is set.

9 LRT/TSEL — Low Receive Threshold (T-MAU Mode only)

Transmit Mode Select (AUI Mode only)

LRT — Low Receive Threshold. When LRT = "1", the internal twisted pair receive thresholds are reduced by 4.5 dB below the standard 10BASE-T value (approximately 3/5) and the unsquelch threshold for the RXD circuit will be 180–312 mV peak.

When LRT = "0", the unsquelch threshold for the RXD circuit will be the standard 10BASE-T value, 300 - 520 mV peak.

In either case, the RXD circuit post squelch threshold will be one half of the unsquelch threshold.

This bit only has meaning when the 10BASE-T network interface is selected.

Read/write accessible only when STOP bit is set. Cleared by H_RESET or S_RESET and is unaffected by STOP.

TSEL — TSEL Transmit Mode Select. TSEL controls the levels at which the AUI drivers rest when the AUI transmit port is idle. When TSEL = 0, DO+ and DO- yield "zero" differential to operate transformer coupled loads (Ethernet 2 and 802.3). When TSEL = 1, the DO+ idles at a higher value with respect to DO-, yielding a logical HIGH state (Ethernet 1).

This bit only has meaning when the AUI network interface is selected.

Read/write accessible only when STOP bit is set. Cleared by H_RESET or S_RESET and is unaffected by STOP.

8-7 PORTSEL[1:0] — Port Select bits allow for software controlled selection of the network medium.

PORTSEL settings of AUI and 10BASE-T are ignored when the ASEL bit of BCR2 (bit 1) has been set to ONE.

The network port configuration is shown in Table 41.

Table 41. Network Port Configuration

PORTSEL[1:0]	ASEL (BCR2[1])	Link Status (of 10BASE-T)	Network Port
0X	1	Fail	AUI
0X	1	Pass	10BASE-T
0 0	0	X	AUI
0 1	0	X	10BASE-T
1 0	X	X	GPSI
1 1	X	X	Reserved

Refer to the section on General Purpose Serial Interface for detailed information on accessing GPSI.

Read/write accessible only when STOP bit is set. Cleared by H_RESET or S_RESET and is unaffected by STOP.

6 INTL — Internal Loopback. See the description of LOOP, CSR15-2.

Read/write accessible only when STOP bit is set.

5 DRTY — Disable Retry. When DRTY = "1", PCnet-32 controller will attempt only one transmission. If DRTY = "0", PCnet-32 controller will attempt 16 retry attempts before signaling a retry error. DRTY is defined when the initialization block is read.

Read/write accessible only when STOP bit is set.

4 FCOLL — Force Collision. This bit allows the collision logic to be tested. PCnet-32 controller must be in internal loopback for FCOLL to be valid. If FCOLL = "1", a collision will be forced during loopback transmission attempts. a Retry Error will ultimately result. If FCOLL = "0", the Force Collision logic will be disabled.

Read/write accessible only when STOP bit is set.

3 DXMTFCS — Disable Transmit CRC (FCS). When DXMTFCS = 0, the

transmitter will generate and append a FCS to the transmitted frame. When DXMTFCS = 1, the FCS logic is allocated to the receiver and no FCS is generated or sent with the transmitted frame. DXMTFCS is overridden when ADD_FCS is set in TMD1.

See also the ADD_FCS bit in TMD1. If DXMTFCS is set and ADD_FCS is clear for a particular frame, no FCS will be generated. The value of ADD_FCS is valid only when STP is set. If ADD_FCS is set for a particular frame, the state of DXMTFCS is ignored and a FCS will be appended on that frame by the transmit circuitry.

In loopback mode, this bit determines if the transmitter appends FCS or if the receiver checks the FCS.

This bit was called DTCR in the LANCE (Am7990).

Read/write accessible only when STOP bit is set.

2 LOOP — Loopback Enable allows PCnet-32 controller to operate in full duplex mode for test purposes. When LOOP = "1", loopback is enabled. In combination with INTL and MENDECL, various loopback modes are defined in Table 42.

Table 42. Loopback Modes

LOOP	INTL	MENDECL	Loopback Mode
0	X	X	Non-Loopback
1	0	X	External Loopback
1	1	0	Internal Loopback Include MENDEC
1	1	1	Internal Loopback Exclude MENDEC

Read/write accessible only when STOP bit is set. LOOP is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.

1 DTX — Disable Transmit results in PCnet-32 controller not accessing the Transmit Descriptor Ring and therefore no transmissions are attempted. DTX = "0", will set TXON bit (CSR0 bit 4) if STRT (CSR0 bit 1) is asserted.

Read/write accessible only when STOP bit is set.

0 DRX — Disable Receiver results in PCnet-32 controller not accessing the Receive Descriptor Ring and therefore all receive frame data are ignored. DRX = "0", will set RXON bit (CSR0 bit 5) if STRT (CSR0 bit 1) is asserted.

Read/write accessible only when STOP bit is set.

CSR16: Initialization Block Address Lower

Bit	Name	Description
		This register is an alias of the location CSR1. Accesses to/from this register are equivalent to access to CSR1.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	IADR	Lower 16 bits of the address of the Initialization Block. This register is an alias of CSR1. Whenever this register is written, CSR1 is updated with CSR16's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR17: Initialization Block Address Upper

Bit	Name	Description
		This register is an alias of the location CSR2. Accesses to/from this register are equivalent to access to CSR2.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	IADR	Upper 16 bits of the address of the Initialization Block. This register is an alias of CSR2. Whenever this register is written, CSR2 is updated with CSR17's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by RESET.

CSR18: Current Receive Buffer Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	CRBA	Contains the lower 16 bits of the current receive buffer address at

which the PCnet-32 controller will store incoming frame data.

Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR19: Current Receive Buffer Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	CRBA	Contains the upper 16 bits of the current receive buffer address at which the PCnet-32 controller will store incoming frame data. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET, or STOP.

CSR20: Current Transmit Buffer Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	CXBA	Contains the lower 16 bits of the current transmit buffer address from which the PCnet-32 controller is transmitting. Read/write accessible only when STOP bit is set.

CSR21: Current Transmit Buffer Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	CXBA	Contains the upper 16 bits of the current transmit buffer address from which the PCnet-32 controller is transmitting. Read/write accessible only when STOP bit is set.

CSR22: Next Receive Buffer Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NRBA	Contains the lower 16 bits of the next receive buffer address to which the PCnet-32 controller will store incoming frame data. Read/write accessible only when STOP bit is set.

CSR23: Next Receive Buffer Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NRBA	Contains the upper 16 bits of the next receive buffer address to which the PCnet-32 controller will store incoming frame data. Read/write accessible only when STOP bit is set.

CSR24: Base Address of Receive Ring Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	BADR	Contains the lower 16 bits of the base address of the Receive Ring. Read/write accessible only when STOP bit is set.

CSR25: Base Address of Receive Ring Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	BADR	Contains the upper 16 bits of the base address of the Receive Ring. Read/write accessible only when STOP bit is set.

CSR26: Next Receive Descriptor Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NRDA	Contains the lower 16 bits of the next RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR27: Next Receive Descriptor Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.

15-0	NRDA	Contains the upper 16 bits of the next RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR28: Current Receive Descriptor Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	CRDA	Contains the lower 16 bits of the current RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR29: Current Receive Descriptor Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	CRDA	Contains the upper 16 bits of the current RDRE address pointer. Read/write accessible only when STOP bit is set.

CSR30: Base Address of Transmit Ring Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	BADX	Contains the lower 16 bits of the base address of the Transmit Ring. Read/write accessible only when STOP bit is set.

CSR31: Base Address of Transmit Ring Upper

Bit	Name	Description
31-16	RES	Reserved locations. Read and written as zero.
15-0	BADX	Contains the upper 16 bits of the base address of the Transmit Ring. Read/write accessible only when STOP bit is set.

CSR32: Next Transmit Descriptor Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NXDA	Contains the lower 16 bits of the next TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR33: Next Transmit Descriptor Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NXDA	Contains the upper 16 bits of the next TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR34: Current Transmit Descriptor Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Read and written as zero.
15-0	CXDA	Contains the lower 16 bits of the current TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR35: Current Transmit Descriptor Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	CXDA	Contains the upper 16 bits of the current TDRE address pointer. Read/write accessible only when STOP bit is set.

CSR36: Next Next Receive Descriptor Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NNRDA	Contains the lower 16 bits of the next next receive descriptor address pointer. Read/write accessible only when STOP bit is set.

CSR37: Next Next Receive Descriptor Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NNRDA	Contains the upper 16 bits of the next next receive descriptor address pointer. Read/write accessible only when STOP bit is set.

CSR38: Next Next Transmit Descriptor Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NNXDA	Contains the lower 16 bits of the next next transmit descriptor address pointer. Read/write accessible only when STOP bit is set.

CSR39: Next Next Transmit Descriptor Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NNXDA	Contains the upper 16 bits of the next next transmit descriptor address pointer. Read/write accessible only when STOP bit is set.

CSR40: Current Receive Status and Byte Count Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-12	RES	Reserved locations. Read and written as zero.
11-0	CRBC	current Receive Byte Count. This field is a copy of the BCNT field of RMD2 of the current receive descriptor. Read/write accessible only when STOP bit is set.

CSR41: Current Receive Status and Byte Count Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	CRST	Current Receive Status. This field is a copy of bits 15:8 of RMD1 of the current receive descriptor. Read/write accessible only when STOP bit is set.
7-0	RES	Reserved locations. Read and written as zero.

CSR42: Current Transmit Status and Byte Count Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-12	RES	Reserved locations. Read and written as zero.
11-0	CXBC	current Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the current transmit descriptor. Read/write accessible only when STOP bit is set.

CSR43: Current Transmit Status and Byte Count Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	CXST	Current Transmit Status. This field is a copy of bits 15:8 of TMD1 of the current transmit descriptor. Read/write accessible only when STOP bit is set.
7-0	RES	Reserved locations. Read and written as zero.

CSR44: Next Receive Status and Byte Count Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-12	RES	Reserved locations. Read and written as zero.
11-0	NRBC	Next Receive Byte Count. This field is a copy of the BCNT field of RMD2 of the next receive descriptor. Read/write accessible only when STOP bit is set.

CSR45: Next Receive Status and Byte Count Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	NRST	Next Receive Status. This field is a copy of bits 15:8 of RMD1 of the next receive descriptor. Read/write accessible only when STOP bit is set.
7-0	RES	Reserved locations. Read and written as zero.

CSR46: Poll Time Counter

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	POLL	Poll Time Counter. This counter is incremented by the PCnet-32 controller microcode and is used to trigger the descriptor ring polling operation of the PCnet-32 controller. Read/write accessible only when STOP bit is set.

CSR47: Polling Interval

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	POLLINT	Polling Interval. This register contains the time that the PCnet-32 controller will wait between successive polling operations. The POLLINT value is expressed as the two's complement of the desired interval, where each bit of POLLINT represents 1 BCLK period of time (486 and VL-Bus modes; 2 BCLK 386 mode). POLLINT[3:0] are ignored. (POLLINT[16] is implied to be a one, so POLLINT[15] is significant, and does not represent the sign of the two's complement POLLINT value.) The default value of this register is 0000. This corresponds to a polling interval of 65,536 BCLK periods (486 and VL-Bus modes; 131,072 BCLK 386 mode). The POLLINT value of 0000 is crxeated during the microcode initialization routine, and therefore might not be seen when reading CSR47 after H_RESET or S_RESET. If the user desires to program a value for POLLINT other than the default, then the correct procedure is to first set *INIT only* in CSR0. Then, when the initialization sequence is complete, the user must set STOP in CSR0. Then the user may write to CSR47 and then set STRT in CSR0. In this way, the default value of 0000 in CSR47 will be overwritten with the desired user value. If the user does NOT use the standard initialization procedure (standard implies use of an initialization block in memory and setting the INIT bit of CSR0), but instead, chooses to write directly to each of the registers that are involved in the INIT operation, then it is imperative that the user

also write 0000 0000 to CSR47 as part of the alternative initialization sequence.

Read/write accessible only when STOP bit is set.

CSR48: Temporary Storage 2 Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP2	Lower 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR49: Temporary Storage 2 Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP2	Upper 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR50: Temporary Storage 3 Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP3	Lower 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR51: Temporary Storage 3 Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP3	Upper 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR52: Temporary Storage 4 Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP4	Lower 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR53: Temporary Storage 4 Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP4	Upper 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR54: Temporary Storage 5 Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP5	Lower 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR55: Temporary Storage 5 Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP5	Upper 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR56: Temporary Storage 6 Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP6	Lower 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR57: Temporary Storage 6 Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP6	Upper 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR58: Software Style

Bit	Name	Description
		This register is an alias of the location BCR20. Accesses to/from this register are equivalent to accesses to BCR20.
31-10	RES	Reserved locations. Written as zeros and read as undefined.
9	CSRPCNET	CSR PCnet-ISA configuration bit. When set, this bit indicates that the PCnet-32 controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the PCnet-ISA (Am79C960) device. When cleared, this bit indicates that PCnet-32 controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the ILACC (Am79C900) device. The value of CSRPCNET is determined by the PCnet-32 controller. CSRPCNET is read only by the host. The PCnet-32 controller uses the setting of the Software Style register (BCR20[7:0]/CSR58[7:0]) to determine the value for this bit. CSRPCNET is set to a ONE by H_RESET or S_RESET and is not affected by STOP.
8	SSIZE32	Software Size 32 bits. When set, this bit indicates that the PCnet-32 controller utilizes AMD 79C900 (ILACC) software structures. In particular, Initialization Block and Transmit and Receive descriptor bit maps are affected. When cleared, this bit indicates that the PCnet-32 controller utilizes AMD PCnet-ISA software structures. ***Note:*** Regardless of the setting of SSIZE32, the Initialization Block must always begin on a double-word boundary. The value of SSIZE32 is determined by the PCnet-32 controller. SSIZE32 is read only by the host. The PCnet-32 controller uses the setting of the Software Style register (BCR20, bits 7-0) to determine the value for this bit. SSIZE32 is cleared by H_RESET or S_RESET and is not affected by STOP. If SSIZE32 is reset, then bits IADR[31:24] of CSR2 will be used to generate values for the upper 8 bits of the 32 bit address bus during master accesses initiated by the PCnet-32 controller. This action is required, since the 16-bit software structures specified by the SSIZE32=0 setting will yield only 24 bits of address for PCnet-32 controller bus master accesses. If SSIZE32 is set, then the software structures that are common to the PCnet-32 controller and the host system will supply a full 32 bits for each address pointer that is needed by the PCnet-32 controller for performing master accesses. The value of the SSIZE32 bit has no effect on the drive of the upper 8 address pins. The upper 8 address pins are always driven, regardless of the state of the SSIZE32 bit. Note that the setting of the SSIZE32 bit has no effect on the defined width for I/O resources. I/O resource width is determined by the state of the DWIO bit.
7-0	SWSTYLE	Software Style register. The value in this register determines the "style" of I/O and memory resources that are used by the PCnet-32 controller. The S/W resource style selection will affect the interpretation of a few bits within the CSR space and the width of the descriptors and initialization block. See Table 43. All PCnet-32 controller CSR bits and BCR bits and all descriptor, buffer and initialization block entries not cited in the table above are unaffected by the Software Style selection and are therefore always fully functional as specified in the BCR and CSR sections. Read/write accessible only when STOP bit is set. The SWSTYLE register will contain the value 00h following H_RESET or S_RESET and will be unaffected by STOP.

Table 43. Software Resource Style Selection

SWSTYLE[7:0] (Hex)	Style Name	CSRPCNET	SSIZE32	Altered Bit Interpretations
00	LANCE/ PCnet-ISA	1	0	ALL CSR4 bits will function as defined in the CSR4 section. TMD1[29] functions as ADD_FCS
01	ILACC	0	1	CSR4[9:8], CSR4[5:4] and CSR4[1:0] will have *no function*, but will be writeable and readable. CSR4[15:10], CSR4[7:6] and CSR4[3:2] will function as defined in the CSR4 section. TMD1[29] becomes NO_FCS.
02	PCnet-32	1	1	ALL CSR4 bits will function as defined in the CSR4 section. TMD1[29] functions as ADD_FCS
All other combinations	Reserved	Undefined	Undefined	Undefined

CSR59: IR Register

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	IRREG	Contains the value 0105. This register always contains the same value. It is not writeable. Read accessible only when STOP bit is set.

CSR60: Previous Transmit Descriptor Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	PXDA	Contains the lower 16 bits of the previous TDRE address pointer. PCnet-32 controller has the capability to stack multiple transmit frames. Read/write accessible only when STOP bit is set.

CSR61: Previous Transmit Descriptor Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	PXDA	Contains the upper 16 bits of the previous TDRE address pointer. PCnet-32 controller has the capability to stack multiple transmit frames. Read/write accessible only when STOP bit is set.

CSR62: Previous Transmit Status and Byte Count Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-12	RES	Reserved locations. Read and written as zero. Accessible only when STOP bit is set.
11-0	PXBC	Previous Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the previous transmit descriptor. Read/write accessible only when STOP bit is set.

CSR63: Previous Transmit Status and Byte Count Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	PXST	Previous Transmit Status. This field is a copy of bits 15:8 of TMD1 of the previous transmit descriptor. Read/write accessible only when STOP bit is set.
7-0	RES	Reserved locations. Read and written as zero. Accessible only when STOP bit is set.

CSR64: Next Transmit Buffer Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NXBA	Contains the lower 16 bits of the next transmit buffer address from which the PCnet-32 controller will transmit an outgoing frame. Read/write accessible only when STOP bit is set.

CSR65: Next Transmit Buffer Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	NXBA	Contains the upper 16 bits of the next transmit buffer address from which the PCnet-32 controller will transmit an outgoing frame. Read/write accessible only when STOP bit is set.

CSR66: Next Transmit Status and Byte Count Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-12	RES	Reserved locations. Read and written as zero. Accessible only when STOP bit is set.
11-0	NXBC	Next Transmit Byte Count. This field is a copy of the BCNT field of TMD2 of the next transmit descriptor. Read/write accessible only when STOP bit is set.

CSR67: Next Transmit Status and Byte Counter Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-8	NXST	Next Transmit Status. This field is a copy of bits 15:8 of TMD1 of the next transmit descriptor. Read/write accessible only when STOP bit is set.
7-0	RES	Reserved locations. Read and written as zero. Accessible only when STOP bit is set.

CSR68: Transmit Status Temporary Storage Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	XSTMP	Lower 16 bits of a Transmit Status Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR69: Transmit Status Temporary Storage Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	XSTMP	Transmit Status Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR70: Temporary Storage Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP8	Lower 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR71: Temporary Storage Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	TMP8	Upper 16 bits of a Temporary Storage location. Read/write accessible only when STOP bit is set.

CSR72: Receive Ring Counter

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RCVRC	Receive Ring Counter location. Contains a Two's complement

binary number used to number the current receive descriptor. This counter interprets the value in CSR76 as pointing to the first descriptor. a counter value of zero corresponds to the last descriptor in the ring.

Read/write accessible only when STOP bit is set.

CSR74: Transmit Ring Counter

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	XMTRC	Transmit Ring Counter location. Contains a Two's complement binary number used to number the current transmit descriptor. This counter interprets the value in CSR78 as pointing to the first descriptor. A counter value of zero corresponds to the last descriptor in the ring. Read/write accessible only when STOP bit is set.

CSR76: Receive Ring Length

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RCVRL	Receive Ring Length. Contains the two's complement of the receive descriptor ring length. This register is initialized during the PCnet-32 controller initialization routine based on the value in the RLEN field of the initialization block. However, this register can be manually altered. the actual receive ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. Read/write accessible only when STOP bit is set.

CSR78: Transmit Ring Length

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	XMTRL	Transmit Ring Length. Contains the two's complement of the transmit descriptor ring length. This register is initialized during the PCnet-32 controller initialization routine based on the value in the TLEN field of the initialization block. However, this register can be manually altered. the actual transmit ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. Read/write accessible only when STOP bit is set.

CSR80: Burst and FIFO Threshold Control

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-14	RES	Reserved locations. Read as ones and written as zero.
13-12	RCVFW[1:0]	Receive FIFO Watermark. RCVFW controls the point at which receive DMA is requested in relation to the number of received bytes in the receive FIFO. RCVFW specifies the number of bytes which must be present (once the frame has been verified as a non-runt) before receive DMA is requested. Note however that in order for receive DMA to be performed for a new frame, at least 64 bytes must have been received. This effectively avoids having to react to receive frames which are runts or suffer a collision during the slot time (512 bit times). If the Runt Packet Accept feature is enabled, receive DMA will be requested as soon as either the RCVFW threshold is reached, or a complete valid receive frame is detected (regardless of length). RCVFW is set to a value of 10 (64 bytes) after H_RESET or S_RESET and is unaffected by STOP.

RCVFW[1:0]	Bytes Received
00	16
01	32
10	64
11	Reserved

Read/write accessible only when STOP bit is set.

Certain combinations of watermark programming and LINBC (BCR18[2-0]) programming may create situations where no linear bursting is possible, or where the

FIFO may be excessively read or excessively written. Such combinations are declared as illegal.

Combinations of watermark settings and LINBC settings must obey the following relationship:

watermark (in bytes) ≥ LINBC (in bytes)

Combinations of watermark and LINBC settings that violate this rule may cause unexpected behavior.

11-10 XMTSP[1:0] Transmit Start Point. XMTSP controls the point at which preamble transmission attempts commence in relation to the number of bytes written to the transmit FIFO for the current transmit frame. When the entire frame is in the FIFO, transmission will start regardless of the value in XMTSP. XMTSP is given a value of 10 (64 bytes) after H_RESET or S_RESET and is unaffected by STOP. Regardless of XMTSP, the FIFO will not internally over write its data until at least 64 bytes (or the entire frame if <64 bytes) have been transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be re-written to the transmit FIFO, and re-tries will be handled autonomously by the MAC. This bit is read/write accessible only when the STOP bit is set.

XMTSP[1:0]	Bytes Written
00	4
01	16
10	64
11	112

9-8 XMTFW[1:0] Transmit FIFO Watermark. XMTFW specifies the point at which transmit DMA stops, based upon the number of write cycles that could be performed to the transmit FIFO without FIFO overflow. Transmit DMA is allowed at any time when the number of write cycles specified by SMTFW could be executed without causing transmit FIFO overflow. XMTFW is set to a value of 00b (8 cycles) after H_RESET or S_RESET and is unaffected by STOP. Read/write accessible only when STOP bit is set.

XMTFW[1:0]	Write Cycles
00	8
01	16
10	32
11	Reserved

Certain combinations of watermark programming and LINBC programming may create situations where no linear bursting is possible, or where the FIFO may be excessively read or excessively written. Such combinations are declared as illegal.

Combinations of watermark settings and LINBC settings must obey the following relationship:

watermark (in bytes) ≥ LINBC (in bytes)

Combinations of watermark and LINBC settings that violate this rule may cause unexpected behavior.

7-0 DMACR[7:0] DMA Cycle Register. This register contains the maximum allowable number of transfers to system memory that the Bus Interface will perform during a single DMA cycle. The Cycle Register is not used to limit the number of transfers during Descriptor transfers. A value of zero will be interpreted as one transfer. During H_RESET or S_RESET a value of 16 is loaded in the BURST register. If the DMAPLUS bit in CSR4 is set, the DMA Cycle Register is disabled. When the ENTST bit in CSR4 is set, all writes to this register will automatically perform a decrement cycle.

When the Cycle Register times out in the middle of a linear burst, the linear burst will continue until a legal starting address is reached, and then the PCnet-32 controller will relinquish the bus.

Therefore, if linear bursting is enabled, and the user wishes the PCnet-32 controller to limit bus activity to desired_max transfers, then the Cycle Register should be programmed to a value of:

Burst count setting = (desired_max DIV (length of linear burst in transfers)) x length of linear burst in transfers where DIV is the operation that yields the

INTEGER portion of the + operation.

Note: If either Linear Burst Write is enabled, the value has to be greater than or equal to 4.

Read/write accessible only when the STOP bit is set.

CSR82: Bus Activity Timer

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	DMABAT	Bus Activity Timer Register. If the TIMER bit in CSR4 is set, this register contains the maximum allowable time that PCnet-32 controller will take up on the system bus during FIFO data transfers for a single DMA cycle. The Bus Activity Timer Register does not limit the number of transfers during Descriptor transfers. The DMABAT value is interpreted as an unsigned number with a resolution of 0.1 µs. For instance, a value of 51 µs would be programmed with a value of 510. If the TIMER bit in CSR4 is set, DMABAT is enabled and must be initialized by the user. The DMABAT register is undefined until written. When the ENTST bit in CSR4 is set, all writes to this register will automatically perform a decrement cycle. If the user has NOT enabled the Linear Burst function and wishes the PCnet-32 controller to limit bus activity to MAX_TIME µs, then the Burst Timer should be programmed to a value of: *MAX_TIME- [(11 + 4w) x (BCLK period)],* *where w = wait states.* If the user has enabled the Linear Burst function and wishes the PCnet-32 controller to limit bus activity to MAX_TIME µs, then the Burst Timer should be programmed to a value of: *MAX_TIME- [((3+lbs) x w + 10 + lbs) x (BCLK period)],* *where w = wait states and lbs = linear burst size in number of transfers per sequence.* This is because the PCnet-32 controller may use as much as one "linear burst size" plus three transfers in order to complete the linear burst before releasing the bus. As an example, if the linear burst size is four transfers, and the number of wait states for the system memory is two, and the BCLK period is 30 ns and the MAX time allowed on the bus is 3 µs, then the Burst Timer should be programmed for: *MAX_TIME- [((3+lbs) x w + 10 + lbs) x (BCLK period)],* *3 µs - [(3 + 4) x 2 +10 + 4) x (30 ns)] = 3 µs - (28 x 30 ns) = 3 - 0.84 µs = 2.16 µs.* Then, if the PCnet-32 controller's Bus ActivityTimer times out after 2.16 µs when the PCnet- 32 controller has completed all but the last three transfers of a linear burst, the PCnet-32 controller may take as much as 0.84 µs to complete the bursts and release the bus. The bus release will occur at 2.16 + 0.84 = 3 µs. A value of zero will in the DMABAT register with the TIMER bit in CSR4 set to ONE will produce single linear burst sequences per bus master period when programmed for linear burst mode, and will yield sets of three transfers when not programmed for linear burst mode. The Bus Activity Timer is set to a value of 00h after H_RESET or S_RESET and is unaffected by STOP. Read/write accessible only when STOP bit is set.

CSR84: DMA Address Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	DMABA	DMA Address Register. This register contains the lower 16 bits of the address of system memory for the current DMA cycle. The Bus Interface Unit controls the Address Register by issuing increment commands to increment the memory address for sequential operations. The DMABA register is undefined

		until the first PCnet-32 controller DMA operation. When the ENTST bit in CSR4 is set, all writes to this register will automatically perform an increment cycle. Read/write accessible only when STOP bit is set.

CSR85: DMA Address Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	DMABA	DMA Address Register. This register contains the upper 16 bits of the address of system memory for the current DMA cycle. The Bus Interface Unit controls the Address Register by issuing increment commands to increment the memory address for sequential operations. The DMABA register is undefined until the first PCnet-32 controller DMA operation. When the ENTST bit in CSR4 is set, all writes to this register will automatically perform an increment cycle. Read/write accessible only when STOP bit is set.

CSR86: Buffer Byte Counter

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-12	RES	Reserved, Read and written with ones.
11-0	DMABC	DMA Byte Count Register. Contains a Two's complement binary number of the current size of the remaining transmit or receive buffer in bytes. This register is incremented by the Bus Interface Unit. The DMABC register is undefined until written. When ENTST (CSR4.15) is asserted, all writes to this register will automatically perform an increment cycle. Read/write accessible only when STOP bit is set.

CSR88: Chip ID Lower

Bit	Name	Description
		This register is exactly the same as the Chip ID register in the JTAG description.
31 - 28		Version. This 4-bit pattern is silicon-revision dependent.
27 - 12		Part number. The 16-bit code for the PCnet-32 controller is 0010 0100 0011 0000b.
11 - 1		Manufacturer ID. The 11-bit manufacturer code for AMD is 00000000001b. This code is per the JEDEC Publication 106-A.
0		Always a logic 1.

CSR89: Chip ID Upper

Bit	Name	Description
		The lower 16 bits of this register are exactly the same as the upper 16 bits of the Chip ID register in the JTAG description, which are exactly the same as the upper 16 bits of CSR88.
31 - 16		Reserved locations. Read as undefined.
15 - 12		Version. This 4-bit pattern is silicon-revision dependent.
11 - 0		Upper 12 bits of the PCnet-32 controller part number, i.e. 0010 0100 0011b.

CSR92: Ring Length Conversion

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RCON	Ring Length Conversion Register. This register performs a ring length conversion from an encoded value as found in the initialization block to a Two's complement value used for internal counting. By writing bits 15-12 with an encoded ring length, a Two's complemented value is read. The RCON register is undefined until written. Read/write accessible only when STOP bit is set.

Read/write accessible only when STOP bit is set.

CSR94: Transmit Time Domain Reflectometry Count

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-10	RES	Reserved locations. Read and written as zero.
9-0	XMTTDR	Time Domain Reflectometry reflects the state of an internal counter that counts from the start of transmission to the occurrence of loss of carrier. TDR is incremented at a rate of 10 MHz. Read accessible only when STOP bit is set. Write operations are ignored. XMTTDR is cleared by H_RESET or S_RESET.

CSR96: Bus Interface Scratch Register 0 Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SCR0	This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. The SCR0 register is undefined until written. Read/write accessible only when STOP bit is set.

CSR97: Bus Interface Scratch Register 0 Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SCR0	This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. The SCR0 register is undefined until written. Read/write accessible only when STOP bit is set.

CSR98: Bus Interface Scratch Register 1 Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SCR1	This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. Read/write accessible only when STOP bit is set.

CSR99: Bus Interface Scratch Register 1 Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SCR1	This register is shared between the Buffer Management Unit and the Bus Interface Unit. All Descriptor Data communications between the BIU and BMU are written and read through SCR0 and SCR1 registers. Read/write accessible only when STOP bit is set.

CSR100: Bus Time-out

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MERRTO	This register contains the value of the longest allowable bus latency (interval between assertion of HOLD and assertion of HLDA) that a slave device may insert into a PCnet-32 controller master transfer. If this value of bus latency is exceeded, then a MERR will be indicated in CSR0, bit 11, and an interrupt may be generated, depending upon the setting of the MERRM bit (CSR3, bit 11) and IENA bit (CSR0[6]). The value in this register is interpreted as a number of XTAL1÷2 clock periods. (I.e. the value in this register is given in 0.1 µs increments.) For example, the value 0200h (512 decimal) will

cause a MERR to be indicated after 51.2 µs of bus latency.

A value of zero will allow an infinitely long bus latency. I.e. a value of zero will never give a bus time-out error. A non-zero value is interpreted as an unsigned number of BCLK cycles.

This register is set to 0200 by H_RESET or S_RESET and is unaffected by STOP.

Read/write accessible only when STOP bit is set.

CSR104: SWAP Register Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SWAP	This register performs word and byte swapping depending upon if 32-bit or 16-bit internal write operations are performed. This register is used internally by the BIU/BMU as a word or byte swapper. The register is externally accessible for test reasons only. CSR104 holds the lower 16 bits and CSR105 holds the upper 16 bits. Read/write accessible only when STOP bit is set.

CSR105: SWAP Register Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	SWAP	This register performs word and byte swapping depending upon if 32-bit or 16-bit internal write operations are performed. This register is used internally by the BIU/BMU as a word or byte swapper. The register is externally accessible for test reasons only. CSR104 holds the lower 16 bits and CSR105 holds the upper 16 bits. Read/write accessible only when STOP bit is set.

CSR108: Buffer Management Scratch Lower

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	BMSCR	The Buffer Management Scratch register is used for assembling Receive and Transmit Status. This register is also used as the primary scan register for Buffer Management Test Modes. BMSCR register is undefined until written. Read/write accessible only when STOP bit is set.

CSR109: Buffer Management Scratch Upper

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	BMSCR	The Buffer Management Scratch register is used for assembling Receive and Transmit Status. This register is also used as the primary scan register for Buffer Management Test Modes. BMSCR register is undefined until written. Read/write accessible only when STOP bit is set.

CSR112: Missed Frame Count

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MFC	Missed Frame Count. Indicates the number of missed frames. MFC will roll over to a count of zero from the value 65535. The MFCO bit of CSR4 (bit 8) will be set each time that this occurs. This register is always readable and is cleared by H_RESET or S_RESET or STOP. A write to this register performs an increment when the ENTST bit in CSR4 is set.

CSR114: Receive Collision Count

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	RCC	Receive Collision Count. Indicates the total number of collisions encountered by the receiver since the last reset of the counter. RCC will roll over to a count of zero from the value 65535. The RCVCCO bit of CSR4 (bit 5) will be set each time that this occurs. The RCC value is read accessible at all times, regardless of the value of the STOP bit. Write operations are ignored. RCC is cleared by H_RESET or S_RESET or by setting the STOP bit. A write to this register performs an increment when the ENTST bit in CSR4 is set.

CSR122: Receive Frame Alignment Control

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-1	RES	Reserved locations, written as zeros and read as undefined.
0	RCVALGN	Receive Frame Align. When set, this bit forces the data field of ISO 8802-3 (IEEE/ANSI 802.3) frames to align to 0 MOD 4 address boundaries (i.e. double word aligned addresses). It is important to note that this feature will only function correctly if all receive buffer boundaries are doubleword aligned and all receive buffers have 0 MOD 4 lengths. In order to accomplish the data alignment, the PCnet-32 controller simply inserts two bytes of random data at the beginning of the receive packet (i.e. before the ISO 8802-3 (IEEE/ANSI 802.3) destination address field). The MCNT field reported to the receive descriptor will not include the extra two bytes. RCVALGN is cleared by H_RESET or S_RESET and is not affected by STOP. Read/write accessible only when STOP bit is set.

CSR124: Buffer Management Test

Bit	Name	Description
		This register is used to place the BMU/BIU into various test mode to support Test/Debug. This register is writeable only when the ENTST bit in CSR4 and the STOP bit of CSR0 are both set. The functions controlled by this register are enabled only if the ENTST bit is set. ENTST should be set before anything in CSR124 can be programmed, including RUNTACC. ENTST must be reset after writing to CSR124 before writing to any other register. If it is done, the PCnet-32 controller will not run. All bits in this register or cleared by H_RESET or S_RESET and are not affected by STOP.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-5	RES	Reserved locations. Read and written as zero.
4	GPSIEN	This bit places the PCnet-32 controller in the GPSI mode. This mode will reconfigure the System Interface Address Pins so that the GPSI port is exposed. This allows bypassing the MENDEC-T-MAU logic. The GPSI mode may also be enabled by test shadow bits setting of BCR18, bits 4 and 3 as in Table 44. See Table 45 for pin reconfiguration in GPSI mode. Note that when the GPSI mode is invoked, only the lower 24 bits of the address bus are available. During Software Relocatable Mode the LED2 pin must be pulled LOW. IOAW24 (BCR21[8]) must be set to allow slave operations. During master accesses in GPSI mode, the PCnet-32 controller will not drive the upper 8 bits of the address bus with address information.
3	RPA	Runt Packet Accept. This bit forces the receive logic to accept runt packets (packets shorter than 64 bytes). The state of the RPA bit can be changed only when the device is in the test mode (when the ENTST bit in CSR4 is set to ONE). To enable

RPA, software must first write a ONE to the ENTST bit. Next, software must write a ONE to the RPA bit. Finally, software must write a ZERO to the ENTST bit to take the device out of test mode operation. Once the RPA bit has been set to ONE, the device will remain in the Runt Packet Accept mode until the RPA bit is cleared to ZERO.

2-0 RES Reserved locations. Written as zeros and read as undefined.

Table 44. GPSI Mode Selection

TSTSHDW Value (BCR18[4:3])	PVALID (BCR19[15])	GPSIEN	Operating Mode
00	X	0	Normal Operating Mode
10	1	X	GPSI Mode
01	1	0	Reserved
11	1	X	Reserved
XX	0	0	Normal Operating Mode
XX	0	1	GPSI Mode

Table 45. GPSI Pin Configurations

GPSI Function	GPSI I/O Type	LANCE GPSI Pin	ILACC GPSI Pin	PCnet-32/ PCnet-ISA GPSI Pin	PCnet-32 Pin Number	PCnet-32 Normal Pin Function
Transmit Data	O	TX	TXD	TXDAT	132	A31
Transmit Enable	O	TENA	RTS	TXEN	133	A30
Transmit Clock	I	TCLK	TXC	STDCLK	134	A29
Collision	I	CLSN	CDT	CLSN	137	A28
Receive Carrier Sense	I	RENA	CRS	RXCRS	138	A27
Receive Clock	I	RCLK	RXC	SRDCLK	140	A26
Receive Data	I	RX	RXD	RXDAT	141	A25

Bus Configuration Registers

The Bus Configuration Registers (BCR) are used to program the configuration of the bus interface and other special features of the PCnet-32 controller that are not related to the IEEE 8802-3 MAC functions. The BCRs are accessed by first setting the appropriate RAP value, and then by performing a slave access to the BDP.

All BCR registers are 16 bits in width in WIO mode and 32 bits in width in DWIO mode. The upper 16 bits of all BCR registers is undefined when in DWIO mode. These bits should be written as ZEROs and should be treated as undefined when read. The "Default" value given for any BCR is the value in the register after H_RESET, and is hexadecimal unless otherwise stated. BCR register values are unaffected by S_RESET and are unaffected by the assertion of the STOP bit.

Note that several registers have no default value. BCR3 and BCR8-BCR15 are reserved and have undefined values. BCR2, BCR16, BCR17 and BCR21 are not observable without first being programmed, either through the EEPROM read operation or through the Software Relocatable Mode. Therefore, the only observable values for these registers are those that have been programmed and a default value is not applicable. See Table 46.

Writes to those registers marked as "Reserved" will have no effect. Reads from these locations will produce *undefined* values.

Table 46. Bus Configuration Registers

RAP Addr.	Mnemonic	Default (Hex)	Name	Programmability		
				User	EEPROM	SRM
0	MSRDA	0005	Master Mode Read Active	No	No	No
1	MSWRA	0005	Master Mode Write Active	No	No	No
2	MC	N/A*	Miscellaneous Configuration	Yes	Yes	Yes
3	Reserved	N/A		No	No	No
4	LNKST	00C0	Link Status (Default)	Yes	No	No
5	LED1	0084	Receive (Default)	Yes	No	No
6	LED2	0088	Receive Polarity (Default)	Yes	No	No
7	LED3	0090	Transmit (Default)	Yes	No	No
8–15	Reserved	N/A		No	No	No
16	IOBASEL	N/A*	I/O Base Address Lower	Yes	Yes	Yes
17	IOBASEU	N/A*	I/O Base Address Upper	Yes	Yes	Yes
18	BSBC	2101	Burst Size and Bus Control	Yes	Yes	No
19	EECAS	0002	EEPROM Control and Status	Yes	No	No
20	SWSTYLE	0000	Software Style	Yes	No	No
21	INTCON	N/A*	Interrupt Control	Yes	Yes	Yes

Key: **SRM** = *Software Relocatable Mode*

* *Registers marked with an asterisk (*) have no default value, since they are not observable without first being programmed, either through the EEPROM read operation or through the Software Relocatable Mode. Therefore, the only observable values for these registers are those that have been programmed and a default value is not applicable.*

BCR0: Master Mode Read Active

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MSRDA	Reserved locations. After H_RESET, the value in this register will be 0005. The settings of this register will have no effect on any PCnet-32 controller function. Writes to this register have no effect on the operation of the PCnet-32 controller and will not alter the value that is read.

BCR1: Master Mode Write Active

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	MSWRA	Reserved locations. After H_RESET, the value in this register will be 0005. The settings of this register will have no effect on any PCnet-32 controller function. Writes to this register have no effect on the operation of the PCnet-32 controller and will not alter the value that is read.

BCR2: Miscellaneous Configuration

Bit	Name	Description
		Note that all bits in this register are programmable through the EEPROM PREAD operation and through the Software Relocatable Mode operation.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	RES	Reserved location. Written and read as zero.
14	T-MAULOOP	When set, this bit allows external loopback packets to pass onto the network through the T-MAU interface, if the T-MAU interface has been selected. If the T-MAU interface has not been selected, then this bit has no effect. This bit is reset to ZERO by H_RESET and is unaffected by S_RESET or STOP.
13-9	RES	Reserved locations. Written and read as zero.
8	IESRWE	IEEE Shadow Ram Write Enable. The PCnet-32 controller contains a shadow RAM on board for storage of the IEEE address following the serial EEPROM read operation. Accesses to APROM I/O Resources will be directed toward this RAM. When IESRWE is set to a ONE, then 32-bit and 16-bit write access to the shadow RAM will be enabled. When IESRWE is set to a ZERO, then 32-bit and 16-bit write access to the shadow RAM will be disabled. At no time are 8- bit write accesses to the shadow RAM allowed. This bit is reset to ZERO by H_RESET and is unaffected by S_RESET or STOP.
7	INTLEVEL	interrupt Level. This bit allows the interrupt output signals to be programmed for edge or level-sensitive applications. When INTLEVEL is set to a ZERO, the selected interrupt pin is configured for edge sensitive operation. In this mode, an interrupt request is signaled by a high level driven on the selected interrupt pin by the PCnet-32 controller. When the interrupt is cleared, the selected interrupt pin is driven to a low level by the PCnet-32 controller. This mode is intended for systems that do not allow interrupt channels to be shared by multiple devices. When INTLEVEL is set to a ONE, the selected interrupt pin is configured for level sensitive operation. In this mode, an interrupt request is signaled by a low level driven on the selected interrupt pin by the PCnet-32 controller. When the interrupt is cleared, the selected interrupt pin is floated by the PCnet-32 controller and allowed to be pulled to a high level by an external pull-up device. This mode is intended for systems which allow the interrupt signal to be shared by multiple devices. This bit is reset to ZERO by H_RESET and is unaffected by R_RESET or STOP.
6-4	RES	Reserved locations. Written and read as zero.
3	EADISEL	EADI Select. When set, this bit configures three of the four LED outputs to function as the outputs of an EADI interface. LED1 becomes SFBD, LED2 becomes SRDCLK and LEDPRE3 becomes SRD. $\overline{\text{LNKST}}$ continues to function as an LED output. In addition to these reassignments, the INTR2 pin will be reassigned to function as the $\overline{\text{EAR}}$ pin. This bit is reset to ZERO by H_RESET and is unaffected by S_RESET or STOP.
2	AWAKE	Auto-Wake. If LNKST is set and AWAKE = "1", the 10BASE-T receive circuitry is active during sleep and listens for Link Pulses. $\overline{\text{LNKST}}$ indicates Link Status and

goes active if the 10BASE-T port comes of out of "link fail" state. This LNKST pin can be used by external circuitry to re-enable the PCnet-32 controller and/or other devices.

When AWAKE = "0", the Auto-Wake circuitry is disabled. This bit only has meaning when the 10BASE-T network interface is selected.

This bit is reset to ZERO by H_RESET and is unaffected by S_RESET or STOP.

1 ASEL

Auto Select. When set, the PCnet-32 controller will automatically select the operating media interface port, unless the user has selected GPSI mode through appropriate programming of the PORTSEL bits of the Mode Register (CSR15). If GPSI mode has not been selected and ASEL has been set to a ONE, then when the 10BASE-T transceiver is in the link pass state (due to receiving valid frame data and/or Link Test pulses or the DLNKTST bit is set), the 10BASE-T port will be used. If GPSI mode has not been selected and ASEL has been set to a ONE, then when the 10BASE-T port is in the link fail state, the AUI port will be used. Switching between the ports will not occur during transmission, to avoid any type of fragment generation.

When ASEL is set to ONE, Link Beat Pulses will be transmitted on the 10BASE-T port, regardless of the state of Link Status. When ASEL is reset to ZERO, Link Beat Pulses will only be transmitted on the 10BASE-T port when the PORTSEL bits of the Mode Register (CSR15) have selected 10BASE-T as the active port.

When ASEL is set to a ZERO, then the selected network port will be determined by the settings of the PORTSEL bits of CSR15.

The ASEL bit is reset to ONE by H_RESET and is unaffected by S_RESET or STOP.

The network port configuration are as follows:

PORTSEL(1:0)	ASEL (BCR2[1])	Link Status (of 10BASE-T)	Network Port
0X	1	0	AUI
0X	1	1	10BASE-T
0 0	0	X	AUI
0 1	0	X	10BASE-T
1 0	X	X	GPSI
1 1	X	X	Reserved

0 RES

Reserved location. The default value of this bit is a ZERO. Writing a ONE to this bit has no effect on device function. Existing drivers may write a ONE to this bit, but new drivers should write a ZERO to this bit.

BCR4: Link Status LED

Bit	Name	Description
		BCR4 controls the function(s) that the LNKST pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR4 defaults to Link Status (LNKST) with pulse stretcher enabled (PSE = 1) and is fully programmable. The default setting after H_RESET for the LNKST register is 00C0h. The LNKST register value is unaffected by S_RESET or STOP.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (Bits 6-0). This bit is READ only by the host, and is unaffected by H_RESET or S_RESET or STOP. This bit is valid only if the network link status is PASS.

14	LEDPOL	LED Polarity. When this bit has the value ZERO, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true and the LED pin will be floated and allowed to float high whenever the OR of the enabled signals is false (i.e. the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit). When this bit has the value ONE, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false (i.e. the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit). The setting of this bit will not affect the polarity of the LEDOUT bit for this register.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value ONE, then the LED output will always be floated. When LEDDIS has the value ZERO, then the LED output value will be governed by the LEDOUT and LEDPOL values.
12-8	RES	Reserved locations. Written as ZEROs, read as undefined.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination time for each new occurrence of the enabled function for this LED output. A value of 0 disables the function. A value of 1 enables the function.
6	LNKSTE	Link Status Enable. Indicates the current link status on the Twisted Pair interface. When this bit is set, a value of ONE will be passed to the LEDOUT signal to indicate that the link status state is PASS. A value of ZERO will be passed to the LEDOUT signal to indicate that the link status state is FAIL. A value of 0 disables the signal. A value of 1 enables the signal.
5	RCVME	Receive Match status Enable. Indicates receive activity on the network that has passed the address match function for this node. All address matching modes are included: Physical, Logical filtering, Promiscuous, Broadcast, and EADI. A value of 0 disables the signal. A value of 1 enables the signal.
4	XMTE	Transmit status Enable. Indicates PCnet-32 controller transmit activity. A value of 0 disables the signal. A value of 1 enables the signal.
3	RXPOLE	Receive Polarity status Enable. Indicates the current Receive Polarity condition on the Twisted Pair interface. A value of ONE indicates that the polarity of the RXD± pair has been reversed. A value of ZERO indicates that the polarity of the RXD± pair has *not* been reversed. Receive polarity indication is valid only if the LNKST bit of BCR4 indicates link PASS status. A value of 0 disables the signal. A value of 1 enables the signal.
2	RCVE	Receive status Enable. Indicates receive activity on the network. A value of 0 disables the signal. A value of 1 enables the signal.
1	JABE	Jabber status Enable. Indicates that the PCnet-32 controller is jabbering on the network. A value of 0 disables the signal. A value of 1 enables the signal.
0	COLE	Collision status Enable. Indicates collision activity on the network. When the AUI port is selected, collision activity during the 4.0 µs internal following a transmit completion (SQE internal) will not activate the LEDOUT bit. A value of 0 disables the signal. A value of 1 enables the signal.

BCR5: LED1 Status

Bit	Name	Description
		BCR5 controls the function(s) that the LED1 pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR5 defaults to Receive Status (RCV) with pulse

stretcher enabled (PSE = 1) and is fully programmable.

The default setting after H_RESET for the LED1 register is 0084h. The LED1 register value is unaffected by S_RESET or STOP.

31-16 RES — Reserved locations. Written as zeros and read as undefined.

15 LEDOUT — This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true.

The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (Bits 6-0).

This bit is READ only by the host, and is unaffected by H_RESET S_RESET or STOP.

This bit is valid only if the network link status is PASS.

14 LEDPOL — LED Polarity. When this bit has the value ZERO, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true and the LED pin will be floated and allowed to float high whenever the OR of the enabled signals is false (i.e. the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit).

When this bit has the value ONE, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false (i.e. the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit).

The setting of this bit will not affect the polarity of the LEDOUT bit for this register.

13 LEDDIS — LED Disable. This bit is used to disable the LED output. When LEDDIS has the value ONE, then the LED output will always be floated. When LEDDIS has the value ZERO, then the LED output value will be governed by the LEDOUT and LEDPOL values.

12-8 RES — Reserved locations. Write as ZEROs, read as undefined.

7 PSE — Pulse Stretcher Enable. Extends the LED illumination time for each new occurrence of the enabled function for this LED output.

A value of 0 disables the function. A value of 1 enables the function.

6 LNKSTE — Link Status Enable. Indicates the current link status on the Twisted Pair interface. When this bit is set, a value of ONE will be passed to the LEDOUT signal to indicate that the link status state is PASS. A value of ZERO will be passed to the LEDOUT signal to indicate that the link status state is FAIL.

A value of 0 disables the signal. A value of 1 enables the signal.

5 RCVME — Receive Match status Enable. Indicates receive activity on the network that has passed the address match function for this node. All address matching modes are included: Physical, Logical filtering, Promiscuous, Broadcast, and EADI.

A value of 0 disables the signal. A value of 1 enables the signal.

4 XMTE — Transmit status Enable. Indicates PCnet-32 controller transmit activity.

A value of 0 disables the signal. A value of 1 enables the signal.

3 RXPOLE — Receive Polarity status Enable. Indicates the current Receive Polarity condition on the Twisted Pair interface. A value of ONE indicates that the polarity of the RXD± pair has been reversed. A value of ZERO indicates that the polarity of the RXD± pair has *not* been reversed.

Receive polarity indication is valid only if the LNKST bit of BCR4 indicates link PASS status.

A value of 0 disables the signal. A value of 1 enables the signal.

2 RCVE — Receive status Enable. Indicates receive activity on the network.

A value of 0 disables the signal. A value of 1 enables the signal.

1	JABE	Jabber status Enable. Indicates that the PCnet-32 controller is jabbering on the network. A value of 0 disables the signal. A value of 1 enables the signal. .
0	COLE	Collision status Enable. Indicates collision activity on the network. When the AUI port is selected, collision activity during the 4.0 µs internal following a transmit completion (SQE internal) will not activate the LEDOUT bit. A value of 0 disables the signal. A value of 1 enables the signal.

BCR6: LED2 Status

Bit	Name	Description
		BCR6 controls the function(s) that the LED2 pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR6 defaults to twisted pair MAU Receive Polarity (RCVPOL) with pulse stretcher enabled (PSE = 1) and is fully programmable. The default setting after H_RESET for the LED2 register is 0088h. The LED2 register value is unaffected by S_RESET or STOP.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (Bits 11-8 and 5-0). This bit is READ only by the host, and is unaffected by H_RESET S_RESET or STOP. This bit is valid only if the network link status is PASS.
14	LEDPOL	LED Polarity. When this bit has the value ZERO, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true and the LED pin will be floated and allowed to float high whenever the OR of the enabled signals is false (i.e. the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit). When this bit has the value ONE, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false (i.e. the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit). The setting of this bit will not affect the polarity of the LEDOUT bit for this register.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value ONE, then the LED output will always be floated. When LEDDIS has the value ZERO, then the LED output value will be governed by the LEDOUT and LEDPOL values.
12-8	RES	Reserved locations. Write as ZEROs, read as undefined.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination time for each new occurrence of the enabled function for this LED output. A value of 0 disables the function. A value of 1 enables the function.
6	LNKSTE	Link Status Enable. Indicates the current link status on the Twisted Pair interface. When this bit is set, a value of ONE will be passed to the LEDOUT signal to indicate that the link status state is PASS. A value of ZERO will be passed to the LEDOUT signal to indicate that the link status state is FAIL. A value of 0 disables the signal. A value of 1 enables the signal.
5	RCVME	Receive Match status Enable. Indicates receive activity on the network that has passed the address match function for this node. All address matching modes are included: Physical, Logical filtering, Promiscuous, Broadcast, and EADI.

		A value of 0 disables the signal. A value of 1 enables the signal.
4	XMTE	Transmit status Enable. Indicates PCnet-32 controller transmit activity. A value of 0 disables the signal. A value of 1 enables the signal.
3	RXPOLE	Receive Polarity status Enable. Indicates the current Receive Polarity condition on the Twisted Pair interface. A value of ONE indicates that the polarity of the RXD± pair has been reversed. A value of ZERO indicates that the polarity of the RXD± pair has *not* been reversed. Receive polarity indication is valid only if the LNKST bit of BCR4 indicates link PASS status. A value of 0 disables the signal. A value of 1 enables the signal.
2	RCVE	Receive status Enable. Indicates receive activity on the network. A value of 0 disables the signal. A value of 1 enables the signal.
1	JABE	Jabber status Enable. Indicates that the PCnet-32 controller is jabbering on the network. A value of 0 disables the signal. A value of 1 enables the signal.
0	COLE	Collision status Enable. Indicates collision activity on the network. When the AUI port is selected, collision activity during the 4.0 µs internal following a transmit completion (SQE internal) will not activate the LEDOUT bit. A value of 0 disables the signal. A value of 1 enables the signal.

BCR7: LED3 Status

Bit	Name	Description
		BCR7 controls the function(s) that the LEDPRE3 pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR7 defaults to Transmit Status (XMT) with pulse stretcher enabled (PSE = 1) and is fully programmable. The default setting after H_RESET for the LED3 register is 0090h. The LED3 register value is unaffected by S_RESET or STOP.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (Bits 11-8 and 5-0). This bit is READ only by the host, and is unaffected by H_RESET S_RESET or STOP. This bit is valid only if the network link status is PASS.
14	LEDPOL	LED Polarity. When this bit has the value ZERO, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true and the LED pin will be floated and allowed to float high whenever the OR of the enabled signals is false (i.e. the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit). When this bit has the value ONE, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false (i.e. the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit). The setting of this bit will not affect the polarity of the LEDOUT bit for this register.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value ONE, then the LED output will always be floated. When LEDDIS has the value ZERO, then the LED output value will be governed by the LEDOUT and LEDPOL values.
12-8	RES	Reserved locations. Write as ZEROs, read as undefined.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination time for each new occurrence of the en-

		abled function for this LED output. A value of 0 disables the function. A value of 1 enables the function.
6	LNKSTE	Link Status Enable. Indicates the current link status on the Twisted Pair interface. When this bit is set, a value of ONE will be passed to the LEDOUT signal to indicate that the link status state is PASS. A value of ZERO will be passed to the LEDOUT signal to indicate that the link status state is FAIL. A value of 0 disables the signal. A value of 1 enables the signal.
5	RCVME	Receive Match status Enable. Indicates receive activity on the network that has passed the address match function for this node. All address matching modes are included: Physical, Logical filtering, Promiscuous, Broadcast, and EADI. A value of 0 disables the signal. A value of 1 enables the signal.
4	XMTE	Transmit status Enable. Indicates PCnet-32 controller transmit activity. A value of 0 disables the signal. A value of 1 enables the signal.
3	RXPOLE	Receive Polarity status Enable. Indicates the current Receive Polarity condition on the Twisted Pair interface. A value of ONE indicates that the polarity of the RXD± pair has been reversed. A value of ZERO indicates that the polarity of the RXD± pair has *not* been reversed. Receive polarity indication is valid only if the LNKST bit of BCR4 indicates link PASS status. A value of 0 disables the signal. A value of 1 enables the signal.
2	RCVE	Receive status Enable. Indicates receive activity on the network. A value of 0 disables the signal. A value of 1 enables the signal.
1	JABE	Jabber status Enable. Indicates that the PCnet-32 controller is jabbering on the network. A value of 0 disables the signal. A value of 1 enables the signal.
0	COLE	Collision status Enable. Indicates collision activity on the network. When the AUI port is selected, collision activity during the 4.0 µs internal following a transmit completion (SQE internal) will not activate the LEDOUT bit. A value of 0 disables the signal. A value of 1 enables the signal.

BCR16: I/O Base Address Lower

Bit	Name	Description
		Note that all bits in this register are programmable through the EEPROM PREAD operation and through the Software Relocatable Mode operation.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-5	IOBASEL	I/O Base Address Lower 16 bits. These bits are used to determine the location of the PCnet-32 controller in all of I/O space. They function as bits 15 through 5 of the I/O address of the PCnet-32 controller. Note that the lowest five bits of the PCnet-32 controller I/O space are not programmable. These bits are assumed to be ZEROs. This means that it is not possible to locate the PCnet-32 controller on a space that does not begin on a 32-byte block boundary. The value of IOBASEL is determined in either of two ways: 1. The IOBASEL value may be set during the EEPROM read. 2. If no EEPROM exists, or if there is an error detected in the EEPROM data, then the PCnet-32 controller will enter Software Relocatable Mode, and a specific sequence of write accesses to I/O address 378h will cause the IOBASEL value to be updated. Refer to the Software Relocatable Mode section of this document for more details. A direct write access to the I/O Base Address Lower register may be performed.

		IOBASEL is not affected by S_RESET or STOP.
4-0	RES	Reserved locations. Written as ZEROs, read as undefined.

BCR17: I/O Base Address Upper

Bit	Name	Description
		Note that all bits in this register are programmable through the EEPROM PREAD operation and software relocatable mode.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-0	IOBASEU	I/O Base Address Upper 16 bits. These bits are used to determine the location of the PCnet-32 controller in all of I/O space. They function as bits 31 through 16 of the I/O address of the PCnet-32 controller. The value of IOBASEU is determined in either of two ways: 1. The IOBASEU value may be set during the EEPROM read. 2. If no EEPROM exists, or if there is an error detected in the EEPROM data, then the PCnet-32 controller will enter Software Relocatable Mode, and a specific sequence of write accesses to I/O address 378h will cause the IOBASEU value to be updated. Refer to the Software Relocatable Mode section of this document for more details. A direct write access to the I/O Base Address Upper register may be performed. IOBASEU is not affected by S_RESET or STOP.

BCR18: Burst Size and Bus Control

Bit	Name	Description
		Note that all bits in this register are programmable through the EEPROM PREAD operation.
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15-11	CLL	Cache Line Length. These bits are used to determine how often a cache invalidation cycle needs to be performed when Generate Cache Invalidation Cycles has been activated by setting the GCIC bit (bit 10 of BCR18) to a one. CLL values are interpreted as multiples of 4 bytes. For example, a CLL value of 00001b means the cache line length is 4 bytes and a cache invalidation cycle (assertion of $\overline{EADS}$) will be performed every 4 bytes. A CLL value of 00010b means the cache line length is 8 bytes, and a cache invalidation cycle will be performed every 8 bytes. A CLL value of 00100 means the cache line length is 16 bytes. A value of 00000 means that the cache line size is "infinite". In other words, a single $\overline{EADS}$ assertion will be performed on the first access at the beginning of each bus mastership period (write accesses only) and no subsequent $\overline{EADS}$ assertions will be made during this bus mastership period. Cache invalidation cycles are performed only during PCnet-32 controller bus master write accesses. Some CLL values are reserved (see chart below). The portion of the Address Bus that will be floated at the time of an address hold operation (AHOLD asserted) will be determined by the value of the Cache Line Length register. The following chart lists all of the legal values of CLL showing the portion of the Address Bus that will become floated during an address hold operation:

CLL Value	Portion of Address Bus Floated During AHOLD
00000	None
00001	A31-A2
00010	A31-A3
00011	Reserved CLL Value
00100	A31-A4
00101–00111	Reserved CLL Values
01000	A31-A5
01001 –01111	Reserved CLL Values
10000	A31-A6
10001– 11111	Reserved CLL Values

Note that the default value of CLL after RESET is 00100b. All timing diagrams in this document are

drawn with the assumption that this is the value of CLL.

When VESA VL-Bus mode is selected, then the LEADS pin functions as if CLL = 00001b regardless of the actual CLL setting.

CLL is set to 00100b by H_RESET and is not affected by S_RESET or STOP.

10 GCIC/IWBACK — Generate Cache Invalidation Cycles. Ignore WBACK signal.

GCIC — Generate Cache Invalidation Cycles. When this bit is set and the Am486 mode has been selected, then the PCnet-32 controller assumes that the host system contains a cache, but the host system does not snoop the local bus master accesses of the PCnet-32 controller, and therefore, that cache invalidation cycles must be generated by the PCnet-32 controller for PCnet-32 controller master accesses. PCnet-32 controller will not perform snoops to invalidate cache lines for local bus accesses that other local bus masters may have executed. When GCIC is a ZERO, the PCnet-32 controller assumes that logic in the host system will create the cache invalidation cycles that may be necessary as a result of PCnet-32 controller local bus master accesses. The cache invalidation logic performs its functions according to the GCIC and CLL bits, regardless of the setting of the Am486/Am386 pin.

IWBACK — Ignore WBACK signal. When this bit is set and the VESA VL-Bus Mode has been selected, then the PCnet-32 device will operate as though the WBACK input pin is always held HIGH, even though the real value of the WBACK input may change. This function is provided to allow the PCnet-32 device to operate in systems which violate VESA VL-Bus WBACK operation by either floating this pin or by always driving this pin LOW.

LEADS is always asserted with each assertion of ADS when VESA VL-Bus mode has been selected, regardless of the setting of the GCIC/IWBACK bit.

GCIC/IWBACK is cleared by H_RESET and is not affected by S_RESET or STOP.

9 PRPCNET — Priority PCnet. This bit is used to set the priority of the daisy chain arbitration logic that resides within the PCnet-32 controller. When PRPCNET = 1, the priority of the daisy chain logic is set to PCnet-32 controller highest priority, External device lowest priority. When PRPCNET = 0, the priority of the daisy chain logic is set to External device highest priority, PCnet-32 controller lowest priority. In the case PRPCNET = 0, where the PCnet- 32 controller has lower priority than the external device and the PCnet-32 controller is preempted due to a HOLDI request from the external device, the PCnet-32 controller will complete the current sequence of accesses and then pass the HLDA to the external device by asserting the HLDAO signal. The HOLD output signal from the PCnet-32 controller will not change state during the HLDAO hand-off. If the PCnet-32 controller is performing a linear burst, then the PCnet-32 controller will complete the linear burst and then pass the HLDA to the external device through the HLDAO signal. For more details on exact timing of preemption events, see the individual descriptions of the various DMA transfers.

PRPCNET is cleared by H_RESET and is not affected by S_RESET or STOP.

The default setting for this bit will be PRPCNET = 0. This default value reflects the nature of the CPU's handling of a HOLD request (i.e. the CPU has lowest priority). By making this the default setting, the PCnet-32 controller response to HOLDI is as close as possible to the timing of the CPU response to HOLD, so that minimal design difficulty will be created by inserting the PCnet-32 controller into the system as the mediating device between the CPU and the extension bus chipset.

8 RES

Reserved bit. Must be written as a ONE. Will be read as a ONE.

This reserved location is SET by H_RESET and is not affected by S_RESET or STOP.

7 DWIO

Double Word I/O. When set, this bit indicates that the PCnet-32 controller is programmed for DWIO mode. When cleared, this bit indicates that the PCnet-32 controller is programmed for Word I/O mode. This bit affects the I/O Resource Offset map and it affects the defined width of the PCnet-32 controller's I/O resources. See the DWIO and WIO sections for more details.

The PCnet-32 controller will set DWIO if it detects a double word *write* access to offset 10h from the PCnet-32 controller I/O Base Address (corresponding to the RDP resource). A double word write access to offset 10h is the only way that the DWIO bit can be set. DWIO cannot be set by a direct write to BCR18.

Once the DWIO bit has been set to a ONE, only a H_RESET can reset it to a ZERO.

DWIO is **read *only*** by the host.

DWIO is cleared by H_RESET and is not affected by S_RESET or STOP.

6 BREADE

Burst Read Enable. When set, this bit enables Linear Bursting during memory read accesses, where Linear Bursting is defined to mean that only the first transfer in the current bus arbitration will contain an address cycle. Subsequent transfers will consist of data only. However, the entire address bus will still be driven with appropriate values during the subsequent cycles, but $\overline{ADS}$ will not be asserted. When cleared, this bit prevents the part from performing linear bursting during read accesses. In no case will the part linearly burst a descriptor access or an initialization access.

BREADE is cleared by H_RESET and is not affected by S_RESET or STOP.

Burst Read activity is not allowed when the BCLK frequency is >33 MHz. Linear bursting is disabled in VL-Bus systems that operate above this frequency by connecting the VLBEN pin to either ID(3) (for VL-Bus version 1.0 systems) or ID(4) AND ID(3) AND ID(1) AND ID(0) (for VL-Bus version 1.1 or 2.0 systems). In Am486-style systems that have BCLK frequencies above 33 MHz, disabling the linear burst capability is ideally carried out through EEPROM bit programming, since the EEPROM programming can be setup for a particular machine's architecture. When the VLBEN pin has been reset to a ZERO, then the BREADE bit will be forced to a value of ZERO. Any attempt to change this value by writing to the BREADE bit location will have no effect.

5 BWRITE

Burst Write Enable. When set, this bit enables Linear Bursting during memory write accesses, where Linear Bursting is defined to mean that only the first transfer in the current bus arbitration will contain an address cycle. Subsequent transfers will consist of data only. However, the entire address bus will still be driven with appropriate values during the subsequent cycles, but $\overline{ADS}$ will not be asserted. When cleared, this bit prevents the part from performing linear bursting during write accesses. In no case will the part linearly burst a descriptor access or an initialization access.

BWRITE is cleared by H_RESET and is not affected by S_RESET or STOP.

Burst Write activity is not allowed when the BCLK frequency is >33 MHz. Linear bursting is disabled in VL-Bus systems that operate above this frequency by connecting the VLBEN pin to either ID(3) (for VL-Bus version 1.0 systems) or ID(4) AND ID(3) AND ID(1) AND ID(0) (for VL-Bus version 1.1 or 2.0 systems). In Am486-style systems that have BCLK frequencies above 33 MHz, disabling the linear burst capability is ideally carried out through EEPROM bit programming, since the EEPROM programming can be setup for a particular machine's architecture. When the VLBEN pin has been reset to a ZERO, then the BWRITE bit will be forced to a

value of ZERO. Any attempt to change this value by writing to the BWRITE bit location will have no effect.

4-3 TSTSHDW Test Shadow bits. These bits are used to place the PCnet-32 controller into GPSI mode. BCR18[3] must be set to ZERO. The operating modes possible are indicated in Table 47.

See Table 48 for pin reconfiguration in GPSI mode.

Table 47. GPSI Mode Selection

TSTSHDW Value (BCR18[4:3])	PVALID (BCR19[15])	GPSIEN (CSR124[4])	Operating Mode
00	X	0	Normal Operating Mode
10	1	X	GPSI Mode
01	1	0	Reserved
11	1	X	Reserved
XX	0	0	Normal Operating Mode
XX	0	1	GPSI Mode

Note that when the GPSI mode is invoked, only the lower 24 bits of the address bus are available. IOAW24 (BCR21[8]) must be set to allow slave operations. During master accesses in GPSI mode, the PCnet-32 controller will not drive the upper 8 bits of the address bus with address information.

These bits are not writeable, regardless of the setting of the ENTST bit in CSR4. Values may only be programmed to these bits through the EEPROM read operation.

BCR18[4:3] are set to 0 by H_RESET and are unaffected by S_RESET or STOP.

2-0 LINBC[2:0] Linear Burst Count. The 3-bit value in this register sets the upper limit for the number of transfer cycles in a Linear Burst. This limit determines how often the PCnet-32 controller will assert the $\overline{ADS}$ signal during linear burst transfers. Each time that the interpreted value of LINBC transfers is reached, the PCnet-32 controller will assert the $\overline{ADS}$ signal with a new valid address. The LINBC value should contain only one active bit. LINBC values with more than one active bit may produce predictable results, but such values will not be compatible with future AMD network controllers. The LINBC entry is shifted by two bits before being used by the PCnet-32 controller. For example, the value LINBC[2:0] = 010 is understood by the PCnet-32 controller to mean 01000 = 8. Therefore, the value LINBC[2:0] = 010 will cause the PCnet-32 controller to issue a new $\overline{ADS}$ every

Table 48. GPSI Pin Configurations

GPSI Function	GPSI I/O Type	LANCE GPSI Pin	ILACC GPSI Pin	PCnet-32/ PCnet-ISA GPSI Pin	PCnet-32 Pin Number	PCnet-32 Normal Pin Function
Transmit Data	O	TX	TXD	TXDAT	132	A31
Transmit Enable	O	TENA	RTS	TXEN	133	A30
Transmit Clock	I	TCLK	TXC	STDCLK	134	A29
Collision	I	CLSN	CDT	CLSN	137	A28
Receive Carrier Sense	I	RENA	CRS	RXCRS	138	A27
Receive Clock	I	RCLK	RXC	SRDCLK	140	A26
Receive Data	I	RX	RXD	RXDAT	141	A25

01000b = 8 transfers. The PCnet-32 controller may linearly burst fewer than the value represented by LINBC, due to other conditions that cause the burst to end prematurely. Therefore, LINBC should be regarded as an upper limit to the length of linear burst.

Note that linear burst operation will only begin on certain addresses. The general rule for linear burst starting addresses is:

A[31:0] MOD (LINBC x 16) = 0.

Table 49 illustrates all possible starting address values for all legal LINBC values (only 1, 2, and 4 are legal; other values are reserved). Note that A[31:8] are don't care values for all addresses. (A[1:0] do not exist within a 32 bit system, however, they are valid bits within the buffer pointer field of descriptor word 0.)

Table 49. Linear Burst Cycles

LINBC[2:0]	LBS = Linear Burst Size (No. of Transfers)	Size of Burst (Byte)	Linear Burst Beginning Addresses (A[31:6] = Don't Care) A[5:0] =
1	4	16	00, 10, 20, 30
2	8	32	00, 20
4	16	64	00

Due to the beginning address restrictions just given, it can be shown that some portion of the address bus will be held stable throughout each linear burst sequence, while the lowest portion of the address bus will change value with each new cycle. The portion of the address bus that will be held stable during a linear burst access is given in Table 50.

Table 50. Linear Burst Address Bus

LINBC Value	Portion of Address Bus Stable During Linear Burst
001	A[31:4]
010	A[31:5]
100	A[31:6]

The assertion of RDYRTN in the place of BRDY within a linear burst cycle will cause the linear burst to be interrupted. In that case, the PCnet-32 controller will revert to ordinary two-cycle transfers, except that BLAST will remain deasserted to show that linear bursting is being requested by the PCnet-32 controller. This situation is defined as interrupted linear burst cycles. If BRDY is sampled as asserted (without also sampling RDYRTN asserted during the same access) during interrupted linear burst cycles, then linear bursting will resume.

There are several events which may cause early termination of linear burst. Among those events are: no more data available for transfer in either a buffer or in the FIFO or if either the Cycle Register (CSR80) or the Bus Activity Timer Register (CSR82) times out. In any of these cases, the PCnet-32 controller will end the Linear Burst by asserting BLAST and then releasing the bus. A Partial Linear Burst may have been sent out before the assertion of BLAST, where "Partial Linear Burst" refers to the case where the number of data words transferred between the last asserted ADS and the assertion of BLAST is less than LINBC.

The value on the address bus will be updated with appropriate values every clock cycle during linear burst operations, even though ADS will not be asserted during every clock cycle.

Certain combinations of watermark programming and LINBC programming may create situations where no linear bursting is possible, or where the FIFO may be excessively read or excessively written. Such combinations are declared as illegal.

Combinations of watermark settings and LINBC settings must obey the following relationship:

watermark (in bytes) ≥ LINBC (in bytes)

Combinations of watermark and LINBC settings that violate this rule may cause unexpected behavior.

LINBC is set to the value of 001 by H_RESET and is not affected by S_RESET or STOP. This gives a default linear burst length of 4 transfers = 001 x 4.

BCR19: EEPROM Control and Status

Bit	Name	Description
31-16	RES	Reserved locations. Written as zeros and read as undefined.
15	PVALID	EEPROM Valid status bit. This bit is read only by the host. A value of ONE in this bit indicates that a PREAD operation has occurred, and that 1) there is an EEPROM connected to the PCnet-32 controller microwire interface pins and 2) the contents read from the EEPROM have passed the checksum verification operation. A value of ZERO in this bit indicates that the contents of the EEPROM are different from the contents of the applicable PCnet-32 controller on-board registers and/or that the checksum for the entire 36 bytes of EEPROM is incorrect or that no EEPROM is connected to the microwire interface pins. PVALID is set to ZERO during H_RESET and is unaffected by S_RESET or the STOP bit. However, following the H_RESET operation, an automatic read of the EEPROM will be performed. Just as is true for the normal PREAD command, at the end of this automatic read operation, the PVALID bit may be set to ONE. Therefore, H_RESET will set the PVALID bit to ZERO at first, but the automatic EEPROM read operation may later set PVALID to a ONE. If PVALID becomes ZERO following an EEPROM read operation (either automatically generated after H_RESET, or requested through PREAD), then all EEPROM-programmable BCR locations will be reset to their H_RESET values. If no EEPROM is present at the EESK, EEDI and EEDO pins, then all attempted PREAD commands will terminate early and PVALID will NOT be set. This applies to the automatic read of the EEPROM after H_RESET as well as to host-initiated PREAD commands.
14	PREAD	EEPROM Read command bit. When this bit is set to a ONE by the host, the PVALID bit (BCR19[15]) will immediately be reset to a ZERO and then the PCnet-32 controller will perform a read operation of 36 bytes from the EEPROM through the microwire interface. The EEPROM data that is fetched during the read will be stored in the appropriate internal registers on board the PCnet-32 controller. Upon completion of the EEPROM read operation, the PCnet-32 controller will assert the PVALID bit. EEPROM contents will be indirectly accessible to the host through I/O read accesses to the Address PROM (offsets 0h through Fh) and through I/O read accesses to other EEPROM programmable registers. Note that I/O read accesses from these locations will not actually access the EEPROM itself, but instead will access the PCnet-32 controller's internal copy of the EEPROM contents. I/O write accesses to these locations may change the PCnet-32 controller register contents, but the EEPROM locations will not be affected. EEPROM locations may be accessed directly through BCR19. At the end of the read operation, the PREAD bit will automatically be reset to a ZERO by the PCnet-32 controller and PVALID will bet set, provided that an EEPROM existed on the microwire interface pins and that the checksum for the entire 36 bytes of EEPROM was correct. Note that when PREAD is set to a ONE, then the PCnet-32 controller will no longer respond to I/O accesses directed toward it, until the PREAD operation has completed successfully. If a PREAD command is given to the PCnet-32 controller but no EEPROM is attached to the microwire interface pins, then the PREAD command will terminate early, the PREAD bit will be cleared to a ZERO and the PVALID bit will remain reset with a value of ZERO. The PCnet-32 controller will then enter Software Relocatable Mode to await further programming. This applies to the automatic read of the EEPROM after H_RESET as

well as to host initiated PREAD commands. EEPROM programmable locations on board the PCnet-32 controller will be set to their default values by such an aborted PREAD operation. For example, if the aborted PREAD operation immediately followed the H_RESET operation, then the final state of the EEPROM programmable locations will be equal to the H_RESET programming for those locations.

If a PREAD command is given to the PCnet-32 controller and the auto-detection pin (EESK/LED1/SFBD) indicates that no EEPROM is present, then the EEPROM read operation will still be attempted.

Note that at the end of the H_RESET operation, a read of the EEPROM will be performed automatically. This H_RESET-generated EEPROM read function will not proceed if the auto-detection pin (EESK/LED1/SFBD) indicates that no EEPROM is present. Instead, Software Relocatable Mode will be entered immediately.

PREAD is set to ZERO during H_RESET and is unaffected by S_RESET or STOP.

PREAD is only writeable when the STOP bit is set to ONE.

13 EEDET — EEPROM Detect. This bit indicates the sampled value of the EESK/LED1/SFBD pin at the end of H_RESET. The value of this bit is independent whether or not an EEPROM is actually present at the EEPROM interface. It is only a function of the sampled value of the EESK/LEDI/SFBD pin at the end of H_RESET.

The value of this bit is determined at the end of the H_RESET operation. It is unaffected by S_RESET or STOP.

This bit is not writeable. It is read only.

Table 51 indicates the possible combinations of EEDET and the existence of an EEPROM and the resulting operations that are possible on the EEPROM microwire interface.

12-5 RES — Reserved locations. Written as ZERO, read as undefined.

4 EEN — EEPROM port enable. When this bit is set to a one, it causes the values of EBUSY, ECS, ESK and EDI to be driven onto the SHFBUSY, EECS, EESK and EEDI pins, respectively. When this bit is reset to a zero, then the SHFBUSY pin will be driven with the inverse of the PVALID (bit 15 of BCR19) value. PVALID is set to "ONE" if the EEPROM read was successful. It is set to "ZERO" otherwise. If EEN = 0

Table 51. EEDET Effects on EEPROM Operation

EEDET Value (BCR19[3])	EEPROM Connected?	Result if PREAD is set to ONE	Result of Automatic EEPROM Read Operation Following H_RESET
0	No	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to ZERO.	First TWO EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to ZERO.
0	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.	First TWO EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to ZERO.
1	No	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to ZERO.	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to ZERO.
1	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.

and no EEPROM read function is currently active, then EECS will be driven LOW. When EEN = 0 and no EEPROM read function is currently active, EESK and EEDI pins will be driven by the LED registers BCR5 and BCR4, respectively. See Table 52.

EEN is set to ZERO by H_RESET and is unaffected by S_RESET or STOP.

3 EBUSY — EEPROM BUSY. This bit controls the value of the SHFBUSY pin of the PCnet-32 controller when the EEN bit is set to ONE and the PREAD bit is set to ZERO. This bit is used to indicate to external EEPROM-programmable logic that an EEPROM access is occurring.

When user programming of the EEPROM is desired through the BCR19 EEPROM Port, then EBUSY should be set to ONE before EEN is set to ONE in systems where EEPROM-programmable external logic exists. At the end of the EEPROM programming operation, EBUSY should either remain set at ONE until after EEN is set to ZERO, or the user may reset EBUSY to ZERO with EEN = 1 immediately following a read of EEPROM byte locations 35 and 36, which should be the last accesses performed during BCR19 accesses to the EEPROM. A programmed PREAD operation following the BCR19 EEPROM programming accesses will cause the SHFBUSY pin to become LOW if the EEPROM checksum is verified.

EBUSY has no effect on the output value of the SHFBUSY pin unless the PREAD bit is set to ZERO and the EEN bit is set to ONE.

EBUSY is set to ZERO by H_RESET and is unaffected by S_RESET or STOP.

2 ECS — EEPROM Chip Select. This bit is used to control the value of the EECS pin of the microwire interface when the EEN bit is set to ONE and the PREAD bit is set to ZERO. If EEN = "1" and PREAD = "0" and ECS is set to a ONE, then the EECS pin will be forced to a HIGH level at the rising edge of the next BCLK following bit programming. If EEN = "1" and PREAD = "0" and ECS is set to a ZERO, then the EECS pin will be forced to a LOW level at the rising edge of the next BCLK following bit programming.

ECS has no effect on the output value of the EECS pin unless the PREAD bit is set to ZERO and the EEN bit is set to ONE.

ECS is set to ZERO by H_RESET and is unaffected by S_RESET or STOP.

1 ESK — EEPROM Serial Clock. This bit and the EDI/EDO bit are used to control host access to the EEPROM. Values programmed to this bit are placed onto the EESK pin at the rising edge of the next BCLK following bit programming, except when the PREAD bit is set to ONE or the EEN bit is set to ZERO. If both the ESK bit and the EDI/EDO bit values are changed during one BCR19 write operation, while EEN = 1, then setup and hold times of the EEDI pin value with respect to the EESK signal edge are not guaranteed.

Table 52. EEPROM Enable

Reset Pin	PREAD or Auto Read in Progress	EEN	EECS	SHFBUSY	EESK	EEDI
High	X	X	0	1	Z	Z
Low	1	X	Active	1	Active	Active
Low	0	1	From ECS Bit of BCR19	From EBUSY Bit of BCR19	From ESK Bit of BCR19	From EEDI Bit of BCR19
Low	0	0	0	$\overline{\text{PVALID}}$	LED1	$\overline{\text{LNKST}}$

		ESK has no effect on the EESK pin unless the PREAD bit is set to ZERO and the EEN bit is set to ONE. ESK is reset to ONE by H_RESET and is unaffected by S_RESET or STOP.
0	EDI/EDO	EEPROM Data In / EEPROM Data Out. Data that is written to this bit will appear on the EEDI output of the microwire interface, except when the PREAD bit is set to ONE or the EEN bit is set to ZERO. Data that is read from this bit reflects the value of the EEDO input of the microwire interface. EDI/EDO has no effect on the EEDI pin unless the PREAD bit is set to ZERO and the EEN bit is set to ONE. EDI/EDO is reset to ZERO by H_RESET and is unaffected by S_RESET or STOP.

BCR20: Software Style

Bit	Name	Description
		This register is an alias of the location CSR58. Accesses to/from this register are equivalent to accesses to CSR58.
31-10	RES	Reserved locations. Written as ZEROs and read as undefined.
9	CSRPCNET	CSR PCnet-ISA configuration bit. When set, this bit indicates that the PCnet-32 controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the PCnet-ISA (Am79C960) device. When cleared, this bit indicates that PCnet-32 controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the ILACC (Am79C900) device. The value of CSRPCNET is determined by the PCnet-32 controller. CSRPCNET is read only by the host. The PCnet-32 controller uses the setting of the Software Style register (BCR20[7:0]) to determine the value for this bit. CSRPCNET is set by H_RESET and is not affected by S_RESET or STOP.
8	SSIZE32	Software Size 32 bits. When set, this bit indicates that the PCnet-32 controller utilizes AMD 79C900 (ILACC) software structures. In particular, Initialization Block and Transmit and Receive descriptor bit maps are affected. When cleared, this bit indicates that the PCnet-32 controller utilizes AMD PCnet-ISA software structures. ***Note:*** *Regardless of the setting of SSIZE32, the Initialization Block must always begin on a double-word boundary.* The value of SSIZE32 is determined by the PCnet-32 controller. SSIZE32 is read only by the host. The PCnet-32 controller uses the setting of the Software Style register (BCR20[7:0]/CSR58[7:0]) to determine the value for this bit. SSIZE32 is cleared by H_RESET and is not affected by S_RESET or STOP. If SSIZE32 is reset, then bits IADR[31:24] of CSR2 will be used to generate values for the upper 8 bits of the 32 bit address bus during master accesses initiated by the PCnet-32 controller. This action is required, since the 16-bit software structures specified by the SSIZE32=0 setting will yield only 24 bits of address for PCnet-32 controller bus master accesses. If SSIZE32 is set, then the software structures that are common to the PCnet-32 controller and the host system will supply a full 32 bits for each address pointer that is needed by the PCnet-32 controller for performing master accesses. The value of the SSIZE32 bit has no effect on the drive of the upper 8 address pins. The upper 8 address pins are always driven, regardless of the state of the SSIZE32 bit. *Note that the setting of the SSIZE32 bit has no effect on the defined width for I/O resources. I/O resource width is determined by the state of the DWIO bit.*
7-0	SWSTYLE	Software Style register. The value in this register determines the "style" of I/O and memory

resources that shall be used by the PCnet-32 controller. The Software Style selection will affect the interpretation of a few bits within the CSR space and the width of the descriptors and initialization block. See Table 53.

All PCnet-32 controller CSR bits and BCR bits and all descriptor, buffer and initialization block entries not cited in the table above are unaffected by the Software Style selection and are therefore always fully functional as specified in the CSR and BCR and descriptor sections.

Read/write accessible only when the STOP bit is set.

The SWSTLYE register will contain the value 00h following H_RESET and is not affected by S_RESET or STOP.

BCR21: Interrupt Control

Bit	Name	Description
		Note that all bits in this register are programmable through the EEPROM PREAD operation and software relocatable mode.
31-9	RES	Reserved locations. Written as ZEROs and read as undefined.
8	IOAW24	I/O Address Width 24 bits. When set to a ONE, the IOAW24 bit will cause the PCnet-32 controller to ignore the upper 8 bits of the address bus when determining whether an I/O address matches PCnet-32 controller I/O space. When IOAW24 is set to a ZERO, then the PCnet-32 controller will examine all 32 bits of the address bus when determining whether an I/O address matches PCnet-32 controller I/O space. Read/write accessible only when the STOP bit is set. The IOAW24 bit will be reset to ZERO by H_RESET and is not affected by S_RESET or STOP.
7	REJECTDIS	Reject Disable. When set to a ONE, the REJECTDIS bit will cause the $\overline{EAR}$ function of the EADI interface to be disabled. Specifically, the INTR2 pin will retain its function as INTR2 and will not function as $\overline{EAR}$, regardless of the setting of the BCR2 EADISEL bit. When reset to a ZERO, the REJECTDIS bit will allow the INTR2 pin to be redefined to function as $\overline{EAR}$ of the EADI interface when the EADISEL bit of BCR2 has been set to a ONE. Read/write accessible only when STOP bit is set. The REJECTDIS bit will be reset to ZERO by H_RESET and is not affected by S_RESET or STOP.
6-2	RES	Reserved locations. Written as ZEROs and read as undefined.
1-0	INTSEL	interrupt Select. The value of these bits determines which of the interrupt pins will be the active interrupt. Table 55 indicates which interrupt will be selected for each combination of INTSEL and JTAGSEL values. Interrupt pins that are not selected will be floated. The INTSEL register will contain the value 00 following H_RESET and is not affected by S_RESET or STOP.

Table 53. Software Style Selection

SWSTYLE[7:0] (Hex)	Style Name	CSRPCNET	SSIZE32	Altered Bit Interpretations
00	LANCE/ PCnet-ISA	1	0	ALL CSR4 bits will function as defined in the CSR4 section. TMD1[29] functions as ADD_FCS
01	ILACC	0	1	CSR4[9:8], CSR4[5:4] and CSR4[1:0] will have *no function*, but will be writeable and readable. CSR4[15:10], CSR4[7:6] and CSR4[3:2] will function as defined in the CSR4 section. TMD1[29] becomes NO_FCS.
02	PCnet-32	1	1	ALL CSR4 bits will function as defined in the CSR4 section. TMD1[29] functions as ADD_FCS
All other combinations	Reserved			Undefined

Table 54. Interrupt Select

INTSEL[1:0]	JTAGSEL	Selected interrupt Pin
00	0	INTR1
01	0	INTR2
10	0	INTR3
11	0	INTR4
00	1	INTR1
01	1	INTR2
10	1	INTR1
11	1	INTR2

Initialization Block

When SSIZE32=0 (BCR20/CSR58, bit 8) the software structures are defined to be 16 bits wide and the initialization block looks as shown in Table 55. When SSIZE32=1, the software structures are defined to be 32 bits wide and the initialization block looks as shown in Table 56. Regardless of the value of SSIZE32, the initialization block must be aligned to a double word boundary, i.e. CSR1[1:0] and CSR16[1:0] must be set to ZERO.

Note that the PCnet-32 device performs doubleword accesses to read the initialization block. This statement is always true, regardless of the setting of the SSIZE32 bit.

Table 55. Initialization Block (when SSIZE = 0)

Address	Bits 15-13	Bit 12	Bits 11-8	Bits 7-4	Bits 3-0
IADR+00	MODE 15-00				
IADR+02	PADR 15-00				
IADR+04	PADR 31-16				
IADR+06	PADR 47-32				
IADR+08	LADRF 15-00				
IADR+0A	LADRF 31-16				
IADR+0C	LADRF 47-32				
IADR+0E	LADRF 63-48				
IADR+10	RDRA 15-00				
IADR+12	RLEN	0	RES	RDRA 23-16	
IADR+14	TDRA 15-00				
IADR+16	TLEN	0	RES	TDRA 23-16	

Table 56. Initialization Block (when SSIZE = 1)

Address	Bits 31-28	Bits 27-24	Bits 23-20	Bits 19-16	Bits 15-12	Bits 11-8	Bits 7-4	Bits 3-0
IADR+00	TLEN	RES	RLEN	RES	MODE			
IADR+04	PADR 31–00							
IADR+08	RES				PADR 47-32			
IADR+0C	LADR 31–00							
IADR+10	LADR 63–32							
IADR+14	RDRA 31–00							
IADR+18	TDRA 31-00							

RLEN and TLEN

When SSIZE32 = 0 (BCR20[8]), then the software structures are defined to be 16 bits wide, and the RLEN and TLEN fields in the initialization block are 3 bits wide, occupying bits 15,14, and 13 and the value in these fields determines the number of Transmit and Receive Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is shown in Table 57.

Table 57. Descriptor Ring Entries (SSIZE32 = 0)

R/TLEN	No. of DREs
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

If a value other than those listed in the above table is desired, CSR76 and CSR78 can be written after initialization is complete. See the description of the appropriate CSRs.

When SSIZE32=1 (BCR20[8]), then the software structures are defined to be 32 bits wide, and the RLEN and TLEN fields in the initialization block are 4 bits wide, occupying bits 15,14, 13, and 12, and the value in these fields determines the number of Transmit and Receive Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is is shown in Table 58.

If a value other than those listed in the above table is desired, CSR76 and CSR78 can be written after initialization is complete. See the description of the appropriate CSRs.

Table 58. Descriptor Ring Entries (SSIZE = 1)

R/TLEN	No. of DREs
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
11XX	512
1X1X	512

RDRA and TDRA

TDRA and RDRA indicate where the transmit and receive descriptor rings, respectively, begin. Each DRE must be located at 0 MOD 16 address values when SSIZE32 = 1 (BCR20[8]). Each DRE must be located at 0 MOD 8 address values when SSIZE32 = 0 (BCR20[8]).

LADRF

The Logical Address Filter (LADRF) is a 64-bit mask that is used to accept incoming Logical Addresses. If the first bit in the incoming address (as transmitted on the wire) is a "1", the address is deemed logical. If the first bit is a "0", it is a physical address and is compared against the physical address that was loaded through the initialization block.

A logical address is passed through the CRC generator, producing a 32 bit result. The high order 6 bits of the CRC is used to select one of the 64 bit positions in the Logical Address Filter. If the selected filter bit is set, the address is accepted and the frame is placed into memory. See Figure 38.

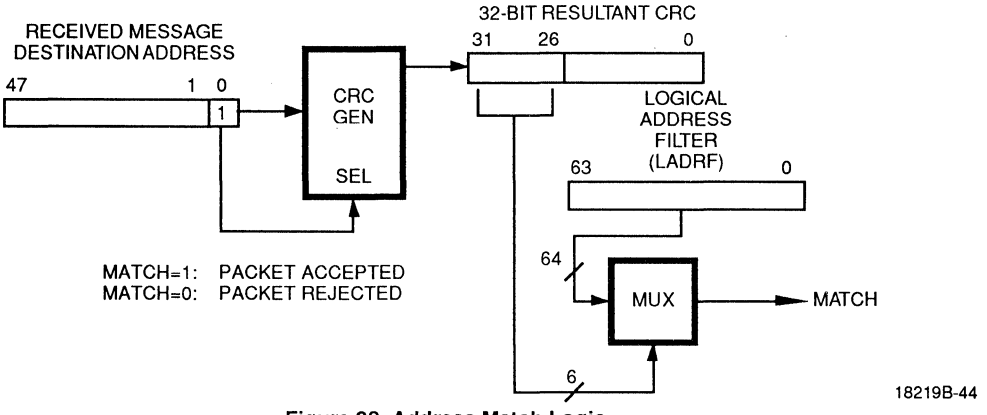


Figure 38. Address Match Logic

The Logical Address Filter is used in multicast addressing schemes. The acceptance of the incoming frame based on the filter value indicates that the message may be intended for the node. It is the node's responsibility to determine if the message is actually intended for the node by comparing the destination address of the stored message with a list of acceptable logical addresses.

If the Logical Address Filter is loaded with all zeroes and promiscuous mode is disabled, all incoming logical addresses except broadcast will be rejected.

The Broadcast address, which is all ones, does not go through the Logical Address Filter and is handled as follows:

- If the Disable Broadcast Bit is cleared, the broadcast address is accepted
- If the Disable Broadcast Bit is set and promiscuous mode is enabled, the broadcast address is accepted.
- If the Disable Broadcast Bit is set and promiscuous mode is disabled, the broadcast address is rejected.

If external loopback is used, the FCS logic must be allocated to the receiver (by setting the DXMTFCS bit in CSR15, and clearing the ADD_FCS bit in TMD1) when using multicast addressing.

PADR

This 48-bit value represents the unique node address assigned by the ISO 8802-3 (IEEE/ANSI 802.3) and used for internal address comparison. PADR[0] is the first address bit transmitted on the wire, and must be zero. The six hex-digit nomenclature used by the ISO 8802-3 (IEEE/ANSI 802.3) maps to the PCnet-32 controller PADR register as follows: the first byte comprises PADR[7:0], with PADR[0] being the least significant bit of the byte. The second ISO 8802-3 (IEEE/ANSI 802.3) byte maps to PADR[15:8], again from LS bit to MS bit, and so on. The sixth byte maps to PADR[47:40], the LS bit being PADR[40].

MODE

The mode register in the initialization block is copied into CSR15 and interpreted according to the description of CSR15.

Receive Descriptors

When SSIZE32 = 0 (BCR20[8]), then the software structures are defined to be 16 bits wide, and receive descriptors look as shown in Table 59.

Table 59. Receive Descriptor (SSIZE32 = 0)

	LANCE/ PCnet-ISA Descriptor Designation	PCnet-32 Descriptor Designation	
Address	**Bits 15-0**	**Bits 15-8**	**Bits 7-0**
CRDA+00	RMD0	RMD0[15:0]	
CRDA+02	RMD1	RMD1[31:24]	RMD0[23:16]
CRDA+04	RMD2	RMD1[15:0]	
CRDA+06	RMD3	RMD2[15:0]	

PCnet-32 reference names within the table above refer to the descriptor definitions given in text below. Since the text descriptions are for 32-bit descriptors, the table above shows the mapping of the 32-bit descriptors into the 16-bit descriptor space. Since 16-bit descriptors are a subset of the 32-bit descriptors, some portions of the 32-bit descriptors may not appear in Table 59.

When SSIZE32 = 1 (BCR 20[8]), then the software structures are defined to be 32 bits wide, and receive descriptors look as shown in Table 60.

Table 60. Receive Descriptor (SSIZE32 =1)

Address	LANCE/PCnet-ISA Descriptor Designation				PCnet-32 Descriptor Designation
	Bits 31-24	Bits 23-16	Bits 15-8	Bits 7-0	Bits 31-0
CRDA+00	*NA	RMD1[7:0]	RMD0[15:8]	RMD0[7:0]	RMD0
CRDA+04	RMD1[15:8]	*NA	RMD2[15:8]	RMD2[7:0]	RMD1
CRDA+08	*NA	*NA	RMD3[15:8]	RMD3[7:0]	RMD2
CRDA+0C	*NA	*NA	*NA	*NA	RMD3

**NA = These 8 bits do not exist in any LANCE descriptor.*

The Receive Descriptor Ring Entries (RDREs) are composed of four receive message descriptors (RMD0–RMD3). Together they contain the following information:

- The address of the actual message data buffer in user (host) memory.
- The length of that message buffer.
- Status information indicating the condition of the buffer. The eight most significant bits of RMD1 (RMD1[31:24]) are collectively termed the STATUS of the receive descriptor.

RMD0

Bit	Name	Description
31-0	RBADR	RECEIVE BUFFER ADDRESS. This field contains the address of the receive buffer that is associated with this descriptor.

RMD1

Bit	Name	Description
31	OWN	This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-32 controller (OWN=1). The PCnet-32 controller clears the OWN bit after filling the buffer pointed to by the descriptor entry. The host sets the OWN bit after emptying the buffer. Once the PCnet-32 controller or host has relinquished ownership of a buffer, it must not change any field in the descriptor entry.
30	ERR	ERR is the OR of FRAM, OFLO, CRC, or BUFF. ERR is set by the PCnet-32 controller and cleared by the host.
29	FRAM	FRAMING ERROR indicates that the incoming frame contained a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is set by the PCnet-32 controller and cleared by the host.
28	OFLO	OVERFLOW error indicates that the receiver has lost all or part of the incoming frame, due to an inability to store the frame in a memory buffer before the internal FIFO overflowed. OFLO is valid only when ENP is not set. OFLO is set by the PCnet-32 controller and cleared by the host.
27	CRC	CRC indicates that the receiver has detected a CRC (FCS) error on the incoming frame. CRC is valid only when ENP is set and OFLO is not. CRC is set by the PCnet-32 controller and cleared by the host.
26	BUFF	BUFFER ERROR is set any time the PCnet-32 controller does not own the next buffer while data chaining a received frame. This can occur in either of two ways: 1) The OWN bit of the next buffer is zero. 2) FIFO overflow occurred before the PCnet-32 controller received the STATUS (RMD1[31:24]) of the next descriptor. If a Buffer Error occurs, an Overflow Error may also occur internally in the FIFO, but will not be reported in the descriptor status entry unless both BUFF and OFLO errors occur at the same

		time. BUFF is set by the PCnet-32 controller and cleared by the host.
25	STP	START OF PACKET indicates that this is the first buffer used by the PCnet-32 controller for this frame. It is used for data chaining buffers. STP is set by the PCnet-32 controller and cleared by the host.
24	ENP	END OF PACKET indicates that this is the last buffer used by the PCnet-32 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits not one buffer and there is no data chaining. ENP is set by the PCnet-32 controller and cleared by the host.
23-16	RES	Reserved locations. These locations should be read and written as ZEROs.
15-12	ONES	These four bits must be written as ONES. They are written by the host and unchanged by the PCnet-32 controller.
11-0	BCNT	BUFFER BYTE COUNT is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and unchanged by the PCnet-32 controller.

RMD2

Bit	Name	Description
31-24	RCC	Receive Collision Count. Indicates the accumulated number of collisions on the network since the last frame was successfully received, excluding collisions that occurred during transmissions from this node. The PCnet-32 controller implementation of this counter may not be compatible with the ILACC RCC definition. If network statistics are to be monitored, then CSR114 should be used for the purpose of monitoring Receive collisions instead of these bits.
23-16	RPC	Runt Packet Count. Indicates the accumulated number of runts that were addressed to this node since the last time that a receive packet was successfully received and its corresponding RMD2 ring entry was written to by the PCnet-32 controller. In order to be included in the RPC value, a runt must be long enough to meet the minimum requirement of the internal address matching logic. The minimum requirement for a runt to pass the internal address matching mechanism is: 18 bits of valid preamble plus a valid SFD detected, followed by 7 bytes of frame data. This requirement is unvarying, regardless of the address matching mechanisms in force at the time of reception (i.e. physical, logical, broadcast or promiscuous). The PCnet-32 controller implementation of this counter may not be compatible with the ILACC RPC definition.
15-12	ZEROs	This field is reserved. PCnet-32 controller will write ZEROs to these locations.
11-0	MCNT	MESSAGE Byte COUNT is the length in bytes of the received message, expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the PCnet-32 controller and cleared by the host.

RMD3

Bit	Name	Description
31-0	RES	Reserved locations. All bits must be ZEROs.

Transmit Descriptors

When SSIZE32 = 0 (BCR 20[8]), then the software structures are defined to be 16 bits wide, and transmit descriptors look as shown in Table 61.

PCnet-32 reference names within the table above refer to the descriptor definitions given in text below. Since the text descriptions are for 32-bit descriptors, the table above shows the mapping of the 32-bit descriptors into the 16-bit descriptor space. Since 16-bit descriptors are a subset of the 32-bit descriptors, some portions of the 32-bit descriptors may not appear in Table 61.

When SSIZE32 = 1 (BCR 20[8]), then the software structures are defined to be 32 bits wide, and transmit descriptors look as shown in Table 62.

Table 61. Transmit Descriptor (SSIZE32 = 0)

	LANCE Descriptor Designation	PCnet-32 Descriptor Designation	
Address	**Bits 15-0**	**Bits 15-8**	**Bits 7-0**
CRDA+00	TMD0	TMD0[15:0]	
CRDA+02	TMD1	TMD1[31:24]	TMD0[23:16]
CRDA+04	TMD2	TMD1[15:0]	
CRDA+06	TMD3	TMD2[15:0]	

Table 62. Transmit Descriptor (SSIZE32 = 1)

	LANCE Descriptor Designation				PCnet-32 Descriptor Designation
Address	**Bits 31-24**	**Bits 23-16**	**Bits 15-8**	**Bits 7-0**	**Bits 31-0**
CTDA+00	*NA	TMD1[7:0]	TMD0[15:8]	TMD0[7:0]	TMD0
CTDA+04	TMD1[15:8]	*NA	TMD2[15:8]	TMD2[7:0]	TMD1
CTDA+08	TMD3[15:8]	TMD3[7:0]	*NA	*NA	TMD2
CTDA+0C	*NA	*NA	*NA	*NA	TMD3

**NA = These 8 bits do not exist in any LANCE descriptor.*

The Transmit Descriptor Ring Entries (TDREs) are composed of 4 transmit message descriptors (TMD0-TMD3). Together they contain the following information:

- The address of the actual message data buffer in user or host memory.
- The length of the message buffer.
- Status information indicating the condition of the buffer. The eight most significant bits of TMD1 (TMD1[31:24]) are collectively termed the STATUS of the transmit descriptor.

TMD0

Bit	Name	Description
31-0	TBADR	Transmit Buffer address. This field contains the address of the transmit buffer that is associated with this descriptor.

TMD1

Bit	Name	Description
31	OWN	This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-32 controller (OWN=1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The PCnet-32 controller clears the OWN bit after transmitting the contents of the buffer. Both the PCnet-32 controller and the host must not alter a descriptor entry after it has relinquished ownership.
30	ERR	ER is the OR of UFLO, LCOL, LCAR, or RTRY. ERR is set by the PCnet-32 controller and cleared by the host. This bit is set in the current descriptor when the error occurs, and therefore may be set in any descriptor of a chained buffer transmission.
29	ADD_FCS / NO_FCS	Bit 29 functions as ADD_FCS when programmed for the default I/O style of PCnet-ISA and when programmed for the I/O style PCnet-32 controller. Bit 29 functions as NO_FCS when programmed for the I/O style ILACC.
	ADD_FCS	ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. It is valid only if the STP bit is set. When ADD_FCS is set, the state of DXMTFCS is ignored and

		transmitter FCS generation is activated. When ADD_FCS = 0, FCS generation is controlled by DXMTFCS. ADD_FCS is set by the host, and unchanged by the PCnet-32 controller. This was a reserved bit in the LANCE (Am7990). *This function differs from the ILACC function for this bit.*
	NO_FCS	NO_FCS dynamically controls the generation of FCS on a frame by frame basis. It is valid only if the ENP bit is set. When NO_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is deactivated. When NO_FCS = 0, FCS generation is controlled by DXMTFCS. NO_FCS is set by the host, and unchanged by the PCnet-32 controller. This was a reserved bit in the LANCE (Am7990). *This function is identical to the ILACC function for this bit.*
28	MORE	MORE indicates that more than one re-try was needed to transmit a frame. The value of MORE is written by the PCnet-32 controller. This bit has meaning only if the ENP bit is set.
27	ONE	ONE indicates that exactly one re-try was needed to transmit a frame. ONE flag is not valid when LCOL is set. The value of the ONE bit is written by the PCnet-32 controller. This bit has meaning only if the ENP bit is set.
26	DEF	DEFERED indicates that the PCnet-32 controller had to defer while trying to transmit a frame. This condition occurs if the channel is busy when the PCnet-32 controller is ready to transmit. DEF is set by the PCnet-32 controller and cleared by the host.
25	STP	START OF PACKET indicates that this is the first buffer to be used by the PCnet-32 controller for this frame. It is used for data chaining buffers. The STP bit must be set in the first buffer of the frame, or the PCnet-32 controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set. STP is set by the host and unchanged by the PCnet-32 controller.
24	ENP	END OF PACKET indicates that this is the last buffer to be used by the PCnet-32 controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the host and unchanged by the PCnet-32 controller.
23-16	RES	Reserved locations.
15-12	ONES	Must be Ones. This field is written by the host and unchanged by the PCnet-32 controller.
11-0	BCNT	BUFFER BYTE COUNT is the usable length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the PCnet-32 controller. This field is written by the host and unchanged by the PCnet-32 controller. There are no minimum buffer size restrictions.

TMD2

Bit	Name	Description
31	BUFF	BUFFER ERROR is set by the PCnet-32 controller during transmission when the PCnet- 32 controller does not find the ENP flag in the current buffer and does not own the next buffer. This can occur in either of two ways: 1. The OWN bit of the next buffer is zero. 2. FIFO underflow occurred before the PCnet-32 controller obtained the STATUS byte (TMD1[31:24]) of the next descriptor. BUFF is set by the PCnet-32 controller and cleared by the host. BUFF error will turn off the transmitter (CSR0, TXON = 0). If a Buffer Error occurs, an Underflow Error will also occur. BUFF is not valid when LCOL or RTRY error is set during transmit data chaining. BUFF is set by the PCnet-32 controller and cleared by the host.
30	UFLO	UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indi-

		cates that the FIFO has emptied before the end of the frame was reached. Upon UFLO error, the transmitter is turned off (CSR0, TXON = 0). UFLO is set by the PCnet-32 controller and cleared by the host.
29	EXDEF	EXCESSIVE DEFERRAL. Indicates that the transmitter has experienced Excessive Deferral on this transmit frame, where Excessive Deferral is defined in ISO 8802-3 (IEEE/ANSI 802.3) .
28	LCOL	LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The PCnet-32 controller does not re-try on late collisions. LCOL is set by the PCnet-32 controller and cleared by the host.
27	LCAR	LOSS OF CARRIER is set in AUI mode when the carrier is lost during an PCnet-32 controller-initiated transmission. The PCnet-32 controller does not re-try upon loss of carrier. It will continue to transmit the whole frame until done. In 10BASE-T mode LCAR will be set when the T-MAU is in Link Fail state. LCAR is not valid in Internal Loopback Mode. LCAR is set by the PCnet-32 controller and cleared by the host.
26	RTRY	RETRY ERROR indicates that the transmitter has failed after 16 attempts to successfully transmit a message, due to repeated collisions on the medium. If DRTY = 1 in the MODE register, RTRY will set after 1 failed transmission attempt. RTRY is set by the PCnet-32 controller and cleared by the host.
25-16	TDR	TIME DOMAIN REFLECTOMETRY reflects the state of an internal PCnet-32 controller counter that counts at a 10 MHz rate from the start of a transmission to the occurrence of a collision or loss of carrier. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the PCnet-32 controller and is valid only if RTRY is set. Note that 10 MHz gives very low resolution and in general has not been found to be particularly useful. This feature is here primarily to maintain full compatibility with the LANCE (Am7990).
15-4	RES	Reserved locations.
3-0	TRC	TRANSMIT RETRY COUNT. Indicates the number of transmit retries of the associated packet. The maximum count is 15. However, if a RETRY error occurs, the count will roll over to zero. In this case only, the Transmit Retry Count value of zero should be interpreted as meaning 16. TRC is written by the PCnet-32 controller into the last transmit descriptor of a frame, or when an error terminates a frame. Valid only when OWN = 0.

TMD3

Bit	Name	Description
31-0	RES	Reserved locations. All bits must be ZEROs.

Register Summary

CSRs — Control and Status Registers

RAP Addr	Symbol	Default Value After H_RESET (Hex)	Comments	Use
00	CSR0	xxxx 0004	PCnet-32 Controller Status	R
01	CSR1	xxxx xxxx	Lower IADR: maps to location 16	S
02	CSR2	xxxx xxxx	Upper IADR: maps to location 17	S
03	CSR3	xxxx 0000	Interrupt Masks and Deferral Control	S
04	CSR4	xxxx 0115	Test and Features Control	R
05	CSR5	xxxx 0000	Reserved	T
06	CSR6	xxxx xxxx	RXTX: RX/TX Descriptor Table Length	T
07	CSR7	xxxx 0000	Reserved	T
08	CSR8	xxxx xxxx	LADR0: Logical Address Filter — LADRF[15:0]	T
09	CSR9	xxxx xxxx	LADR1: Logical Address Filter — LADRF[31:16]	T
10	CSR10	xxxx xxxx	LADR2: Logical Address Filter — LADRF[47:32]	T
11	CSR11	xxxx xxxx	LADR3: Logical Address Filter — LADRF[63:48]	T
12	CSR12	xxxx xxxx	PADR0: Physical Address Register — PADR[15:0][	T
13	CSR13	xxxx xxxx	PADR1: Physical Address Register — PADR[31:16]	T
14	CSR14	xxxx xxxx	PADR2: Physical Address Register — PADR[47:32]	T
15	CSR15	See Register Desc.	MODE: Mode Register	S
16	CSR16	xxxx xxxx	IADR: Base Address of INIT Block Lower (Copy)	T
17	CSR17	xxxx xxxx	IADR: Base Address of INIT Block Upper (Copy)	T
18	CSR18	xxxx xxxx	CRBA: Current RCV Buffer Address Lower	T
19	CSR22	xxxx xxxx	CRBA: Current RCV Buffer Address Upper	T
20	CSR20	xxxx xxxx	CXBA: Current XMT Buffer Address Lower	T
21	CSR21	xxxx xxxx	CXBA: Current XMT Buffer Address Upper	T
22	CSR22	xxxx xxxx	NRBA: Next RCV Buffer Address Lower	T
23	CSR23	xxxx xxxx	NRBA: Next RCV Buffer Address Upper	T
24	CSR24	xxxx xxxx	BADR: Base Address of RCV Ring Lower	S
25	CSR25	xxxx xxxx	BADR: Base Address of RCV Ring Upper	S
26	CSR26	xxxx xxxx	NRDA: Next RCV Descriptor Address Lower	T
27	CSR27	xxxx xxxx	NRDA: Next RCV Descriptor Address Upper	T
28	CSR28	xxxx xxxx	CRDA: Current RCV Descriptor Address Lower	T
29	CSR29	xxxx xxxx	CRDA: Current RCV Descriptor Address Upper	T
30	CSR30	xxxx xxxx	BADX: Base Address of XMT Ring Lower	S
31	CSR31	xxxx xxxx	BADX: Base Address of XMT Ring Upper	S
32	CSR32	xxxx xxxx	NXDA: Next XMT Descriptor Address Lower	T
33	CSR33	xxxx xxxx	NXDA: Next XMT Descriptor Address Upper	T
34	CSR34	xxxx xxxx	CXDA: Current XMT Descriptor Address Lower	T
35	CSR35	xxxx xxxx	CXDA: Current XMT Descriptor Address Upper	T
36	CSR36	xxxx xxxx	NNRDA: Next Next Receive Descriptor Address Lower	T

Key: ***x*** *= undefined* ***R*** *= Running Register* ***S*** *= Setup Register* ***T*** *= Test Register*

Register Summary (continued)

CSRs — Control and Status Registers

RAP Addr	Symbol	Default Value After H_RESET (Hex)	Comments	Use
37	CSR37	xxxx xxxx	NNRDA: Next Next Receive Descriptor Address Upper	T
38	CSR38	xxxx xxxx	NNXDA: Next Next Transmit Descriptor Address Lower	T
39	CSR39	xxxx xxxx	NNXDA: Next Next Transmit Descriptor Address Upper	T
40	CSR40	xxxx xxxx	CRBC: Current RCV Status and Byte Count Lower	T
41	CSR41	xxxx xxxx	CRBC: Current RCV Status and Byte Count Upper	T
42	CSR42	xxxx xxxx	CXBC: Current XMT Status and Byte Count Lower	T
43	CSR43	xxxx xxxx	CXBC: Current XMT Status and Byte Count Upper	T
44	CSR44	xxxx xxxx	NRBC: Next RCV Status and Byte Count Lower	T
45	CSR45	xxxx xxxx	NRBC: Next RCV Status and Byte Count Upper	T
46	CSR46	xxxx xxxx	POLL: Poll Time Counter	T
47	CSR47	xxxx xxxx	PI: Polling Interval	S
48	CSR48	xxxx xxxx	TMP2: Temporary Storage Lower	T
49	CSR48	xxxx xxxx	TMP2: Temporary Storage Upper	T
50	CSR50	xxxx xxxx	TMP3: Temporary Storage Lower	T
51	CSR50	xxxx xxxx	TMP3: Temporary Storage Upper	T
52	CSR52	xxxx xxxx	TMP4: Temporary Storage Lower	T
53	CSR52	xxxx xxxx	TMP4: Temporary Storage Upper	T
54	CSR54	xxxx xxxx	TMP5: Temporary Storage Lower	T
55	CSR54	xxxx xxxx	TMP5: Temporary Storage Upper	T
56	CSR56	xxxx xxxx	TMP6: Temporary Storage Lower	T
57	CSR56	xxxx xxxx	TMP6: Temporary Storage Upper	T
58	CSR58	See Register Desc.	SWS: Software Style	S
59	CSR59	xxxx 0105	IR: IR Register	T
60	CSR60	xxxx xxxx	PXDA: Previous XMT Descriptor Address Lower	T
61	CSR61	xxxx xxxx	PXDA: Previous XMT Descriptor Address Upper	T
62	CSR62	xxxx xxxx	PXBC: Previous XMT Status and Byte Count Lower	T
63	CSR63	xxxx xxxx	PXBC: Previous XMT Status and Byte Count Upper	T
64	CSR64	xxxx xxxx	NXBA: Next XMT Buffer Address Lower	T
65	CSR65	xxxx xxxx	NXBA: Next XMT Buffer Address Upper	T
66	CSR66	xxxx xxxx	NXBC: Next XMT Status and Byte Count Lower	T
67	CSR67	xxxx xxxx	NXBC: Next XMT Status and Byte Count Upper	T
68	CSR68	xxxx xxxx	XSTMP: XMT Status Temporary Storage Lower	T
69	CSR69	xxxx xxxx	XSTMP: XMT Status Temporary Storage Upper	T
70	CSR70	xxxx xxxx	RSTMP: RCV Status Temporary Storage Lower	T
71	CSR71	xxxx xxxx	RSTMP: RCV Status Temporary Storage Upper	T
72	CSR72	xxxx xxxx	RCVRC: RCV Ring Counter	T

Key: ***x*** *= undefined* ***R*** *= Running Register* ***S*** *= Setup Register* ***T*** *= Test Register*

Register Summary (continued)

CSRs — Control and Status Registers

RAP Addr	Symbol	Default Value After H_RESET (Hex)	Comments	Use
73	CSR73	xxxx xxxx	Reserved	T
74	CSR74	xxxx xxxx	XMTRC: XMT Ring Counter	T
75	CSR75	xxxx xxxx	Reserved	T
76	CSR76	xxxx xxxx	RCVRL: RCV Ring Length	S
77	CSR77	xxxx xxxx	Reserved	T
78	CSR78	xxxx xxxx	XMTRL: XMT Ring Length	S
79	CSR79	xxxx xxxx	Reserved	T
80	CSR80	xxxx E810	Burst and FIFO Threshold Control	S
81	CSR81	xxxx xxxx	Reserved	T
82	CSR82	xxxx 0000	DMABAT: Bus Acitivty Timer	S
83	CSR83	xxxx xxxx	Reserved	T
84	CSR84	xxxx xxxx	DMABA: DMA Address Lower	T
85	CSR85	xxxx xxxx	DMABA: DMA Address Upper	T
86	CSR86	xxxx xxxx	DMABC: Buffer Byte Counter	T
87	CSR87	xxxx xxxx	Reserved	T
88	CSR88	0243 0003	Chip ID Lower	T
89	CSR89	xxxx 0243	Chip ID Upper	T
90	CSR90	xxxx xxxx	Reserved	T
91	CSR91	xxxx xxxx	Reserved	T
92	CSR92	xxxx xxxx	RCON: Ring Length Conversion	T
93	CSR93	xxxx xxxx	Reserved	T
94	CSR94	xxxx 0000	XMTTDR: Transmit Time Domain Reflectometry Count	T
95	CSR95	xxxx xxxx	Reserved	T
96	CSR96	xxxx xxxx	SCR0: BIU Scratch Register 0 Lower	T
97	CSR97	xxxx xxxx	SCR0: BIU Scratch Register 0 Upper	T
98	CSR98	xxxx xxxx	SCR1: BIU Scratch Register 1 Lower	T
99	CSR99	xxxx xxxx	SCR1: BIU Scratch Register 1 Upper	T
100	CSR100	xxxx 0200	Bus Time-Out	S
101	CSR101	xxxx xxxx	Reserved	T
102	CSR102	xxxx xxxx	Reserved	T
103	CSR103	xxxx 0105	Reserved	T
104	CSR104	xxxx xxxx	SWAP: Swap Register Lower	T
105	CSR105	xxxx xxxx	SWAP: Swap Register Upper	T
106	CSR106	xxxx xxxx	Reserved	T
107	CSR107	xxxx xxxx	Reserved	T
108	CSR108	xxxx xxxx	BMSCR: BMU Scratch Register Lower	T
109	CSR109	xxxx xxxx	BMSCR: BMU Scratch Register Upper	T

Key: ***x*** *= undefined* ***R*** *= Running Register* ***S*** *= Setup Register* ***T*** *= Test Register*

Register Summary (continued)

CSRs — Control and Status Registers

RAP Addr	Symbol	Default Value After H_RESET (Hex)	Comments	Use
110	CSR110	xxxx xxxx	Reserved	T
111	CSR111	xxxx xxxx	Reserved	T
112	CSR112	xxxx 0000	Missed Frame Count	R
113	CSR113	xxxx xxxx	Reserved	T
114	CSR114	xxxx 0000	Receive Collision Count	R
115	CSR115	xxxx xxxx	Reserved	T
116	CSR116	xxxx xxxx	Reserved	T
117	CSR117	xxxx xxxx	Reserved	T
118	CSR118	xxxx xxxx	Reserved	T
119	CSR119	xxxx xxxx	Reserved	T
120	CSR120	xxxx xxxx	Reserved	T
121	CSR121	xxxx xxxx	Reserved	T
122	CSR122	See Register Desc.	Receive Frame Alignment Control	S
123	CSR123	xxxx xxxx	Reserved	T
124	CSR124	See Register Desc.	BMU Test	T
125	CSR125	xxxx xxxx	Reserved	T
126	CSR126	xxxx xxxx	Reserved	T
127	CSR127	xxxx xxxx	Reserved	T

Key: ***x*** *= undefined* ***R*** *= Running Register* ***S*** *= Setup Register* ***T*** *= Test Register*

Register Summary (continued)

BCR — Bus Configuration Registers

RAP Addr.	Mnemonic	Default (Hex)	Name	Programmability		
				User	EEPROM	SRM
0	MSRDA	0005	Master Mode Read Active	No	No	No
1	MSWRA	0005	Master Mode Write Active	No	No	No
2	MC	N/A*	Miscellaneous Configuration	Yes	Yes	Yes
3	Reserved	N/A		No	No	No
4	$\overline{\text{LNKST}}$	00C0	Link Status (Default)	Yes	No	No
5	LED1	0084	Receive (Default)	Yes	No	No
6	LED2	0088	Receive Polarity (Default)	Yes	No	No
7	LED3	0090	Transmit (Default)	Yes	No	No
8–15	Reserved	N/A		No	No	No
16	IOBASEL	N/A*	I/O Base Address Lower	Yes	Yes	Yes
17	IOBASEU	N/A*	I/O Base Address Upper	Yes	Yes	Yes
18	BSBC	2101	Burst Size and Bus Control	Yes	Yes	No
19	EECAS	0002	EEPROM Control and Status	Yes	No	No
20	SWSTYLE	0000	Software Style	Yes	No	No
21	INTCON	N/A*	Interrupt Control	Yes	Yes	Yes

Key: ***SRM*** = *Software Relocatable Mode*

* *Registers marked with an asterisk (*) have no default value, since they are not observable without first being programmed, either through the EEPROM read operation or through the Software Relocatable Mode. Therefore, the only observable values for these registers are those that have been programmed and a default value is not applicable.*

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C

Ambient Temperature
Under Bias ..... −65°C to +125°C

Supply Voltage to AVss
or DVss (AV_{DD}, DV_{DD}) ..... −0.3 V to +6.0 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A) ..... 0°C to +70°C

Supply Voltages
(AV_{DD}, DV_{DD}) ..... 5 V ±5%

All inputs within the range: ..... $AV_{SS} - 0.5\ V \le V_{in} \le AV_{DD} + 0.5\ V$, or $DV_{SS} - 0.5\ V \le V_{in} \le DV_{DD} + 0.5\ V$

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions		Min	Max	Unit
Digital Input Voltage						
V_{IL}	Input LOW Voltage				0.8	V
V_{IH}	Input HIGH Voltage			2.0		V
Digital Ouput Voltage						
V_{OL}	Output LOW Voltage	I_{OL1} = 8 mA, I_{OL2} = 4 mA (Notes 1 and 5)			0.45	V
V_{OH}	Output HIGH Voltage (Note 2)	I_{OH} = −4 mA (Note 5)		2.4		V
Digital Input Leakage Current						
I_{IX}	Input Leakage Current (Note 3)	V_{DD} = 5 V, V_{IN} = 0 V		−10	10	µA
Digital Ouput Leakage Current						
I_{OZL}	Output Low Leakage Current (Note 4)	V_{OUT} = 0 V		−10		µA
I_{OZH}	Output High Leakage Current (Note 4)	V_{OUT} = V_{DD}			10	µA
Crystal Input						
V_{ILX}	XTAL1 Input LOW Threshold Voltage	V_{IN} = External Clock		−0.5	0.8	V
V_{ILHX}	XTAL1 Input HIGH Threshold Voltage	V_{IN} = External Clock		V_{DD}−0.8	V_{DD} + 0.5	V
I_{ILX}	XTAL1 Input LOW Current	VIN = 0 V	Active	−120	0	µA
			Sleep	−10	+10	µA
I_{IHX}	XTAL1 Input HIGH Current	VIN = VDD	Active	0	120	µA
			Sleep		400	µA
Attachment Unit Interface						
I_{IAXD}	Input Current at DI+ and DI−	−1 V < VIN < AV_{DD} +0.5 V		−500	+500	µA
I_{IAXC}	Input Current at CI+ and CI−	−1 V < VIN < AV_{DD} +0.5 V		−500	+500	µA
V_{AOD}	Differential Output Voltage |(DO+)−(DO−)|	R_L = 78 Ω		630	1200	mV
V_{AODOFF}	Transmit Differential Output Idle Voltage	R_L = 78 Ω (Note 9)		−40	+40	mV

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Attachment Unit Interface (Continued)					
I_{AODOFF}	Transmit Differential Output Idle Current	R_L = 78 Ω (Note 8)	–1	+1	mA
V_{CMT}	Transmit Output Common Mode Voltage	R_L = 78 Ω	2.5	AV_{DD}	V
V_{ODI}	DO± Transmit Differential Output Voltage Imbalance	R_L = 78 Ω (Note 7)		25	mV
V_{ATH}	Receive Data Differential Input Threshold		–35	35	mV
V_{ASQ}	DI± and CI± Differential Input Threshold (Squelch)		–275	–160	mV
V_{IRDVD}	DI± and CI± Differential Mode Input Voltage Range		–1.5	+1.5	V
V_{ICM}	DI± and CI± Input Bias Voltage	I_{IN} = 0 mA	AV_{DD}–3.0	AV_{DD}–1.0	V
V_{OPD}	DO± Undershoot Voltage at Zero Differential on Transmit Return to Zero (ETD)	(Note 9)		–100	mV
Twisted Pair Interface					
I_{IRXD}	Input Current at RXD±	$AV_{SS} < V_{IN} < AV_{DD}$	–500	500	µA
R_{RXD}	RXD± Differential Input Resistance		10		KΩ
V_{TIVB}	RXD+, RXD– Open Circuit Input Voltage (Bias)	I_{IN} = 0 mA	AV_{DD} – 3.0	AV_{DD} – 1.5	V
V_{TIDV}	Differential Mode Input Voltage Range (RXD±)	AV_{DD} = 5 V	–3.1	+3.1	V
V_{TSQ+}	RXD Positive Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	300	520	mV
V_{TSQ-}	RXD Negative Squelch Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	–520	–300	mV
V_{THS+}	RXD Post-Squelch Positive Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	150	293	mV
V_{THS-}	RXD Post-Squelch Negative Threshold (Peak)	Sinusoid 5 MHz ≤ f ≤10 MHz	–293	–150	mV
V_{LTSQ+}	RXD Positive Squelch Threshold (Peak)	$\overline{LRT}$ = LOW	180	312	mV
V_{LTSQ-}	RXD Negative Squelch Threshold (Peak)	$\overline{LRT}$ = LOW	–312	–180	mV
V_{LTHS+}	RXD Post-Squelch Positive Threshold (Peak)	$\overline{LRT}$ = LOW	90	176	mV
V_{LTHS-}	RXD Post-Squelch Negative Threshold (Peak)	$\overline{LRT}$ = LOW	–176	–90	mV

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Twisted Pair Interface (continued)					
V_{RXDTH}	RXD Switching Threshold	(Note 4)	–35	35	mV
V_{TXH}	TXD± and TXP± Output HIGH Voltage	DV_{SS} = 0 V	DV_{DD} – 0.6	DV_{DD}	V
V_{TXL}	TXD± and TXP± Output LOW Voltage	DV_{DD} = +5 V	DV_{SS}	DV_{SS} + 0.6	V
V_{TXI}	TXD± and TXP± Differential Output Voltage Imbalance		–40	+40	mV
V_{TXOFF}	TXD± and TXP± Idle Output Voltage			40	mV
R_{TX}	TXD± Differential Driver Output Impedance	(Note 4)		40	Ω
	TXP± Differential Driver Output Impedance	(Note 4)		80	Ω
IEEE 1149.1 (JTAG) Test Port					
V_{IL}	TCK, TMS, TDI			0.8	V
V_{IH}	TCK, TMS, TDI		2.0		V
V_{OL}	TDO	I_{OL} = 2.0 mA		0.4	V
V_{OH}	TDO	I_{OH} = –0.4 mA	2.4		V
I_{IL}	TCK, TMS, TDI	V_{DD} = 5.5 V, V_I = 0.5 V		–200	µA
I_{IH}	TCK, TMS, TDI	V_{DD} =5.5 V, V_I = 2.7V		–100	µA
I_{OZ}	TDO	V_{OUT} = 0 V, V_{OUT} = V_{DD}	–10	+10	µA
Power Supply Current					
I_{DD}	Active Power Supply Current	XTAL1 = 20 MHz		100	mA
I_{DDCOMA}	Coma Mode Power Supply Current	$\overline{SLEEP}$ active		200	µA
$I_{DDSNOOZE}$	Snooze Mode Power Supply Current	AWAKE bit set active		10	mA
Pin Capacitance					
C_{IN}	Input Pin Capacitance	F_C = 1 MHz (Note 6)		10	pF
C_O	I/O or Output Pin Capacitance	F_C = 1 MHz (Note 6)		10	pF
C_{CLK}	BCLK Pin Capacitance	F_C = 1 MHz (Note 6)		10	pF

Notes:

1. *I_{OL1} = 8 mA applies to all output and I/O pins except HOLD and $\overline{LDEV}$. I_{OL2} = 4 mA applies to HOLD and $\overline{LDEV}$ only.*
2. *V_{OH} does not apply to open-drain output pins.*
3. *I_{IX} applies to all input only pins except D±, C±, and XTAL1.*
4. *I_{OZL} and I_{OZH} apply to all three-state output pins and bi-directional pins.*
5. *Outputs are CMOS and will be driven to rail if the load is not resistive.*
6. *Parameter not tested; value determined by characterization.*
7. *Tested, but to values in excess of limits. Test accuracy not sufficient to allow screening guard bands.*
8. *Correlated to other tested parameters – not tested directly.*
9. *Test not implemented to data sheet specification.*

SWITCHING CHARACTERISTICS: BUS INTERFACE

Parameter Symbol	Parameter Description	Test Conditions (C_L = 50 pF Unless Otherwise Noted)	Min	Max	Unit
Clock Timing					
	CLK Frequency		1	33	MHz
t_1	CLK Period		30	1000	ns
t_2	CLK High time	@ 2.0 V	14		ns
t_3	CLK Low Time	@ 0.8 V	14		ns
t_4	CLK Fall Time	2.0 V to 0.8 V (Note 1)		3	ns
t_5	CLK Rise Time	0.8 V to 2.0 V (Note 1)		3	ns
Output and Float Delay Timing					
t_6	A2–A31, $\overline{BE0}$-$\overline{BE3}$, M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$, $\overline{ADS}$, HOLD, HLDAO, INTR, $\overline{EADS}$, $\overline{RDY}$ Valid Delay		3	12	ns
t_7	A2–A31, $\overline{BE0}$-$\overline{BE3}$, M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$, $\overline{ADS}$, $\overline{RDY}$ Float Delay			18	ns
t_8					ns
t_{8a}	$\overline{BLAST}$ Valid Delay		3	12	ns
t_9	$\overline{BLAST}$ Float Delay			18	ns
t_{10}	D0–D31 Data Valid Delay		3	12	ns
t_{11}	D0–D31 Data Float Delay			18	ns
t_{12}	(This symbol is not used)				
t_{13}	(This symbol is not used)				
Setup and Hold Timing					
t_{14}	$\overline{LBS16}$ Setup time		5		ns
t_{15}	$\overline{LBS16}$ Hold Time		2		ns
t_{16}	$\overline{BRDY}$, $\overline{RDYRTN}$ Setup Time		5		ns
t_{17}	$\overline{BRDY}$, $\overline{RDYRTN}$ Hold Time		2		ns
t_{18}	HOLDI, AHOLD, HLDA, $\overline{SLEEP}$ Setup Time		6		ns
t_{18a}	$\overline{BOFF}$ Setup Time		8		ns
t_{19}	HOLDI, $\overline{BOFF}$, AHOLD, HLDA, $\overline{SLEEP}$ Hold Time		2		ns
t_{20}	RESET, CLKRESET Setup Time		5		ns
t_{21}	RESET, CLKRESET Hold Time		2		ns
t_{22}	D0–D31, A2–A31, $\overline{BE0}$-$\overline{BE3}$, $\overline{ADS}$, M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$ Setup Time		4		ns
t_{23}	D0–D31, A2–A31, $\overline{BE0}$-$\overline{BE3}$, $\overline{ADS}$, M/$\overline{IO}$, D/$\overline{C}$, W/$\overline{R}$ Hold Time		2		ns

SWITCHING CHARACTERISTICS: BUS INTERFACE (continued)

Parameter Symbol	Parameter Description	Test Conditions (C_L = 50 pF Unless Otherwise Noted)	Min	Max	Unit
JTAG (IEEE 1149.1) Test Signal Timing					
t_{24}	TCK Frequency			10	MHz
t_{25}	TCK Period		100		ns
t_{26}	TCK High Time	@ 2.0 V	45		ns
t_{27}	TCK Low Time	@ 0.8 V	45		ns
t_{28}	TCK Rise Time			4	ns
t_{29}	TCK Fall Time			4	ns
t_{30}	TDI, TMS Setup Time		16		ns
t_{31}	TDI, TMS Hold Time		10		ns
t_{32}	TDO Valid Delay		3	60	ns
t_{33}	TDO Float Delay			50	ns
t_{34}	All Outputs (Non-Test) Valid Delay		3	25	ns
t_{35}	All Outputs (Non-Test) Float Delay			36	ns
t_{36}	All Inputs (Non-Test)) Setup Time		8		ns
t_{37}	All Inputs (Non-Test) Hold Time		7		ns
$\overline{\text{LDEV}}$ Timing					
t_{38}	$\overline{\text{ADS}}$ Asserted to $\overline{\text{LDEV}}$ Asserted			20	ns
t_{39}	$\overline{\text{LDEV}}$ Valid Pulse Width		t_{BCLK}		
Data Bus Activation Timing (Slave)					
t_{40}	Data Bus Driven After $\overline{\text{ADS}}$ Sampled Asserted		1		BCLK
HOLD Inactive Timing					
t_{41}	HOLD Deasserted to HOLD Asserted		2 BCLK –15 ns		
EEPROM Timing					
t_{42}	EESK Frequency	(Note 2)		650	kHz
t_{43}	EESK HIGH Time	(Note 2)	780		ns
t_{44}	EESK LOW Time	(Note 2)	780		ns
t_{45}	EECS, EEDI, SHFBUSY Valid Output Delay from EESK	(Note 2)	–15	+15	ns
t_{46}	EECS LOW Time	(Note 2)	1550		ns
t_{47}	EEDO Setup Time to EESK		50		ns
t_{48}	EEDO Hold Time to EESK		0		ns

Notes:

1. *Not 100% tested; guaranteed by design characterization.*
2. *Parameter value is given for automatic EEPROM read operation. When EEPROM port (BCR19) is used to access the EEPROM, software is responsible for meeting EEPROM timing requirements.*

SWITCHING CHARACTERISTICS: 10BASE-T INTERFACE

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Transmit Timing					
t_{TETD}	Transmit Start of Idle		250	350	ns
t_{TR}	Transmitter Rise Time	(10% to 90%)		5.5	ns
t_{TF}	Transmitter Fall Time	(90% to 10%)		5.5	ns
t_{TM}	Transmitter Rise and Fall Time Mismatch			1	ns
t_{PERLP}	Idle Signal Period		8	24	ms
t_{PWLP}	Idle Link Pulse Width	(Note 1)	75	120	ns
t_{PWPLP}	Predistortion Idle Link Pulse Width	(Note 1)	45	55	ns
t_{JA}	Transmit Jabber Activation Time		20	150	ms
t_{JR}	Transmit Jabber Reset Time		250	750	ms
t_{JREC}	Transmit Jabber Recovery Time (minimum time gap between transmitted frames to prevent jabber activation)		1.0		µs
Receive Timing					
t_{PWNRD}	RXD Pulse Width Not to Turn Off Internal Carrier Sense	VIN > VTHS (min)	136	–	ns
t_{PWROFF}	RXD Pulse Width to Turn Off	VIN > VTHS (min)		200	ns
t_{RETD}	Receive Start of Idle		200		ns
t_{RCVON}	$\overline{RCV}$ Asserted Delay		T_{RON} - 50	T_{RON} + 100	ns
t_{RCVOFF}	$\overline{RCV}$ De-Asserted Delay		20	62	ms
Collision Detection and SQE Test					
t_{COLON}	$\overline{COL}$ Asserted Delay		750	900	ns
t_{COLOFF}	$\overline{COL}$ De-Asserted Delay		20	62	ms

Note:

1. Not tested; parameter guaranteed by characterization.

SWITCHING CHARACTERISTICS: AUI

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
AUI Port					
t_{DOTR}	DO+,DO- rise time (10% to 90%)		2.5	5.0	ns
t_{DOTF}	DO+,DO- fall time (90% to 10%)		2.5	5.0	ns
t_{DORM}	DO+,DO- rise and fall time mismatch		–	1.0	ns
t_{DOETD}	DO+/- End of Transmission		200	375	ns
t_{PWODI}	DI pulse width accept/reject threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 1)	15	45	ns
t_{PWKDI}	DI pulse width maintain/turn-off threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 2)	136	200	ns
t_{PWOCI}	CI pulse width accept/reject threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 3)	10	26	ns
t_{PWKCI}	CI pulse width maintain/turn-off threshold	$\|V_{IN}\| > \|V_{ASQ}\|$ (Note 4)	90	160	ns
Internal MENDEC Clock Timing					
t_{X1}	XTAL1 period	V_{IN} = External Clock	49.995	50.005	ns
t_{X1H}	XTAL1 HIGH pulse width	V_{IN} = External Clock	20		ns
t_{X1L}	XTAL1 LOW pulse width	V_{IN} = External Clock	20		ns
t_{X1R}	XTAL1 rise time	V_{IN} = External Clock		5	ns
t_{X1F}	XTAL1 fall time	V_{IN} = External Clock		5	ns

Notes:

1. *DI pulses narrower than t_{PWODI} (min) will be rejected; pulses wider than t_{PWODI} (max) will turn internal DI carrier sense on.*
2. *DI pulses narrower than t_{PWKDI} (min) will maintain internal DI carrier sense on; pulses wider than t_{PWKDI} (max) will turn internal DI carrier sense off.*
3. *CI pulses narrower than t_{PWOCI} (min) will be rejected; pulses wider than t_{PWOCI} (max) will turn internal CI carrier sense on.*
4. *CI pulses narrower than t_{PWKCI} (min) will maintain internal CI carrier sense on; pulses wider than t_{PWKCI} (max) will turn internal CI carrier sense off.*

SWITCHING CHARACTERISTICS: GPSI

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Transmit Timing					
t_{GPT1}	STDCLK period (802.3 Compliant)		99.99	100.01	ns
t_{GPT2}	STDCLK High Time		40	60	ns
t_{GPT3}	TXDAT and TXEN delay from ↑ STDCLK		0	70	ns
t_{GPT4}	RXCRS setup before ↑ STDCLK (Last Bit)		210		ns
t_{GPT5}	RXCRS hold after ↓ TXEN		0		ns
t_{GPT6}	CLSN active time to trigger collision	(Note 1)	110		ns
t_{GPT7}	CLSN active to ↓ RXCRS to prevent LCAR assertion		0		ns
t_{GPT8}	CLSN active to ↓ RXCRS for SQE Hearbeat Window		0	4.0	µs
t_{GPT9}	CLSN active to ↑ RXCRS for normal collision		0	51.2	µs
Receive Timing					
t_{GPR1}	SRDCLK period	(Note 2)	80	120	ns
t_{GPR2}	SRDCLK High Time	(Note 2)	30	80	ns
t_{GPR3}	SRDCLK Low Time	(Note 2)	30	80	ns
t_{GPR4}	RXDAT and RXCRS setup to ↑ SRDCLK		15		ns
t_{GPR5}	RXDAT hold after ↑ SRDCLK		15		ns
t_{GPR6}	RXCRS hold after ↓ SRDCLK		0		ns
t_{GPR7}	CLSN active to first ↑ SRDCLK (Collision Recognition)		0		ns
t_{GPR8}	CLSN active to ↑ SRDCLK for Address Type Designation Bit	(Note 3)	51.2		µs
t_{GPR9}	CLSN setup to last ↑ SRDCLK for Collision Recognition		210		ns
t_{GPR10}	CLSN active		110		ns
t_{GPR11}	CLSN inactive setup to first ↑ SRDCLK		300		ns
t_{GPR12}	CLSN inactive hold to last ↑ SRDCLK		300		ns

Notes:

1. *CLSN must be asserted for a continuous period of 110 nsec or more. Assertion for less than 110 nsec period may or may not result in CLSN recognition.*
2. *RXCRS should meet jitter requirements of IEEE 802.3 specification.*
3. *CLSN assertion before 51.2 µsec will be indicated as a normal collision. CLSN assertion after 51.2 µsec will be considered as a Late Receive Collision.*

SWITCHING CHARACTERISTICS: EADI

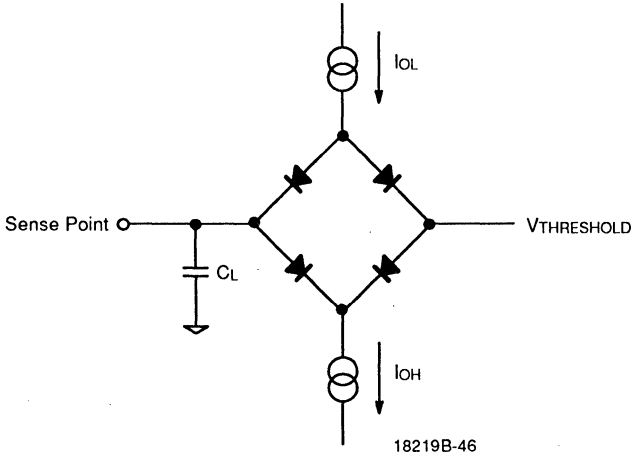
Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
t_{EAD1}	SRD setup to ↑ SRDCLK		40		ns
t_{EAD2}	SRD hold to ↑ SRDCLK		40		ns
t_{EAD3}	$\overline{SF/BD}$ change to ↓ SRDCLK		−15	+15	ns
t_{EAD4}	$\overline{EAR}$ deassertion to ↑ SRDCLK (first rising edge)		50		ns
t_{EAD5}	$\overline{EAR}$ assertion after SFD event (packet rejection)		0	51,090	ns
t_{EAD6}	$\overline{EAR}$ assertion		110		ns

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	Must Be Steady	Will Be Steady
	May Change from H to L	Will Be Changing from H to L
	May Change from L to H	Will Be Changing from L to H
	Don't Care Any Change Permitted	Changing State Unknown
	Does Not Apply	Center Line is High Impedance "Off" State

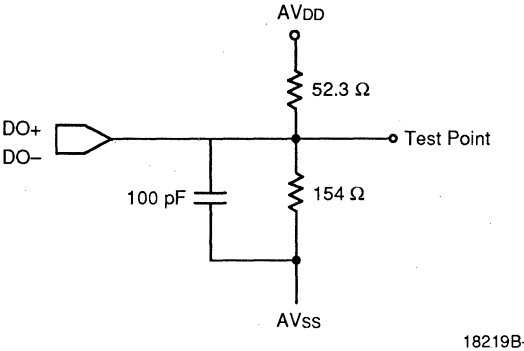
KS000010

SWITCHING TEST CIRCUITS

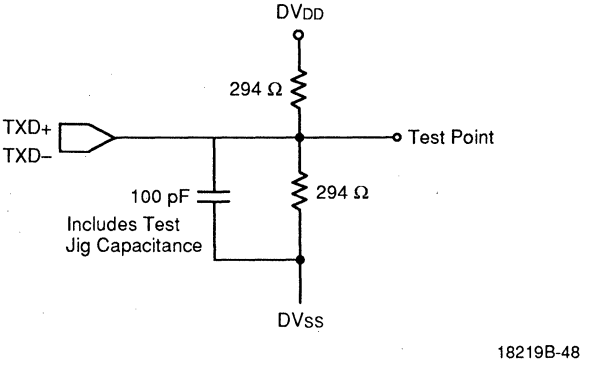


Normal and Three-State Outputs

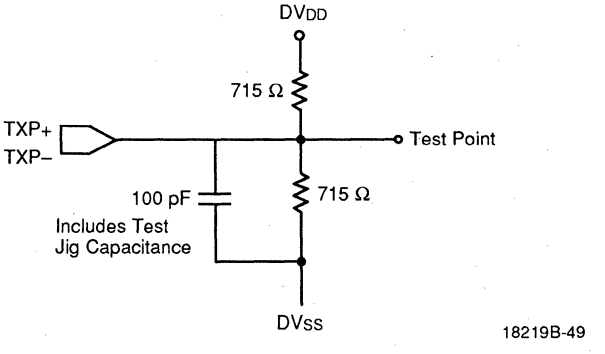
SWITCHING TEST CIRCUITS



AUI DO Switching Test Circuit



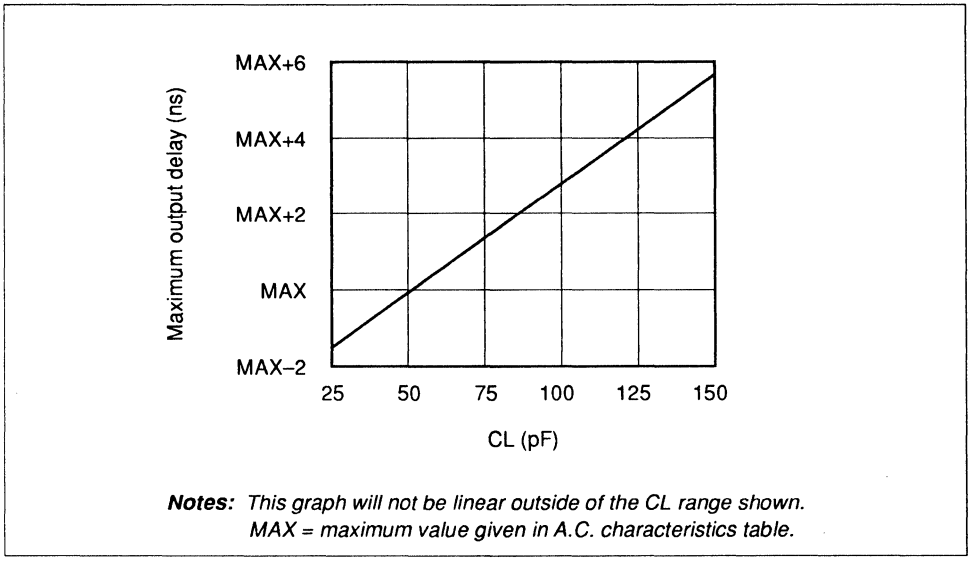
TXD Switching Test Circuit



TXP Outputs Test Circuit

ESTIMATED OUTPUT VALID DELAY VS. LOAD CAPACITANCE

Note that this graph represents change in output delay for estimated conditions.

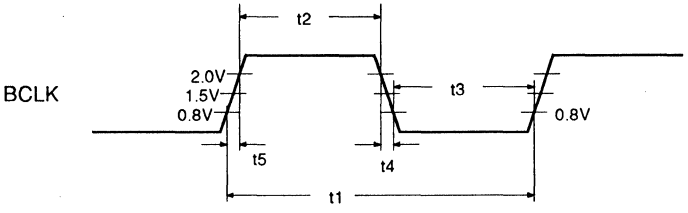


Notes: *This graph will not be linear outside of the CL range shown.*
MAX = maximum value given in A.C. characteristics table.

18219B-50

Output Delay Versus Load Capacitance Under Estimated Conditions

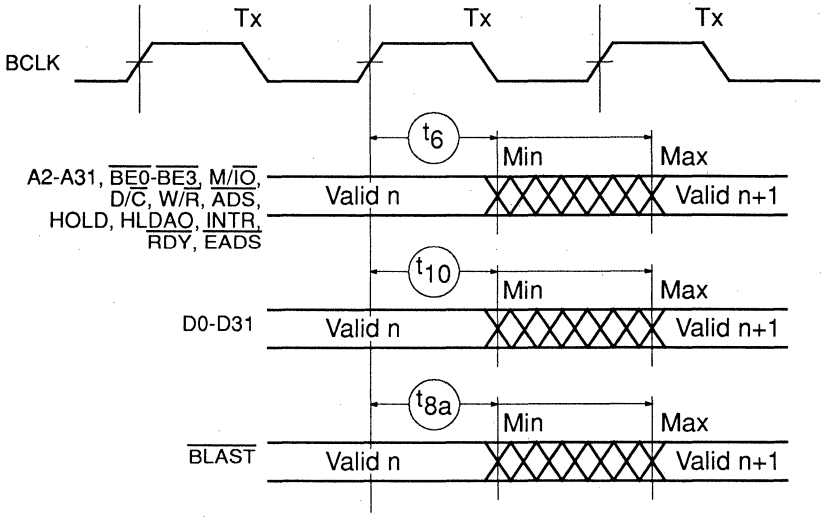
SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE



18219B-51

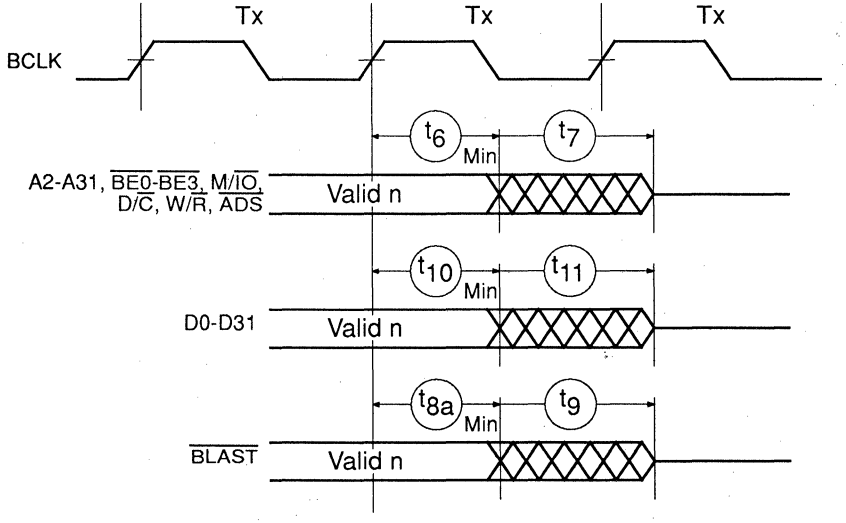
BCLK Waveform

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE



18219B-52

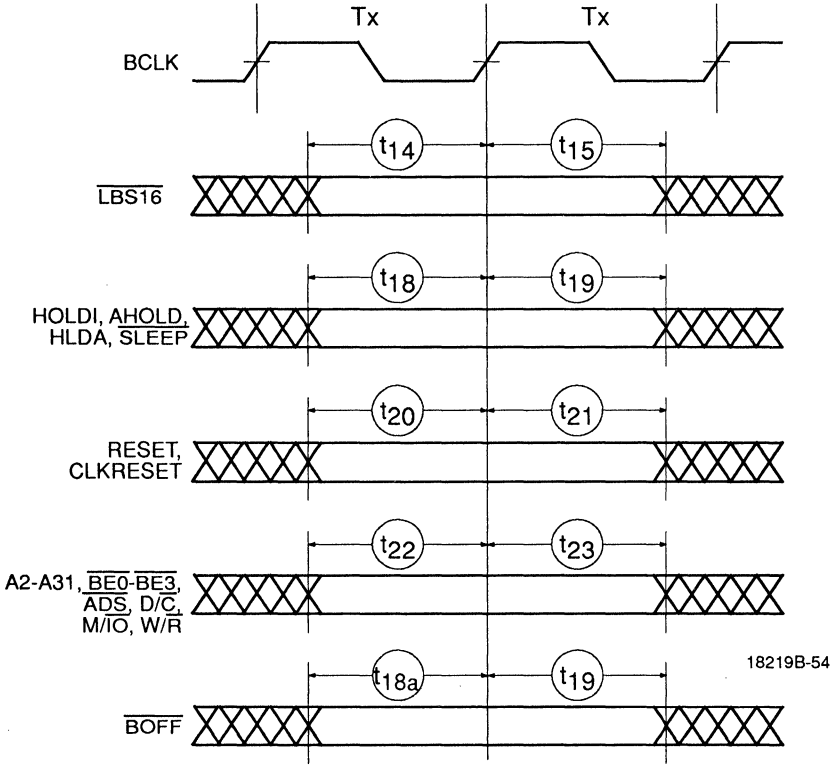
Output Valid Delay Timing



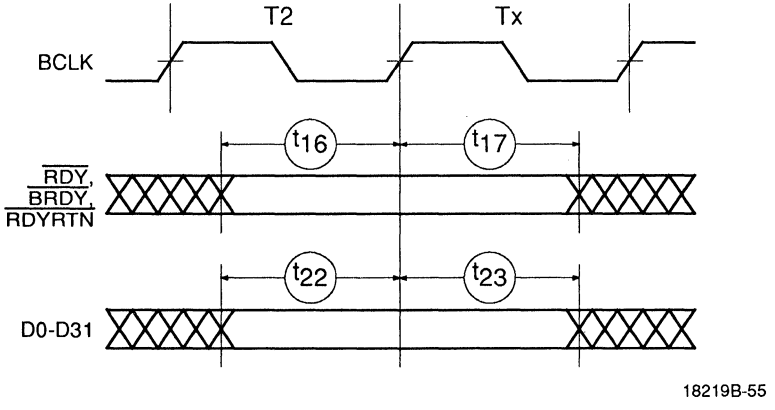
18219B-53

Maximum Float Delay Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE

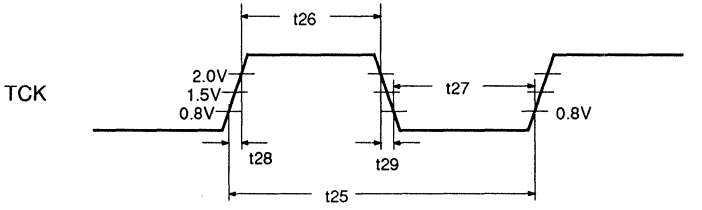


Input Setup and Hold Timing



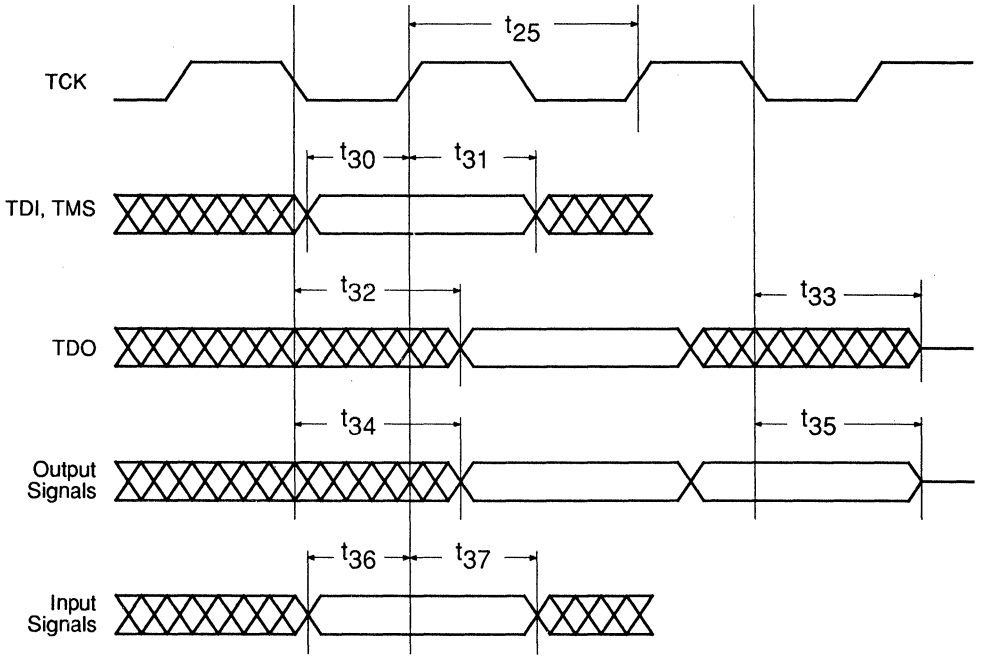
Ready and Data Setup and Hold Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE



18219B-56

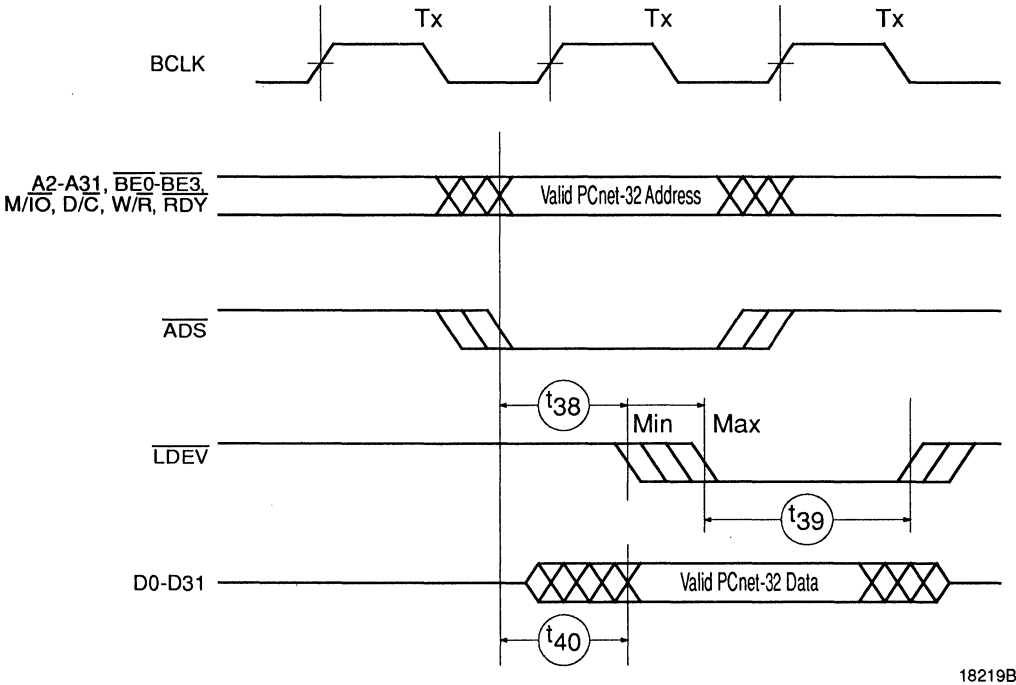
JTAG (IEEE 1149.1) TCK Waveform



18219B-57

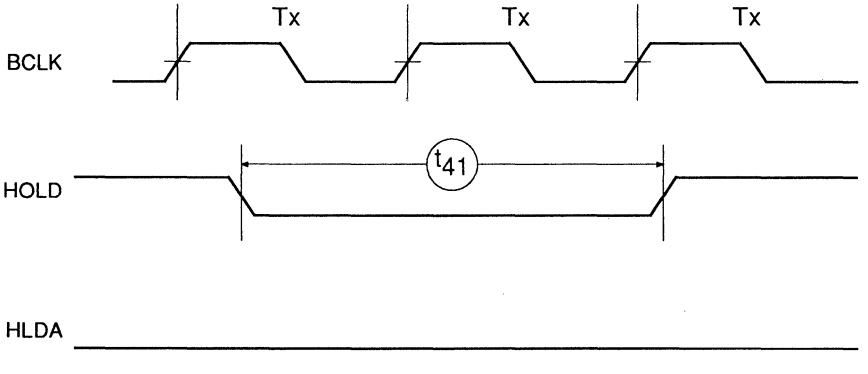
JTAG (IEEE 1149.1) Test Signal Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE



18219B-58

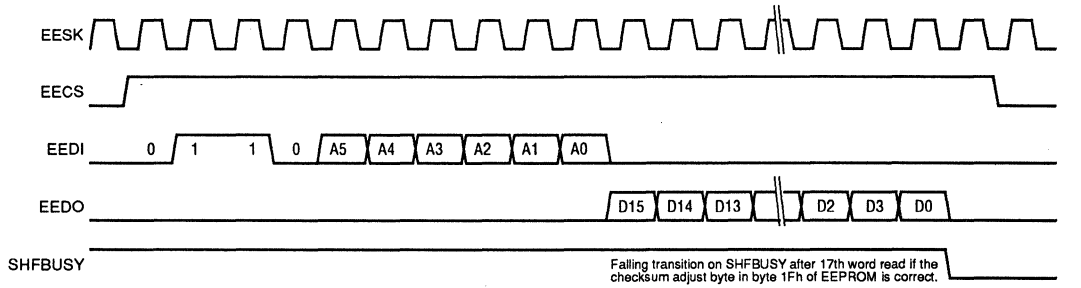
LDEV Timing and Data Bus Activation Timing



18219B-59

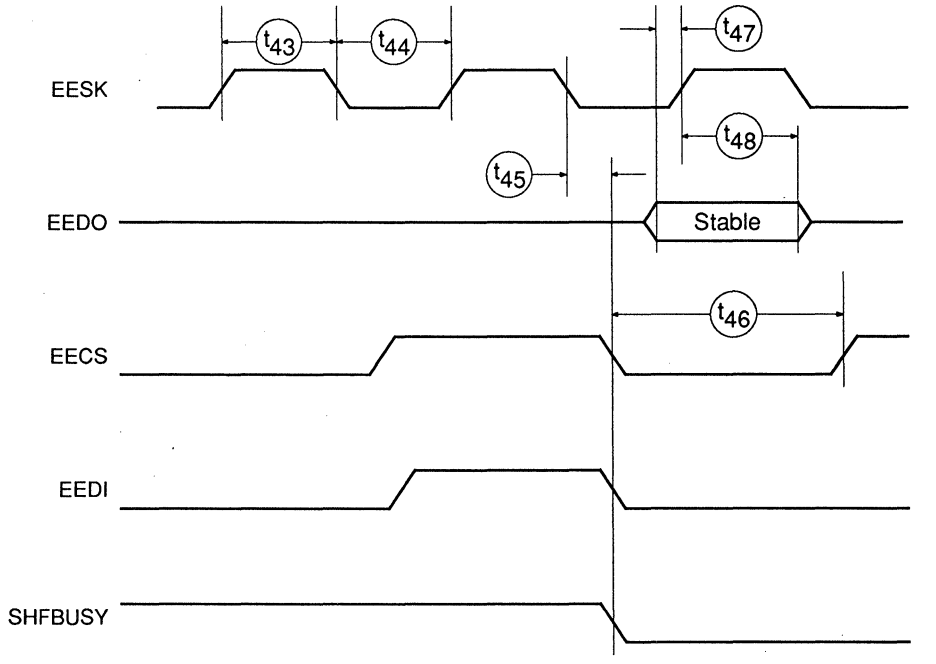
HOLD Inactive Timing

SWITCHING WAVEFORMS: SYSTEM BUS INTERFACE



18219B-60

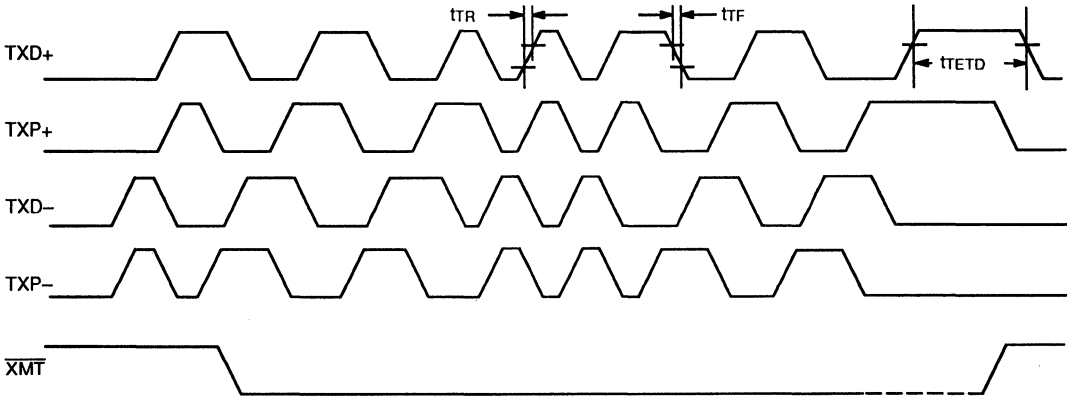
Automatic EEPROM Read Functional Timing



18219B-61

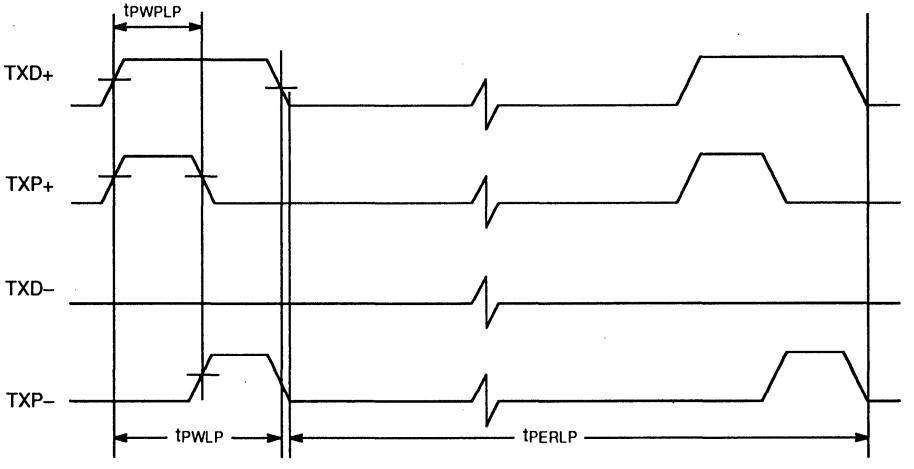
Automatic EEPROM Read Timing

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



18219B-62

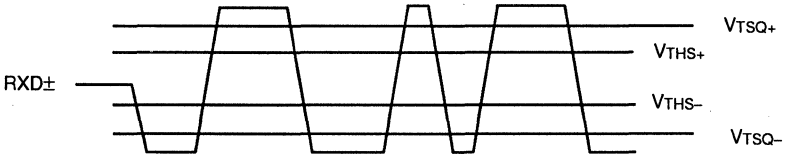
Transmit Timing



18219B-63

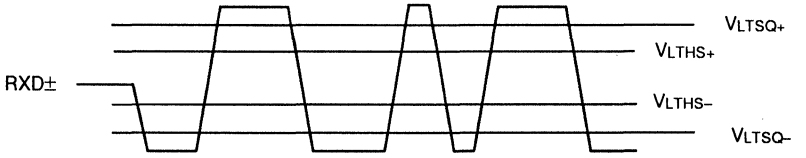
Idle Link Test Pulse

SWITCHING WAVEFORMS: 10BASE-T INTERFACE



18219B-64

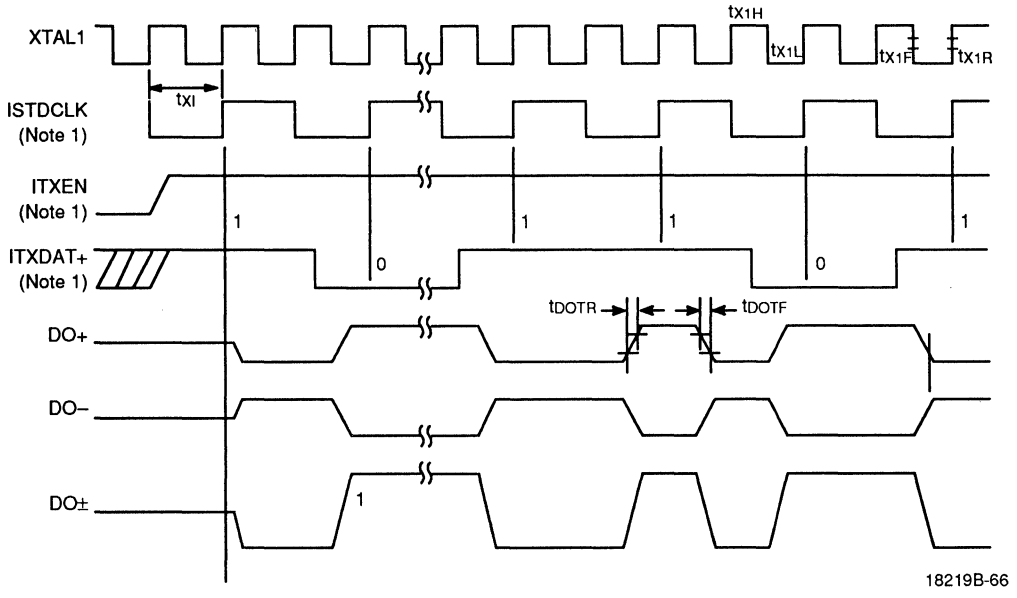
Receive Thresholds (LRT = 0)



18219B-65

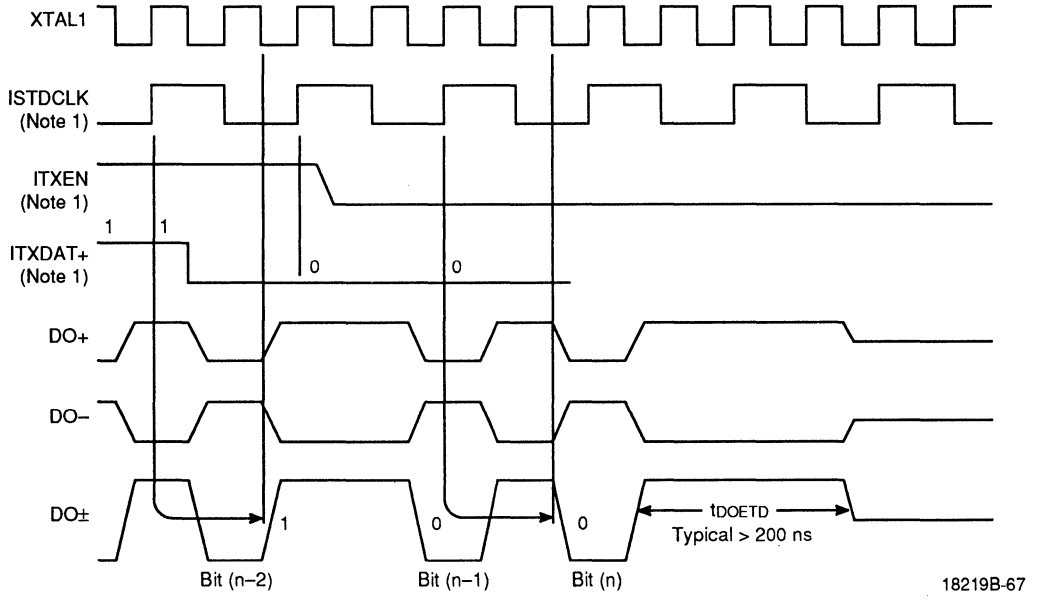
Receive Thresholds (LRT = 1)

SWITCHING WAVEFORMS: AUI



Note 1:
Internal signal and is shown for clarification only.

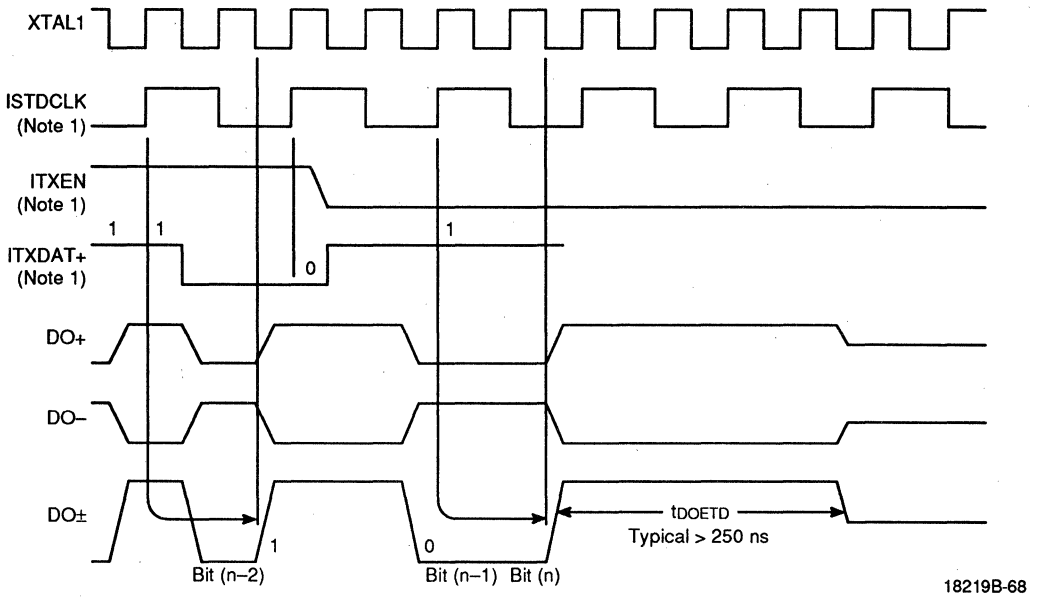
Transmit Timing—Start of Packet



Note 1:
Internal signal and is shown for clarification only.

Transmit Timing—End of Packet (Last Bit = 0)

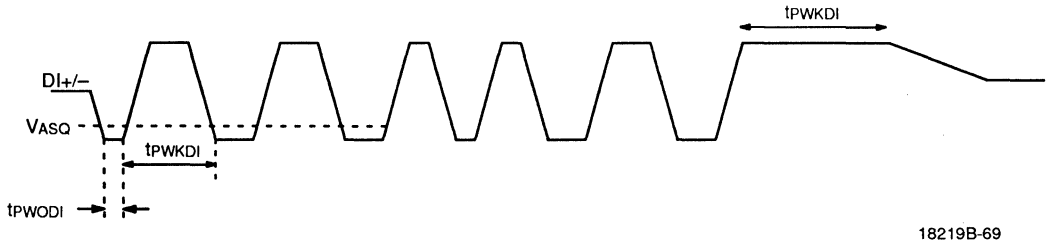
SWITCHING WAVEFORMS: AUI



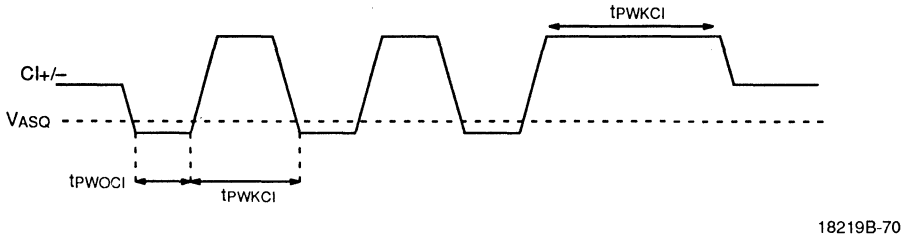
Note 1:
Internal signal and is shown for clarification only.

Transmit Timing—End of Packet (Last Bit = 1)

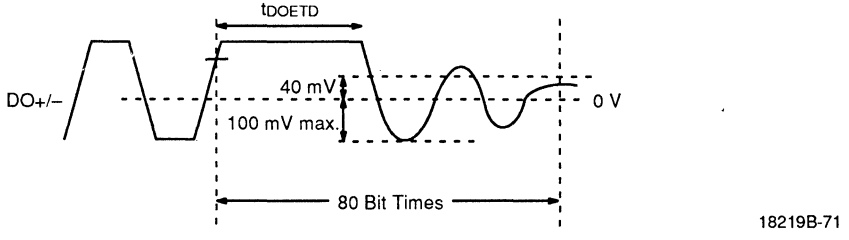
SWITCHING WAVEFORMS: AUI



Receive Timing Diagram

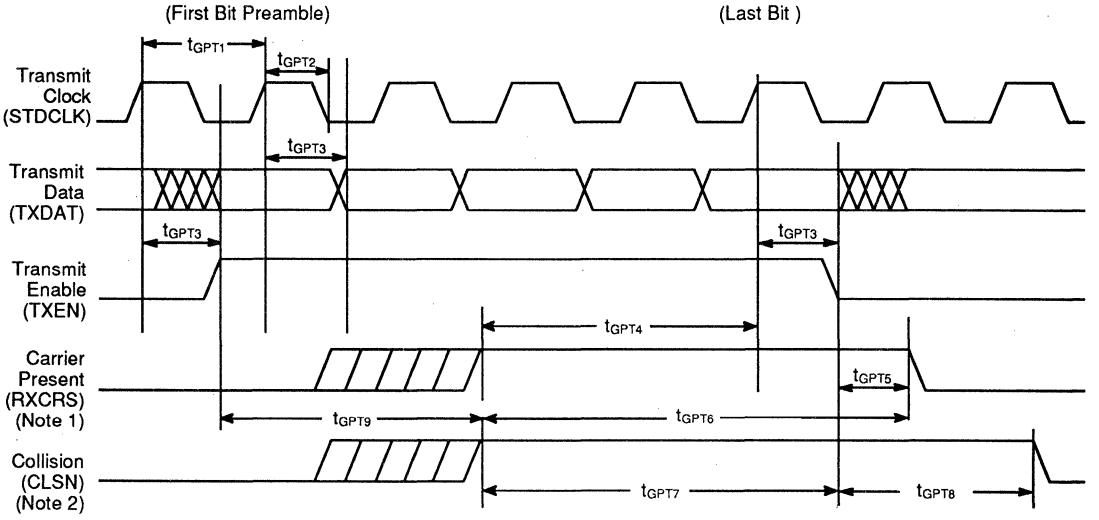


Collision Timing Diagram



Port DO ETD Waveform

SWITCHING WAVEFORMS: GPSI

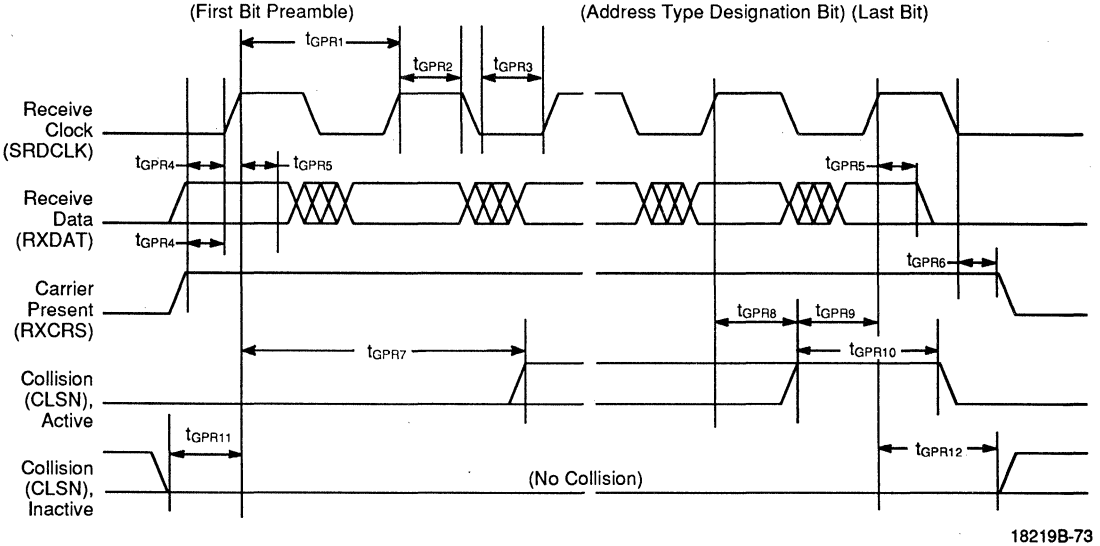


Notes:

18219B-72

1. *If RXCRS is not present during transmission, LCAR bit in TMD2 will be set.*
2. *If CLSN is not present during or shortly after transmission, CERR in CSR0 will be set.*

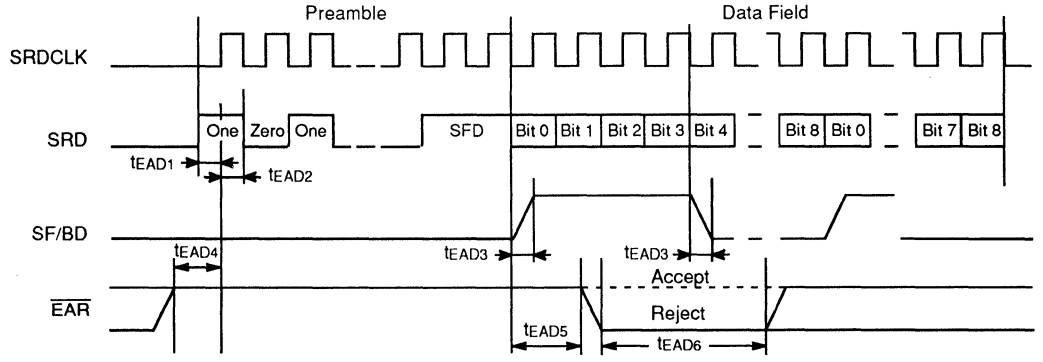
Transmit Timing



18219B-73

Receive Timing

SWITCHING WAVEFORMS: EADI



EADI Reject Timing

APPENDIX A

PCnet-32 Compatible Media Interface Modules

PCnet-32 COMPATIBLE 10BASE-T FILTERS AND TRANSFORMERS

The table below provides a sample list of PCnet-32 compatible 10BASE-T filter and transformer modules available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Filters and Transformers	Filters Transformers and Choke	Filters Transformers Dual Choke	Filters Transformers Dual Chokes
Bel Fuse	A556-2006-DE	16-pin 0.3" DIL	√			
Bel Fuse	0556-2006-00	14-pin SIP	√			
Bel Fuse	0556-2006-01	14-pin SIP			√	
Bel Fuse	0556-6392-00	16-pin 0.5" DIL			√	
Halo Electronics	FD02-101G	16-pin 0.3" DIL	√			
Halo Electronics	FD12-101G	16-pin 0.3" DIL		√		
Halo Electronics	FD22-101G	16-pin 0.3" DIL			√	
PCA Electronics	EPA1990A	16-pin 0.3" DIL	√			
PCA Electronics	EPA2013D	16-pin 0.3" DIL		√		
PCA Electronics	EPA2162	16-pin 0.3" SIP			√	
Pulse Engineering	PE-65421	16-pin 0.3" DIL	√			
Pulse Engineering	PE-65434	16-pin 0.3" SIL			√	
Pulse Engineering	PE-65445	16-pin 0.3" DIL			√	
Pulse Engineering	PE-65467	12-pin 0.5" SMT				√
Valor Electronics	PT3877	16-pin 0.3" DIL	√			
Valor Electronics	FL1043	16-pin 0.3" DIL			√	

PCnet-32 Compatible AUI Isolation Transformers

The table below provides a sample list of PCnet-32 compatible AUI isolation transformers available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3" DIL	50 µH
Bel Fuse	S553-0756-AE	16-pin 0.3" SMD	75 µH
Halo Electronics	TD01-0756K	16-pin 0.3" DIL	75 µH
Halo Electronics	TG01-0756W	16-pin 0.3" SMD	75 µH
PCA Electronics	EP9531-4	16-pin 0.3" DIL	50 µH
Pulse Engineering	PE64106	16-pin 0.3" DIL	50 µH
Pulse Engineering	PE65723	16-pin 0.3" SMT	75 µH
Valor Electronics	LT6032	16-pin 0.3" DIL	75 µH
Valor Electronics	ST7032	16-pin 0.3" SMD	75 µH

PCnet-32 Compatible DC/DC Converters

The table below provides a sample list of PCnet-32 compatible DC/DC converters available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part No.	Package	Voltage	Remote On/Off
Halo Electronics	DCU0-0509D	24-pin DIP	5/-9	No
Halo Electronics	DCU0-0509E	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1007P	24-pin DIP	5/-9	No
PCA Electronics	EPC1054P	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1078	24-pin DIP	5/-9	Yes
Valor Electronics	PM7202	24-pin DIP	5/-9	No
Valor Electronics	PM7222	24-pin DIP	5/-9	Yes

MANUFACTURER CONTACT INFORMATION

Contact the following companies for further information on their products:

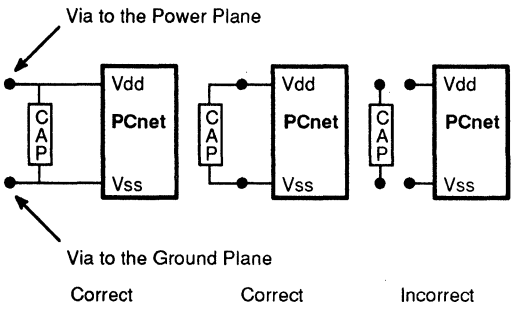
Company		U.S. and Domestic	Asia	Europe
Bel Fuse	Phone: FAX:	(201) 432-0463 (201) 432-9542	852-328-5515 852-352-3706	33-1-69410402 33-1-69413320
Halo Electronics	Phone: FAX:	(415) 969-7313 (415) 367-7158	65-285-1566 65-284-9466	
PCA Electronics (HPC in Hong Kong)	Phone: FAX:	(818)-892-0761 (818)-894-5791	852-553-0165 852-873-1550	33-1-44894800 33-1-42051579
Pulse Engineering	Phone: FAX:	(619) 674-8100 (619) 675-8262	852-425-1651 852-480-5974	353-093-24107 353-093-24459
Valor Electronics	Phone: FAX:	(619) 537-2500 (619) 537-2525	852-513-8210 852-513-8214	49-89-6923122 49-89-6926542

APPENDIX B

Recommendation for Power and Ground Decoupling

The mixed analog/digital circuitry in the PCnet-32 make it imperative to provide noise-free power and ground connections to the device. Without clean power and ground connections, a design may suffer from high bit error rates or may not function at all. Hence, it is highly recommended that the guidelines presented here are followed to ensure a reliable design.

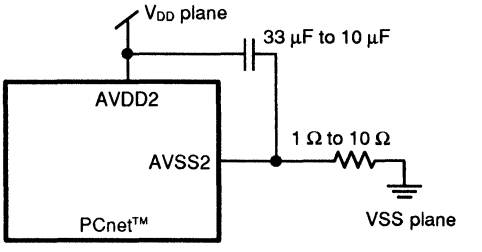
Decoupling/Bypass Capacitors: Adequate decoupling of the power and ground pins and planes is required by all PCnet-32 designs. This includes both low-frequency bulk capacitors and high frequency capacitors. It is recommended that **at least one** low-frequency bulk (e.g. 22 µF) decoupling capacitor be used in the area of the PCnet-32 device. The bulk capacitor(s) should be connected directly to the power and ground planes. In addition, **at least 8** high frequency decoupling capacitors (e.g. 0.1 µF multilayer ceramic capacitors) should be used around the periphery of the PCnet-32 device to prevent power and ground bounce from affecting device operation. To reduce the inductance between the power and ground pins and the capacitors, the pins should be connected directly to the capacitors, rather than through the planes to the capacitors. The suggested connection scheme for the capacitors is shown in the figure below. Note also that the traces connecting these pins to the capacitors should be as wide as possible to reduce inductance (15 mils is desirable).



The most critical pins in the layout of a PCnet-32 design are the 4 analog power and 2 analog ground pins, AVDD[1-4] and AVSS[1-2], respectively. All of these pins are located in one corner of the device, the "analog corner". Specific functions and layout requirements of the analog power and ground pins are given below.

AVSS1 and AVDD3: These pins provide the power and ground for the Twisted Pair and AUI drivers. In addition AVSS1 serves as the ground for the logic interfaces in the 20 MHz Crystal Oscillator. Hence, these pins can be very noisy. A dedicated 0.1 µF capacitor between these pins is recommended.

AVSS2 and AVDD2: These pins are the **most critical** pins on the PCnet-32 device because they provide the power and ground for the phase-lock loop (PLL) portion of the chip. The voltage-controlled oscillator (VCO) portion of the PLL is sensitive to noise in the 60 kHz – 200 kHz range. To prevent noise in this frequency range from disrupting the VCO, it is **strongly recommended** that the low-pass filter shown below be implemented on these pins.



To determine the value for the resistor and capacitor, the formula is:

$$R * C \geq 88$$

where R is in ohms and C is in microfarads. Some possible combinations are given below. To minimize the voltage drop across the resistor, the R value should not be more than 10 Ω.

R	C
2.7 Ω	33 µF
4.3 Ω	22 µF
6.8 Ω	15 µF
10 Ω	10 µF

AVSS2 and AVDD2/AVDD4: These pins provide power and ground for the AUI and twisted pair receive circuitry. In addition, as mentioned earlier, AVSS2 and AVDD2 provide power and ground for the phase-lock loop portion of the chip. Except for the filter circuit already mentioned, no specific decoupling is necessary on these pins.

AVDD1: AVDD1 provides power for the control and interface logic in the PLL. Ground for this logic is provided by digital ground pins. No specific decoupling is necessary on this pin.

Special Note for Adapter Cards: In adapter card designs, it is important to utilize all available power and ground pins available on the bus edge connector. In addition, the connection from the bus edge connector to the power or ground plane should be made through more than one via and with wide traces (15 mils desirable) wherever possible. Following these recommendations results in minimal inductance in the power and ground paths. By minimizing this inductance, ground bounce is minimized.

APPENDIX C

Alternative Method for Initialization

The PCnet-32 controller may be initialized by performing I/O writes only. That is, data can be written directly to the appropriate control and status registers (CSR) instead of reading from the Initialization Block in memory. The registers that must be written are shown in the table below. These books are followed by writing the START bit in CSR0.

Control and Status Register	Comment
CSR8	LADRF[15:0]
CSR9	LADRF[31:16]
CSR10	LADRF[47:32]
CSR11	LADRF[63:48]
CSR12	PADR[15:0]
CSR13	PADR[31:16]
CSR14	PADR[47:32]
CSR15	Mode
CSR24-25	BADR
CSR30-31	BADX
CSR47	POLLINT
CSR76	RCVRL
CSR78	XMTRL

Note: *The INIT bit must not be set or the initialization block will be accessed instead.*

Introduction of the Look-Ahead Packet Processing (LAPP) Concept

A driver for the PCnet-32 controller would normally require that the CPU copy receive frame data from the controller's buffer space to the application's buffer space *after* the entire frame has been received by the controller. For applications that use a ping-pong windowing style, the traffic on the network will be halted until the current frame has been completely processed by the entire application stack. This means that the time between last byte of a receive frame arriving at the client's Ethernet controller and the client's transmission of the first byte of the next outgoing frame will be separated by:

1) the time that it takes the client's CPU's interrupt procedure to pass software control from the current task to the driver
2) plus the time that it takes the client driver to pass the header data to the application and request an application buffer
3) plus the time that it takes the application to generate the buffer pointer and then return the buffer pointer to the driver
4) plus the time that it takes the client driver to transfer all of the frame data from the controller's buffer space into the application's buffer space and then call the application again to process the complete frame
5) plus the time that it takes the application to process the frame and generate the next outgoing frame
6) plus the time that it takes the client driver to set up the descriptor for the controller and then write a TDMD bit to CSR0

The sum of these times can often be about the same as the time taken to actually transmit the frames on the wire, thereby yielding a network utilization rate of less than 50%.

An important thing to note is that the PCnet-32 controller's data transfers to its buffer space are such that the system bus is needed by the PCnet-32 controller for approximately 4% of the time. This leaves 96% of the sytem bus bandwidth for the CPU to perform some of the inter–frame operations *in advance of the completion of network receive activity*, if possible. The question then becomes: how much of the tasks that need to be performed between reception of a frame and transmission of the next frame can be performed *before* the reception of the frame actually ends at the network, and how can the CPU be instructed to perform these tasks during the network reception time?

The answer depends upon exactly what is happening in the driver and application code, but the steps that can be performed at the same time as the receive data are arriving include as much as the first three steps and part of the fourth step shown in the sequence above. By performing these steps before the entire frame has arrived, the frame throughput can be substantially increased.

A good increase in performance can be expected when the first three steps are performed before the end of the network receive operation. A much more significant performance increase could be realized if the PCnet-32 controller could place the frame data directly into the application's buffer space; (i.e. eliminate the need for step four.) In order to make this work, it is necessary that the application buffer pointer be determined before the frame has completely arrived, then the buffer pointer in the next desriptor for the receive frame would need to be modified in order to direct the PCnet-32 controller to write directly to the application buffer. More details on this operation will be given later.

An alternative modification to the existing system can gain a smaller, but still significant improvement in performance. This alternative leaves step four unchanged in that the CPU is still required to perform the copy operation, but it allows a large portion of the copy operation to be done before the frame has been completely received by the controller, (i.e. the CPU can perform the copy operation of the receive data from the PCnet-32 controller's buffer space into the application buffer space *before* the frame data has completely arrived from the network.) This allows the copy operation of step four to be performed concurrently with the arrival of network data, rather than sequentially, following the end of network receive activity.

Outline of the LAPP Flow:

This section gives a suggested outline for a driver that utilizes the LAPP feature of the PCnet-32 controller.

Note: *The labels in the following text are used as references in the timeline diagram that follows.*

SETUP:

The driver should set up descriptors in groups of 3, with the OWN and STP bits of each set of three descriptors to read as follows: 11b, 10b, 00b.

An option bit (LAPPEN) exists in CSR3, bit position 5. The software should set this bit. When set, the LAPPEN bit directs the PCnet-32 controller to generate an INTERRUPT when STP has been written to a receive descriptor by the PCnet-32 controller.

FLOW:

The PCnet-32 controller polls the current receive descriptor at some point in time before a message arrives. The PCnet-32 controller determines that this receive buffer is OWNed by the PCnet-32 controller and it stores the descriptor information to be used when a message does arrive.

N0: Frame preamble appears on the wire, followed by SFD and destination address.

N1: The 64th byte of frame data arrives from the wire. This causes the PCnet-32 controller to begin frame data DMA operations to the first buffer.

C0: When the 64th byte of the message arrives, the PCnet-32 controller performs a lookahead operation to the next receive descriptor. This descriptor should be owned by the PCnet-32 controller.

C1: The PCnet-32 controller intermittently requests the bus to transfer frame data to the first buffer as it arrives on the wire.

S0: The driver remains idle.

C2: When the PCnet-32 controller has completely filled the first buffer, it writes status to the first descriptor.

C3: When the first descriptor for the frame has been written, changing ownership from the PCnet-32 controller to the CPU, the PCnet-32 controller will generate an SRP INTERRUPT. (This interrupt appears as a RINT interrupt in CSR0.)

S1: The SRP INTERRUPT causes the CPU to switch tasks to allow the PCnet-32 controller's driver to run.

C4: During the CPU interrupt–generated task switching, the PCnet-32 controller is performing a lookahead operation to the third descriptor. At this point in time, the third descriptor is owned by the CPU. [Note: *Even though the third buffer is not owned by the PCnet-32 controller, existing AMD Ethernet controllers will continue to perform data DMA into the buffer space that the controller already owns (i.e. buffer number 2). The controller does not know if buffer space in buffer number 2 will be sufficient or not, for this frame, but it has no way to tell except by trying to move the entire message into that space. Only when the message does not fit will it signal a buffer error condition – there is no need to panic at the point that it discovers that it does not yet own descriptor number 3.*]

S2: The first task of the driver's interrupt service routine is to collect the header information from the PCnet-32 controller's first buffer and pass it to the application.

S3: The application will return an application buffer pointer to the driver. The driver will add an offset to the application data buffer pointer, since the PCnet-32 controller will be placing the first portion of the message into the first and second buffers. (The modified application data buffer pointer will only be directly used by the PCnet-32 controller when it reaches the third buffer.) The driver will place the modified data buffer pointer into the final descriptor of the group (#3) and will grant ownership of this descriptor to the PCnet-32 controller.

C5: Interleaved with S2, S3 and S4 driver activity, the PCnet-32 controller will write frame data to buffer number 2.

S4: The driver will next proceed to copy the contents of the PCnet-32 controller's first buffer to the *beginning* of the application space. This copy will be to the exact (unmodified) buffer pointer that was passed by the application.

S5: After copying all of the data from the first buffer into the beginning of the application data buffer, the driver will begin to poll the ownership bit of the second descriptor. The driver is waiting for the PCnet-32 controller to finish filling the second buffer.

C6: At this point, knowing that it had not previously owned the third descriptor, and knowing that the current message has not ended (there is more data in the fifo), the PCnet-32 controller will make a "last ditch lookahead" to the final (third) descriptor; This time, the ownership will be TRUE (i.e. the descriptor belongs to the controller), because the driver wrote the application pointer into this descriptor and then changed the ownership to give the descriptor to the PCnet-32 controller back at S3. Note that if steps S1, S2 and S3 have not completed at this time, a BUFF error will result.

C7: After filling the second buffer and performing the last chance lookahead to the next descriptor, the PCnet-32 controller will write the status and change the ownership bit of descriptor number 2.

S6: After the ownership of descriptor number 2 has been changed by the PCnet-32 controller, the *next* **driver** poll of the 2nd descriptor will show ownership granted to the CPU. The driver now copies the data from buffer number 2 into the "middle section"

of the application buffer space. This operation is interleaved with the C7 and C8 operations.

C8: The PCnet-32 controller will perform data DMA to the last buffer, whose pointer is pointing to application space. Data entering the last buffer will not need the infamous "double copy" that is required by existing drivers, since it is being placed directly into the application buffer space.

N2: The message on the wire ends.

S7: When the driver completes the copy of buffer number 2 data to the application buffer space, it begins polling descriptor number 3.

C9: When the PCnet-32 controller has finished all data DMA operations, it writes status and changes ownership of descriptor number 3.

S8: The driver sees that the ownership of descriptor number 3 has changed, and it calls the application to tell the application that a frame has arrived.

S9: The application processes the received frame and generates the next TX frame, placing it into a TX buffer.

S10: The driver sets up the TX descriptor for the PCnet-32 controller.

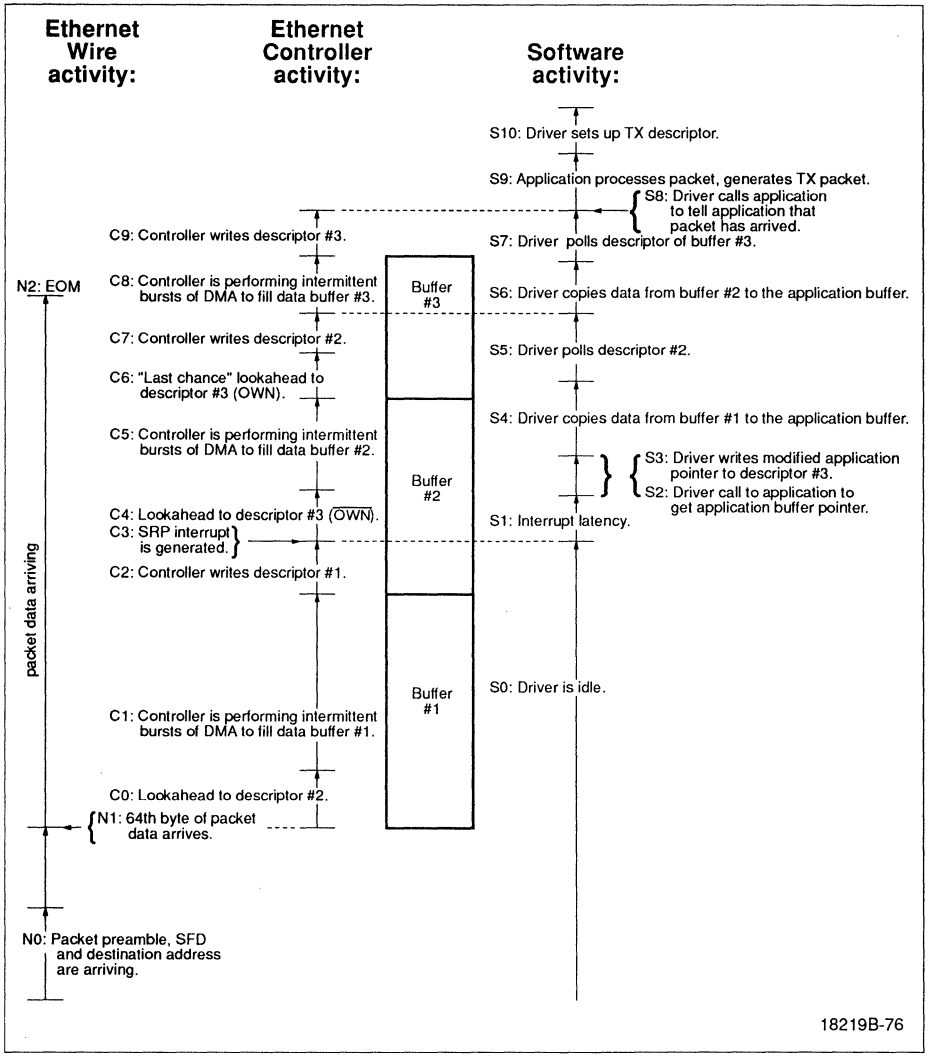


Figure 1. LAPP Timeline

LAPP Software Requirements

Software needs to set up a receive ring with descriptors formed into groups of 3. The first descriptor *of each group* should have OWN = 1 and STP = 1, the second descriptor *of each group* should have OWN = 1 and STP = 0. The third descriptor *of each group* should have OWN = 0 and STP = 0. The size of the first buffer (as indicated in the first descriptor), should be ***at least*** equal to the largest expected header size; However, for maximum efficiency of CPU utilization, the first buffer size should be larger than the header size. It should be equal to the expected number of message bytes, minus the time needed for Interrupt latency and minus the application call latency, minus the time needed for the driver to write to the third descriptor, minus the time needed for the driver to copy data from buffer #1 to the application buffer space, and minus the time needed for the driver to copy data from buffer #2 to the application buffer space. Note that the time needed for the copies performed by the driver depends upon the sizes of the 2nd and 3rd buffers, and that the sizes of the second and third buffers need to be set accoring to the time needed for the data copy operations! This means that an iterative self–adjusting mechanism needs to be placed into the software to determine the correct buffer sizing for optimal operation. Fixed values for buffer sizes may be used; In such a case, the LAPP method will still provide a significant performance increase, but the performance increase will not be maximized.

The following diagram illustrates this setup for a receive ring size of 9:

LAPP Rules for Parsing of Descriptors

When using the LAPP method, software must use a modified form of descriptor *parsing* as follows:

Software will examine OWN and STP to determine where a RCV frame begins. RCV frames will only begin in buffers that have OWN = 0 and STP = 1.

Software shall assume that a frame continues until it finds *either* ENP = 1 or ERR= 1.

Software must discard all descriptors with OWN = 0 and STP = 0 and move to the next descriptor when searching for the beginning of a new frame; ENP and ERR should be ignored by software during this search.

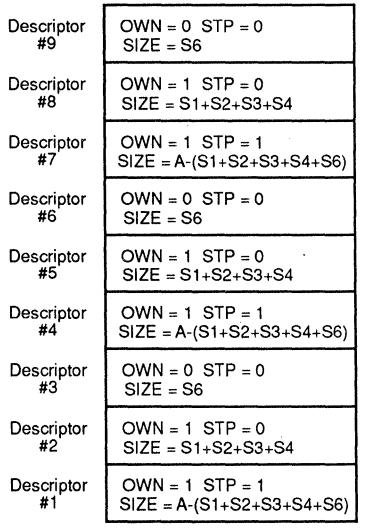
Software cannot change an STP value in the receive descriptor ring after the initial setup of the ring is complete, even if software has ownership of the STP descriptor *unless* the previous STP descriptor in the ring is *also* OWNED by the software.

When LAPPEN = 1, then hardware will use a modified form of descriptor *parsing* as follows:

The controller will examine OWN and STP to determine where to begin placing a RCV frame. A new RCV frame will only begin in a buffer that has OWN = 1 and STP = 1.

The controller will always obey the OWN bit for determining whether or not it may use the next buffer for a chain.

The controller will always mark the end of a frame with *either* ENP = 1 or ERR= 1.



A = Expected message size in bytes
S1 = Interrupt latency
S2 = Application call latency
S3 = Time needed for driver to write to third descriptor
S4 = Time needed for driver to copy data from buffer #1 to application buffer space
S6 = Time needed for driver to copy data from buffer #2 to application buffer space

Note that the times needed for tasks S1, S2, S3, S4, and S6 should be divided by 0.8 microseconds to yield an equivalent number of network byte times before subtracting these quantities from the expected message size A.

18219B-77

Figure 2. LAPP 3 Buffer Grouping

The controller will *discard* all descriptors with **OWN = 1** and STP = 0 and move to the next descriptor ***when searching for a place to begin a new frame***. It discards these desciptors by simply changing the ownership bit from OWN=1 to OWN = 0. Such a descriptor is unused for receive purposes by the controller, and the driver must recognize this. (The driver will recognize this if it follows the software rules.)

The controller will *ignore* all descriptors with **OWN = 0** and STP = 0 and move to the next descriptor ***when searching for a place to begin a new frame***. In other words, the controller is allowed to skip entries in the ring that it does not own, but only when it is looking for a place to begin a new frame.

Some Examples of LAPP Descriptor Interaction

Choose an expected frame size of 1060 bytes.

Choose buffer sizes of 800, 200 and 200 bytes.

1) Assume that a 1060 byte frame arrives correctly, and that the timing of the early interrupt and the software is smooth. The descriptors will have changed from:

Descriptor Number	Before the Frame Arrived			After the Frame has Arrived			Comments (After Frame Arrival)
	OWN	STP	ENP*	OWN	STP	ENP*	
1	1	1	X	0	1	0	Bytes 1–800
2	1	0	X	0	0	0	Bytes 801–1000
3	0	0	X	0	0	1	Bytes 1001–1060
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Not yet used

**ENP or ERR*

2) Assume that instead of the expected 1060 byte frame, a 900 byte frame arrives, either because there was an error in the network, or because this is the last frame in a file transmission sequence.

Descriptor Number	Before the Frame Arrived			After the Frame has Arrived			Comments (After Frame Arrival)
	OWN	STP	ENP*	OWN	STP	ENP*	
1	1	1	X	0	1	0	Bytes 1–800
2	1	0	X	0	0	1	Bytes 801–900
3	0	0	X	0	0	?**	Discarded buffer
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Not yet used

**ENP or ERR*

*** Note that the PCnet-32 controller* might *write a ZERO to ENP location in the 3rd descriptor. Here are the two possibilities:*

1) If the controller finishes the data transfers into buffer number 2 after *the driver writes the application's modified buffer pointer into the third descriptor, then the controller will write a ZERO to ENP for this buffer and will write a ZERO to OWN and STP.*

2) If the controller finishes the data transfers into buffer number 2 before *the driver writes the application's modified buffer pointer into the third descriptor, then the controller will complete the frame in buffer number two* and then skip the then unowned third buffer. *In this case, the PCnet-32 controller will not have had the opportunity to RESET the ENP bit in this descriptor, and it is possible that the software left this bit as ENP=1 from the last time through the ring. Therefore, the software* **must** *treat the location as a don't care; The rule is, after finding ENP=1 (or ERR=1) in descriptor number 2, the software* must *ignore ENP bits until it finds the next STP=1.*

3) Assume that instead of the expected 1060 byte frame, a 100 byte frame arrives, because there was an error in the network, or because this is the last frame in a file transmission sequence, or perhaps because it is an acknowledge frame.

Descriptor Number	Before the Frame Arrived			After the Frame has Arrived			Comments (After Frame Arrival)
	OWN	STP	ENP*	OWN	STP	ENP*	
1	1	1	X	0	1	1	Bytes 1–100
2	1	0	X	0	0	0***	Discarded buffer
3	0	0	X	0	0	?**	Discarded buffer
4	1	1	X	1	1	X	Controller's current location
5	1	0	X	1	0	X	Not yet used
6	0	0	X	0	0	X	Not yet used
etc.	1	1	X	1	1	X	Not yet used

**ENP or ERR*

*** Same as note in case 2 above, except that in this case, it is very unlikely that the driver can respond to the interrupt and get the pointer from the application before the PCnet-32 controller has completed its poll of the next descriptors. This means that for almost all occurrences of this case, the PCnet-32 controller will not find the OWN bit set for this descriptor and therefore, the ENP bit will almost always contain the old value, since the PCnet-32 controller will not have had an opportunity to modify it.*

**** Note that even though the PCnet-32 controller will write a ZERO to this ENP location, the software* should *treat the location as a don't care, since after finding the ENP=1 in descriptor number 2, the software should ignore ENP bits until it finds the next STP=1.*

Buffer Size Tuning

For maximum performance, buffer sizes should be adjusted depending upon the expected frame size and the values of the interrupt latency and application call latency. The best driver code will minimize the CPU utilization while also minimizing the latency from frame end on the network to frame sent to application from driver (frame latency). These objectives are aimed at increasing throughput on the network while decreasing CPU utilization.

Note that the buffer sizes in the ring may be altered at any time that the CPU has ownership of the corresponding descriptor. The best choice for buffer sizes will maximize the time that the driver is swapped out, while minimizing the time from the last byte written by the PCnet-32 controller to the time that the data is passed from the driver to the application. In the diagram, this corresponds to maximizing S0, while minimizing the time between C9 and S8. (The timeline happens to show a minimal time from C9 to S8.)

Note that by increasing the size of buffer number 1, we increase the value of S0. However, when we increase the size of buffer number 1, we also increase the value of S4. If the size of buffer number 1 is too large, then the driver will not have enough time to perform tasks S2, S3, S4, S5 and S6. The result is that there will be delay from the execution of task C9 until the execution of task S8. A perfectly timed system will have the values for S5 and S7 at a minimum.

An average increase in performance can be achieved if the general guidelines of buffer sizes in figure 2 is followed. However, as was noted earlier, the correct sizing for buffers will depend upon the expected message size. There are two problems with relating expected message size with the correct buffer sizing:

1) Message sizes cannot always be accurately predicted, since a single application may expect different message sizes at different times, therefore, the buffer sizes chosen will not always maximize throughput.

2) Within a single application, message sizes might be somewhat predicatable, but when the same driver is to be shared with multiple applications, there may not be a common predictable message size.

Additional problems occur when trying to define the correct sizing because the correct size also depends upon the interrupt latency, which may vary from system to system, depending upon both the hardware and the software installed in each system.

In order to deal with the unpredictable nature of the message size, the driver can implement a self tuning

mechanism that examines the amount of time spent in tasks S5 and S7 as such: While the driver is polling for each descriptor, it could count the number of poll operations performed and then adjust the number 1 buffer size to a larger value, by adding "t" bytes to the buffer count, if the number of poll operations was greater than "x". If fewer than "x" poll operations were needed for each of S5 and S7, then the software should adjust the buffer size to a smaller value by, subtracting "y" bytes from the buffer count. Experiments with such a tuning mechanism must be performed to determine the best values for "X" and "y."

Note whenever the size of buffer number 1 is adjusted, buffer sizes for buffer number 2 and buffer 3 should also be adjusted.

In some systems the typical mix of receive frames on a network for a client application consists mostly of large data frames, with very few small frames. In this case, for maximum efficiency of buffer sizing, *when a frame arrives under a certain size limit, the driver should* ***not*** *adjust the buffer sizes in response to the short frame.*

An Alternative LAPP Flow – the TWO Interrupt Method

An alternative to the above suggested flow is to use two interrupts, one at the start of the Receive frame and the other at the end of the receive frame, instead of just looking for the SRP interupt as was described above. This alternative attempts to reduce the amount of time that the software "wastes" while polling for descriptor own bits. This time would then be available for other CPU tasks. It also minimizes the amount of time the CPU needs for data copying. This savings can be applied to other CPU tasks.

The time from the end of frame arrival on the wire to delivery of the frame to the application is labeled as frame latency. For the one-interrupt method, frame latency is minimized, while CPU utilization increases. For the two-interrupt method, frame latency becomes greater, while CPU utilization decreases.

Note that some of the CPU time that can be applied to non-Ethernet tasks is used for task switching in the CPU. One task switch is required to swap a non-Ethernet task into the CPU (after S7A) and a second task switch is needed to swap the Ethernet driver back in again (at S8A). If the time needed to perform these task switches exceeds the time saved by not polling descriptors, then there is a net loss in performance with this method. Therefore, the SPRINT method implemented should be carefully chosen.

Figure 3 shows the event flow for the two-interrupt method.

Figure 4 shows the buffer sizing for the two-interrupt method. Note that the second buffer size will be about the same for each method.

There is another alternative which is a marriage of the two previous methods. This third possibility would use the buffer sizes set by the two-interrupt method, but would use the polling method of determining frame end. This will give good frame latency but at the price of very high CPU utilization.

And still, there are even more compromise positions that use various fixed buffer sizes and effectively, the flow of the one-interrupt method. All of these compromises will reduce the complexity of the one-interrupt method by removing the heuristic buffer sizing code, but they all become less efficient than heuristic code would allow.

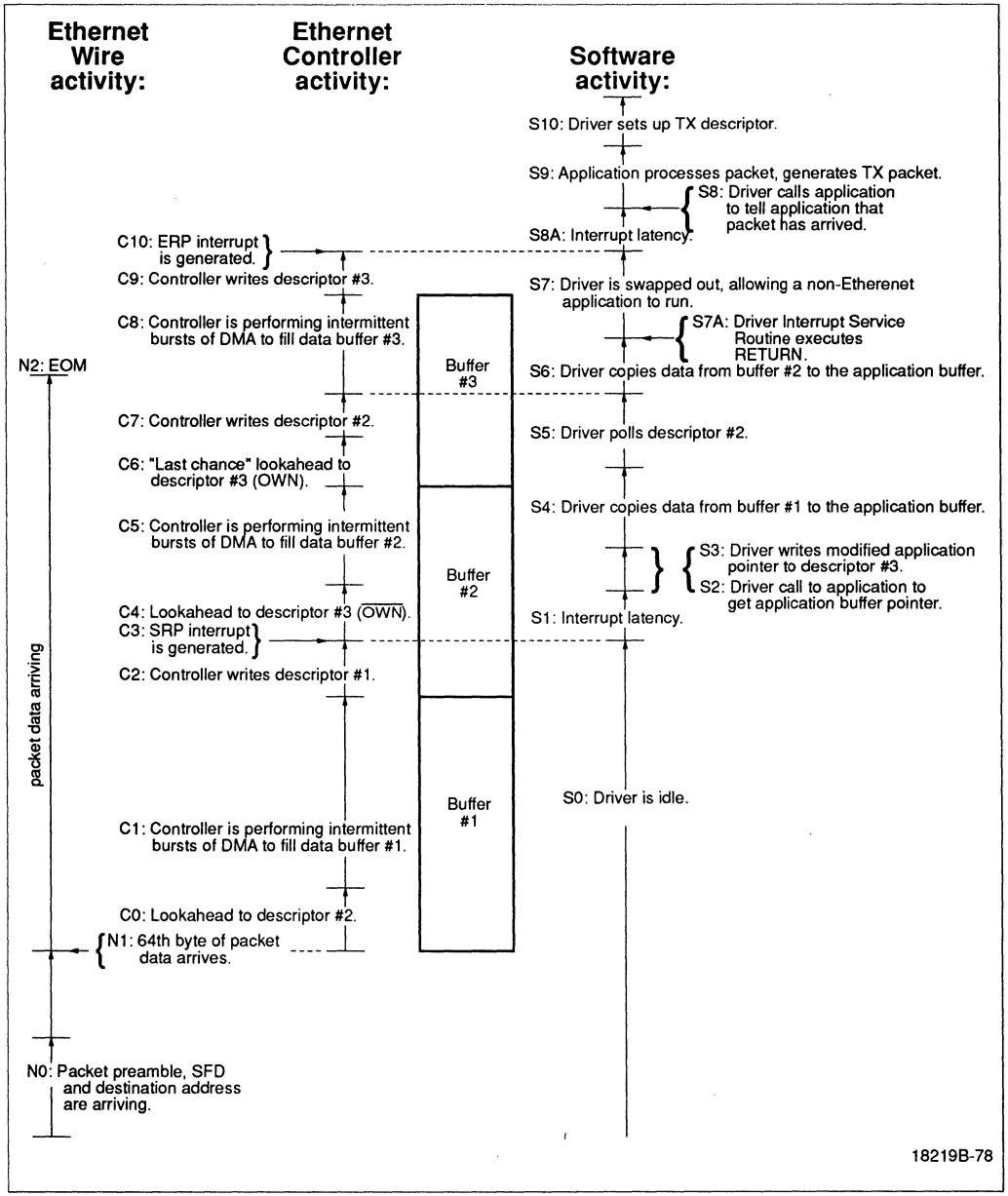
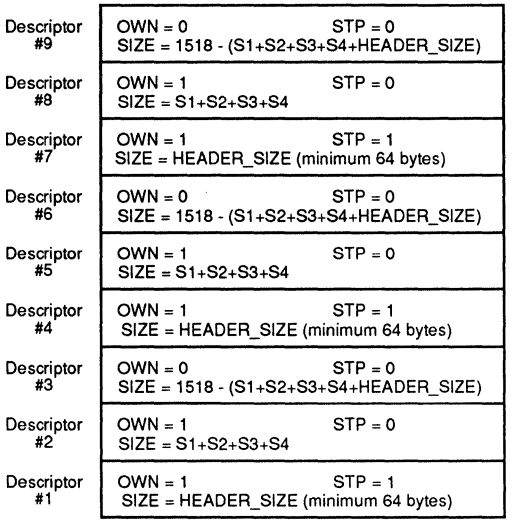


Figure 3. LAPP Timeline for TWO-INTERRUPT Method



A = Expected message size in bytes
S1 = Interrupt latency
S2 = Application call latency
S3 = Time needed for driver to write to third descriptor
S4 = Time needed for driver to copy data from buffer #1 to application buffer space
S6 = Time needed for driver to copy data from buffer #2 to application buffer space

Note that the times needed for tasks S1, S2, S3, S4, and S6 should be divided by 0.8 microseconds to yield an equivalent number of network byte times before subtracting these quantities from the expected message size A.

18219B-79

Figure 4. LAPP 3 Buffer Grouping for TWO-INTERRUPT Method

DATA SHEET REVISION SUMMARY

The following list represents the key differences between revision A (October 1993) and revision B (May 1994).

Global Change

Look-Ahead Packet Processing (LAPP) is the name for the early protocol analysis.

Support for 386DX mode is removed from the device. Descriptions for 386 mode of operation are deleted throughout the data sheet.

DWIO mode cannot be set by reading the EEPROM or writing directly to BCR18.

The Initialization Block must be on a double-word boundary.

The acronym DMABR (DMA Burst Register) is changed to DMACR (DMA Cycle Register).

Pin Description—VL-Bus Mode

Page 1-670:
The description for $\overline{\text{WBACK}}$ is rewritten for clarity.

Pin Description—Local Bus Mode

Page 1-684:
The description for $\overline{\text{WBACK}}$ is rewritten for clarity.

The descriptions for pins $\overline{\text{Am486}}$, $\overline{\text{AHOLD}}$, BCLK, $\overline{\text{BOFF}}$, $\overline{\text{EADS}}$, RESET, $\overline{\text{RDY}}$, and $\overline{\text{RDYRTN}}$ are rewritten for clarity.

Detailed Functions

Bus Interface Unit

Page 1-694:
The description for $\overline{\text{WBACK}}$ is rewritten for clarity.

The following timing figures have been changed to correct typographical errors. Signals A2-A31, $\overline{\text{BE0}}$–$\overline{\text{BE3}}$, M/$\overline{\text{IO}}$, D/$\overline{\text{C}}$, and W/$\overline{\text{R}}$ are now shown to be driven one clock cycle before $\overline{\text{ADS}}$ becomes asserted.

Figures 1–19, 21, 23, 24, 25, 26, and 27.

Manchester Encoder/Decoder

Page 1-743:

PLL Tracking

The number for VCO bit cell phase corrections is now corrected to 10%.

General Purpose Serial Interface

Page 1-748:

This section is rewritten for clarity.

Software Access

Page 1-750:

The section on PCnet-32 Controller I/O Resource Mapping is rewritten to reflect changes in methods of setting DWIO mode.

Hardware Access

Page 1-756

The section on EEPROM microwire access is rewritten for clarity.

User Accessible Registers

Page 1-769

CSR1
The description is rewritten for clarity.

CSR3
A new bit, DXSUFLO (bit 6) is added.

The bit name for bit 5 is now LAPPEN.

CSR4
The bit description for bit 15 (ENTST) is rewritten for clarity.

CSR15
The description for bits 8–7 (PORTSEL) is rewritten for clarity.

CSR58
The description for bit 8 (SSIZE32) is rewritten for clarity.

CSR80
The description for bits 9–8 (XMTFW) is rewritten for clarity.

BCR4–7
The description for bit 0 (COLE) is rewritten for clarity.

BCR18
Bit 10 now contains description for two function: GCIC and IWBACK.

The description for bit 7 (DWIO) is rewritten for clarity.

BCR19
The description for bit 13 (EEDET) is rewritten for clarity.

BCR20
The description for bit 8 (SSIZE32) is rewritten for clarity.

DC Characteristics

Page 1-830:

C_{IN}, C_O, C_{CLK}
The maximum value is now 10 pF.

Switching Characteristics

Page 1-832:

t_{45}
The maximum value is corrected to +15 ns.

Appendices

Appendix A

This section is updated with the latest information.

Appendix B

This section is rewritten for clarity.

Appendix D

This is a new section on the LAPP concept.

PRELIMINARY

Am79C970

PCnet™-PCI Single-Chip Ethernet Controller for PCI Local Bus

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Single-chip Ethernet controller for the Peripheral Component Interconnect (PCI) local bus**
- **Supports ISO 8802-3 (IEEE/ANSI 802.3) and Ethernet Standards**
- **Direct interface to the PCI local bus**
- **Compliant to PCI local bus specification (Revision 2.0)**
- **Software compatible with AMD's Am7990 LANCE, Am79C90 C-LANCE, Am79C960 PCnet-ISA, Am79C961 PCnet-ISA+, Am79C965 PCnet-32, and Am79C900 ILACC™ register and descriptor architecture**
- **Compatible with Am2100/Am1500T and Novell® NE2100/NE1500 driver software**
- **High-performance Bus Master architecture with integrated DMA Buffer Management Unit for low CPU and bus utilization**
- **Big endian byte alignment supported**
- **Single +5 V power supply operation**
- **Low-power, CMOS design with sleep modes allows reduced power consumption for critical battery powered applications and Green PCs**
- **Microwire™ EEPROM interface supports jumperless design**
- **Individual 136-byte transmit and 128-byte receive FIFOs provide frame buffering for increased system latency, and support the following features:**
  - — Automatic retransmission with no FIFO reload
  - — Automatic receive stripping and transmit padding (individually programmable)
  - — Automatic runt packet rejection
  - — Automatic deletion of received collision frames
- **Look-Ahead Packet Processing (LAPP) concept allows protocol analysis to begin before end of receive frame**
- **Integrated Manchester Encoder/Decoder**
- **Provides integrated Attachment Unit Interface (AUI) and 10BASE-T transceiver with automatic port selection**
- **Automatic Twisted-Pair receive polarity detection and automatic correction of the receive polarity**
- **Optional byte padding to long-word boundary on receive**
- **Dynamic transmit FCS generation programmable on a frame-by-frame basis**
- **Internal/external loopback capabilities**
- **Supports the following types of network interfaces:**
  - — AUI to external 10BASE2, 10BASE5,10BASE-T or 10BASE-F MAU
  - — Internal 10BASE-T transceiver with Smart Squelch to Twisted-Pair medium
- **NAND Tree test mode for connectivity testing on printed circuit boards**
- **132-pin PQFP package**

GENERAL DESCRIPTION

The PCnet-PCI single-chip 32-bit Ethernet controller is a highly integrated Ethernet system solution designed to address high-performance system application requirements. It is a flexible bus-mastering device that can be used in any application, including network-ready PCs, printers, fax modems, and bridge/router designs. The bus-master architecture provides high data throughput in the system and low CPU and system bus utilization. The PCnet-PCI controller is fabricated with AMD's advanced low-power CMOS process to provide low operating and standby current for power sensitive applications.

The PCnet-PCI controller is a complete Ethernet node integrated into a single VLSI device. It contains a bus interface unit, a DMA buffer management unit, an IEEE 802.3-defined Media Access Control (MAC) function, individual 136-byte transmit and 128-byte receive FIFOs, an IEEE 802.3-defined Attachment Unit Interface (AUI) and Twisted-Pair Transceiver Media Attachment Unit (10BASE-T MAU), and a Microwire EEPROM interface. The PCnet-PCI controller is also register compatible with the LANCE (Am7990) Ethernet controller, the C-LANCE (Am79C90) Ethernet controller, the ILACC (Am79C900) Ethernet controller, and all Ethernet

Publication# 18220 Rev. C Amendment/0
Issue Date: June 1994

controllers in the PCnet Family, including the PCnet-ISA controller (Am79C960), the PCnet-ISA+ controller (Am79C961), and the PCnet-32 controller (Am79C965). The buffer management unit supports the LANCE, ILACC, and PCnet descriptor software models. The PCnet-PCI controller is software compatible with the Novell NE2100 and NE1500 Ethernet adapter card architectures. In addition, a Sleep function has been incorporated to provide low standby current, excellent for notebooks and Green PCs.

The 32-bit multiplexed bus interface unit provides a direct interface to the PCI local bus applications, simplifying the design of an Ethernet node in a PC system. With its built-in support for both little and big endian byte alignment, this controller also addresses proprietary non-PC applications.

The PCnet-PCI controller supports auto configuration in the PCI configuration space. Additional PCnet-PCI controller configuration parameters, including the unique IEEE physical address, can be read from an external non-volatile memory (serial EEPROM) immediately following system RESET.

The controller also has the capability to automatically select either the AUI port or the Twisted-Pair transceiver. Only one interface is active at any one time. The individual transmit and receive FIFOs optimize system overhead, providing sufficient latency during frame transmission and reception, and minimizing intervention during normal network error recovery. The integrated Manchester encoder/decoder (MENDEC) eliminates the need for an external Serial Interface Adapter (SIA) in the system. In addition, the device provides programmable on-chip LED drivers for transmit, receive, collision, receive polarity, link integrity or jabber status.

BLOCK DIAGRAM

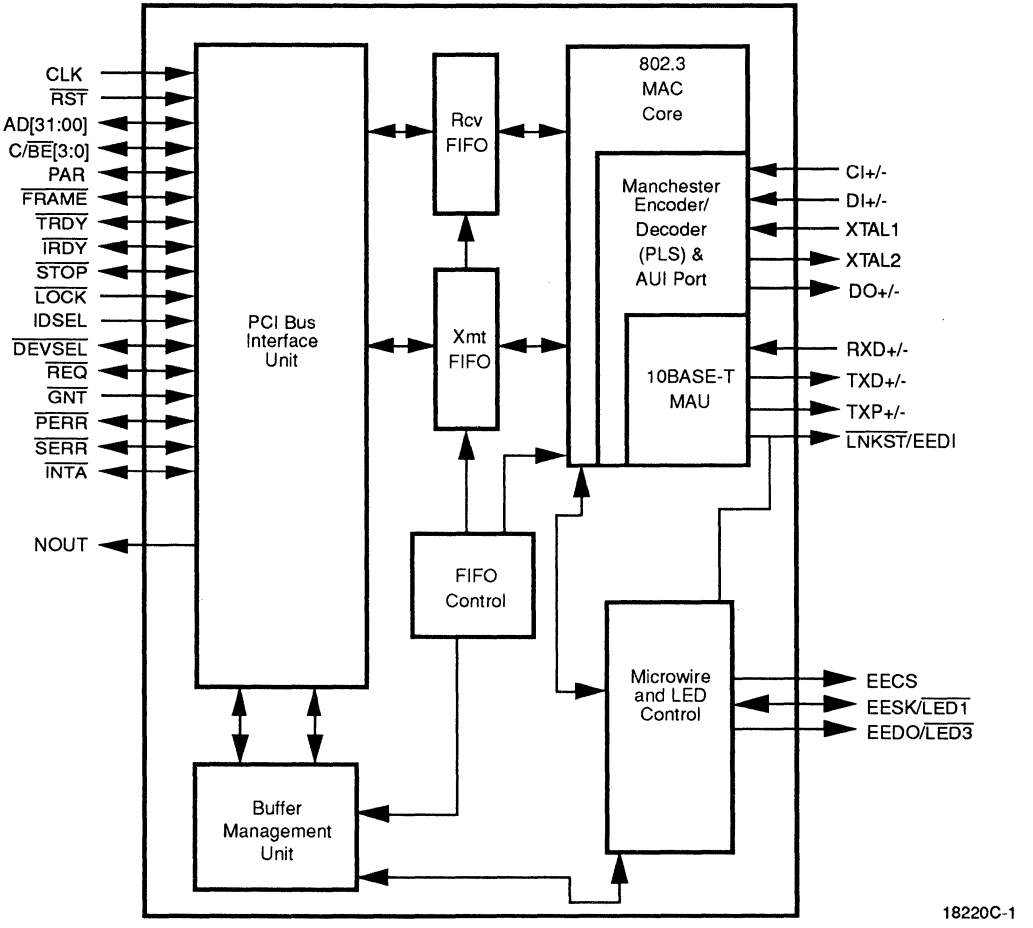
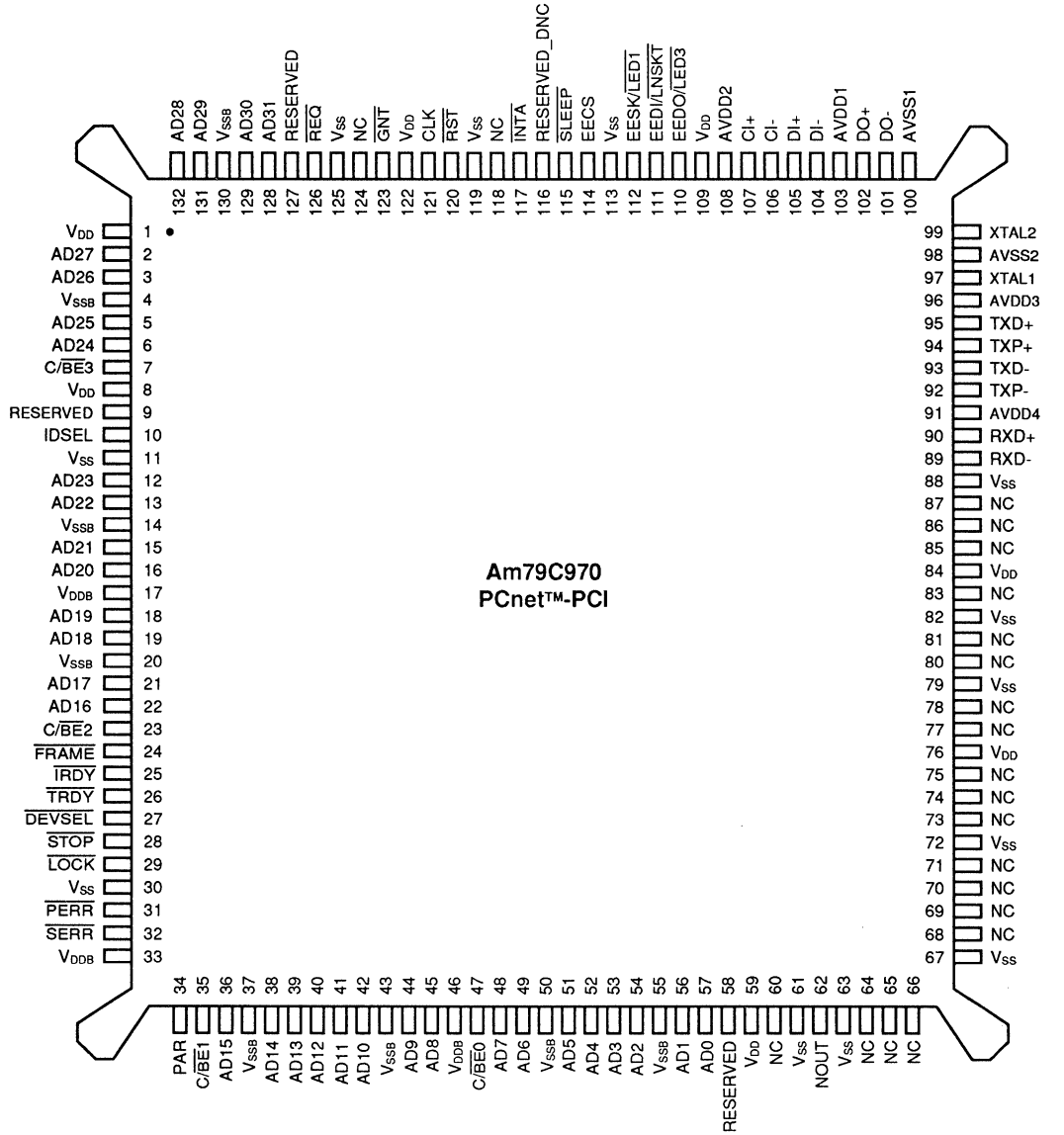


TABLE OF CONTENTS

RELATED PRODUCTS

Part No.	Description
Am79C98	Twisted-Pair Ethernet Transceiver (TPEX)
Am79C100	Twisted-Pair Ethernet Transceiver Plus (TPEX+)
Am7996	IEEE 802.3/Ethernet/Cheapernet Tap Transceiver
Am79C981	Integrated Multiport Repeater Plus™ (IMR+™)
Am79C987	Hardware Implemented Management Information Base™ (HIMIB™)
Am79C940	Media Access Controller for Ethernet (MACE™)
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am79C90	CMOS Local Area Network Controller for Ethernet (C-LANCE)
Am79C900	Integrated Local Area Communications Controller™ (ILACC™)
Am79C960	PCnet-ISA Single-Chip Ethernet Controller (for ISA bus)
Am79C961	PCnet-ISA+ Single-Chip Ethernet Controller (with Microsoft® Plug n' Play support)
Am79C965	PCnet-32 Single-Chip 32-Bit Ethernet Controller (for 486 and VL buses)
Am79C974	PCnet-SCSI Combination Ethernet and SCSI Controller for PCI Systems

CONNECTION DIAGRAM



Pin 1 is marked for orientation.

18220C-2

NC = No Connection, reserved for future use.

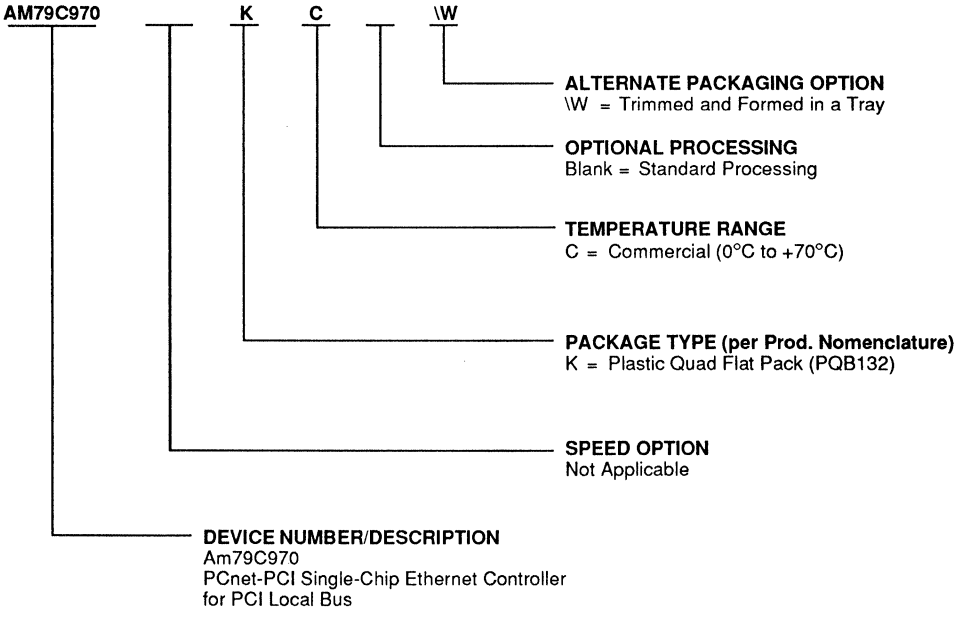
RESERVED = Internally bonded, for AMD internal test only; should not be connected.

RESERVED_DNC = Reserved, don't connect.

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:



Valid Combinations	
AM79C970	KC, KC\W

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

PIN DESIGNATIONS

Listed by Pin Number

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	V_{DDB}	34	PAR	67	V_{SS}	100	AVSS1
2	AD27	35	C/$\overline{BE}$1	68	NC	101	DO–
3	AD26	36	AD15	69	NC	102	DO+
4	V_{SSB}	37	V_{SSB}	70	NC	103	AVDD1
5	AD25	38	AD14	71	NC	104	DI–
6	AD24	39	AD13	72	V_{SS}	105	DI+
7	C/$\overline{BE}$3	40	AD12	73	NC	106	CI–
8	V_{DD}	41	AD11	74	NC	107	CI+
9	RESERVED	42	AD10	75	NC	108	AVDD2
10	IDSEL	43	V_{SSB}	76	V_{DD}	109	V_{DD}
11	V_{SS}	44	AD9	77	NC	110	EEDO/$\overline{LED3}$
12	AD23	45	AD8	78	NC	111	EEDI/$\overline{LNKST}$
13	AD22	46	V_{DDB}	79	V_{SS}	112	EESK/$\overline{LED1}$
14	V_{SSB}	47	C/$\overline{BE}$0	80	NC	113	V_{SS}
15	AD21	48	AD7	81	NC	114	EECS
16	AD20	49	AD6	82	V_{SS}	115	$\overline{SLEEP}$
17	V_{DDB}	50	V_{SSB}	83	NC	116	RESERVED_DNC
18	AD19	51	AD5	84	V_{DD}	117	$\overline{INTA}$
19	AD18	52	AD4	85	NC	118	NC
20	V_{SSB}	53	AD3	86	NC	119	V_{SS}
21	AD17	54	AD2	87	NC	120	$\overline{RST}$
22	AD16	55	V_{SSB}	88	V_{SS}	121	CLK
23	C/$\overline{BE}$2	56	AD1	89	RXD–	122	V_{DD}
24	$\overline{FRAME}$	57	AD0	90	RXD+	123	$\overline{GNT}$
25	$\overline{IRDY}$	58	RESERVED	91	AVDD4	124	RESERVED
26	$\overline{TRDY}$	59	V_{DD}	92	TXP–	125	V_{SS}
27	$\overline{DEVSEL}$	60	NC	93	TXD–	126	$\overline{REQ}$
28	$\overline{STOP}$	61	V_{SS}	94	TXP+	127	RESERVED
29	$\overline{LOCK}$	62	NOUT	95	TXD+	128	AD31
30	V_{SS}	63	V_{SS}	96	AVDD3	129	AD30
31	$\overline{PERR}$	64	NC	97	XTAL1	130	V_{SSB}
32	$\overline{SERR}$	65	NC	98	AVSS2	131	AD29
33	V_{DDB}	66	NC	99	XTAL2	132	AD28

PIN DESIGNATIONS

Listed by Group

Pin Name	Pin Function	Type	Driver	# Pins
PCI Bus Interface				
AD[31:00]	Address/Data Bus	IO	TS3	32
C/$\overline{BE}$[3:0]	Bus Command/Byte Enable	IO	TS3	4
CLK	Bus Clock	I	NA	1
$\overline{DEVSEL}$	Device Select	IO	TS6	1
$\overline{FRAME}$	Cycle Frame	IO	TS6	1
$\overline{GNT}$	Bus Grant	I	NA	1
IDSEL	Initialization Device Select	I	ns	1
$\overline{INTA}$	Interrupt	IO	OD6	1
$\overline{IRDY}$	Initiator Ready	IO	TS6	1
$\overline{LOCK}$	Bus Lock	I	NA	1
PAR	Parity	IO	TS6	1
$\overline{PERR}$	Parity Error	IO	TS6	1
$\overline{REQ}$	Bus Request	IO	TS3	1
$\overline{RST}$	Reset	I	NA	1
$\overline{SERR}$	System Error	IO	OD6	1
$\overline{STOP}$	Stop	IO	TS6	1
$\overline{TRDY}$	Target Ready	IO	TS6	1
Board Interface				
EECS	Microwire Serial PROM Chip Select	O	O8	1
EEDI/$\overline{LNKST}$	Microwire Serial EEPROM Data In/Link Status	O	LED	1
EEDO/$\overline{LED3}$	Microwire APROM Data Out/LED predriver	IO	LED	1
EESK/$\overline{LED1}$	Microwire Serial PROM Clock/LED1	IO	LED	1
SLEEP	Sleep Mode	I	NA	1
XTAL1–2	Crystal Input/Output	IO	NA	2
Attachment Unit Interface (AUI)				
CI+/CI–	AUI Collision Differential Pair	I	NA	2
DI+/DI–	AUI Data In Differential Pair	I	NA	2
DO+/DO–	AUI Data Out Differential Pair	O	DO	2
10BASE-T Interface				
RXD+/RXD–	Receive Differential Pair	I	NA	2
TXD+/TXD–	Transmit Differential Pair	O	TDO	2
TXP+/TXP–	Transmit Pre-distortion Differential Pair	O	TPO	2
$\overline{LNKST}$/EEDI	Link Status/Microwire Serial EEPROM Data In	O	LED	1
Test Interface				
NOUT	NAND Tree Test Output	O	O3	1
Power Supplies				
AVDD	Analog Power	P	NA	4
AVSS	Analog Ground	P	NA	2
V_{DD}	Digital Power	P	NA	6
V_{SS}	Digital Ground	P	NA	12
V_{DDB}	I/O Buffer Power	P	NA	4
V_{SSB}	I/O Buffer Ground	P	NA	8

PIN DESIGNATIONS

Listed by Driver Type

The next table describes the various types of drivers that are implemented in the PCnet-PCI controller. Current is given as milliamperes:

Name	Type	I_{OL} (mA)	I_{OH} (mA)	pF
TS3	Tri-State™	3	–2	50
TS6	Tri-State	6	–2	50
O3	Totem Pole	3	–0.4	50
O8	Totem Pole	8	–0.4	50
OD6	Open Drain	6	NA	50
LED	LED	12	–0.4	50

PIN DESCRIPTION

PCI Interface

AD[31:00]
Address and Data Input/Output

These signals are multiplexed on the same PCI pins. During the first clock of a transaction AD[31:00] contain the physical byte address (32 bits). During the subsequent clocks AD[31:00] contain data. Byte ordering is little endian by default. AD[07:00] are defined as least significant byte and AD[31:24] are defined as the most significant byte. For FIFO data transfers, the PCnet-PCI controller can be programmed for big endian byte ordering. See CSR3, bit 2 (BSWP) for more details.

During the address phase of the transaction, when the PCnet-PCI controller is a bus master, AD[31:2] will address the active DWORD (double-word). The PCnet-PCI controller always drives AD[1:0] to '00' during the address phase indicating linear burst order. When the PCnet-PCI controller is not a bus master, the AD[31:00] lines are continuously monitored to determine if an address match exists for I/O slave transfers.

During the data phase of the transaction, AD[31:00] are driven by the PCnet-PCI controller when performing bus master writes and slave read operations. Data on AD[31:00] is latched by the PCnet-PCI controller when performing bus master reads and slave write operations.

When $\overline{RST}$ is active, AD[31:0] are inputs for NAND tree testing.

C/$\overline{BE}$[3:0]
Bus Command and Byte Enables
Input/Output

These signals are multiplexed on the same PCI pins. During the address phase of the transaction, C/$\overline{BE}$[3:0] define the bus command. During the data phase C/$\overline{BE}$[3:0] are used as Byte Enables. The Byte Enables define which physical byte lanes carry meaningful data. C/$\overline{BE}$0 applies to byte 0 (AD[07:00]) and C/$\overline{BE}$3 applies to byte 3 (AD[31:24]). The function of the Byte Enables is independent of the byte ordering mode (CSR3, bit 2).

When $\overline{RST}$ is active, C/$\overline{BE}$[3:0] are inputs for NAND tree testing.

CLK
Clock
Input

This signal provides timing for all the transactions on the PCI bus and all PCI devices on the bus including the PCnet-PCI controller. All bus signals are sampled on the rising edge of CLK and all parameters are defined with respect to this edge. The PCnet-PCI controller operates over a range of 0 to 33 MHz.

When $\overline{RST}$ is active, CLK is an input for NAND tree testing.

$\overline{DEVSEL}$
Device Select
Input/Output

This signal when actively driven by the PCnet-PCI controller as a slave device signals to the master device that the PCnet-PCI controller has decoded its address as the target of the current access. As an input it indicates whether any device on the bus has been selected.

When $\overline{RST}$ is active, $\overline{DEVSEL}$ is an input for NAND tree testing.

$\overline{FRAME}$
Cycle Frame
Input/Output

This signal is driven by the PCnet-PCI controller when it is the bus master to indicate the beginning and duration of the access. $\overline{FRAME}$ is asserted to indicate a bus transaction is beginning. $\overline{FRAME}$ is asserted while data transfers continue. $\overline{FRAME}$ is deasserted when the transaction is in the final data phase.

When $\overline{RST}$ is active, $\overline{FRAME}$ is an input for NAND tree testing.

$\overline{GNT}$
Bus Grant
Input

This signal indicates that the access to the bus has been granted to the PCnet-PCI controller.

The PCnet-PCI controller supports bus parking. When the PCI bus is idle and the system arbiter asserts $\overline{GNT}$ without an active $\overline{REQ}$ from the PCnet-PCI controller, the PCnet-PCI controller will actively drive the AD, C/$\overline{BE}$ and PAR lines.

When $\overline{RST}$ is active, $\overline{GNT}$ is an input for NAND tree testing.

IDSEL
Initialization Device Select
Input

This signal is used as a chip select for the PCnet-PCI controller during configuration read and write transaction.

When $\overline{RST}$ is active, IDSEL is an input for NAND tree testing.

$\overline{\text{INTA}}$
Interrupt Request
Input/Output

An asynchronous attention signal which indicates that one or more of the following status flags is set: BABL, MISS, MERR, RINT, IDON, RCVCCO, RPCO, JAB, MPCO, or TXSTRT. Each status flag has a mask bit which allows for suppression of $\overline{\text{INTA}}$ assertion. The flags have the following meaning:

BABL	Babble
RCVCCO	Receive Collision Count Overflow
RPCO	Runt Packet Count Overflow
JAB	Jabber
MISS	Missed Frame
MERR	Memory Error
MPCO	Missed Packet Count Overflow
RINT	Receive Interrupt
IDON	Initialization Done
TXSTRT	Transmit Start

When $\overline{\text{RST}}$ is active, $\overline{\text{INTA}}$ is an input for NAND tree testing.

$\overline{\text{IRDY}}$
Initiator Ready
Input/Output

This signal indicates PCnet-PCI controllers ability, as a master device, to complete the current data phase of the transaction. $\overline{\text{IRDY}}$ is used in conjunction with the $\overline{\text{TRDY}}$. A data phase is completed on any clock when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. During a write $\overline{\text{IRDY}}$ indicates that valid data is present on AD[31:00]. During a read $\overline{\text{IRDY}}$ indicates that data is accepted by the PCnet-PCI controller as a bus master. Wait states are inserted until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted simultaneously.

When $\overline{\text{RST}}$ is active, $\overline{\text{IRDY}}$ is an input for NAND tree testing.

$\overline{\text{LOCK}}$
Lock
Input

$\overline{\text{LOCK}}$ is used by the current bus master to indicate an atomic operation that may require multiple transfers.

As a slave device, the PCnet-PCI controller can be locked by any master device. When another master attempts to access the PCnet-PCI while it is locked, the PCnet-PCI controller will respond by asserting $\overline{\text{DEVSEL}}$ and $\overline{\text{STOP}}$ with $\overline{\text{TRDY}}$ deasserted (PCI retry).

The PCnet-PCI controller will never assert $\overline{\text{LOCK}}$ as a master.

When $\overline{\text{RST}}$ is active, $\overline{\text{LOCK}}$ is an input for NAND tree testing.

PAR
Parity
Input/Output

Parity is even parity across AD[31:00] and C/$\overline{\text{BE}}$[3:0]. When the PCnet-PCI controller is a bus master, it generates parity during the address and write data phases. It checks parity during read data phases. When the PCnet-PCI controller operates in slave mode and is the target of the current cycle, it generates parity during read data phases. It checks parity during address and write data phases.

When $\overline{\text{RST}}$ is active, PAR is an input for NAND tree testing.

$\overline{\text{PERR}}$
Parity Error
Input/Output

This signal is asserted by the PCnet-PCI controller when it checks for parity error during any data phase when its AD[31:00] lines are inputs. The $\overline{\text{PERR}}$ pin is only active when PERREN (bit 6) in the PCI command register is set.

The PCnet-PCI controller monitors the $\overline{\text{PERR}}$ input during a bus master write cycle. It will assert the Data Parity Reported bit in the Status register of the Configuration Space when a parity error is reported by the target device.

When $\overline{\text{RST}}$ is active, $\overline{\text{PERR}}$ is an input for NAND tree testing.

$\overline{\text{REQ}}$
Bus Request
Input/Output

The PCnet-PCI controller asserts $\overline{\text{REQ}}$ pin as a signal that it wishes to become a bus master. Once asserted, $\overline{\text{REQ}}$ remains active until $\overline{\text{GNT}}$ has become active, independent of subsequent assertion of $\overline{\text{SLEEP}}$ or setting of the STOP bit or access to the S_RESET port (offset 14h).

When $\overline{\text{RST}}$ is active, $\overline{\text{REQ}}$ is an input for NAND tree testing.

$\overline{\text{RST}}$
Reset
Input

When $\overline{\text{RST}}$ is asserted low, then the PCnet-PCI controller performs an internal system reset of the type H_RESET (HARDWARE_RESET). $\overline{\text{RST}}$ must be held for a minimum of 30 CLK periods. While in the H_RESET state, the PCnet-PCI controller will disable or deassert all outputs. $\overline{\text{RST}}$ may be asynchronous to the CLK when asserted or deasserted. It is recommended that the deassertion be synchronous to guarantee clean and bounce free edge.

When RST is active, NAND tree testing is enabled. All PCI interface pins are in input mode. The result of the NAND tree testing can be observed on the NOUT output (pin 62).

SERR
System Error
Input/Output

This signal is asserted for one CLK by the PCnet-PCI controller when it detects a parity error during the address phase when its AD[31:00] lines are inputs.

The SERR pin is only active when SERREN (bit 8) and PERREN (bit 6) in the PCI command register are set.

When RST is active, SERR is an input for NAND tree testing.

STOP
Stop
Input/Output

In the slave role, the PCnet-PCI controller drives the STOP signal to inform the bus master to stop the current transaction. In the bus master role, the PCnet-PCI controller receives the STOP signal and stops the current transaction.

When RST is active, STOP is an input for NAND tree testing.

TRDY
Target Ready
Input/Output

This signal indicates that the PCnet-PCI controllers ability as a selected device to complete the current data phase of the transaction. TRDY is used in conjunction with the IRDY. A data phase is completed on any clock both TRDY and IRDY are asserted. During a read TRDY indicates that valid data is present on AD[31:00]. During a write, TRDY indicates that data has been accepted. Wait states are inserted until both IRDY and TRDY are asserted simultaneously.

When RST is active, TRDY is an input for NAND tree testing.

Board Interface

LED1
LED1
Output

This pin is shared with the EESK function. As LED1, the function and polarity of this pin are programmable through BCR5. By default, LED1 is active LOW and it indicates receive activity on the network. The LED1 output from the PCnet-PCI controller is capable of sinking the necessary 12 mA of current to drive an LED directly.

The LED1 pin is also used during EEPROM Auto-detection to determine whether or not an EEPROM is present at the PCnet-PCI controller Microwire interface. At the trailing edge of the RST pin, LED1 is sampled to determine the value of the EEDET bit in BCR19. A sampled HIGH value means that an EEPROM is present, and EEDET will be set to ONE. A sampled LOW value means that an EEPROM is not present, and EEDET will be set to ZERO. See the EEPROM Auto-detection section for more details.

If no LED circuit is to be attached to this pin, then a pull up or pull down resistor must be attached instead, in order to resolve the EEDET setting.

LED3
LED3
Output

This pin is shared with the EEDO function. When functioning as LED3, the signal on this pin is programmable through BCR7. By default, LED3 is active LOW and it indicates transmit activity on the network. Special attention must be given to the external circuitry attached to this pin. If an LED circuit were directly attached to this pin, it would create an IOL requirement that could not be met by the serial EEPROM that would also be attached to this pin. (This pin is multifunctioned with the EEDO function of the Microwire serial EEPROM interface.)

Therefore, if this pin is to be used as an additional LED output while an EEPROM is used in the system, then buffering is required between the LED3 pin and the LED circuit. If no EEPROM is included in the system design, then the LED3 signal may be directly connected to an LED without buffering. The LED3 output from the PCnet-PCI controller is capable of sinking the necessary 12 mA of current to drive an LED in this case. For more details regarding LED connection, see the section on LEDs.

LNKST
LINK Status
Output

This pin provides 12 mA for driving an LED. By default, it indicates an active link connection on the 10BASE-T interface. This pin can also be programmed to indicate other network status (see BCR4). The LNKST pin polarity is programmable, but by default, it is active LOW. Note that this pin is multiplexed with the EEDI function.

SLEEP
Sleep
Input

When SLEEP is asserted (active LOW), the PCnet-PCI controller performs an internal system reset of the S_RESET type and then proceeds into a power savings mode. (The reset operation caused by SLEEP assertion will not affect BCR registers.) The PCI interface section is not effected by SLEEP. In particular, access to the PCI configuration space remains possible. None of the

configuration registers will be reset by SLEEP. All I/O accesses to the PCnet-PCI controller will result in a PCI target abort response. The PCnet-PCI controller will not assert REQ while in sleep mode. When SLEEP is asserted, all non-PCI interface outputs will be placed in their normal S_RESET condition. All non-PCI interface inputs will be ignored except for the SLEEP pin itself. De-assertion of SLEEP results in wake-up. The system must refrain from starting the network operations of the PCnet-PCI device for 0.5 seconds following the deassertion of the SLEEP signal in order to allow internal analog circuits to stabilize.

Both CLK and XTAL1 inputs must have valid clock signals present in order for the SLEEP command to take effect. If SLEEP is asserted while REQ is asserted, then the PCnet-PCI controller will wait for the assertion of GNT. When GNT is asserted, the REQ signal will be deasserted and then the PCnet-PCI controller will proceed to the power savings mode.

The SLEEP pin should not be asserted during power supply ramp-up. If it is desired that SLEEP be asserted at power up time, then the system must delay the assertion of SLEEP until three CLK cycles after the completion of a valid pin RST operation.

$XTAL_{1-2}$
Crystal Oscillator Inputs
Input/Output

The crystal frequency determines the network data rate. The PCnet-PCI controller supports the use of quartz crystals to generate a 20 MHz frequency compatible with the ISO 8802-3 (IEEE/ANSI 802.3) network frequency tolerance and jitter specifications. See the section External Crystal Characteristics (in section Manchester Encoder/Decoder) for more detail.

The network data rate is one-half of the crystal frequency. XTAL1 may alternatively be driven using an external CMOS level source, in which case XTAL2 must be left unconnected. Note that when the PCnet-PCI controller is in comma mode, there is an internal 22 KΩ resistor from XTAL1 to ground. If an external source drives XTAL1, some power will be consumed driving this resistor. If XTAL1 is driven LOW at this time power consumption will be minimized. In this case, XTAL1 must remain active for at least 30 cycles after the assertion of SLEEP and deassertion of REQ.

Microwire EEPROM Interface

EESK
EEPROM Serial clock
Input/Output

The EESK signal is used to access the external ISO 8802-3 (IEEE/ANSI 802.3) address PROM. This pin is designed to directly interface to a serial EEPROM that uses the Microwire interface protocol. EESK is connected to the Microwire EEPROMs Clock pin. It is controlled by either the PCnet-PCI controller directly during a read of the entire EEPROM, or indirectly by the host system by writing to BCR19, bit 1.

The EESK pin is also used during EEPROM Auto-detection to determine whether or not an EEPROM is present at the PCnet-PCI controller Microwire interface. At the trailing edge of the RST signal, LED1 is sampled to determine the value of the EEDET bit in BCR19. A sampled HIGH value means that an EEPROM is present, and EEDET will be set to ONE. A sampled LOW value means that an EEPROM is not present, and EEDET will be set to ZERO. See the EEPROM Auto-detection section for more details.

EESK is shared with the LED1 function. If no LED circuit is to be attached to this pin, then a pull up or pull down resistor must be attached instead, in order to resolve the EEDET setting.

EEDO
EEPROM Data Out
Input

The EEDO signal is used to access the external ISO 8802-3 (IEEE/ANSI 802.3) address PROM. This pin is designed to directly interface to a serial EEPROM that uses the Microwire interface protocol. EEDO is connected to the Microwire EEPROMs Data Output pin. It is controlled by the EEPROM during reads. It may be read by the host system by reading BCR19 bit 0.

EEDO is shared with the LED3 function.

EECS
EEPROM Chip Select
Output

The function of the EECS signal is to indicate to the Microwire EEPROM device that it is being accessed. The EECS signal is active high. It is controlled by either the PCnet-PCI controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19 bit 2.

EEDI
EEPROM Data In
Output

The EEDI signal is used to access the external ISO 8802-3 (IEEE/ANSI 802.3) address PROM. EEDI functions as an output. This pin is designed to directly interface to a serial EEPROM that uses the Microwire interface protocol. EEDI is connected to the Microwire EEPROMs Data Input pin. It is controlled by either the PCnet-PCI controller during command portions of a read of the entire EEPROM, or indirectly by the host system by writing to BCR19 bit 0.

EEDI is shared with the LNKST function.

Attachment Unit Interface

CI±
Collision In
Input

A differential input pair signaling the PCnet-PCI controller that a collision has been detected on the network media, indicated by the CI± inputs being driven with a 10 MHz pattern of sufficient amplitude and pulse width to meet ISO 8802-3 (IEEE/ANSI 802.3) standards. Operates at pseudo ECL levels.

DI±
Data In
Input

A differential input pair to the PCnet-PCI controller carrying Manchester encoded data from the network. Operates at pseudo ECL levels.

DO±
Data Out
Output

A differential output pair from the PCnet-PCI controller for transmitting Manchester encoded data to the network. Operates at pseudo ECL levels.

Twisted-Pair Interface

RXD±
10BASE-T Receive Data
Input

10BASE-T port differential receivers.

TXD±
10BASE-T Transmit Data
Output

10BASE-T port differential drivers.

TXP±
10BASE-T Pre-Distortion Control
Output

These outputs provide transmit pre-distortion control in conjunction with the 10BASE-T port differential drivers.

Test Interface

NOUT
NAND Tree Out
Output

The results of the NAND tree testing can be observed on the NOUT pin. NOUT will be constantly high, when $\overline{RST}$ is deasserted.

Power Supply Pins

Analog Power Supply Pins

AV_{DD}
Analog Power (4 Pins)
Power

There are four analog +5 V supply pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to Appendix B and the *PCnet Family Technical Manual* (PID #18216A) for details.

AV_{SS}
Analog Ground (2 Pins)
Power

There are two analog ground pins. Special attention should be paid to the printed circuit board layout to avoid excessive noise on these lines. Refer to Appendix B and the *PCnet Family Technical Manual* (PID #18216A) for details.

Digital Power Supply Pins

V_{DD}
Digital Power (6 Pins)
Power

There are 6 power supply pins that are used by the internal digital circuitry. All V_{DD} pins must be connected to a +5 V supply.

V_{DDB}
I/O Buffer Power (4 Pins)
Power

There are 4 power supply pins that are used by the PCI bus Input/Output buffer drivers. All V_{DDB} pins must be connected to a +5 V supply.

V_{SS}
Digital Ground (12 Pins)
Ground

There are 12 ground pins that are used by the internal digital circuitry.

V_{SSB}
I/O Buffer Ground (8 Pins)
Ground

There are 8 ground pins that are used by the PCI bus Input/Output buffer drivers.

BASIC FUNCTIONS

System Bus Interface Function

The PCnet-PCI controller is designed to operate as a Bus Master during normal operations. Some slave I/O accesses to the PCnet-PCI controller are required in normal operations as well. Initialization of the PCnet-PCI controller is achieved through a combination of PCI Configuration Space accesses, Bus Slave accesses, Bus Master accesses and an optional read of a serial EEPROM that is performed by the PCnet-PCI controller. The EEPROM read operation is performed through the Microwire interface. The ISO 8802-3 (IEEE/ANSI 802.3) Ethernet Address may reside within the serial EEPROM. Some PCnet-PCI controller configuration registers may also be programmed by the EEPROM read operation.

The APROM, on-chip board-configuration registers, and the Ethernet controller registers occupy 32-bytes of I/O space which can be located on a wide variety of starting addresses by modifying the Base Address Register in the PCI Configuration Space.

Software Interface

The software interface to the PCnet-PCI controller is divided into three parts. One part is the PCI configuration registers. They are used to identify the PCnet-PCI controller, and are also used to setup the configuration of the device. The setup information includes the I/O base address and the routing of the PCnet-PCI controller interrupt channel. This allows for a jumperless implementation.

The second portion of the software interface is the direct access to the I/O resources of the PCnet-PCI controller. The PCnet-PCI controller occupies 32-bytes of I/O space that must begin on a 32-byte block boundary. The I/O base address can be changed to any 32-bit quantity that begins on a 32-bit block boundary by modifying the Base Address Register in the PCI Configuration Space. The 32-byte I/O space is used by the software to program the PCnet-PCI controller operating mode, to enable and disable various features, to monitor operating status and to request particular functions to be executed by the PCnet-PCI controller.

The third portion of the software interface is the descriptor and buffer areas that are shared between the software and the PCnet-PCI controller during normal network operations. The descriptor area boundaries are set by the software and do not change during normal network operations. There is one descriptor area for receive activity and there is a separate area for transmit activity. The descriptor space contains relocatable pointers to the network packet data and it is used to transfer packet status from the PCnet-PCI controller to the software. The buffer areas are locations that hold packet data for transmission or that accept packet data that has been received.

Network Interfaces

The PCnet-PCI controller can be connected to an 802.3 network via one of two network interfaces. The Attachment Unit Interface (AUI) provides an ISO 8802-3 (IEEE/ANSI 802.3) compliant differential interface to a remote MAU or an on-board transceiver. The 10BASE-T interface provides a twisted-pair Ethernet port. While in auto-selection mode, the interface in use is determined by an auto-sensing mechanism which checks the link status on the 10BASE-T port. If there is no active link status, then the device assumes an AUI connection.

DETAILED FUNCTIONS

Bus Interface Unit (BIU)

The bus interface unit is built of several state machines that run synchronously to CLK. One bus interface unit state machine handles accesses where the PCnet-PCI controller is the bus slave, and another handles accesses where the PCnet-PCI controller is the bus master. All inputs are synchronously sampled. All outputs are synchronously generated on the rising edge of CLK.

Bus Acquisition

The PCnet-PCI microcode (in the buffer management section) will determine when a DMA transfer should be initiated. The first step in any PCnet-PCI bus master transfer is to acquire ownership of the bus. This task is handled by synchronous logic within the BIU. Bus ownership is requested with the $\overline{REQ}$ signal and ownership is granted by the arbiter through the $\overline{GNT}$ signal.

Figure 1 shows the PCnet-PCI controller bus acquisition. $\overline{GNT}$ is asserted at clock 3. The PCnet-PCI controller starts driving AD[31:00] and C/$\overline{BE}$[3:0] prior to clock 4. $\overline{FRAME}$ is asserted at clock 5 indicating a valid address and command on AD[31:00] and C/$\overline{BE}$[3:0]. ADSTEP (bit 7) in the PCI Command register is set to ONE to indicated that the PCnet-PCI controller uses address stepping. Address stepping in only used for the first address phase of a bus master period.

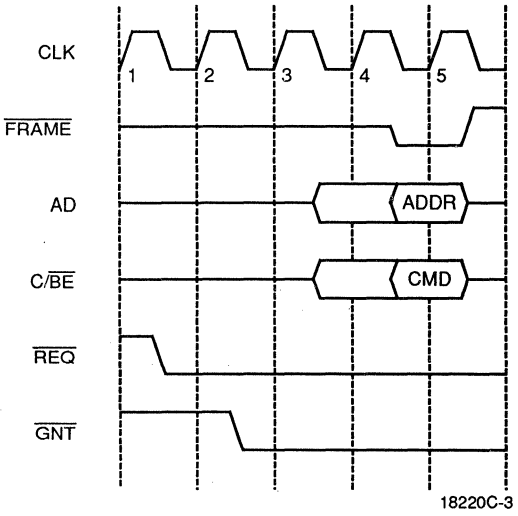


Figure 1. Bus Acquisition

Note that assertion of the STOP bit in CSR0 will not cause a deassertion of the $\overline{REQ}$ signal. Note also that a read of the RESET register, (I/O resource at offset 14h from the PCnet-PCI I/O base address) will not cause a deassertion of the $\overline{REQ}$ signal. Either of these actions will cause the internal master state machine logic to cease operations, but the $\overline{REQ}$ signal will remain active until the $\overline{GNT}$ signal is asserted. Following either of the above actions, on the next clock cycle after the $\overline{GNT}$ signal is asserted, the PCnet-PCI controller will deassert the $\overline{REQ}$ signal.

Assertion of a minimum-width pulse on the $\overline{RST}$ pin will cause the $\overline{REQ}$ signal to deassert immediately following the assertion of the $\overline{RST}$ pin. In this case, the PCnet-PCI controller will not wait for the assertion of the $\overline{GNT}$ signal before deasserting the $\overline{REQ}$ signal.

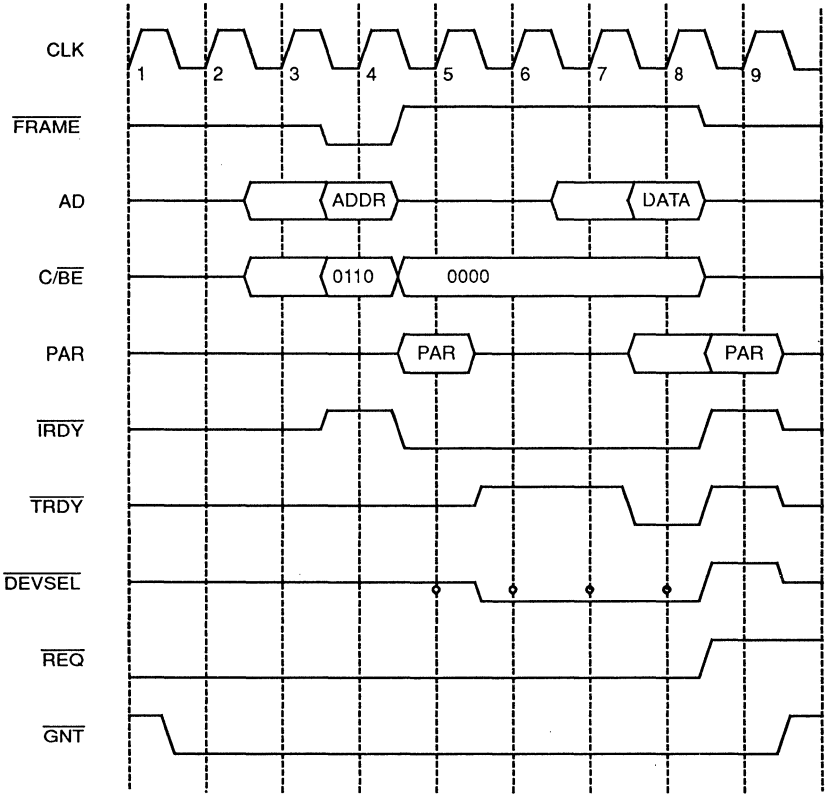
Bus Master DMA Transfers

There are four primary types of DMA transfers. The PCnet-PCI controller uses non-burst as well as burst cycles for read and write access to the main memory.

Basic Non-Burst Read Cycles

The PCnet-PCI controller uses non-burst read cycles to access the initialization block and the receive and transmit descriptor entries. Some of the read accesses to the transmit buffer memory are also in non-burst mode. All PCnet-PCI controller non-burst read accesses are of the PCI command type Memory Read (type 6). Note that during all non-burst read operations, the PCnet-PCI controller will always activate all byte enables, even though some byte lanes may not contain valid data as indicated by a buffer pointer value. In such instances, the PCnet-PCI controller will internally discard unneeded bytes.

Figure 2 shows a typical non-burst read access. The PCnet-PCI controller asserts $\overline{IRDY}$ at clock 5 immediately after the address phase and starts sampling $\overline{DEVSEL}$. The target extends the cycle by asserting $\overline{DEVSEL}$ not until clock 6. Additionally, the target inserts one wait state by asserting its ready ($\overline{TRDY}$) at clock 8.

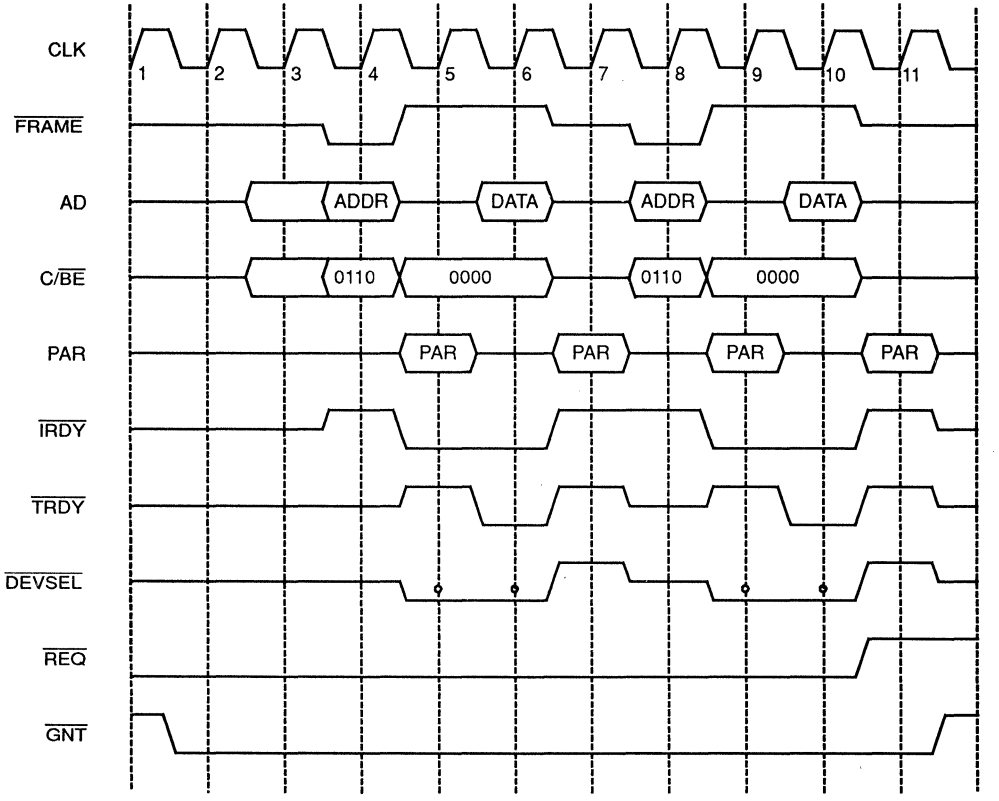


o *DEVSEL is sampled by the PCnet-PCI controller.*

18220C-4

Figure 2. Non-Burst Read Cycle With Wait States

Figure 3 shows two non-burst read access within one arbitration cycle. The PCnet-PCI controller will drop FRAME between two consecutive non-burst read cycles. The PCnet-PCI controller will re-request the bus right again if it is preempted before starting the second access. The example below also shows a target that can respond to the PCnet-PCI controller read cycles without wait states.



o *DEVSEL is sampled by the PCnet-PCI controller.*

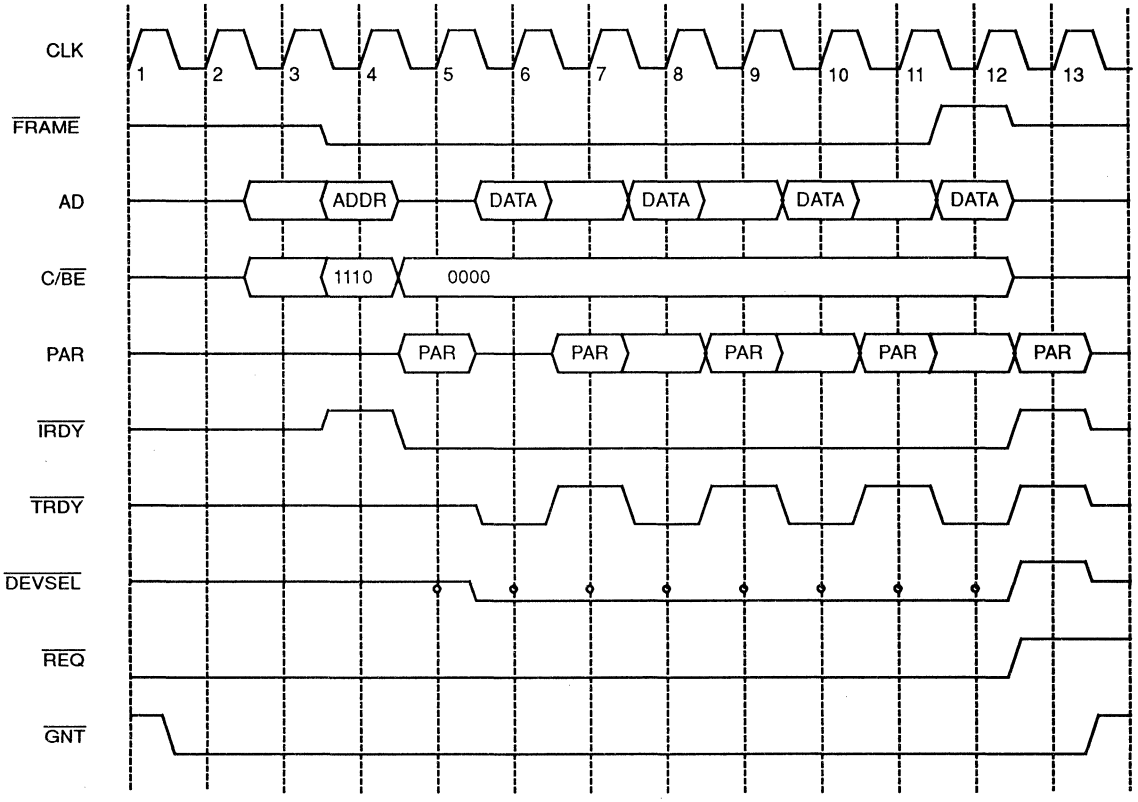
18220C-5

Figure 3. Non-Burst Read Cycles Without Wait States

Basic Burst Read Cycles

The PCnet-PCI controller provides a burst mode to read data from the transmit buffer. The burst mode must be enabled by setting BREADE in BCR18. All PCnet-PCI controller burst read transfers are of the PCI command type Memory Read Line (type15). AD[1:0] will both be ZERO during the address phase indicating a linear burst order. All four byte enable signals will be ZERO during the data phase as the PCnet-PCI controller always reads a full 32-bit word when in burst mode.

Figure 4 shows a typical burst read access. The PCnet-PCI controller arbitrates for the bus, is granted access, and reads four 32-bit words (DWORD) from system memory and then releases the bus. All four data phases in this example take two clock cycles each, which is determined by the timing of $\overline{TRDY}$.



o *$\overline{DEVSEL}$ is sampled by the PCnet-PCI controller.*

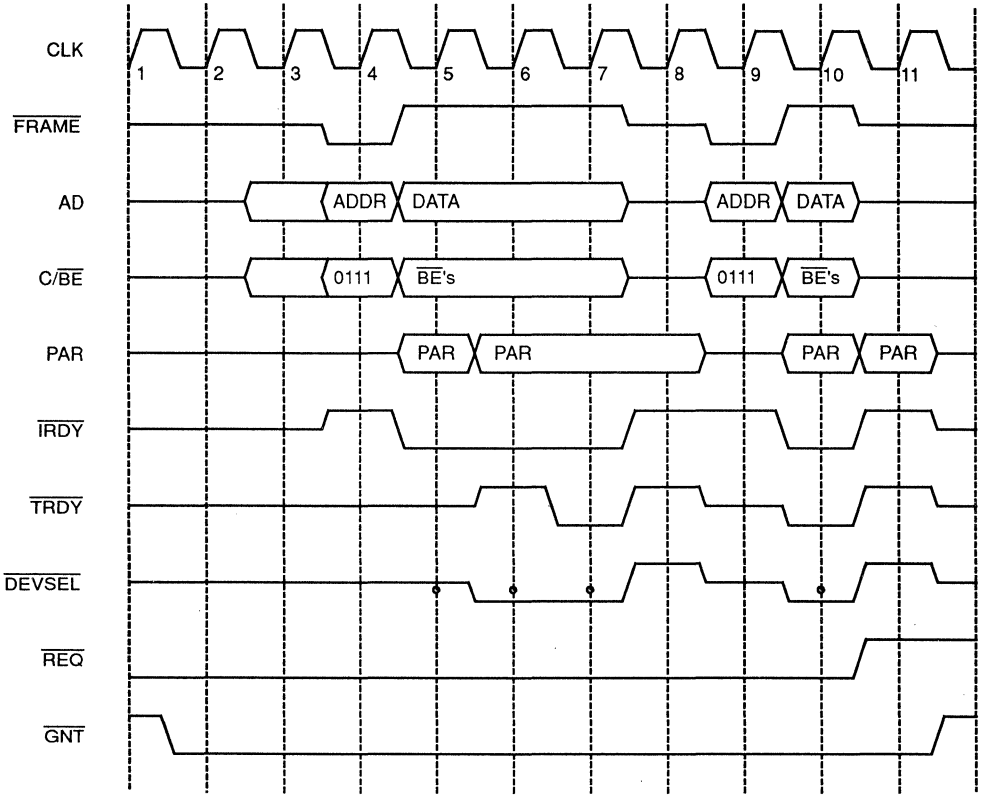
18220C-6

Figure 4. Burst Read Cycles

Basic Non-Burst Write

The PCnet-PCI controller uses non-burst write cycles to access the receive and transmit descriptor entries. Some of the write accesses to the receive buffer memory are also in non-burst mode. All PCnet-PCI controller non-burst write accesses are of the PCI command type Memory Write (type 7).

Figure 5 shows two non-burst write access within one arbitration cycle. The PCnet-PCI controller will drop FRAME between two consecutive non-burst write cycles. The PCnet-PCI controller will re-request the bus immediately if it is preempted before starting the second access. The example below shows an extended cycle for the first access. The target asserts DEVSEL 2 clock cycles after the address phase (FRAME asserted) and adds one extra wait state by asserting TRDY only on clock 7. The second write cycle in the example shows a ZERO wait state access.



•DEVSEL is sampled by the PCnet-PCI controller.

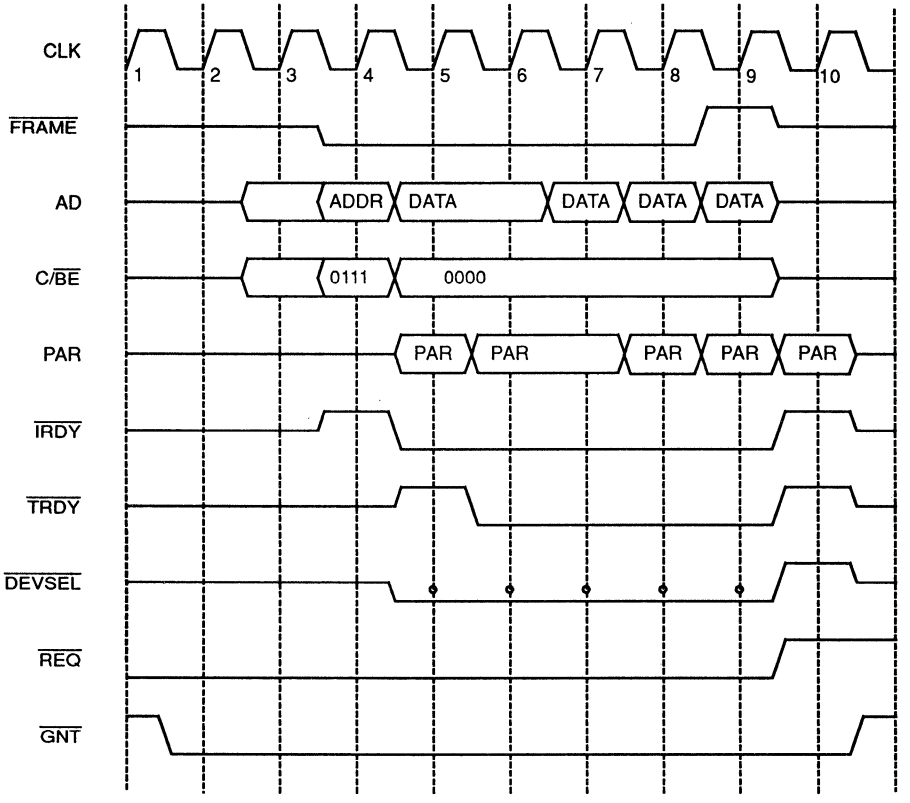
18220C-7

Figure 5. Non-Burst Write Cycles With and Without Wait States

Basic Burst Write Cycles

The PCnet-PCI controller provides a burst mode to write data to the receive buffer. The burst mode must be enabled by setting BWRITE in BCR18. All PCnet-PCI controller burst write transfers are of the PCI command type Memory Write (type 7). AD[1:0] will both be ZERO during the address phase indicating a linear burst order. All four byte enable signals will be ZERO during the data phase as the PCnet-PCI controller always writes a full 32-bit word when in burst mode.

Figure 6 shows a typical burst write access. The PCnet-PCI controller arbitrates for the bus, is granted access, and writes four 32-bit words (DWORDs) from system memory and then releases the bus. In this example, the memory system extends the data phase of the first access by one wait state. The following three data phases take one clock cycle each, which is determined by the timing of $\overline{TRDY}$.



o *DEVSEL is sampled by the PCnet-PCI controller.*

18220C-8

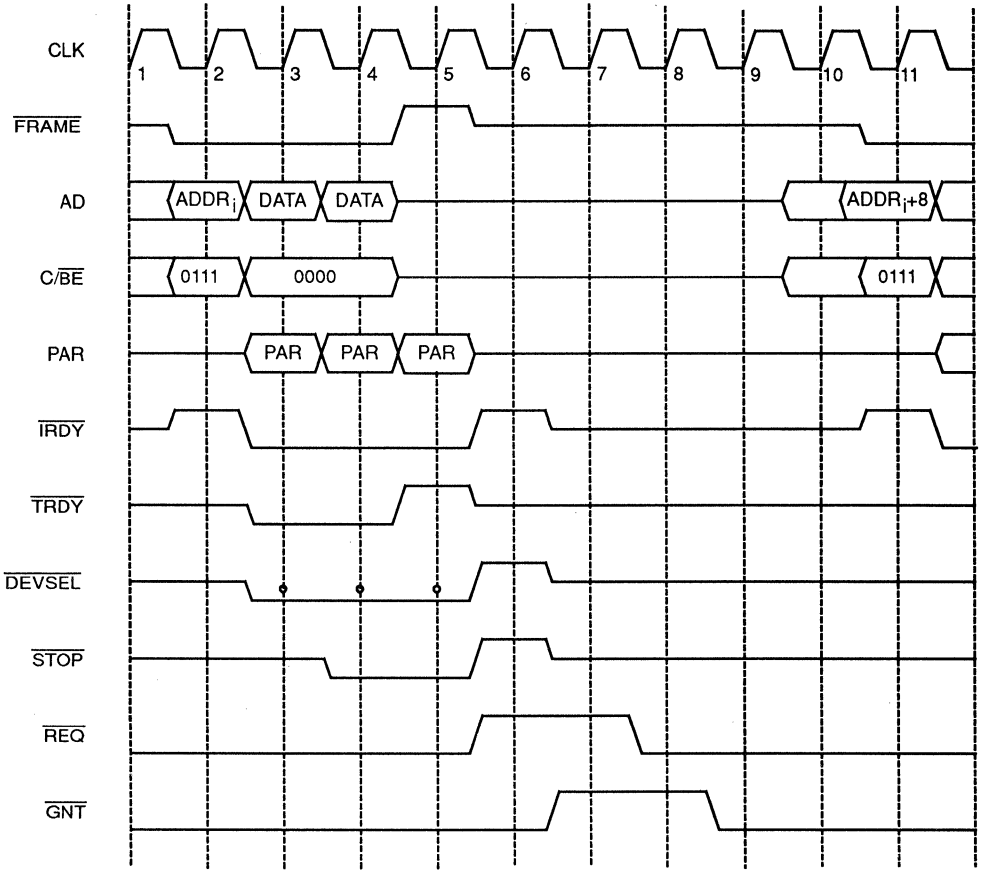
Figure 6. Burst Write Cycles

Target Initiated Termination

When the PCnet-PCI controller is a bus master, the cycles it produces on the PCI bus may be terminated by the target in one of three different ways.

Disconnect With Data Transfer

Figure 7 shows a disconnection in which one last data transfer occurs after the target asserted $\overline{STOP}$. $\overline{STOP}$ is asserted on clock 4 to start the termination sequence. Data is still transferred during this cycles, since both $\overline{IRDY}$ and $\overline{TRDY}$ are asserted. The PCnet-PCI controller terminates the current transfer with the deassertion of $\overline{FRAME}$ on clock 5 and then one clock cycle later with the deassertion $\overline{IRDY}$. It finally releases the bus on clock 6. The PCnet-PCI controller will re-request the bus after 2 clock cycles, if it wants to transfer more data. The starting address of the new transfer will be the address of the next untransferred data.



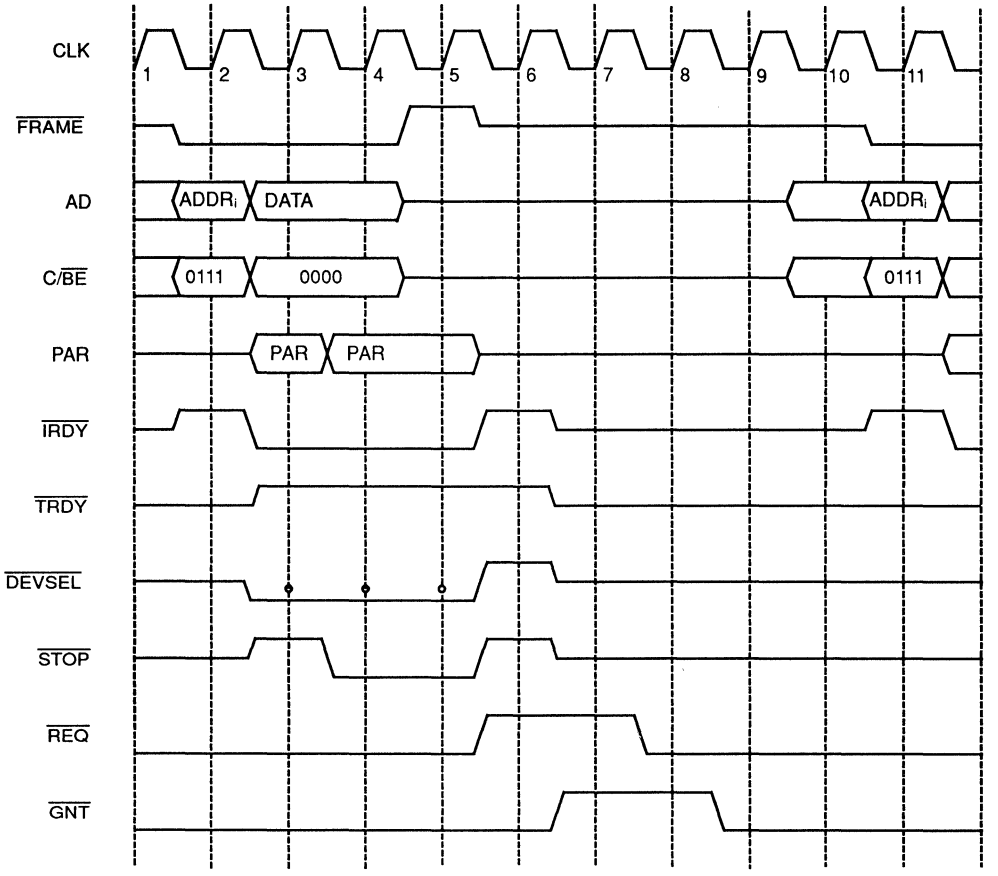
o *DEVSEL is sampled by the PCnet-PCI controller.*

18220C-9

Figure 7. Disconnect with Data Transfer

Disconnect Without Data Transfer

Figure 8 shows a target disconnect sequence during which no data is transferred. $\overline{STOP}$ is asserted on clock 4 without $\overline{TRDY}$ being asserted at the same time. The PCnet-PCI controller terminates the current transfer with the deassertion of $\overline{FRAME}$ on clock 5 and one clock cycle later with the deassertion of $\overline{IRDY}$. It finally releases the bus on clock 6. The PCnet-PCI controller will re-request the bus after 2 clock cycles to retry the last transfer. The starting address of the new transfer will be the same address as of the last untransferred data.



o *DEVSEL is sampled by the PCnet-PCI controller.*

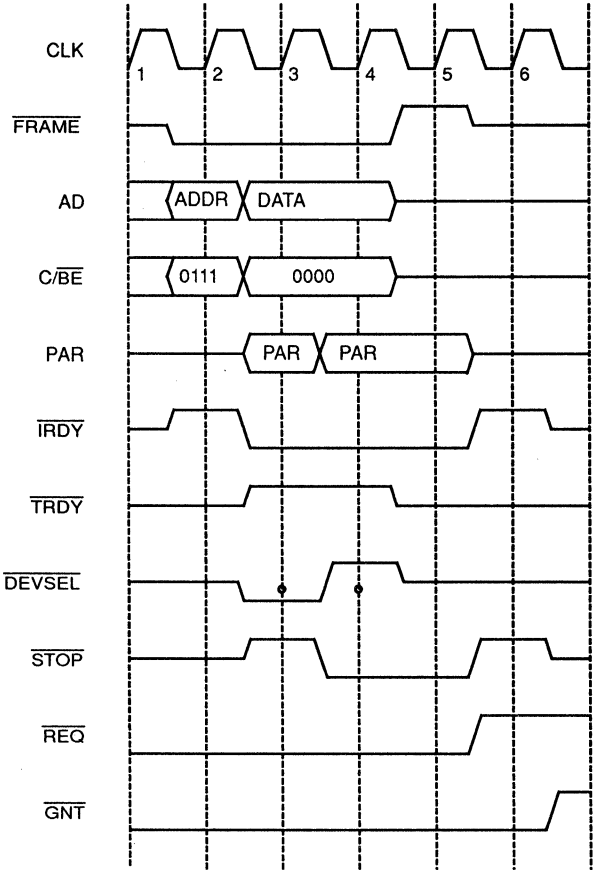
18220C-10

Figure 8. Disconnect Without Data Transfer

Target Abort

Figure 9 shows a target abort sequence. The target asserts DEVSEL for one clock. It then deasserts DEVSEL and asserts STOP on clock 4. A target can use the target abort sequence to indicate that it cannot service the data transfer and that it does not want the transaction to be retried. Additionally, the PCnet-PCI controller cannot make any assumption about the success of the previous data transfers in the current transaction. The PCnet-PCI controller terminates the current transfer with the deassertion of FRAME on clock 5 and one clock cycle later with the deassertion of IRDY. It finally releases the bus on clock 6.

Since data integrity is not guaranteed, the PCnet-PCI controller cannot recover from a target abort event. The PCnet-PCI controller will reset all CSR and BCR locations to their H_RESET values. Any on-going network activity will be stopped immediately. The PCI configuration registers will not be cleared. RTABORT (bit 12) in the Status register will be set to indicate that the PCnet-PCI controller has received a target abort.



o *DEVSEL is sampled by the PCnet-PCI controller.*

18220C-11

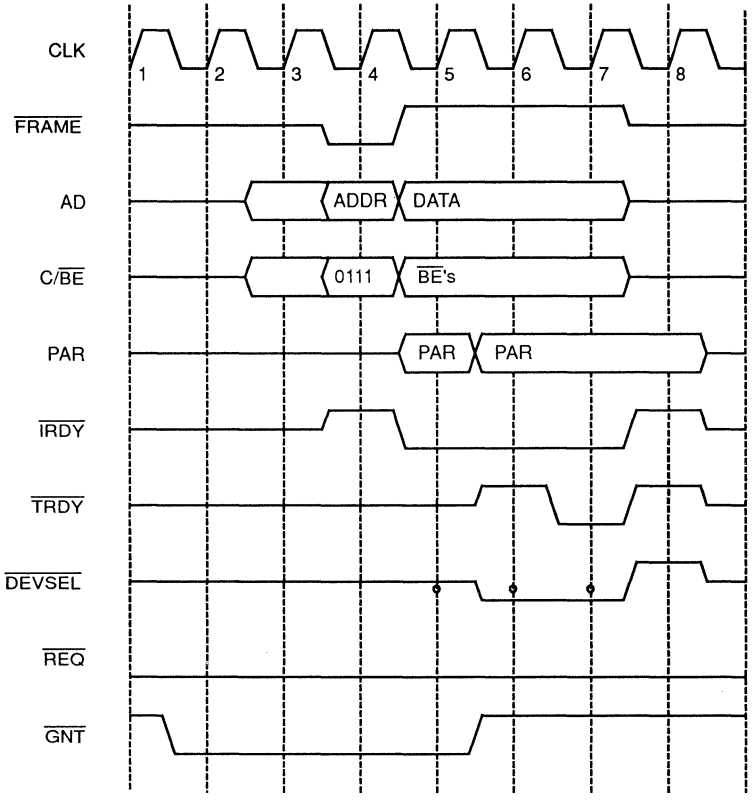
Figure 9. Target Abort

Master Initiated Termination

There are three scenarios besides normal completion of a transaction where the PCnet-PCI controller will terminate the cycles it produces on the PCI bus.

Preemption When FRAME is Deasserted

The PCnet-PCI controller will generate multiple address phases during a single bus ownership period when it is accessing the initialization block, the descriptor ring entries and the data buffers in main memory. FRAME is deasserted in between two address phases. While FRAME is deasserted, the central arbiter can remove GNT to the PCnet-PCI controller at any time to service another master. When GNT is removed, the PCnet-PCI controller will finish the current transfer and then release the bus. It will keep REQ asserted to regain bus ownership as soon as possible.



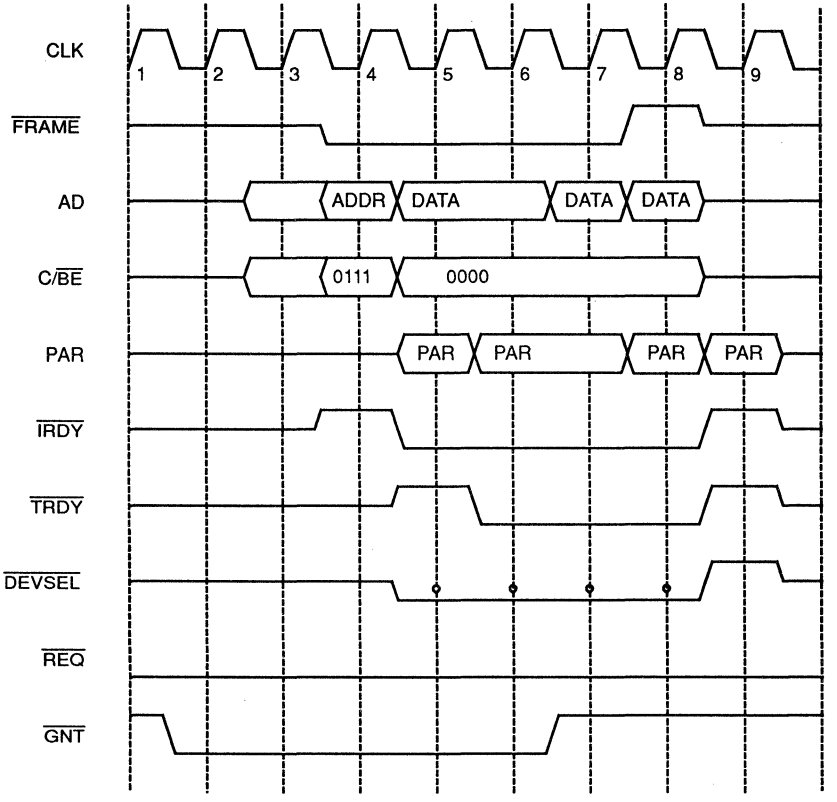
o *DEVSEL is sampled by the PCnet-PCI controller.*

18220C-12

Figure 10. Preemption When FRAME is Deasserted

Preemption When $\overline{\text{FRAME}}$ is Asserted

The central arbiter can take $\overline{\text{GNT}}$ to the PCnet-PCI controller away if the current bus operation takes too long. This may happen e.g. when the PCnet-PCI controller tries to fill the whole transmit FIFO and the target inserts extra wait states for every data phase. When $\overline{\text{GNT}}$ is taken away, the PCnet-PCI controller will finish the current transfer and then immediately release the bus. The Latency Timer in PCI configuration space of the PCnet-PCI controller is always set to ZERO. The PCnet-PCI controller will keep $\overline{\text{REQ}}$ asserted to regain bus ownership as soon as possible.



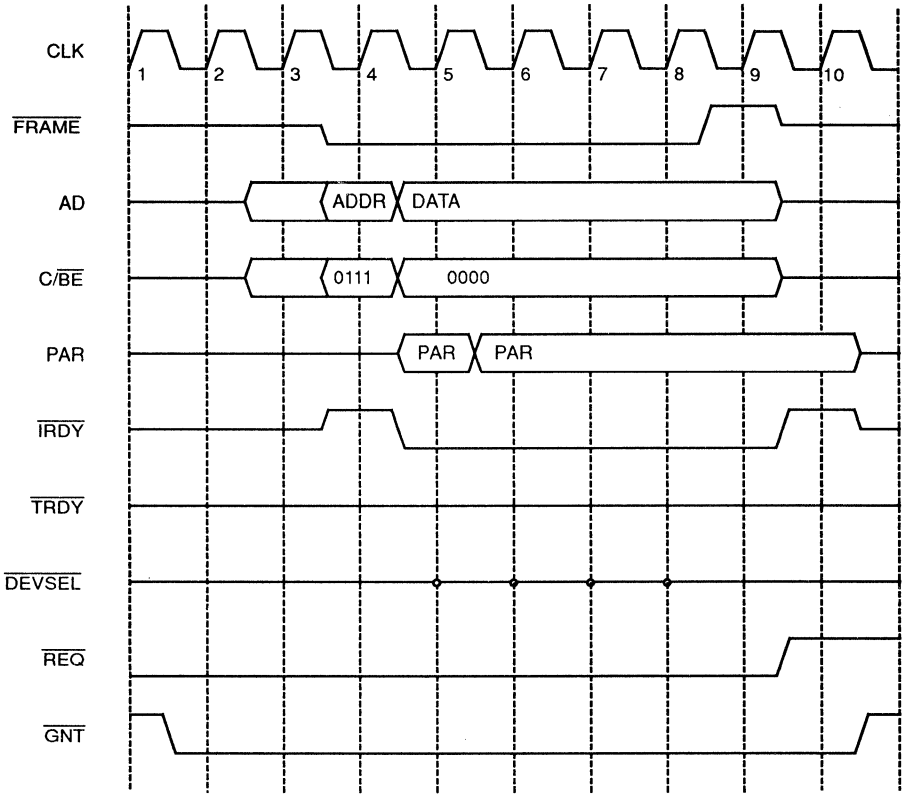
o *DEVSEL is sampled by the PCnet-PCI controller.*

18220C-13

Figure 11. Preemption When $\overline{\text{FRAME}}$ is Asserted

Master Abort

The PCnet-PCI controller will terminate its cycle with a Master Abort sequence if $\overline{\text{DEVSEL}}$ is not asserted within 4 clocks after $\overline{\text{FRAME}}$ is asserted. Master Abort is treated as a fatal error by the PCnet-PCI controller. The PCnet-PCI controller will reset all CSR and BCR locations to their H_RESET values. Any on-going network activity will be stopped immediately. The PCI configuration registers will not be cleared. RMABORT (bit 13) in the Status register will be set to indicate that the PCnet-PCI controller has terminated its transaction with a master abort.



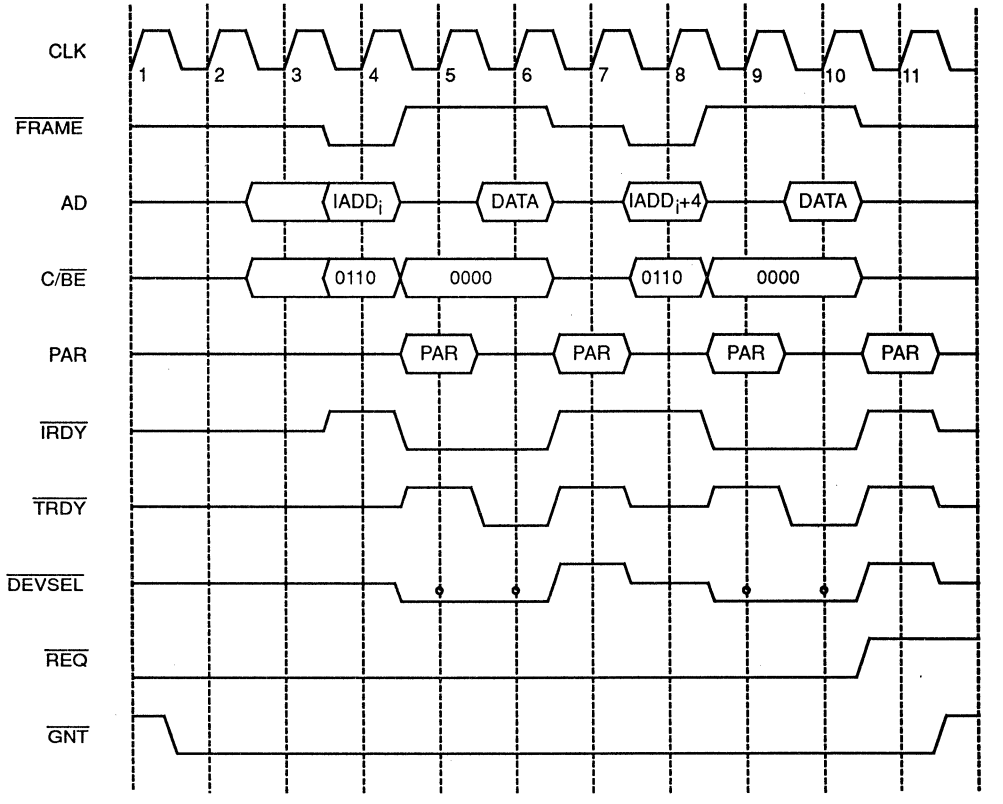
o *$\overline{\text{DEVSEL}}$ is sampled by the PCnet-PCI controller.*

18220C-14

Figure 12. Master Abort

Initialization Block DMA Transfers

During execution of the PCnet-PCI controller bus master initialization procedure, the PCnet-PCI microcode will repeatedly request DMA transfers from the BIU. During each of these initialization block DMA transfers, the BIU will perform two data transfer cycles (eight bytes) and then it will relinquish the bus. The two transfers within the mastership period will always be read cycles to ascending contiguous addresses. When SSIZE32 = 1 (BCR20, bit 8), there are 7 DWORDs to transfer during the bus master initialization procedure, so four bus mastership periods are needed in order to complete the initialization sequence. Note that the last DWORD transfer of the last bus mastership period of the initialization sequence accesses an unneeded location. Data from this transfer is discarded internally. When SSIZE32 = 0 (BCR20, bit 8), then three bus mastership periods are needed to complete the initialization sequence.



o *DEVSEL is sampled by the PCnet-PCI controller.*

18220C-15

Figure 13. Initialization Block Read

Descriptor DMA Transfers

PCnet-PCI microcode will determine when a descriptor access is required. A descriptor DMA read will consist of two DWORD (double-word) transfers. A descriptor DMA write will consist of one or two DWORD transfers. (The transfers within a descriptor DMA transfer mastership period will always be of the same type (either all read or all write)).

If buffer chaining is used, writes to the descriptors of all intermediate buffers consist of only one DWORD to return OWNership of the buffer to the system. On all single buffer transmit or receive descriptors, as well as on the last buffer in chain, writes to the descriptor consist of two DWORDs. The first DWORD containing status information. The second DWORD containing additional status and the OWNership bit (i.e. MD1[31]).

The transfers will be addressed as specified in tables 1 and 2.

Table 1. Bus Master Reads of Descriptors

16-Bit Software Mode			32-Bit Software Mode		
Address Sequence AD[7:0]*	**LANCE/ PCnet-ISA Item Accessed**	**PCnet-PCI Item Accessed**	**Address Sequence AD[7:0]***	**LANCE/ PCnet-ISA Item Accessed**	**PCnet-PCI Item Accessed**
00	MD1[15:0], MD0[15:0]	MD1[31:24], MD0[23:0]	04	MD1[15:8], MD2[15:0]	MD1[31:0]
04	MD3[15:0], MD2[15:0]	MD2[15:0], MD1[15:0]	00	MD1[7:0], MD0[15:0]	MD0[31:0]
Bus Break			Bus Break		

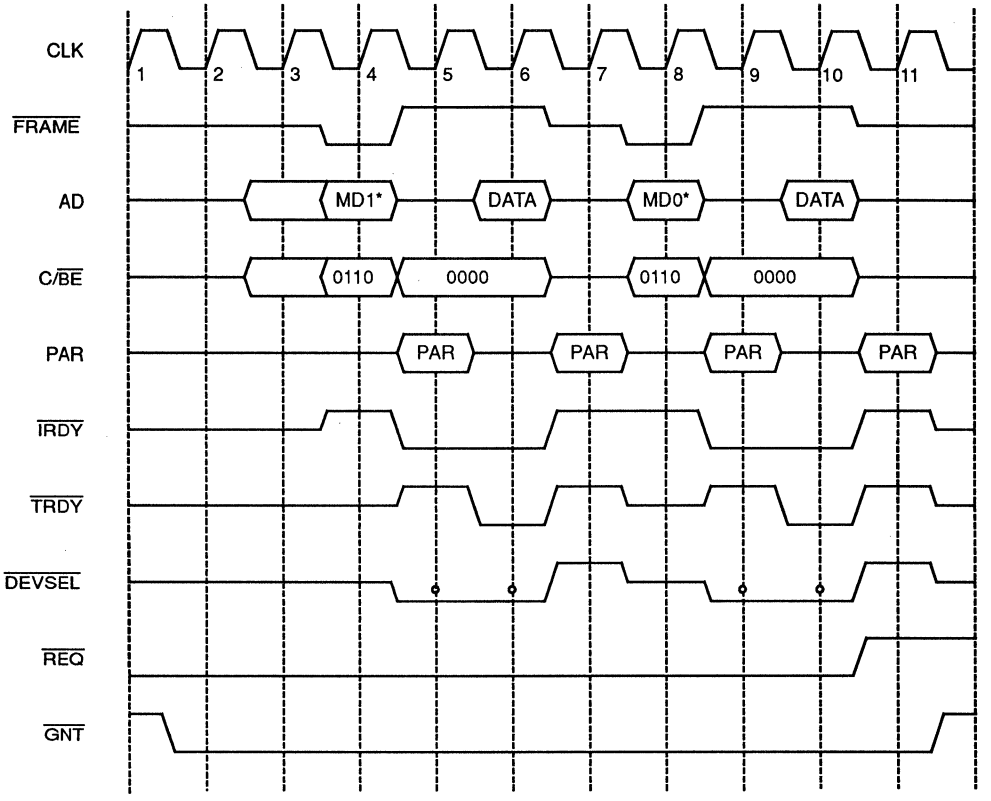
Table 2. Bus Master Writes to Descriptors

16-Bit Software Mode			32-Bit Software Mode		
Address Sequence AD[7:0]*	**LANCE/ PCnet-ISA Item Accessed**	**PCnet-PCI Item Accessed**	**Address Sequence AD[7:0]***	**LANCE/ PCnet-ISA Item Accessed**	**PCnet-PCI Item Accessed**
04	MD3[15:0], MD2[15:0]	MD2[15:0], MD1[15:0]	08	MD3[15:0]	MD2[31:0]
00	MD1[15:0], MD0[15:0]	MD1[31:24], MD0[23:0]	04	MD1[15:8], MD2[15:0]	MD1[31:0]
Bus Break			Bus Break		

** Address values for AD[31:08] are constant throughout any single descriptor DMA transfer. AD[1:0] must be set to ZERO in the descriptor base address.*

During descriptor read accesses, the byte enable signals will indicate that all byte lanes are active. Should some of the bytes not be needed, then the PCnet-PCI controller will internally discard the extraneous information that was gathered during such a read. During write accesses, only the bytes which need to be written are enabled, by activating the corresponding byte enable pins.

The only significant differences between descriptor DMA transfers and initialization DMA transfers are that the addresses of the accesses follow different ordering.

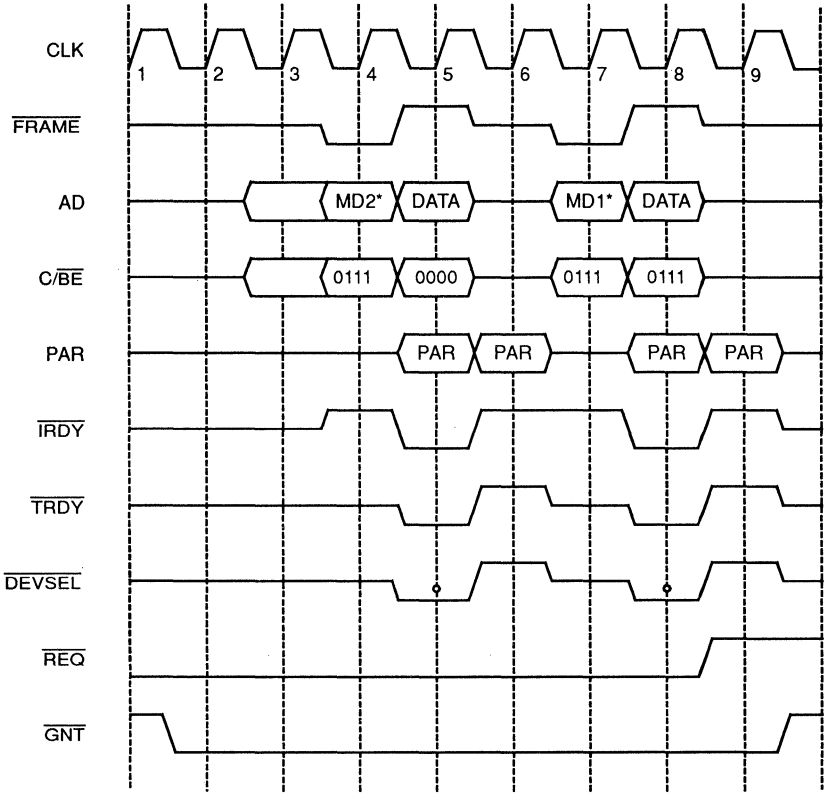


o *DEVSEL is sampled by the PCnet-PCI controller.*

* *Note that Message Descriptor addresses 1 and 0 are in descending order.*

18220C-16

Figure 14. Descriptor Ring Read



o *DEVSEL is sampled by the PCnet-PCI controller.*

** Note that Message Descriptor addresses 2 and 1 are in descending order.*

18220C-17

Figure 15. Descriptor Ring Write

FIFO DMA Transfers

PCnet-PCI microcode will determine when a FIFO DMA transfer is required. This transfer mode will be used for transfers of data to and from the PCnet-PCI FIFOs. Once the PCnet-PCI BIU has been granted bus mastership, it will perform a series of consecutive transfer cycles before relinquishing the bus. All transfers within the master cycle will be either read or write cycles, and all transfers will be to contiguous, ascending addresses. Both non-burst and burst cycles are used.

Non-Burst FIFO DMA Transfers

Non-burst FIFO DMA transfers is the default mode the PCnet-PCI controller uses to read and write data when accessing the FIFOs. Each non-burst transfer will be performed sequentially, with the issue of an address, and the transfer of the corresponding data with appropriate output signals to indicate selection of the active data bytes during the transfer. $\overline{\text{FRAME}}$ will be dropped after every address phase. (See figures 2,3 and 5.) The number of data transfer cycles contained within a single bus mastership period is in general dependent on the programming of the DMAPLUS option (CSR4, bit 14). Several other factors will also affect the length of the bus mastership period. The possibilities are as follows:

- If DMAPLUS = 0, a maximum of 16 transfers will be performed by default. This default value may be changed by writing to the DMA Transfer Counter (CSR80). Note that DMAPLUS = 0 merely sets a maximum value. The minimum number of transfers in the bus mastership period will be determined by all of the following variables: the settings of the FIFO watermarks and the conditions of the FIFOs, the value of the DMA Transfer Counter (CSR80), the value of the DMA Bus Timer (CSR82), and any occurrence of preemption that takes place during the bus mastership period.
- If DMAPLUS = 1, the bus cycle will continue until the transmit FIFO is filled to its high threshold (read transfers) or the receive FIFO is emptied to its low threshold (write transfers), or until the DMA Bus Timer value (CSR82) has expired. Other variables may also affect the end point of the bus mastership period in this mode. Among those variables are the particular conditions existing within the FIFOs, receive and transmit status conditions, and bus preemption events.

The FIFO thresholds are programmable (see description of CSR80), as are the DMA Transfer Counter and Bus Timer values. The exact number of transfer cycles in the case of DMAPLUS = 1 will be dependent on the latency of the system bus to the PCnet-PCI controller's mastership request and the speed of bus operation, but will be limited by the value in the Bus Timer register, the FIFO condition, receive and transmit status, and by preemption events. Barring a time-out by either of these registers, or a bus preemption by another mastering device, or exceptional receive and transmit events, or an end of packet signal from the FIFO, the FIFO watermark settings and the extent of Bus Grant latency will be the major factors determining the number of accesses performed during any given arbitration cycle when DMAPLUS = 1.

The $\overline{\text{TRDY}}$ response of the memory device will also affect the number of transfers when DMAPLUS = 1, since the speed of the accesses will affect the state of the FIFO. (During accesses, the FIFO may be filling or emptying on the network end. A slower memory response will allow additional data to accumulate inside of the FIFO (during write transfers from the receive FIFO). If the accesses are slow enough, a complete DWORD may become available before the end of the arbitration cycle and thereby increase the number of transfers in that cycle.) The general rule is that the longer the Bus Grant latency or the slower the bus transfer operations (or clock speed) or the higher the transmit watermark or the lower the receive watermark or any combination thereof the longer will be the average bus mastership period.

Burst FIFO DMA Transfers

Bursting is only performed by the PCnet-PCI controller if the BREADE and/or BWRITE bits of BCR18 are set. These bits individually enable/disable the ability of the PCnet-PCI controller to perform burst accesses during master read operations and master write operations, respectively. Only FIFO data transfers will make use of the burst mode.

The first transfer in the burst will consist of both an address and a data phase. Subsequent transfers will contain data only. AD[1:0] will always be ZERO during the address phase indicating a linear burst order. Note, that the terms 'burst' and 'linear burst' are used interchangeably throughout this document.

The number of data phases within the burst transfer is determined by the LINBC value from the BCR18 register. The burst upper limit is calculated by taking the BCR18 LINBC[2:0] value and multiplying it by 4. The result is the number of transfers that will be performed within a single linear burst sequence. When the LINBC upper limit of data transfers have been performed, a new $\overline{\text{FRAME}}$ may be asserted (if there is more data to be transferred), with a new address on the AD pins. Following the assertion of a new $\overline{\text{FRAME}}$, the linear bursting of data will resume. All byte lanes will always be active during all burst transfers as reflected by the C/$\overline{\text{BE}}$[3:0] signals.

The number of data transfer cycles within the total bus mastership period is dependent on the programming of the DMAPLUS option (CSR4, bit 14). The possibilities are as follows:

- If DMAPLUS = 0, a maximum of 16 transfers will be performed by default. This default value may be changed by writing to the DMA Transfer Counter (CSR80). Note that DMAPLUS = 0 merely sets a maximum value. The minimum number of transfers in the bus mastership period will be determined by all of the following variables: the settings of the FIFO watermarks and the conditions of the FIFOs, the value of the DMA Transfer Counter (CSR80), the value of the DMA Bus Timer (CSR82), and any occurrence of preemption that takes place during the bus mastership period.
- If DMAPLUS = 1, linear bursting will continue until the transmit FIFO is filled to its high threshold (read transfers) or the receive FIFO is emptied to its low threshold (write transfers), or until the DMA Bus Timer value (CSR82) has expired. A bus preemption event is another cause of termination of cycles. The FIFO thresholds are programmable (see description of CSR80), as are the DMA Transfer Counter and Bus Timer values. The exact number of total transfer cycles in the case of DMAPLUS = 1 will be dependent on the latency of the system bus to the PCnet-PCI controller's mastership request and the speed of bus operation, but will be limited by the value in the Bus Timer Register, the FIFO condition and by preemption occurrences, if any.

Note that the number of transfer cycles for each $\overline{\text{FRAME}}$ assertion will always only be controlled by LINBC, Bus Grant and FIFO conditions. The number of transfer cycles for each $\overline{\text{FRAME}}$ assertions will not be affected by DMAPLUS or by the values in the DMA Transfer Count register and Bus Timer register. However, these factors can influence the number of transfers that is performed during any given bus mastership period.

Barring a time-out by the DMA Transfer Count register or the Bus Timer register or a bus preemption by another mastering device, the FIFO watermark settings and the extent of Bus Grant latency will be the major factors in determining the number of accesses performed during any given bus mastership period. The $\overline{\text{TRDY}}$ response time of the memory device will also affect the number of transfers, since the speed of the accesses will affect the state of the FIFO. (During accesses, the FIFO may be filling or emptying on the network end. For example, on a Receive operation, a slower device will allow additional data to accumulate inside of the FIFO. If the accesses are slow enough, a complete DWORD may become available before the end of the bus mastership period and thereby increase the number of transfers in that period.) The general rule is that the longer the Bus Grant latency or the slower the bus transfer operations or the slower the clock speed or the higher the transmit watermark or the lower the receive watermark or any combination thereof, will produce longer total burst lengths.

Linear Burst DMA Starting Address Restrictions

A PCnet-PCI controller linear burst will begin only when the address of the current transfer meets the following condition:

AD[31:00] MOD (LINBC x 16) = 0,

The following table illustrates all possible starting address values for all legal LINBC values. Note that AD[31:06] are don't care values for all addresses. Also note that while AD[1:0] do not physically exist within a 32 bit system (the PCnet-PCI controller always drives AD[1:0] to ZERO during the address phase to indicate a linear burst order), they are valid bits within the buffer pointer field of descriptor word 0. Thus, where AD[1:0] are listed, they refer to the lowest two bits of the descriptors buffer pointer field. These bits will have an affect on determining when a PCnet-PCI controller linear burst operation may legally begin and they will affect the output values of the byte enable pins, therefore they have been included in the table as AD[1:0].

Table 3. Linear Burst DMA Starting Address Values

LINBC[2:0]	LBS = Linear Burst Size (number of transfers)	Size of Burst (bytes)	Linear Burst Beginning Addresses AD[5:0] = (Hex) (AD[31:06] = (don't care)
1	4	16	00, 10, 20, 30
2	8	32	00, 20
4	16	64	00
0, 3, 5, 7	Reserved	Reserved	Not Applicable

It is not necessary for the software to insure that the buffer address pointer contained in descriptor word 0 matches the address restrictions given in the table. If the buffer pointer does not meet the conditions set forth in the table, then the PCnet-PCI controller will simply postpone the start of linear bursting until enough non-burst FIFO DMA transfers have been performed to bring the current working buffer pointer value to a valid linear burst starting address. This operation is referred to as aligning the buffer address to a valid linear burst starting address. Once this has been done, the PCnet-PCI controller will recognize that the address for the current access is a valid linear burst starting address, and it will automatically begin to perform linear burst accesses at that time, provided of course that the software has enabled the linear burst mode.

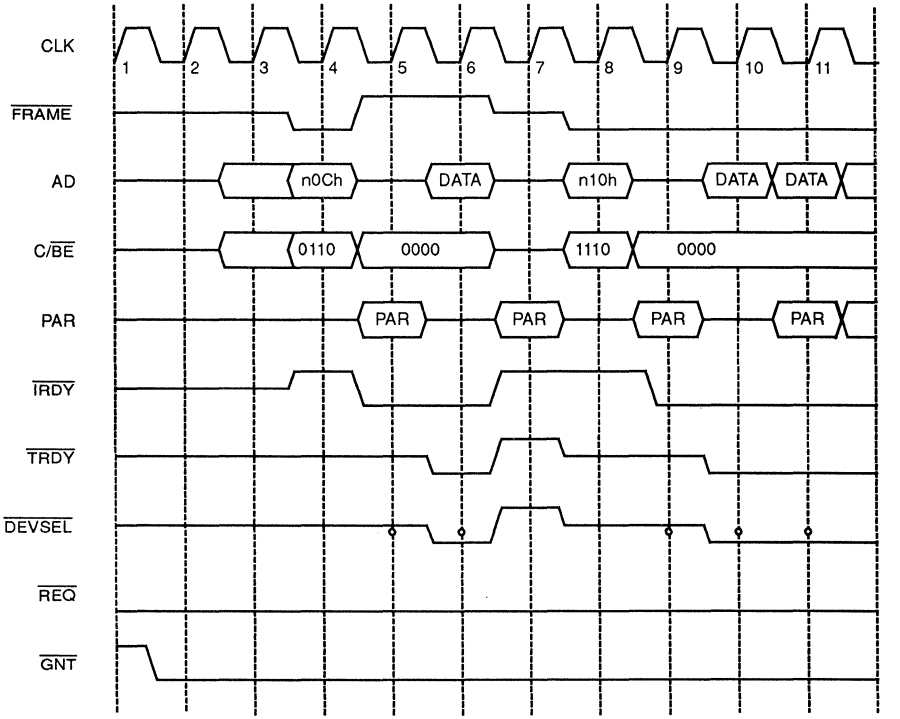
Note that if the software would provide only valid linear burst starting addresses in the buffer pointer, then the PCnet-PCI controller could avoid performing the alignment operation. It would begin linear burst accesses on the very first of the buffer transfers thereby allowing a slight gain in bus bandwidth efficiency.

Linear Burst DMA Address Alignment

Linear bursting may begin during a bus mastership period which was initially performing only non-burst operations. A change from non-burst operation to linear bursting will normally occur during linear burst DMA address alignment operations.

If the PCnet-PCI controller is programmed for burst mode (i.e. BREADE and/or BWRITE bits of BCR18 are set to ONE), and the PCnet-PCI controller requests the bus, but the starting address of the first transaction does not meet the conditions as specified in the table above, then the PCnet-PCI controller will perform non-burst accesses until it arrives at an address that does meet the conditions described in the table. At that time, and without releasing the bus, the PCnet-PCI controller will invoke the linear burst mode.

Figure 16 shows an example of a linear burst DMA alignment operation being performed. The first access to the transmit buffer is in non-burst mode, because the current address does not align with a linear burst boundary. The PCnet-PCI controller switches to burst mode beginning with the second transfer. $\overline{REQ}$ stays asserted during all transfers.



o *DEVSEL is sampled by the PCnet-PCI controller.*

18220C-18

Figure 16. Burst Alignment

Partial Linear Burst

Certain factors may cause the PCnet-PCI controller to burst fewer than the LINBC limit during a single burst sequence. Factors that could generate a partial linear burst include:

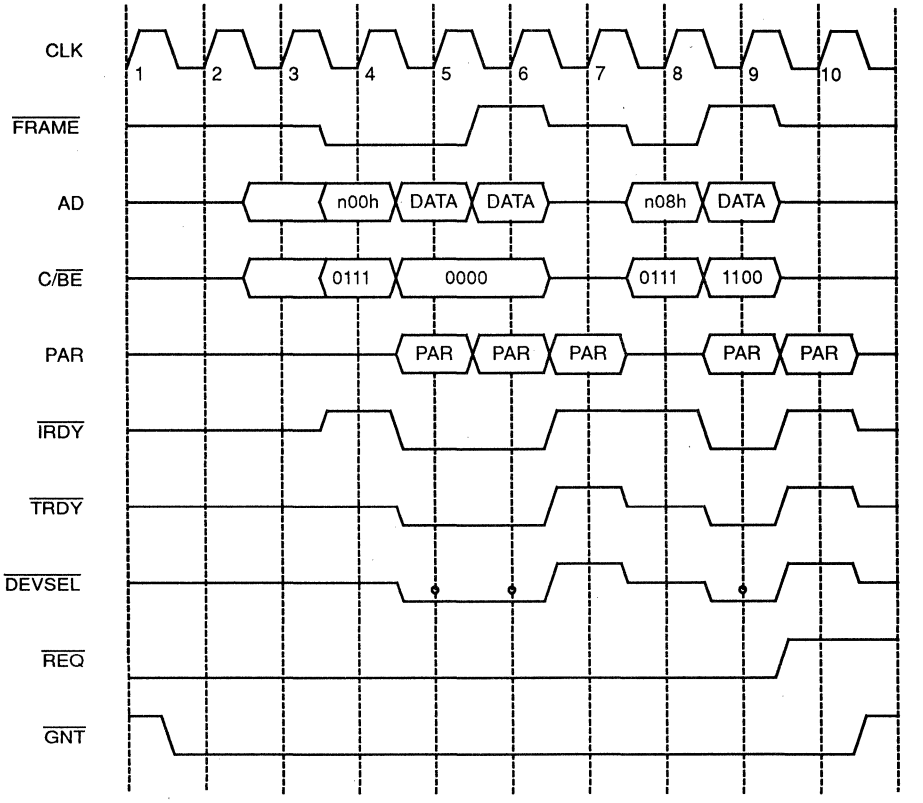
- No more data available for transfers from the current TX buffer
- No more data available for transfer from the RX FIFO for this packet
- No more space available for transfers to the current RX buffer
- Preemption

Typically, during the case of a master read operation (for TX buffer transfers), the last transfer in the linear burst sequence will be the last transfer executed before the PCnet-PCI controller releases the bus. This is true of both partial and completed linear burst sequences.

During the case of a master write operation (for RX buffer transfers) when RX packet data has ended, the last transfer in the linear burst sequence will be the last transfer executed before the PCnet-PCI controller releases the bus. This is true of both partial and completed linear burst sequences.

However, if the next transfer that the PCnet-PCI controller is scheduled to execute will be to the last available location of a RX buffer, then the PCnet-PCI controller will use a non-burst cycle to make the last transfer to the buffer. This event occurs because of the restrictions placed upon the byte enable signals during the linear burst operation. As mentioned in the initial description of linear burst accesses, all byte lanes of the data bus are always enabled during linear burst operations. Note, however, that in the case of the last RX buffer location, the PCnet-PCI controller may own only a portion of the DWORD location. In such cases, it is necessary to discontinue linear burst accesses on the second from last RX buffer location so that an ordinary transfer with some byte lanes disabled can be used for the final transfer.

Figure 17 shows a partial linear burst that occurred while approaching the transfer of the last bytes of data to a RX buffer. The linear burst begins when 10 bytes of space still remain in the RX buffer. (The number of spaces remaining for the figure as drawn could be anywhere from 9 to 12. The value of 10-byte spaces has been chosen just for purposes of illustration.) After the first linear burst transfer, 6 byte spaces remain. Knowing that the second transfer will use another 4 bytes of space, the PCnet-PCI controller is able to predict that the third transfer will be the last. Therefore, it de-asserts FRAME on the second transfer to terminate the linear burst operation. However, the PCnet-PCI controller retains ownership of the bus so that it may, immediately, make non-burst transfer(s) to the last two spaces in the buffer.



o *DEVSEL is sampled by the PCnet-PCI controller.*

18220C-19

Figure 17. Partial Linear Burst at the End of Buffer

If a burst cycle is preempted before the last data phase, the PCnet-PCI controller will finish the current data phase and release the bus. REQ will stay asserted. The PCnet-PCI controller will revert to non-burst cycle accesses following the preemption event. In this case, linear bursting will next occur when the memory address being accessed next meets the linear burst starting address requirements.

Slave I/O Transfers

After the PCnet-PCI controller is configured as I/O device (by setting IOEN in the PCI Command register), it starts monitoring the PCI bus for access to its internal registers. The PCnet-PCI controller will look for an address that falls within its 32 bytes of I/O address space (starting from the I/O base address). The PCnet-PCI controller will assert $\overline{\text{DEVSEL}}$ if it detects an address match and the access is an I/O cycle. $\overline{\text{DEVSEL}}$ is asserted two clock cycles after the host has asserted $\overline{\text{FRAME}}$. The PCnet-PCI controller will not assert $\overline{\text{DEVSEL}}$ if it detects an address match, but the PCI command is not of the type I/O read or I/O write. The PCnet-PCI controller will suspend looking for I/O cycles while being a bus master.

Slave I/O Read

The Slave I/O Read command is used by the host CPU to read the PCnet-PCI's CSRs, BCRs and EEPROM locations. It is a single cycle, non-burst 8-bit,16-bit or 32-bit transfer which is initiated by the host CPU. The typical number of wait states added to a slave I/O read access on the part of the PCnet-PCI controller is 6 to 7 clock cycles, depending upon the relative phases of the internal Buffer Management Unit clock and the CLK signal, since the internal Buffer Management Unit clock is a divide-by-two version of the CLK signal. The PCnet-PCI controller will not produce Slave I/O Read commands while being a bus master.

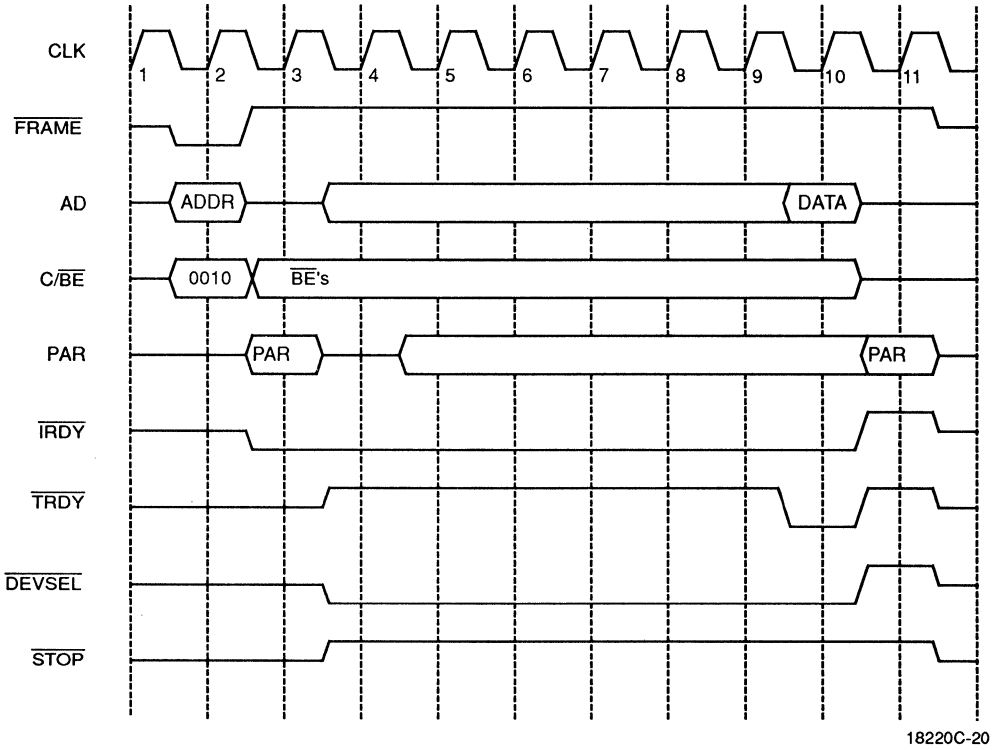


Figure 18. Slave I/O Read

Slave I/O Write

The Slave I/O Write command is used by the host CPU to write to the PCnet-PCI's CSRs, BCRs and EEPROM locations. It is a single cycle, non-burst 16-bit or 32-bit transfer which is initiated by the host CPU. The typical number of wait states added to a slave I/O write access on the part of the PCnet-PCI controller is 6 to 7 clock cycles, depending upon the relative phases of the internal Buffer Management Unit clock and the CLK signal, since the internal Buffer Management Unit clock is a divide-by-two version of the CLK signal. The PCnet-PCI controller will not produce Slave I/O write commands while being a bus master.

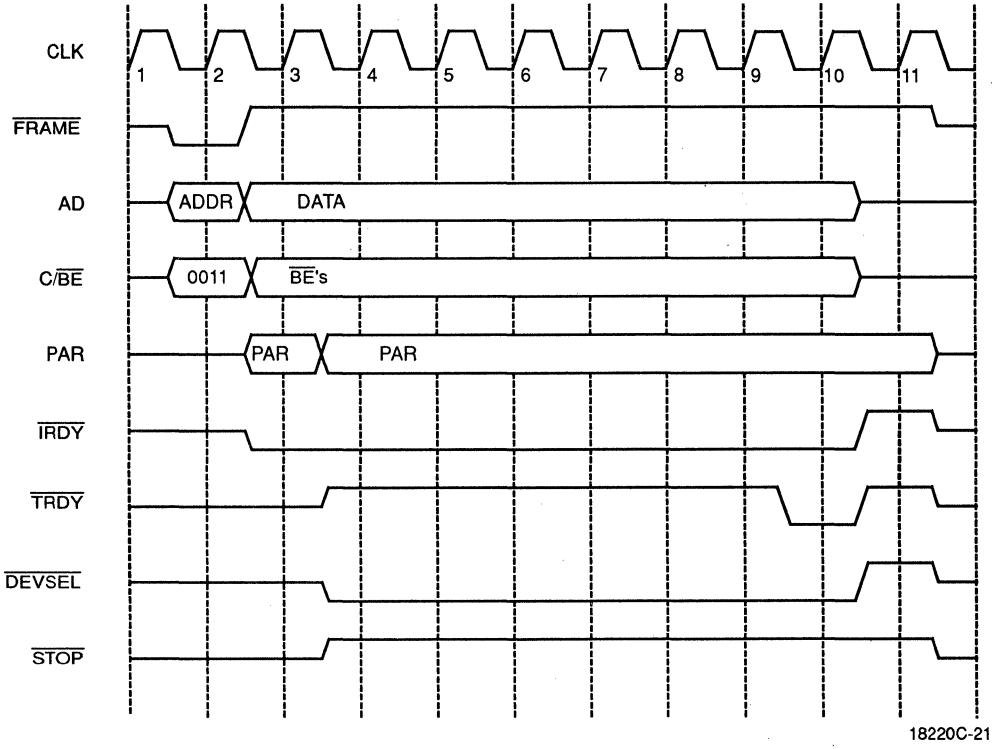


Figure 19. Slave I/O Write

Slave Configuration Transfers

The host can access the PCnet-PCI PCI configuration space with a configuration read or write command. The PCnet-PCI controller will assert $\overline{DEVSEL}$ if the IDSEL input is asserted during the address phase and if the access is a configuration cycle. $\overline{DEVSEL}$ is asserted two clock cycles after the host has asserted $\overline{FRAME}$. All configuration cycles are of fixed length. The PCnet-PCI controller will assert $\overline{TRDY}$ on the 3rd clock of the data phase.

Slave Configuration Read

The Slave Configuration Read command is used by the host CPU to read the configuration space in the PCnet-PCI controller. This provides the host CPU with information concerning the device and its capabilities. This is a single cycle, non-burst 8-bit, 16-bit, or 32-bit transfer.

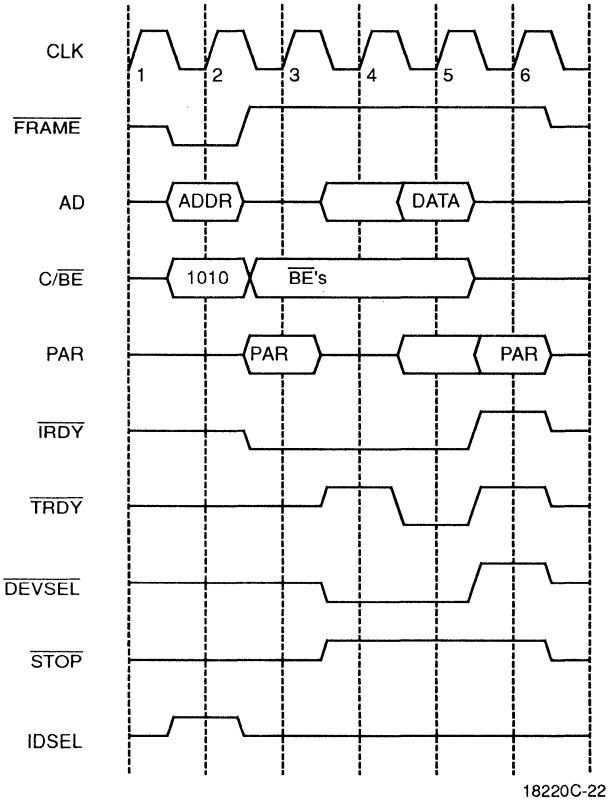


Figure 20. Slave Configuration Read

Slave Configuration Write

The Slave Configuration Write command is used by the host CPU to write the configuration space in the PCnet-PCI controller. This allows the host CPU to control basic activity of the device, such as enable/disable, change I/O location, etc. This is a single cycle, non-burst 8-bit, 16-bit, or 32-bit transfer.

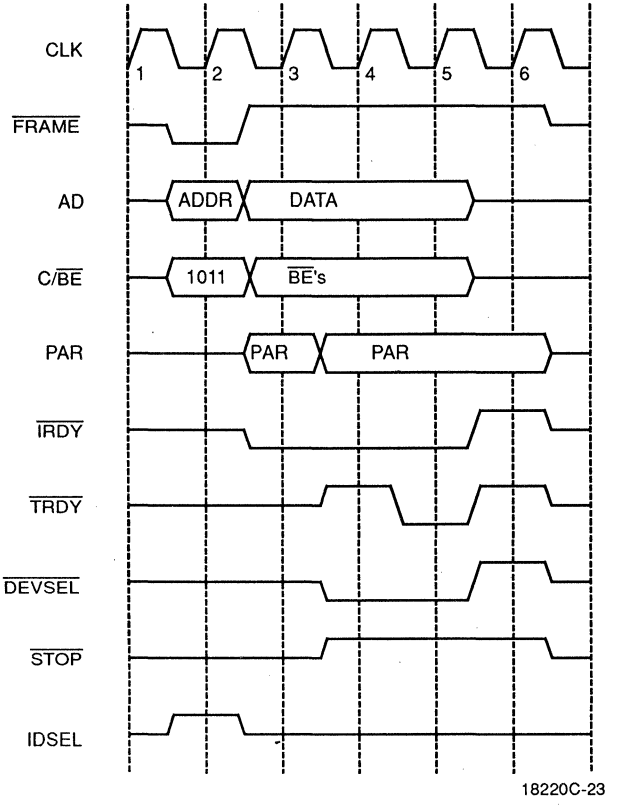


Figure 21. Slave Configuration Write

Buffer Management Unit (BMU)

The buffer management unit is a micro-coded state machine which implements the initialization procedure and manages the descriptors and buffers. The buffer management unit operates at half the speed of the CLK input.

Initialization

PCnet-PCI initialization includes the reading of the initialization block in memory to obtain the operating parameters. The initialization block is read when the INIT bit in CSR0 is set. The INIT bit should be set before or concurrent with the STRT bit to insure correct operation. Two DWORDs are read during each period of bus mastership. When SSIZE32 = 1 (BCR20, bit 8), this results in a total of 4 arbitration cycles (3 arbitration cycles if SSIZE32 = 0). Once the initialization block has been completely read in and internal registers have been updated, IDON will be set in CSR0, and an interrupt generated (if IENA is set). At this point, the BMU knows where the receive and transmit descriptor rings and hence, normal network operations will begin.

The PCnet-PCI controller obtains the start address of the Initialization Block from the contents of CSR1 (least significant 16 bits of address) and CSR2 (most significant 16 bits of address). The host must write CSR1 and CSR2 before setting the INIT bit. The block contains the user defined conditions for PCnet-PCI operation, together with the base addresses and length information of the transmit and receive descriptor rings.

There is an alternative method to initialize the PCnet-PCI controller. Instead of initialization via the initialization block in memory, data can be written directly into the appropriate registers. Either method may be used at the discretion of the programmer. If the registers are written to directly, the INIT bit must not be set, or the initialization block will be read in, thus overwriting the previously written information. Please refer to Appendix C for details on this alternative method.

If initialization is done by writing directly to registers, the Polling Interval register (CSR47) must be initialized in addition to those registers that can be loaded automatically from the initialization block.

Re-Initialization

The transmitter and receiver sections of the PCnet-PCI controller can be turned on via the initialization block (MODE Register DTX, DRX bits; CSR15[1:0]). The states of the transmitter and receiver are monitored by the host through CSR0 (RXON, TXON bits). The PCnet-PCI controller should be reinitialized if the transmitter and/or the receiver were not turned on during the original initialization, and it was subsequently required to activate them or if either section was shut off due to the detection of an error condition (MERR, UFLO, TX BUFF error).

Reinitialization may be done via the initialization block or by setting the STOP bit in CSR0, followed by writing to CSR15, and then setting the START bit in CSR0. Note that this form of restart will not perform the same in the PCnet-PCI controller as in the LANCE. In particular, upon restart, the PCnet-PCI controller reloads the transmit and receive descriptor pointers with their respective base addresses. This means that the software must clear the descriptor own bits and reset its descriptor ring pointers before the restart of the PCnet-PCI controller. The reload of descriptor base addresses is performed in the LANCE only after initialization, so a restart of the LANCE without initialization leaves the LANCE pointing at the same descriptor locations as before the restart.

Buffer Management

Buffer management is accomplished through message descriptor entries organized as ring structures in memory. There are two rings, a receive ring and a transmit ring. The size of a message descriptor entry is 4 DWORDs, or 16 bytes, when SSIZE32 = 1. The size of a message descriptor entry is 4 words, or 8 bytes, when SSIZE32 = 0.

Descriptor Rings

Each descriptor ring must be organized in a contiguous area of memory. At initialization time (setting the INIT bit in CSR0), the PCnet-PCI controller reads the user-defined base address for the transmit and receive descriptor rings, as well as the number of entries contained in the descriptor rings. Descriptor ring base addresses must be on a 16-byte boundary when SSIZE32=1, and 8-byte boundary when SSIZE=0. A maximum of 128 (or 512, depending upon the value of SSIZE32) ring entries is allowed when the ring length is set through the TLEN and RLEN fields of the initialization block. However, the ring lengths can be set beyond this range (up to 65535) by writing the transmit and receive ring length registers (CSR76, CSR78) directly.

Each ring entry contains the following information:

1. The address of the actual message data buffer in user or host memory
2. The length of the message buffer
3. Status information indicating the condition of the buffer

To permit the queuing and de-queuing of message buffers, ownership of each buffer is allocated to either the PCnet-PCI controller or the host. The OWN bit within the descriptor status information, either TMD or RMD (see section on TMD or RMD), is used for this purpose. OWN = "1" signifies that the PCnet-PCI controller currently has ownership of this ring descriptor and its associated buffer. Only the owner is permitted to relinquish ownership or to write to any field in the descriptor entry. A device that is not the current owner of a descriptor entry cannot assume ownership or change any field in the

entry. A device may, however, read from a descriptor that it does not currently own. Software should always read descriptor entries in sequential order. When software finds that the current descriptor is owned by the PCnet-PCI controller, then the software must not read "ahead" to the next descriptor. The software should wait at the unOWNed descriptor until ownership has been granted to the software (when SPRINTEN = 1 (CSR3, bit 5), then this rule is modified. See the SPRINTEN description). Strict adherence to these rules insures that "Deadly Embrace" conditions are avoided.

Descriptor Ring Access Mechanism

At initialization, the PCnet-PCI controller reads the base address of both the transmit and receive descriptor rings into CSRs for use by the PCnet-PCI controller during subsequent operations.

As the final step in the self-initialization process, the base address of each ring is loaded into each of the current descriptor address registers and the address of the next descriptor entry in the transmit and receive rings is computed and loaded into each of the next descriptor address registers.

When SSIZE32 = 0, software data structures are 16 bits wide. The following diagram, Figure 22, illustrates the relationship between the Initialization Base Address, the Initialization Block, the Receive and Transmit Descriptor Ring Base Addresses, the Receive and Transmit Descriptors and the Receive and Transmit Data Buffers, for the case of SSIZE32 = 0.

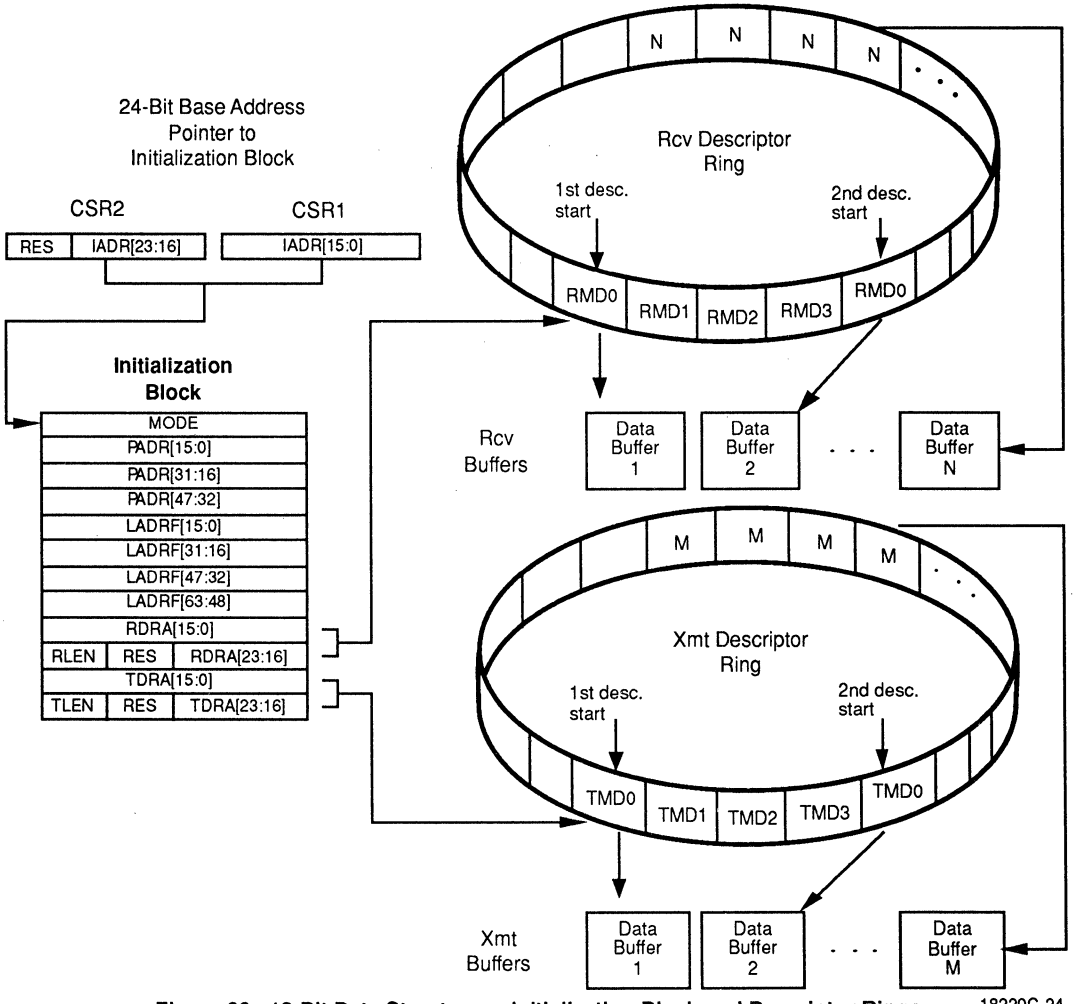


Figure 22. 16-Bit Data Structures: Initialization Block and Descriptor Rings 18220C-24

When SSIZE32 = 1, software data structures are 32 bits wide. The following diagram illustrates, Figure 23, the relationship between the Initialization Base Address, the Initialization Block, the Receive and Transmit Descriptor Ring Base Addresses, the Receive and Transmit Descriptors and the Receive and Transmit Data Buffers, for the case of SSIZE32 = 1.

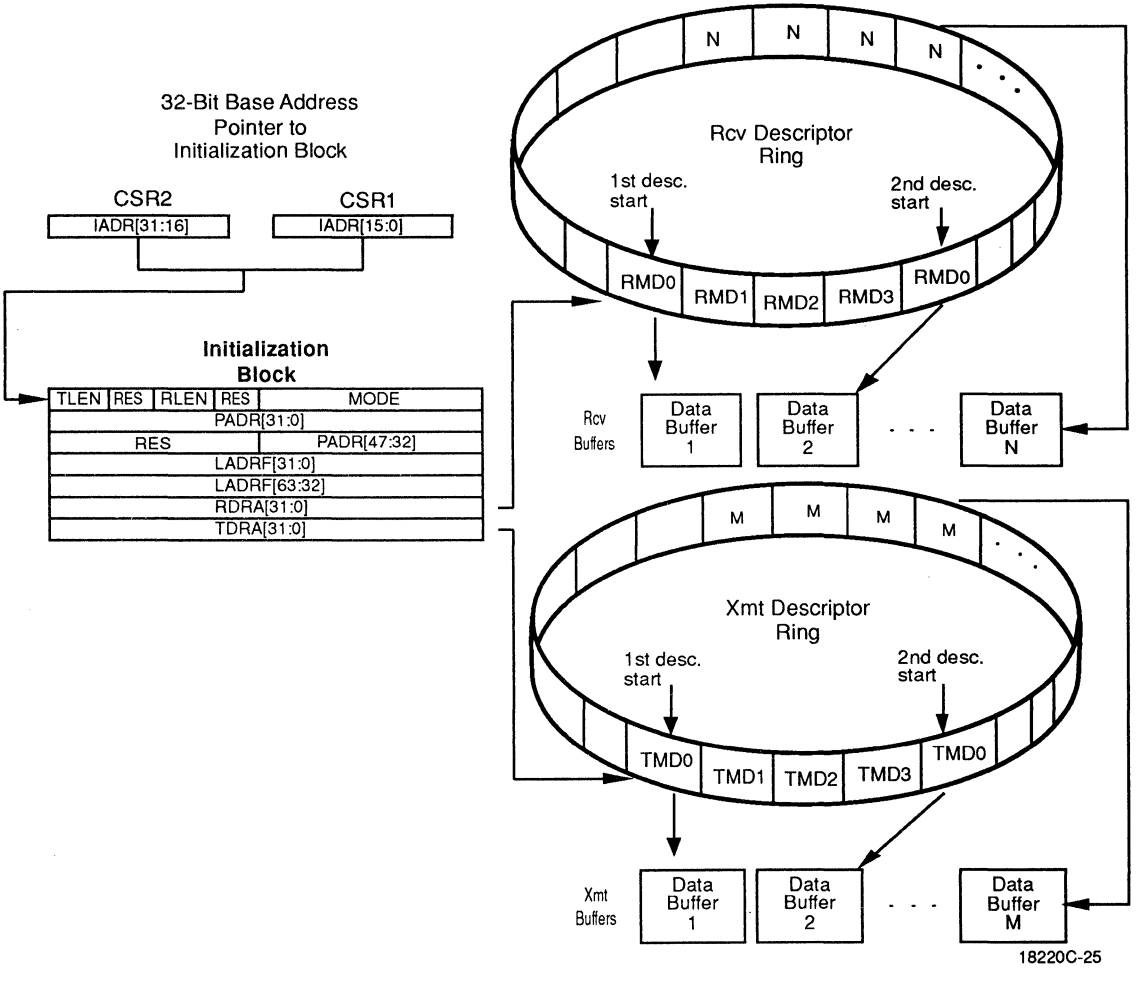


Figure 23. 32-Bit Data Structures: Initialization Block and Descriptor Rings

Polling

If there is no network channel activity and there is no pre- or post-receive or pre- or post-transmit activity being performed by the PCnet-PCI controller, then the PCnet-PCI controller will periodically poll the current receive and transmit descriptor entries in order to ascertain their ownership. If the DPOLL bit in CSR4 is set, then the transmit polling function is disabled.

A typical polling operation consists of the following: The PCnet-PCI controller will use the current receive descriptor address stored internally to vector to the appropriate Receive Descriptor Table Entry (RDTE). It will then use the current transmit descriptor address (stored internally) to vector to the appropriate Transmit Descriptor Table Entry (TDTE). The accesses will be made in the following order: RMD1, then RMD0 of the current RDTE during one bus arbitration, and after that, TMD1, then TMD0 of the current TDTE during a second bus arbitration. All information collected during polling activity will be stored internally in the appropriate CSRs. (i.e. CSR18, CSR19, CSR20, CSR21, CSR40, CSR42, CSR50, CSR52). UnOWNed descriptor status will be internally ignored.

A typical receive poll is the product of the following conditions:

1. PCnet-PCI controller does not possess ownership of the current RDTE and the poll time has elapsed and RXON=1 (CSR0, bit 5), or
2. PCnet-PCI controller does not possess ownership of the next RDTE the poll time has elapsed and RXON=1.

If RXON=0 the PCnet-PCI controller will never poll RDTE locations.

The ideal system should always have at least one RDTE available for the possibility of an unpredictable receive event. (This condition is not a requirement. If this condition is not met, it simply means that frames will be missed by the system because there was no buffer space available.) But the typical system usually has at least one or two RDTEs available for the possibility of an unpredictable receive event. Given that this condition is satisfied, the current and next RDTE polls are rarely seen and hence, the typical poll operation simply consists of a check of the status of the current TDTE. When there is only one RDTE (because the RLEN was set to ZERO), then there is no "next RDTE" and ownership of "next RDTE" cannot be checked. If there is at least one RDTE, the RDTE poll will rarely be seen and the typical poll operation simply consists of a check of the current TDTE.

A typical transmit poll is the product of the following conditions:

1. PCnet-PCI controller does not possess ownership of the current TDTE and
   DPOLL=0 (CSR4, bit 2) and
   TXON=1 (CSR0, bit 4) and
   the poll time has elapsed, or
2. PCnet-PCI controller does not possess ownership of the current TDTE and
   DPOLL=0 and
   TXON=1 and
   a frame has just been received, or
3. PCnet-PCI controller does not possess ownership of the current TDTE and
   DPOLL=0 and
   TXON=1 and
   a frame has just been transmitted.

Setting the TDMD bit of CSR0 will cause the microcode controller to exit the poll counting code and immediately perform a polling operation. If RDTE ownership has not been previously established, then an RDTE poll will be performed ahead of the TDTE poll. If the microcode is not executing the poll counting code when the TDMD bit is set, then the demanded poll of the TDTE will be delayed until the microcode returns to the poll counting code.

The user may change the poll time value from the default of 65,536 clock periods by modifying the value in the Polling Interval register (CSR47). Note that if a non–default value is desired, then a strict sequence of setting the INIT bit in CSR0, waiting for IDONE, then writing to CSR47, and then setting STRT in CSR0 must be observed, otherwise the default value will not be overwritten. See the CSR47 section for details.

Transmit Descriptor Table Entry (TDTE)

If, after a TDTE access, the PCnet-PCI controller finds that the OWN bit of that TDTE is not set, then the PCnet-PCI controller resumes the poll time count and reexamines the same TDTE at the next expiration of the poll time count.

If the OWN bit of the TDTE is set, but Start of Frame (STP) bit is not set, the PCnet-PCI controller will immediately request the bus in order to reset the OWN bit of this descriptor. (This condition would normally be found following a LCOL or RETRY error that occurred in the middle of a transmit frame chain of buffers.) After resetting the OWN bit of this descriptor, the PCnet-PCI controller will again immediately request the bus in order to access the next TDTE location in the ring.

If the OWN bit is set and the buffer length is 0, the OWN bit will be reset. In the LANCE the buffer length of 0 is interpreted as a 4096-byte buffer. It is acceptable to have a 0 length buffer on transmit with STP = 1 or STP = 1 *and* ENP = 1. It is not acceptable to have 0 length buffer with STP = 0 and ENP =1.

If the OWN bit is set and the start of frame (STP) bit is set, then microcode control proceeds to a routine that will enable transmit data transfers to the FIFO. The PCnet-PCI controller will look ahead to the next transmit descriptor after it has performed at least one transmit data transfer from the first buffer. (More than one transmit data transfer may possibly take place, depending upon the state of the transmitter.) The contents of TMD0 and TMD1 will be stored in Next Xmt Buffer Address (CSR64 and CSR65), Next Xmt Byte Count (CSR66) and Next Xmt Status (CSR67) regardless of the state of the OWN bit. This transmit descriptor lookahead operation is performed only once.

If the PCnet-PCI controller does not own the next TDTE (i.e. the second TDTE for this frame), then it will complete transmission of the current buffer and then update the status of the current (first) TDTE with the BUFF and UFLO bits being set. This will cause the transmitter to be disabled (CSR0, TXON=0). The PCnet-PCI controller will have to be re-initialized to restore the transmit function. The situation that matches this description implies that the system has not been able to stay ahead of the PCnet-PCI controller in the transmit descriptor ring and therefore, the condition is treated as a fatal error. (To avoid this situation, the system should always set the transmit chain descriptor own bits in reverse order.)

If the PCnet-PCI controller does own the second TDTE in a chain, it will gradually empty the contents of the first buffer (as the bytes are needed by the transmit operation), perform a single-cycle DMA transfer to update the status of the first descriptor (reset the OWN bit in TMD1), and then it may perform one data DMA access on the second buffer in the chain before executing another lookahead operation. (i.e. a lookahead to the third descriptor.)

The PCnet-PCI controller can queue up to two frames in the transmit FIFO. Call them frame "X" and frame "Y", where "Y" is after "X". Assume that frame "X" is currently being transmitted. Because the PCnet-PCI controller can perform lookahead data transfer past the ENP of frame "X", it is possible for the PCnet-PCI controller to completely transfer the data from a buffer belonging to frame "Y" into the FIFO even though frame "X" has not yet been completely transmitted. At the end of this "Y" buffer data transfer, the PCnet-PCI controller will write intermediate status (change the OWN bit to a ZERO) for the "Y" frame buffer, if frame "Y" uses data chaining. The last TDTE for the "X" frame (containing ENP) has not yet been written, since the "X" frame has not yet been completely transmitted. Note that the PCnet-PCI controller has, in this instance, returned ownership of a TDTE to the host out of a "normal" sequence.

For this reason, it becomes imperative that the host system should never read the Transmit DTE ownership bits out of order. Software should always process buffers in sequence, waiting for the ownership before proceeding.

There should be no problems for software which processes buffers in sequence, waiting for ownership before proceeding.

If an error occurs in the transmission before all of the bytes of the current buffer have been transferred, then TMD2 and TMD1 of the current buffer will be written; In such a case, data transfers from the next buffer will not commence. Instead, following the TMD2/TMD1 update, the PCnet-PCI controller will go to the next transmit frame, if any, skipping over the rest of the frame which experienced an error, including chained buffers. This is done by returning to the polling microcode where PCnet-PCI controller will immediately access the next descriptor and find the condition OWN=1 and STP=0 as described earlier. As described for that case, the PCnet-PCI controller will reset the own bit for this descriptor and continue in like manner until a descriptor with OWN=0 (no more transmit frames in the ring) or OWN=1 and STP=1 (the first buffer of a new frame) is reached.

At the end of any transmit operation, whether successful or with errors, immediately following the completion of the descriptor updates, the PCnet-PCI controller will always perform another poll operation. As described earlier, this poll operation will begin with a check of the current RDTE, unless the PCnet-PCI controller already owns that descriptor. Then the PCnet-PCI controller will proceed to polling the next TDTE. If the transmit descriptor OWN bit has a ZERO value, then the PCnet-PCI controller will resume poll time count incrementing. If the transmit descriptor OWN bit has a value of ONE, then the PCnet-PCI controller will begin filling the FIFO with transmit data and initiate a transmission. This end–of–operation poll coupled with the TDTE lookahead operation allows the PCnet-PCI controller to avoid inserting poll time counts between successive transmit frames.

Whenever the PCnet-PCI controller completes a transmit frame (either with or without error) and writes the status information to the current descriptor, then the TINT bit of CSR0 is set to indicate the completion of a transmission. This causes an interrupt signal if the IENA bit of CSR0 has been set and the TINTM bit of CSR3 is reset.

Receive Descriptor Table Entry (RDTE)

If the PCnet-PCI controller does not own both the current and the next Receive Descriptor Table Entry then the PCnet-PCI controller will continue to poll according to the polling sequence described above. If the receive descriptor ring length is 1, then there is no next descriptor to be polled.

If a poll operation has revealed that the current and the next RDTE belong to the PCnet-PCI controller then additional poll accesses are not necessary. Future poll operations will not include RDTE accesses as long as the PCnet-PCI controller retains ownership of the current and the next RDTE.

When receive activity is present on the channel, the PCnet-PCI controller waits for the complete address of the message to arrive. It then decides whether to accept or reject the frame based on all active addressing schemes. If the frame is accepted the PCnet-PCI controller checks the current receive buffer status register CRST (CSR41) to determine the ownership of the current buffer.

If ownership is lacking, then the PCnet-PCI controller will immediately perform a (last ditch) poll of the current RDTE. If ownership is still denied, then the PCnet-PCI controller has no buffer in which to store the incoming message. The MISS bit will be set in CSR0 and an interrupt will be generated if IENA=1 (CSR0) and MISSM=0 (CSR3). Another poll of the current RDTE will not occur until the frame has finished.

If the PCnet-PCI controller sees that the last poll (either a normal poll, or the last-ditch effort described in the above paragraph) of the current RDTE shows valid ownership, then it proceeds to a poll of the next RDTE. Following this poll, and regardless of the outcome of this poll, transfers of receive data from the FIFO may begin.

Regardless of ownership of the second receive descriptor, the PCnet-PCI controller will continue to perform receive data DMA transfers to the first buffer. If the frame length exceeds the length of the first buffer, and the PCnet-PCI controller does not own the second buffer, ownership of the current descriptor will be passed back to the system by writing a ZERO to the OWN bit of RMD1 and status will be written indicating buffer (BUFF=1) and possibly overflow (OFLO=1) errors.

If the frame length exceeds the length of the first (current) buffer, and the PCnet-PCI controller does own the second (next) buffer, ownership will be passed back to the system by writing a ZERO to the OWN bit of RMD1 when the first buffer is full. Receive data transfers to the second buffer may occur before the PCnet-PCI controller proceeds to look ahead to the ownership of the third buffer. Such action will depend upon the state of the FIFO when the status has been updated on the first descriptor. In any case, lookahead will be performed to the third buffer and the information gathered will be stored in the chip, regardless of the state of the ownership bit. As in the transmit flow, lookahead operations are performed only once.

This activity continues until the PCnet-PCI controller recognizes the completion of the frame (the last byte of this receive message has been removed from the FIFO). The PCnet-PCI controller will subsequently update the current RDTE status with the end of frame (ENP) indication set, write the message byte count (MCNT) of the complete frame into RMD2 and overwrite the "current" entries in the CSRs with the "next" entries.

Media Access Control

The Media Access Control engine incorporates the essential protocol requirements for operation of a compliant Ethernet/802.3 node, and provides the interface between the FIFO sub-system and the Manchester Encoder/Decoder (MENDEC).

The MAC engine is fully compliant to Section 4 of ISO/IEC 8802-3 (ANSI/IEEE Standard 1990 Second edition) and ANSI/IEEE 802.3 (1985).

The MAC engine provides programmable enhanced features designed to minimize host supervision, bus utilization, and pre- or post- message processing. These include the ability to disable retries after a collision, dynamic FCS generation on a frame-by-frame basis, and automatic pad field insertion and deletion to enforce minimum frame size attributes, automatic retransmission without reloading the FIFO, automatic deletion of collision fragments, and reduces bus bandwidth use.

The two primary attributes of the MAC engine are:

- Transmit and receive message data encapsulation.
  - — Framing (frame boundary delimitation, frame synchronization).
  - — Addressing (source and destination address handling).
  - — Error detection (physical medium transmission errors).
- Media access management.
  - — Medium allocation (collision avoidance).
  - — Contention resolution (collision handling).

Transmit and Receive Message Data Encapsulation

The MAC engine provides minimum frame size enforcement for transmit and receive frames. When APAD_XMT = 1 (CSR, bit 11), transmit messages will be padded with sufficient bytes (containing 00h) to ensure that the receiving station will observe an information field (destination address, source address, length/type, data and FCS) of 64-bytes. When ASTRP_RCV = 1 (CSR4, bit 10), the receiver will

automatically strip pad bytes from the received message by observing the value in the length field, and stripping excess bytes if this value is below the minimum data size (46 bytes). Both features can be independently over-ridden to allow illegally short (less than 64 bytes of frame data) messages to be transmitted and/or received. The use of this feature reduces bus utilization because the pad bytes are not transferred into or out of main memory.

Framing (Frame Boundary Delimitation, Frame Synchronization)

The MAC engine will autonomously handle the construction of the transmit frame. Once the Transmit FIFO has been filled to the predetermined threshold (set by XMTSP in CSR80), and providing access to the channel is currently permitted, the MAC engine will commence the 7 byte preamble sequence (10101010b, where first bit transmitted is a 1). The MAC engine will subsequently append the Start Frame Delimiter (SFD) byte (10101011b) followed by the serialized data from the Transmit FIFO. Once the data has been completed, the MAC engine will append the FCS (most significant bit first) which was computed on the entire data portion of the frame. The data portion of the frame consists of destination address, source address, length/type, and frame data.

The user is responsible for the correct ordering and content in each of the fields in the frame.

The receive section of the MAC engine will detect an incoming preamble sequence and lock to the encoded clock. The internal MENDEC will decode the serial bit stream and present this to the MAC engine. The MAC will discard the first 8-bits of information before searching for the SFD sequence. Once the SFD is detected, all subsequent bits are treated as part of the frame. The MAC engine will inspect the length field to ensure minimum frame size, strip unnecessary pad characters (if enabled), and pass the remaining bytes through the Receive FIFO to the host. If pad stripping is performed, the MAC engine will also strip the received FCS bytes, although the normal FCS computation and checking will occur. Note that apart from pad stripping, the frame will be passed unmodified to the host. If the length field has a value of 46 or greater, the MAC engine will not attempt to validate the length against the number of bytes contained in the message.

If the frame terminates or suffers a collision before 64-bytes of information (after SFD) have been received, the MAC engine will automatically delete the frame from the Receive FIFO, without host intervention. The PCnet-PCI controller has the ability to accept runt packets for diagnostics purposes and proprietary networks.

Addressing (Source and Destination Address Handling)

The first 6-bytes of information after SFD will be interpreted as the destination address field. The MAC engine provides facilities for physical, logical (multicast) and broadcast address reception.

Error Detection (Physical Medium Transmission Errors)

The MAC engine provides several facilities which report and recover from errors on the medium. In addition, the network is protected from gross errors due to inability of the host to keep pace with the MAC engine activity.

On completion of transmission, the following transmit status is available in the appropriate TMD and CSR areas:

- The exact number of transmission retry attempts (ONE, MORE, RTRY or TRC).
- Whether the MAC engine had to Defer (DEF) due to channel activity.
- Excessive deferral (EXDEF), indicating that the transmitter has experienced Excessive Deferral on this transmit frame, where Excessive Deferral is defined in ISO 8802-3 (IEEE/ANSI 802.3).
- Loss of Carrier (LCAR), indicating that there was an interruption in the ability of the MAC engine to monitor its own transmission. Repeated LCAR errors indicate a potentially faulty transceiver or network connection.
- Late Collision (LCOL) indicates that the transmission suffered a collision after the slot time. This is indicative of a badly configured network. Late collisions should not occur in a normal operating network.
- Collision Error (CERR) indicates that the transceiver did not respond with an SQE Test message within the predetermined time after a transmission completed. This may be due to a failed transceiver, disconnected or faulty transceiver drop cable, or the fact the transceiver does not support this feature (or it is disabled).

In addition to the reporting of network errors, the MAC engine will also attempt to prevent the creation of any network error due to the inability of the host to service the MAC engine. During transmission, if the host fails to keep the Transmit FIFO filled sufficiently, causing an underflow, the MAC engine will guarantee the message is either sent as a runt packet (which will be deleted by the receiving station) or has an invalid FCS (which will also cause the receiver to reject the message).

The status of each receive message is available in the appropriate RMD and CSR areas. FCS and Framing errors (FRAM) are reported, although the received frame is still passed to the host. The FRAM error will only be reported if an FCS error is detected and there are a non integral number of bytes in the message. The MAC engine will ignore up to 7 additional bits at the end of a message (dribbling bits), which can occur under normal network operating conditions. The reception of 8 additional bits will cause the MAC engine to de-serialize the entire byte, and will result in the received message and FCS being modified.

The PCnet-PCI controller can handle up to 7 dribbling bits when a received frame terminates. During the reception, the FCS is generated on every serial bit (including the dribbling bits) coming from the cable, although the internally saved FCS value is only updated on the eighth bit (on each byte boundary). The framing error is reported to the user as follows:

- If the number of dribbling bits are 1 to 7 and there is no CRC (FCS) error, then there is no Framing error (FRAM = 0).
- If the number of dribbling bits are 1 to 7 and there is a CRC (FCS) error, then there is also a Framing error (FRAM = 1).
- If the number of dribbling bits = 0, then there is no Framing error. There may or may not be a CRC (FCS) error.

Counters are provided to report the Receive Collision Count and Runt Packet Count for network statistics and utilization calculations.

Note that if the MAC engine detects a received frame which has a 00b pattern in the preamble (after the first 8 bits which are ignored), the entire frame will be ignored. The MAC engine will wait for the network to go inactive before attempting to receive additional frames.

Media Access Management

The basic requirement for all stations on the network is to provide fairness of channel allocation. The 802.3/Ethernet protocols define a media access mechanism which permits all stations to access the channel with equality. Any node can attempt to contend for the channel by waiting for a predetermined time (Inter Packet Gap internal) after the last activity, before transmitting on the media. The channel is a multidrop communications media (with various topological configurations permitted) which allows a single station to transmit and all other stations to receive. If two nodes simultaneously contend for the channel, their signals will interact causing loss of data, defined as a collision. It is the responsibility of the MAC to attempt to avoid and recover from a collision, to guarantee data integrity for the end-to-end transmission to the receiving station.

Medium Allocation

The IEEE/ANSI 802.3 Standard (ISO/IEC 8802-3 1990) requires that the CSMA/CD MAC monitor the medium for traffic by watching for carrier activity. When carrier is detected, the media is considered busy, and the MAC should defer to the existing message.

The ISO 8802-3 (IEEE/ANSI 802.3) Standard also allows optional two part deferral after a receive message.

See ANSI/IEEE Std 802.3 –1990 Edition, 4.2.3.2.1:

***Note**: It is possible for the PLS carrier sense indication to fail to be asserted during a collision on the media. If the deference process simply times the interFrame gap based on this indication it is possible for a short interFrame gap to be generated, leading to a potential reception failure of a subsequent frame. To enhance system robustness the following optional measures, as specified in 4.2.8, are recommended when InterFrameSpacingPart1 is other than ZERO:*

1. *Upon completing a transmission, start timing the interpacket gap, as soon as transmitting and carrier Sense are both false.*
2. *When timing an interFrame gap following reception, reset the interFrame gap timing if carrier Sense becomes true during the first 2/3 of the interFrame gap timing interval. During the final 1/3 of the interval the timer shall not be reset to ensure fair access to the medium. An initial period shorter than 2/3 of the interval is permissible including ZERO.*

The MAC engine implements the optional receive two part deferral algorithm, with a first part inter-frame-spacing time of 6.0 µs. The second part of the inter-frame-spacing interval is therefore 3.6 µs.

The PCnet-PCI controller will perform the two part deferral algorithm as specified in Section 4.2.8 (Process Deference). The Inter Packet Gap (IPG) timer will start timing the 9.6 µs InterFrameSpacing after the receive carrier is de-asserted. During the first part deferral (InterFrameSpacingPart1 – IFS1) the PCnet-PCI controller will defer any pending transmit frame and respond to the receive message. The IPG counter will be reset to ZERO continuously until the carrier de-asserts, at which point the IPG counter will resume the 9.6 µs count once again. Once the IFS1 period of 6.0 µs has elapsed, the PCnet-PCI controller will begin timing the second part deferral (InterFrame Spacing Part 2 – IFS2) of 3.6 µs. Once IFS1 has completed, and IFS2 has commenced, the PCnet-PCI controller will not defer to a receive frame if a transmit frame is pending. This means that the PCnet-PCI controller will not attempt to receive the receive frame, since it will start to transmit, and generate a collision at 9.6 µs. The PCnet-PCI controller will guarantee to complete the preamble (64-bit) and jam (32-bit)

sequence before ceasing transmission and invoking the random backoff algorithm.

This transmit two part deferral algorithm is implemented as an option which can be disabled using the DXMT2PD bit in CSR3. Two part deferral after transmission is useful for ensuring that severe IPG shrinkage cannot occur in specific circumstances, causing a transmit message to follow a receive message so closely as to make them indistinguishable.

During the time period immediately after a transmission has been completed, the external transceiver (in the case of a standard AUI connected device), should generate the SQE Test message (a nominal 10 MHz burst of 5 –15 Bit Times duration) on the CI± pair (within 0.6 µs – 1.6 µs after the transmission ceases). During the time period in which the SQE Test message is expected the PCnet-PCI controller will not respond to receive carrier sense.

See ANSI/IEEE Std 802.3—1990 Edition, 7.2.4.6 (1):

"At the conclusion of the output function, the DTE opens a time window during which it expects to see the signal_quality_error *signal asserted on the Control In circuit. The time window begins when the CARRIER_STATUS becomes CARRIER_OFF. If execution of the output function does not cause CARRIER_ON to occur, no SQE test occurs in the DTE. The duration of the window shall be at least 4.0 µs but no more than 8.0 µs. During the time window the Carrier Sense Function is inhibited."*

The PCnet-PCI controller implements a carrier sense "blinding" period within 0 µs – 4.0 µs from de-assertion of carrier sense after transmission. This effectively means that when transmit two part deferral is enabled (DXMT2PD is cleared) the IFS1 time is from 4 µs to 6 µs after a transmission. However, since IPG shrinkage below 4 µs will rarely be encountered on a correctly configured networks, and since the fragment size will be larger than the 4 µs blinding window, then the IPG counter will be reset by a worst case IPG shrinkage/fragment scenario and the PCnet-PCI controller will defer its transmission. In addition, the PCnet-PCI controller will not restart the "blinding" period if carrier is detected within the 4.0 µs – 6.0 µs IFS1 period, but will commence timing of the entire IFS1 period.

Contention Resolution (Collision Handling)

Collision detection is performed and reported to the MAC engine by the integrated Manchester Encoder/Decoder (MENDEC). If a collision is detected before the complete preamble/SFD sequence has been transmitted, the MAC Engine will complete the preamble/SFD before appending the jam sequence. If a collision is detected after the preamble/SFD has been completed, but prior to 512 bits being transmitted, the MAC Engine will abort the transmission, and append the jam sequence immediately. The jam sequence is a 32-bit all Zeros pattern.

The MAC Engine will attempt to transmit a frame a total of 16 times (initial attempt plus 15 retries) due to normal collisions (those within the slot time). Detection of collision will cause the transmission to be re-scheduled, dependent on the backoff time that the MAC Engine computes. If a single retry was required, the ONE bit will be set in the Transmit Frame Status. If more than one retry was required, the MORE bit will be set. If all 16 attempts experienced collisions, the RTRY bit will be set (ONE and MORE will be clear), and the transmit message will be flushed from the FIFO. If retries have been disabled by setting the DRTY bit in CSR15, the MAC Engine will abandon transmission of the frame on detection of the first collision. In this case, only the RTRY bit will be set and the transmit message will be flushed from the FIFO.

If a collision is detected after 512 bit times have been transmitted, the collision is termed a late collision. The MAC Engine will abort the transmission, append the jam sequence and set the LCOL bit. No retry attempt will be scheduled on detection of a late collision, and the transmit message will be flushed from the FIFO.

The ISO 8802-3 (IEEE/ANSI 802.3) Standard requires use of a "truncated binary exponential backoff" algorithm which provides a controlled pseudo random mechanism to enforce the collision backoff interval, before re-transmission is attempted.

See ANSI/IEEE Std 802.3—1990 Edition, 4.2.3.2.5:

"At the end of enforcing a collision (jamming), the CSMA/CD sublayer delays before attempting to retransmit the frame. The delay is an integer multiple of slot Time. The number of slot times to delay before the nth re-transmission attempt is chosen as a uniformly distributed random integer r in the range:

$$0 \leq r < 2k$$

where

$$k = \min(n, 10).$$

"

The PCnet-PCI controller provides an alternative algorithm, which suspends the counting of the slot time/IPG during the time that receive carrier sense is detected. This aids in networks where large numbers of nodes are present, and numerous nodes can be in collision. It effectively accelerates the increase in the backoff time in busy networks, and allows nodes not involved in the collision to access the channel whilst the colliding nodes await a reduction in channel activity. Once channel activity is reduced, the nodes resolving the collision time out their slot time counters as normal.

Manchester Encoder/Decoder (MENDEC)

The integrated Manchester Encoder/Decoder provides the PLS (Physical Layer Signaling) functions required for a fully compliant ISO 8802-3 (IEEE/ANSI 802.3) station. The MENDEC provides the encoding function for data to be transmitted on the network using the high accuracy on-board oscillator, driven by either the crystal oscillator or an external CMOS level compatible clock. The MENDEC also provides the decoding function from data received from the network. The MENDEC contains a Power On Reset (POR) circuit, which ensures that all analog portions of the PCnet-PCI controller are forced into their correct state during power up, and prevents erroneous data transmission and/or reception during this time.

External Crystal Characteristics

When using a crystal to drive the oscillator, the following crystal specification may be used to ensure less than ±0.5 ns jitter at DO±. See Table 4 below.

Table 4. Crystal Specification

Parameter	Min	Nom	Max	Unit
1. Parallel Resonant Frequency		20		MHz
2. Resonant Frequency Error	–50		+50	PPM
3. Change in Resonant Frequency With Respect to Temperature (0 – 70°C)*	–40		+40	PPM
4. Crystal Load Capacitance	20		50	pF
5. Motional Crystal Capacitance (C1)		0.022		pF
6. Series Resistance			35	Ω
7. Shunt Capacitance			7	pF
8. Drive Level			TBD	mW

** Requires trimming specification; not trim is 50 PPM total.*

External Clock Drive Characteristics

When driving the oscillator from a CMOS level external clock source, XTAL2 must be left floating (unconnected). An external clock having the following characteristics must be used to ensure less than ±0.5 ns jitter at DO±. See Table 5.

Table 5. Clock Drive Characteristics

Clock Frequency:	20 MHz ±0.01%
Rise/Fall Time (t_R/t_F):	<= 6 ns from 0.5 V to V_{DD} –0.5 V
XTAL1 HIGH/LOW Time (tHIGH/tLOW):	20 ns min
XTAL1 Falling Edge to Falling Edge Jitter:	< ±0.2 ns at 2.5 V input ($V_{DD/2}$)

MENDEC Transmit Path

The transmit section encodes separate clock and NRZ data input signals into a standard Manchester encoded serial bit stream. The transmit outputs (DO±) are designed to operate into terminated transmission lines. When operating into a 78 Ω terminated transmission line, the transmit signaling meets the required output levels and skew for Cheapernet, Ethernet and IEEE-802.3.

Transmitter Timing and Operation

A 20 MHz fundamental mode crystal oscillator provides the basic timing reference for the MENDEC portion of the PCnet-PCI controller. The crystal is divided by two, to create the internal transmit clock reference. Both clocks are fed into the MENDECs Manchester Encoder to generate the transitions in the encoded data stream. The internal transmit clock is used by the MENDEC to internally synchronize the Internal Transmit Data (ITXDAT) from the controller and Internal Transmit Enable (ITXEN). The internal transmit clock is also used as a stable bit rate clock by the receive section of the MENDEC and controller.

The oscillator requires an external 0.01% timing reference. The accuracy requirements, if an external crystal is used are tighter because allowance for the on-board parasitics must be made to deliver a final accuracy of 0.01%.

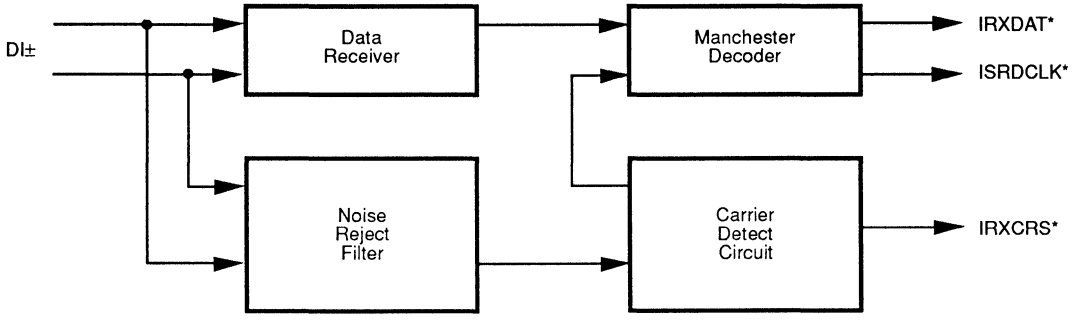
Transmission is enabled by the controller. As long as the ITXEN request remains active, the serial output of the controller will be Manchester encoded and appear at DO±. When the internal request is dropped by the controller, the differential transmit outputs go to one of two idle states, dependent on TSEL in the Mode Register (CSR15, bit 9):

TSEL LOW:	The idle state of DO± yields "ZERO" differential to operate transformer-coupled loads.
TSEL HIGH:	In this idle state, DO+ is positive with respect to DO– (logical HIGH).

Receiver Path

The principal functions of the Receiver are to signal the PCnet-PCI controller that there is information on the receive pair, and separate the incoming Manchester encoded data stream into clock and NRZ data.

The Receiver section (see Receiver Block Diagram) consists of two parallel paths. The receive data path is a ZERO threshold, wide bandwidth line receiver. The carrier path is an offset threshold bandpass detecting line receiver. Both receivers share common bias networks to allow operation over a wide input common mode range.



*Internal signal

18220C-26

Figure 24. Receiver Block Diagram

Input Signal Conditioning

Transient noise pulses at the input data stream are rejected by the Noise Rejection Filter. Pulse width rejection is proportional to transmit data rate.

The Carrier Detection circuitry detects the presence of an incoming data frame by discerning and rejecting noise from expected Manchester data, and controls the stop and start of the phase-lock loop during clock acquisition. Clock acquisition requires a valid Manchester bit pattern of 1010b to lock onto the incoming message.

When input amplitude and pulse width conditions are met at DI±, the internal enable signal from the MENDEC to controller (IRXCRS) is asserted and a clock acquisition cycle is initiated.

Clock Acquisition

When there is no activity at DI± (receiver is idle), the receive oscillator is phase locked to internal transmit clock. The first negative clock transition (bit cell center of first valid Manchester "0") after IRXCRS is asserted interrupts the receive oscillator. The oscillator is then restarted at the second Manchester "0" (bit time 4) and is phase locked to it. As a result, the MENDEC acquires the clock from the incoming Manchester bit pattern in 4 bit times with a 1010b Manchester bit pattern.

ISRDCLK and IRXDAT are enabled 1/4 bit time after clock acquisition in bit cell 5. IRXDAT is at a HIGH state when the receiver is idle (no ISRDCLK). IRXDAT however, is undefined when clock is acquired and may remain HIGH or change to LOW state whenever ISRDCLK is enabled. At 1/4 bit time through bit cell 5, the controller portion of the PCnet-PCI controller sees the first ISRDCLK transition. This also strobes in the incoming fifth bit to the MENDEC as Manchester "1". IRXDAT may make a transition after the ISRDCLK rising edge in bit cell 5, but its state is still undefined. The Manchester "1" at bit 5 is clocked to IRXDAT output at 1/4 bit time in bit cell 6.

PLL Tracking

After clock acquisition, the phase-locked clock is compared to the incoming transition at the bit cell center (BCC) and the resulting phase error is applied to a correction circuit. This circuit ensures that the phase-locked clock remains locked on the received signal. Individual bit cell phase corrections of the Voltage Controlled Oscillator (VCO) are limited to 100% of the phase

difference between BCC and phase-locked clock. Hence, input data jitter is reduced in ISRDCLK by 10 to 1.

Carrier Tracking and End of Message

The carrier detection circuit monitors the DI± inputs after IRXCRS is asserted for an end of message. IRXCRS de-asserts 1 to 2 bit times after the last positive transition on the incoming message. This initiates the end of reception cycle. The time delay from the last rising edge of the message to IRXCRS de-assert allows the last bit to be strobed by ISRDCLK and transferred to the controller section, but prevents any extra bit(s) at the end of message.

Data Decoding

The data receiver is a comparator with clocked output to minimize noise sensitivity to the DI± inputs. Input error is less than ±35 mV to minimize sensitivity to input rise and fall time. ISRDCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit, and clocks the data out on IRXDAT on the following ISRDCLK. The data receiver also generates the signal used for phase detector comparison to the internal MENDEC voltage controlled oscillator (VCO).

Differential Input Terminations

The differential input for the Manchester data (DI±) is externally terminated by two 40.2 Ω ±1% resistors and one optional common-mode bypass capacitor, as shown in the Differential Input Termination diagram below. The differential input impedance, Z_{IDF}, and the common-mode input impedance, Z_{ICM}, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. If SIP devices are used, 39 Ω is also a suitable value. The CI± differential inputs are terminated in exactly the same way as the DI± pair.

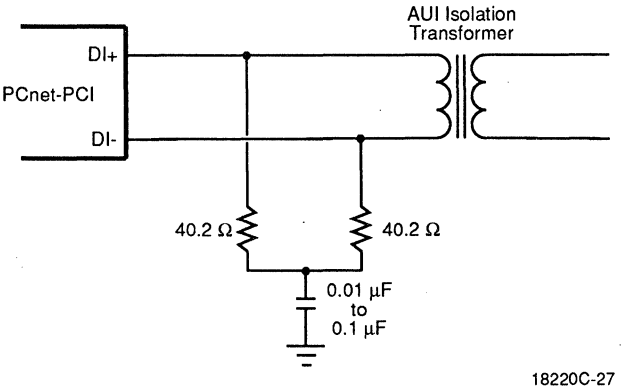


Figure 25. Differential Input Termination

Collision Detection

A MAU detects the collision condition on the network and generates a differential signal at the CI± inputs. This collision signal passes through an input stage which detects signal levels and pulse duration. When the signal is detected by the MENDEC it sets the ICLSN line HIGH. The condition continues for approximately 1.5 bit times after the last LOW-to-HIGH transition on CI±.

Jitter Tolerance Definition

The MENDEC utilizes a clock capture circuit to align its internal data strobe with an incoming bit stream. The clock acquisition circuitry requires four valid bits with the values 1010b. Clock is phase-locked to the negative transition at the bit cell center of the second "0" in the pattern.

Since data is strobed at 1/4 bit time, Manchester transitions which shift from their nominal placement through 1/4 bit time will result in improperly decoded data. With this as the criteria for an error, a definition of Jitter Handling is:

> The peak deviation approaching or crossing 1/4 bit cell position from nominal input transition, for which the MENDEC section will properly decode data.

Attachment Unit Interface (AUI)

The AUI is the PLS (Physical Layer Signaling) to PMA (Physical Medium Attachment) interface which effectively connects the DTE to a MAU. The differential interface provided by the PCnet-PCI controller is fully compliant to Section 7 of ISO 8802-3 (ANSI/IEEE 802.3).

After the PCnet-PCI controller initiates a transmission it will expect to see data "looped-back" on the DI± pair (when the AUI port is selected). This will internally generate a "carrier sense", indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted within TBD bit times after the first transmitted bit on DO± (when using the AUI port). If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in the Transmit Descriptor Ring (TMD2, bit 27) after the frame has been transmitted.

Twisted-Pair Transceiver (T-MAU)

The T-MAU implements the Medium Attachment Unit (MAU) functions for the Twisted-Pair Medium, as specified by the supplement to ISO 8802-3 (IEEE/ANSI 802.3) standard (Type 10BASE-T). The T-MAU provides twisted pair driver and receiver circuits, including on-board transmit digital predistortion and receiver squelch and a number of additional features including Link Status indication, Automatic Twisted-Pair Receive Polarity Detection/Correction and Indication, Receive Carrier Sense, Transmit Active and Collision Present indication.

Twisted-Pair Transmit Function

The differential driver circuitry in the TXD± and TXP± pins provides the necessary electrical driving capability and the pre-distortion control for transmitting signals over maximum length Twisted-Pair cable, as specified by the 10BASE-T supplement to the ISO 8802-3 (IEEE/ANSI 802.3) Standard. The transmit function for data output meets the propagation delays and jitter specified by the standard.

Twisted-Pair Receive Function

The receiver complies with the receiver specifications of the ISO 8802-3 (IEEE/ANSI 802.3) 10BASE-T Standard, including noise immunity and received signal rejection criteria ('Smart Squelch'). Signals meeting this criteria appearing at the RXD± differential input pair are routed to the MENDEC. The receiver function meets the propagation delays and jitter requirements specified by the standard. The receiver squelch level drops to half its threshold value after unsquelch to allow reception of minimum amplitude signals and to offset carrier fade in the event of worst case signal attenuation and crosstalk noise conditions.

Note that the 10BASE-T Standard defines the receive input amplitude at the external Media Dependent Interface (MDI). Filter and transformer loss are not specified. The T-MAU receiver squelch levels are defined to account for a 1 dB insertion loss at 10 MHz, which is typical for the type of receive filters/transformers employed.

Normal 10BASE-T compatible receive thresholds are employed when the LRT bit (CSR15[9]) is LOW. When the LRT bit is set (HIGH), the Low Receive Threshold option is invoked, and the sensitivity of the T-MAU receiver is increased. This allows longer line lengths to be employed, exceeding the 100 m target distance of normal 10BASE-T (assuming typical 24 AWG cable). The increased receiver sensitivity compensates for the increased signal attenuation caused by the additional cable distance.

However, making the receiver more sensitive means that it is also more susceptible to extraneous noise, primarily caused by coupling from co-resident services (crosstalk). For this reason, it is recommended that when using the Low Receive Threshold option that the service should be installed on 4-pair cable only. Multipair cables within the same outer sheath have lower crosstalk attenuation, and may allow noise emitted from adjacent pairs to couple into the receive pair, and be of sufficient amplitude to falsely unsquelch the T-MAU.

Link Test Function

The link test function is implemented as specified by 10BASE-T standard. During periods of transmit pair

inactivity, 'Link beat pulses' will be periodically sent over the twisted pair medium to constantly monitor medium integrity.

When the link test function is enabled (DLNKTST bit in CSR15 is cleared), the absence of link beat pulses and receive data on the RXD± pair will cause the TMAU to go into a link fail state. In the link fail state, data transmission, data reception, data loopback and the collision detection functions are disabled, and remain disabled until valid data or >5 consecutive link pulses appear on the RXD± pair. During link fail, the Link Status ($\overline{\text{LNKST}}$ pin) signal is inactive. When the link is identified as functional, the Link Status signal is asserted. The $\overline{\text{LNKST}}$ pin displays the Link Status signal by default.

Transmission attempts during Link Fail state will produce no network activity and will produce LCAR and CERR error indications.

In order to inter-operate with systems which do not implement Link Test, this function can be disabled by setting the DLNKTST bit in CSR15. With link test disabled, the data driver, receiver and loopback functions as well as collision detection remain enabled irrespective of the presence or absence of data or link pulses on the RXD± pair. Link Test pulses continue to be sent regardless of the state of the DLNKTST bit.

Polarity Detection and Reversal

The T-MAU receive function includes the ability to invert the polarity of the signals appearing at the RXD± pair if the polarity of the received signal is reversed (such as in the case of a wiring error). This feature allows data frames received from a reverse wired RXD± input pair to be corrected in the T-MAU prior to transfer to the MENDEC. The polarity detection function is activated following H_RESET or Link Fail, and will reverse the receive polarity based on both the polarity of any previous link beat pulses and the polarity of subsequent frames with a valid End Transmit Delimiter (ETD).

When in the Link Fail state, the T-MAU will recognize link beat pulses of either positive or negative polarity. Exit from the Link Fail state is made due to the reception of 5 – 6 consecutive link beat pulses of identical polarity. On entry to the Link Pass state, the polarity of the last 5 link beat pulses is used to determine the initial receive polarity configuration and the receiver is reconfigured to subsequently recognize only link beat pulses of the previously recognized polarity.

Positive link beat pulses are defined as received signal with a positive amplitude greater than 585 mV (LRT = HIGH) with a pulse width of 60 ns – 200 ns. This positive excursion may be followed by a negative excursion. This definition is consistent with the expected received signal at a correctly wired receiver, when a link beat pulse which fits the template of Figure 14-12 of the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

Negative link beat pulses are defined as received signals with a negative amplitude greater than 585 mV with a pulse width of 60 ns – 200 ns. This negative excursion may be followed by a positive excursion. This definition is consistent with the expected received signal at a reverse wired receiver, when a link beat pulse which fits the template of Figure 14-12 in the 10BASE-T Standard is generated at a transmitter and passed through 100 m of twisted pair cable.

The polarity detection/correction algorithm will remain "armed" until two consecutive frames with valid ETD of identical polarity are detected. When "armed", the receiver is capable of changing the initial or previous polarity configuration based on the ETD polarity.

On receipt of the first frame with valid ETD following H_RESET or link fail, the T-MAU will utilize the inferred polarity information to configure its RXD± input, regardless of its previous state. On receipt of a second frame with a valid ETD with correct polarity, the detection/correction algorithm will "lock-in" the received polarity. If the second (or subsequent) frame is not detected as confirming the previous polarity decision, the most recently detected ETD polarity will be used as the default. Note that frames with invalid ETD have no effect on updating the previous polarity decision. Once two consecutive frames with valid ETD have been received, the T-MAU will disable the detection/correction algorithm until either a Link Fail condition occurs or H_RESET is activated.

During polarity reversal, an internal POL signal will be active. During normal polarity conditions, this internal POL signal is inactive. The state of this signal can be read by software and/or displayed by LED when enabled by the LED control bits in the Bus Configuration Registers (BCR4–BCR7).

Twisted-Pair Interface Status

Three signals (XMT, RCV and COL) indicate whether the T-MAU is transmitting, receiving, or in a collision state with both functions active simultaneously. These signals are internal signals and the behavior of the LED outputs depends on how the LED output circuitry is programmed.

The T-MAU will power up in the Link Fail state and normal algorithm will apply to allow it to enter the Link Pass state. In the Link Pass state, transmit or receive activity will be indicated by assertion of RCV signal going active. If T-MAU is selected using the PORTSEL bits in CSR15, then when moving from AUI to T-MAU selection the T-MAU will be forced into the LINK Fail state.

In the Link Fail state, XMT, RCV and COL are inactive.

Collision Detect Function

Activity on both twisted pair signals RXD± and TXD± constitutes a collision, thereby causing the COL signal to be activated. (COL is used by the LED control circuits) COL will remain active until one of the two colliding signals changes from active to idle. However, transmission attempt in Link Fail state results in LCAR and CERR indication. COL stays active for 2 bit times at the end of a collision.

Signal Quality Error (SQE) Test (Heartbeat) Function

The SQE function is disabled when the 10BASE-T port is selected.

Jabber Function

The Jabber function inhibits the twisted pair transmit function of the T-MAU TXD± is active for an excessive period (20 ms – 150 ms). This prevents any one node from disrupting the network due to a 'stuck-on' or faulty transmitter. If this maximum transmit time is exceeded, the T-MAU transmitter circuitry is disabled, the JAB bit is set (CSR4, bit 1) the COL signal is asserted. Once the transmit data stream to the T-MAU is removed, an "unjab" time of 250 ms – 750 ms will elapse before the T-MAU COL and re-enables the transmit circuitry.

Power Down

The T-MAU circuitry can be made to go into power savings mode. This feature is useful in battery powered or low duty cycle systems. The T-MAU will go into power down mode when H_RESET is active, coma mode is active, or the T-MAU is not selected. Refer to the Power Savings Modes section for descriptions of the various power down modes.

Any of the three conditions listed above resets the internal logic of the T-MAU and places the device into power down mode. In this mode, the Twisted-Pair driver pins (TXD±, TXP±) are driven LOW, and the internal T-MAU status signals ($\overline{\text{LNKST}}$, RCVPOL, XMT, RCV and COL) signals are inactive.

Once H_RESET ends, coma mode is disabled, and the T-MAU is selected. The T-MAU will remain in the reset state for up to 10 µs. Immediately after the reset condition is removed, the T-MAU will be forced into the Link Fail state. The T-MAU will move to the Link Pass state only after 5 – 6 link beat pulses and/or a single received message is detected on the RD± pair.

In snooze mode, the T-MAU receive circuitry will remain enabled even while the $\overline{\text{SLEEP}}$ pin is driven LOW.

The T-MAU circuitry will always go into power down mode if H_RESET is asserted, coma mode is enabled, or the T-MAU is not selected.

10BASE-T Interface Connection

Figure 26 shows the proper 10BASE-T network interface design. Refer to the *PCnet Family Technical Manual* (PID #18216A) for more design details, and refer to Appendix A for a list of compatible 10BASE-T filter/transformer modules.

Note that the recommended resistor values and filter and transformer modules are the same as those used by the IMR (Am79C980) and the IMR+ (Am79C981).

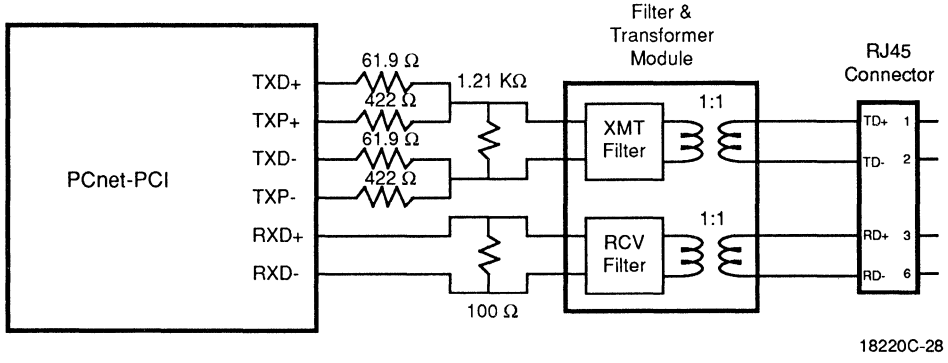


Figure 26. 10BASE-T Interface Connection

Power Savings Modes

The PCnet-PCI controller supports two hardware power savings modes. Both are entered by driving the $\overline{\text{SLEEP}}$ pin LOW.

The PCI interface section is not effected by $\overline{\text{SLEEP}}$. In particular, access to the PCI configuration space remains possible. None of the configuration registers will be reset by $\overline{\text{SLEEP}}$. All I/O accesses to the PCnet-PCI controller will result in a PCI target abort response.

The first power saving mode is called coma mode. In coma mode, the PCnet-PCI controller has no means to use the network to automatically wake itself up. Coma mode is enabled when the AWAKE bit in BCR2 is reset. Coma mode is the default power down mode.

The second power saving mode is called snooze mode. In snooze mode, enabled by setting the AWAKE bit in BCR2 and driving the $\overline{\text{SLEEP}}$ pin LOW, the T-MAU receive circuitry will remain enabled even while the $\overline{\text{SLEEP}}$ pin is driven LOW. The $\overline{\text{LNKST}}$ output is the only one of the LED pins that continues to function. The LNKSTE bit must be set in BCR4 to enable indication of a good 10BASE-T link if there are link beat pulses or valid frames present. This $\overline{\text{LNKST}}$ pin can be used to drive an LED and/or external hardware that directly controls the $\overline{\text{SLEEP}}$ pin of the PCnet-PCI controller. This configuration effectively wakes the system when there is any activity on the 10BASE-T link. Snooze mode can be used only if the T-MAU is the selected network port. Link beat pulses are not transmitted during snooze mode.

If the $\overline{\text{REQ}}$ output is active when the $\overline{\text{SLEEP}}$ pin is asserted, then the PCnet-PCI controller will wait until the $\overline{\text{GNT}}$ input is asserted. Next, the PCnet-PCI controller will deassert the $\overline{\text{REQ}}$ pin and finally, it will internally enter either the coma or snooze sleep mode.

Before the sleep mode is invoked, the PCnet-PCI controller will perform an internal S_RESET. This S_RESET operation will not affect the values of the BCR registers or the PCI configuration space.

The $\overline{\text{SLEEP}}$ pin should not be asserted during power supply ramp-up. If it is desired that $\overline{\text{SLEEP}}$ be asserted at power up time, then the system must delay the assertion of $\overline{\text{SLEEP}}$ until three CLK cycles after the completion of a valid pin $\overline{\text{RST}}$ operation.

Software Access

Configuration Registers

The PCnet-PCI controller supports the 64-byte header portion of the configuration space as predefined by the PCI specification revision 2.0. None of the device specific registers in locations 64 – 255 are used. The layout of the configuration registers in the header region is shown in the table below. All registers required to identify the PCnet-PCI controller and its function are implemented. Additional registers are used to setup the configuration of the PCnet-PCI controller in a system.

31 24	23 16	15 8	7 0	Offset
Device ID		Vendor ID		00h
Status		Command		04h
Base-Class	Sub-Class	Programming IF	Revision ID	08h
Reserved	Header Type	Latency Timer	Reserved	0Ch
Base Address				10h
Reserved				14h
Reserved				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
Reserved				28h
Reserved				2Ch
Reserved				30h
Reserved				34h
Reserved				38h
Reserved	Reserved	Interrupt Pin	Interrupt Line	3Ch

The configuration registers are accessible only by PCI configuration cycles. They can be accessed right after the PCnet-PCI controller is powered-on, even if the read operation of the serial EEPROM is still on-going. All multi-byte numeric fields follow little endian byte ordering. The Command register is the only register cleared by H_RESET. S_RESET as well as asserting $\overline{\text{SLEEP}}$ have no effect on the value of the PCI configuration registers. All write accesses to Reserved locations have no effect, reads from these locations will return a data value of ZERO.

When the PCnet-PCI controller samples its IDSEL input asserted during a configuration cycle, it will acknowledge the cycle by asserting its $\overline{\text{DEVSEL}}$ output. The content of AD[31:00] during the address phase of the configuration cycles must meet the format as shown below:

31 11	10 8	7	6	5 2	1	0
Don't Care	Don't Care	0	0	DWORD Index	0	0

AD[1:0] must both be ZEROs, since the PCnet-PCI controller is not a bridge device. It only recognizes configuration cycles of Type 0 (as defined by the PCI specification revision 2.0). AD[7:2] specify the selected DWORD in the configuration space. AD[7:6] must both be ZERO, since the PCnet-PCI controller does not implement any of the device specific registers in locations 64 – 255. Since AD[1:0] and AD[7:6] must all be ZERO, the lower 8 bits of the address for a configuration cycle are equal to the offset of the DWORD counting from the beginning of the PCI configuration space. AD[10:8] specify one of eight possible functions of a PCI device. The PCnet-PCI controller is a single function device, as indicated in the Header Type register of its PCI configuration space (bit 7, FUNCT = 0). Therefore, the PCnet-PCI controller ignores AD[10:8] during the address phase of a configuration cycle. AD[31:11] are typically used to generate the IDSEL signal. The PCnet-PCI controller ignores all upper address bits.

PCI configuration registers can be accessed with 8-bit, 16-bit or 32-bit transfers. The active bytes within a DWORD are determined by the byte enable signals. E.G. a read of the Sub-Class register can be performed by reading from offset 08h with only $\overline{\text{BE2}}$ being active.

I/O Resources

PCnet-PCI Controller I/O Resource Mapping

The PCnet-PCI controller has several I/O resources. These resources use 32 bytes of I/O space that begin at the PCnet-PCI controller I/O base address.

The PCnet-PCI controller allows two modes of slave access. Word I/O mode treats all PCnet-PCI controller I/O Resources as two-byte entities spaced at two-byte address intervals. Double Word I/O mode treats all PCnet-PCI controller I/O Resources as four-byte entities spaced at four-byte address intervals. The selection of WIO or DWIO mode is accomplished by one of two ways:

- H_RESET function.
- Automatic determination of DWIO mode due to DWORD (double-word) I/O write access to offset 10h.

The PCnet-PCI controller I/O mode setting will default to WIO after H_RESET (i.e. DWIO = 0).

Software may invoke DWIO mode by performing a Double Word write access to the I/O location at offset 10h (RDP). Note that even though the I/O resource mapping changes when the I/O mode setting changes, the RDP location offset is the same for both modes.

1-, 2- and 3-byte accesses to PCnet-PCI controller I/O resources are not allowed during DWIO mode.

The mapping of the PCnet-PCI controller resources into the 32-byte I/O space varies depending upon the setting of the DWIO bit of BCR18. Depending upon the setting of this variable, the 32-byte I/O space will be either Word I/O mapped (WIO) or Double Word I/O mapped (DWIO). A DWIO setting of 0 produces Word I/O mode, while a DWIO setting of 1 produces Double Word I/O mapping.

DWIO is automatically programmed as active when the system attempts a DWORD write access to offset 10h of the PCnet-PCI controller I/O space. The power up reset value of DWIO will be ZERO, and this value will be maintained until a DWORD access is performed to PCnet-PCI controller I/O space.

Therefore, if DWIO mode is desired, it is imperative that the first access to the PCnet-PCI controller be a DWORD write access to offset 10h.

Alternatively, if DWIO mode is not desired, then it is imperative that the software never executes a DWORD write access to offset 10h of the PCnet-PCI controller I/O space.

Once the DWIO bit has been set to a ONE, only a hardware H_RESET can reset it to a ZERO.

The DWIO mode setting is unaffected by the S_RESET or setting the STOP bit.

WIO I/O Resource Map

When the PCnet-PCI controller I/O space is mapped as Word I/O, then the resources that are allotted to the PCnet-PCI controller occur on word boundaries that are offset from the PCnet-PCI controller I/O base address as shown in the following table:

Offset	No. of Bytes	Register
0h	2	APROM
2h	2	APROM
4h	2	APROM
6h	2	APROM
8h	2	APROM
Ah	2	APROM
Ch	2	APROM
Eh	2	APROM
10h	2	RDP
12h	2	RAP (shared by RDP and BDP)
14h	2	Reset Register
16h	2	BDP
18h	2	Vendor Specific Word
1Ah	2	Reserved
1Ch	2	Reserved
1Eh	2	Reserved

When PCnet-PCI controller I/O space is Word mapped, all I/O resources fall on word boundaries and all I/O resources are word quantities. However, while in Word I/O mode, APROM locations may also be accessed as individual bytes either on odd or even byte addresses.

Attempts to write to any PCnet-PCI controller I/O resources *(except to offset 10h, RDP)* as 32 bit quantities while in Word I/O mode are illegal and may cause unexpected reprogramming of the PCnet-PCI controller control registers. Attempts to *read* from any PCnet-PCI controller I/O resources as 32-bit quantities while in Word I/O mode are illegal and will yield undefined values.

An attempt to write to offset 10h (RDP) as a 32 bit quantity while in Word I/O mode will cause the PCnet-PCI controller to exit WIO mode and immediately thereafter, to enter DWIO mode.

Word accesses to non word address boundaries are not allowed while in WIO mode. (A write access may cause unexpected reprogramming of the PCnet-PCI controller control registers. A read access will yield undefined values.)

Accesses of non word quantities to any I/O resource are not allowed while in WIO mode, with the exception of a read to APROM locations. (A write access may cause unexpected reprogramming of the PCnet-PCI controller control registers; a read access will yield undefined values.)

The Vendor Specific Word (VSW) is not implemented by the PCnet-PCI controller. This particular I/O address is reserved for customer use and will not be used by future AMD Ethernet controller products.

DWIO I/O Resource Map

When the PCnet-PCI controller I/O space is mapped as Double Word I/O, then all of the resources that are allotted to the PCnet-PCI controller occur on DWORD boundaries that are offset from the PCnet-PCI controller I/O base address as shown in the table below:

Offset	No. of Bytes	Register
0h	4	APROM
4h	4	APROM
8h	4	APROM
Ch	4	APROM
10h	4	RDP
14h	4	RAP (shared by RDP and BDP)
18h	4	Reset Register
1Ch	4	BDP

When PCnet-PCI I/O space is Double Word mapped, all I/O resources fall on DWORD boundaries. APROM resources are DWORD quantities in DWIO mode. RDP, RAP and BDP contain only two bytes of valid data; the other two bytes of these resources are reserved for future use. (Note that CSR88 is an exception to this rule.) The reserved bits must be written as ZEROs, and when read, are considered undefined.

Accesses to non-doubleword address boundaries are not allowed while in DWIO mode. (A write access may cause unexpected reprogramming of the PCnet-PCI controller control registers; a read access will yield undefined values.)

Accesses of less than 4 bytes to any I/O resource are not allowed while in DWIO mode. (A write access may cause unexpected reprogramming of the PCnet-PCI controller control registers; a read access will yield undefined values.)

A DWORD write access to the RDP offset of 10h will automatically program DWIO mode.

Note that in all cases when I/O resource width is defined as 32 bits, the upper 16 bits of the I/O resource is reserved and written as ZEROS and read as undefined, except for the APROM locations and CSR88.

DWIO mode is exited by asserting the $\overline{RST}$ pin. Assertion of S_RESET or setting the STOP bit of CSR0 will have no effect on the DWIO mode setting.

I/O Space Comments

The following statements apply to both WIO and DWIO mapping:

The RAP is shared by the RDP and the BDP.

The PCnet-PCI controller does not respond to any addresses outside of the offset range 0h–17h when DWIO = 0 or 0h–1Fh when DWIO = 1. I/O offsets 18h through 1Fh are not used by the PCnet-PCI controller when programmed for DWIO = 0 mode; locations 1Ah through 1Fh are reserved for future AMD use and therefore should not be implemented by the user if upward compatibility to future AMD devices is desired.

Note that APROM accesses do not directly access the EEPROM, but are redirected to a set of shadow registers on board the PCnet-PCI controller that contain a copy of the EEPROM contents that was obtained during the automatic EEPROM read operation that follows the H_RESET operation.

PCnet-PCI Controller I/O Base Address

The PCI Configuration Space Base Address register defines what I/O base address the PCnet-PCI controller uses. This register is typically programmed by the PCI configuration utility after system power-up. The PCI configuration utility must also set the IOEN bit in the COMMAND register to enable I/O accesses to the PCnet-PCI controller.

The content of the PCnet-PCI I/O Base Address Registers (BCR16 and BCR17) are ignored.

I/O Register Access

All I/O resources are accessed with similar I/O bus cycles.

I/O accesses to the PCnet-PCI controller begin with a valid $\overline{FRAME}$ signal, the C/$\overline{BE}$[3:0] lines signaling an I/O read or I/O write operation and an address on the AD[31:00] lines that falls within the I/O space of the PCnet-PCI controller. The PCnet-PCI I/O space will be determined by the Base Address Register in the PCI Configuration Space.

The PCnet-PCI controller will respond to an access to its I/O space by asserting the $\overline{DEVSEL}$ signal and eventually, by asserting the $\overline{TRDY}$ signal.

Typical I/O access times are 6 or 7 clock cycles.

APROM Access

The APROM space is a convenient place to store the value of the 48-bit IEEE station address. This space is automatically loaded from the serial EEPROM, if an EEPROM is present. It can be overwritten by the host computer. Its contents have no effect on the operation of the controller. The software must copy the station address from the APROM space to the initialization block or to CSR12-14 in order for the receiver to accept unicast frames directed to this station.

When programmed for WIO mode, any byte or word address from an offset of 0h to an offset of Fh may be read. An appropriate byte or word of APROM contents will be delivered by the PCnet-PCI controller in response to accesses that fall within the APROM range of 0h to Fh.

When programmed for DWIO mode, only DWORD addresses from an offset of 0h to an offset of Fh may be read. An appropriate DWORD of APROM contents will be delivered in response to accesses that fall within the APROM range of 0h to Fh.

Accesses of non-DWORD *quantities* are not allowed in DWIO mode, even though such an access may be properly aligned to a DWORD address boundary.

Write access to any of the APROM locations is allowed, but only 4 bytes on DWORD boundaries in DWIO mode or 2 bytes on word boundaries in WIO mode (only read accesses to the APROM locations can be in 8-bit quantities while in WIO mode). The IESRWE bit (see BCR2) must be set in order to enable a write operation to the APROM. Only the PCnet-PCI controller on-board IEEE Shadow registers are modified by writes to APROM locations. The EEPROM is unaffected by writes to APROM locations.

Note that the APROM locations occupy 16 bytes of space, yet the IEEE station address requirement is for 6 bytes. The 6 bytes of IEEE station address occupy the first 6 locations of the APROM space. The next six bytes are reserved. Bytes 12 and 13 should match the value of the checksum of bytes 1 through 11 and 14 and 15. Bytes 14 and 15 should each be ASCII W (57h). The above requirements must be met in order to be compatible with AMD driver software.

RDP Access (CSR Register Space)

RDP = Register Data Port. The RDP is used with the RAP to gain access to any of the PCnet-PCI controller CSR locations.

Access to any of the CSR locations of the PCnet-PCI controller is performed through the PCnet-PCI controllers Register Data Port (RDP). In order to access a particular CSR location, the Register Address Port (RAP) should first be written with the appropriate CSR address. The RDP now points to the selected CSR. A read of the RDP will yield the selected CSRs data. A write to the RDP will write to the selected CSR.

When programmed for WIO mode, the RDP has a width of 16 bits, hence, all CSR locations have 16 bits of width. Note that when accessing RDP, the upper two bytes of the data bus will be undefined since the byte masks will not be active for those bytes.

If DWIO mode has been invoked, then the RDP has a width of 32 bits, hence, all CSR locations have 32 bits of width and the upper two bytes of the data bus will be active, as indicated by the byte mask. In this case, note that the upper 16 bits of all CSR locations (except CSR88) are reserved and written as ZEROs and read as undefined values. Therefore, during RDP write operations in DWIO mode, the upper 16 bits of all CSR locations should be written as ZEROs.

RAP Access

RAP = Register Address Port. The RAP is used with the RDP and with the BDP to gain access to any of the CSR and BCR register locations, respectively. The RAP contains the address pointer that will be used by an access to either the RDP or BDP. Therefore, it is necessary to set the RAP value before accessing a specific CSR or BCR location. Once the RAP has been written with a value, the RAP value remains unchanged until another RAP write occurs, or until an H_RESET or S_RESET occurs. RAP is set to all ZEROs when an H_RESET or S_RESET occurs. RAP is unaffected by the STOP bit.

When programmed for WIO mode, the RAP has a width of 16 bits. Note that when accessing RAP, the lower two bytes of the data bus will be undefined since the byte masks will not be active for those bytes

When programmed for DWIO mode, the RAP has a width of 32 bits. In DWIO mode, the upper 16 bits of the RAP are reserved and written as ZEROs and read as undefined. These bits should be written as ZEROs.

BDP Access (BCR Register Space)

BDP = Bus Configuration Register Data Port. The BDP is used with the RAP to gain access to any of the PCnet-PCI controller BCR locations.

Access to any of the BCR locations of the PCnet-PCI controller is performed through the PCnet-PCI controllers BCR Data Port (BDP); in order to access a particular BCR location, the Register Address Port (RAP) should first be written with the appropriate BCR address. The BDP now points to the selected BCR. A read of the BDP will yield the selected BCR s data. A write to the BDP will write to the selected BCR.

When programmed for WIO mode, the BDP has a width of 16 bits, hence, all BCR locations have 16 bits of width in WIO mode. Note that when operating in WIO mode,

the upper two bytes of the data bus will be undefined since the byte mask will not be active for those bytes.

If DWIO mode has been invoked, then the BDP has a width of 32 bits, hence, all BCR locations have 32 bits of width and the upper two bytes of the data bus will be active, as indicated by the byte mask. In this case, note that the upper 16 bits of all BCR locations are reserved and written as ZEROs and read as undefined. Therefore, during BDP write operations in DWIO mode, the upper 16 bits of all BCR locations should be written as ZEROs.

RESET Register (S_RESET)

A read of the reset register creates an internal S_RESET pulse in the PCnet-PCI controller. This read access cycle must be 16 bits wide in WIO mode and 32 bits wide in DWIO mode. The internal S_RESET pulse that is generated by this access is different from both the assertion of the hardware $\overline{RST}$ pin (H_RESET) and from the assertion of the software STOP bit. Specifically, the reset registers S_RESET will be the equivalent of the assertion of the $\overline{RST}$ pin (H_RESET) assertion for all CSR locations, but S_RESET will have no effect at all on the BCR or PCI configuration space locations, and S_RESET will not cause a deassertion of the $\overline{REQ}$ pin.

The NE2100 LANCE based family of Ethernet cards requires that a write access to the reset register follows each read access to the reset register. The PCnet-PCI controller does not have a similar requirement. The write access is not required but it does not have any harmful effects.

Write accesses to the reset register will have no effect on the PCnet-PCI controller.

Note that a read access of the reset register will take longer than the normal I/O access time of the PCnet-PCI controller. This is because an internal S_RESET pulse will be generated due to this access, and the access will not be allowed to complete on the system bus until the internal S_RESET operation has been completed. This is to avoid the problem of allowing a new I/O access to proceed while the S_RESET operation has not yet completed, which would result in erroneous data being returned by (or written into) the PCnet-PCI controller. The length of a read of the Reset register can be as long as 64 clock cycles.

Note that a read of the reset register will **not** cause a deassertion of the $\overline{REQ}$ signal, if it happens to be active at the time of the read of the reset register. The $\overline{REQ}$ signal will remain active until the $\overline{GNT}$ signal is asserted. Following the read of the reset register, on the next clock cycle after the $\overline{GNT}$ signal is asserted, the PCnet-PCI controller will deassert the $\overline{REQ}$ signal. No bus master accesses will have been performed during this brief bus ownership period.

Note that this behavior differs from that which occurs following the assertion of a minimum-width pulse on the $\overline{RST}$ pin (H_RESET). A $\overline{RST}$ pin assertion will cause the $\overline{REQ}$ signal to deassert within six clock cycles following the assertion. In the $\overline{RST}$ pin case, the PCnet-PCI controller will not wait for the assertion of the $\overline{GNT}$ signal before deasserting the $\overline{REQ}$ signal.

Vendor Specific Word

This I/O offset is reserved for use by the system designer. The PCnet-PCI controller will not respond to accesses directed toward this offset. The Vendor Specific Word is only available when the PCnet-PCI controller is programmed to word I/O mode (DWIO = 0).

If more than one Vendor Specific Word is needed, it is suggested that the VSW location should be divided into a VSW Register Address Pointer (VSWRAP) at one location (e.g. VSWRAP at byte location 18h or word location 30h, depending upon DWIO state) and a VSW Data Port (VSWDP) at the other location (e.g. VSWDP at byte location 19h or word location 32h, depending upon DWIO state). Alternatively, the system may capture RAP data accesses in parallel with the PCnet-PCI controller and therefore share the PCnet-PCI controller RAP to allow expanded VSW space. PCnet-PCI controller will not respond to access to the VSW I/O address.

Reserved I/O Space

These locations are reserved for future use by AMD. The PCnet-PCI controller does not respond to accesses directed toward these locations, but future AMD products that are intended to be upward compatible with the PCnet-PCI controller device may decode accesses to these locations. Therefore, the system designer may not utilize these I/O locations.

Hardware Access

PCnet-PCI Controller Master Accesses

The PCnet-PCI controller has a bus interface compatible with PCI specification revision 2.0.

Complete descriptions of the signals involved in bus master transactions for each mode may be found in the pin description section of this document. Timing diagrams for master accesses may be found in the block description section for the Bus Interface Unit. This section simply lists the types of master accesses that will be performed by the PCnet-PCI controller with respect to data size and address information.

The PCnet-PCI controller will support master accesses only to 32-bit peripherals. The PCnet-PCI controller does not support master accesses to 8-bit or 16-bit memory. The PCnet-PCI controller is not compatible with 8-bit systems, since there is no mode that supports PCnet-PCI controller accesses to 8-bit peripherals.

Table 6 describes all possible bus master accesses that the PCnet-PCI controller will perform. The right most column lists all operations that may execute the given access:

Table 6. Bus Master Accesses

Access	Mode	$\overline{BE}$[3:0]	Operation
4-byte read	Read	0000	descriptor read or initialization block read or transmit data buffer read
4-byte write	Write	0000	descriptor write or receive data buffer write
3-byte write	Write	1000	receive data buffer write
3-byte write	Write	0001	receive data buffer write
2-byte write	Write	1100	receive data buffer write
2-byte write	Write	1001*	receive data buffer write
2-byte write	Write	0011	receive data buffer write
1-byte write	Write	1110	receive data buffer write
1-byte write	Write	1101*	receive data buffer write
1-byte write	Write	1011*	receive data buffer write
1-byte write	Write	0111	descriptor write or receive data buffer write

** Cases marked with an asterisk represent extreme boundary conditions that are the result of programming one- and two-byte buffer sizes, and therefore will not be seen under normal circumstances.*

Note that all PCnet-PCI controller master read operations will always activate all byte enables. Therefore, no one-, two- or three-byte read operations are indicated in the table.

In the instance where a transmit buffer pointer address begins on a non-DWORD boundary, the pointer will be truncated to the next DWORD boundary address that lies below the given pointer address and the first read access from the transmit buffer will be indicated on the byte enable signals as a four-byte read from this address. Any data from byte lanes that lie outside of the boundary indicated by the buffer pointer will be discarded inside of the PCnet-PCI controller. Similarly, if the end of a transmit buffer occurs on a non-DWORD boundary, then all byte lanes will be indicated as active by the byte enable signals, and any data from byte lanes that lie outside of the boundary indicated by the buffer pointer will be discarded inside of the PCnet-PCI controller.

Slave Access to I/O Resources

The PCnet-PCI device is always a 32-bit peripheral on the system bus. However, the width of individual software resources on board the PCnet-PCI controller may be either 16-bits or 32-bits. The PCnet-PCI controller I/O resource widths are determined by the setting of the DWIO bit as indicated in the following table:

DWIO Setting	PCnet-PCI Controller I/O Resource Width	Example Application
DWIO = 0	16-bit	Existing PCnet-ISA driver that assumes 16-bit I/O mapping and 16-bit resource widths
DWIO = 1	32-bit	New drivers written specifically for the PCnet-PCI controller

Note that when I/O resource width is defined as 32 bits (DWIO mode), the upper 16 bits of the I/O resource is reserved and written as ZEROS and read as undefined, except for the APROM locations and CSR88. The APROM locations and CSR88 are the only I/O resources for which all 32 bits will have defined values. However, this is true only when the PCnet-PCI controller is in DWIO mode.

Configuring the PCnet-PCI controller for DWIO mode is accomplished whenever there is any attempt to perform a 32-bit write access to the RDP location (offset 10h). See the DWIO section for more details.

Table 7 describes all possible bus slave accesses that may be directed toward the PCnet-PCI controller. (i.e., the PCnet-PCI controller is the target device during the transfer.) The first column indicates the type of slave access. RD stands for READ, WR for a WRITE operation. The second column indicates the value of the C/$\overline{BE}$[3:0] lines during the data phase of the transfer. The four byte columns (AD[31:24], AD[23:16], AD[15:8], AD[7:0]) indicate the value on the address/data bus during the data phase of the access. "data" indicates the position of the active bytes; "copy" indicates the positions of copies of the active bytes; "undef" indicates byte locations that are undefined during the transfer.

Table 7. Bus Slave Accesses

TYPE	$\overline{BE}$[3:0]	AD [31:24]	AD [23:16]	AD [15:8]	AD [7:0]	Comments
RD	0000	data	data	data	data	DWORD access to DWORD address, e.g. 300h, 30Ch, 310h (DWIO mode only)
RD	1100	undef	undef	data	data	word access to even word address, e.g. 300h, 30Ch, 310h (WIO mode only)
RD	0011	data	data	copy	copy	word access to odd word address, e.g. 302h, 30Eh, 312h (WIO mode only)
RD	1110	undef	undef	undef	data	byte access to lower byte of even word address, e.g. 300h, 304h (WIO mode only, APROM accesses only)
RD	1101	undef	undef	data	undef	byte access to upper byte of even word address, e.g. 301h, 305h (WIO mode only, APROM accesses only)
RD	1011	undef	data	undef	copy	byte access to lower byte of odd word address, e.g. 302h, 306h (WIO mode only, APROM accesses only)
RD	0111	data	undef	copy	undef	byte access to upper byte of odd word address, e.g. 303h, 307h (WIO mode only, APROM accesses only)
WR	0000	data	data	data	data	DWORD access to DWORD address, e.g. 300h, 30Ch, 310h (DWIO mode only)
WR	1100	undef	undef	data	data	word access to even word address, e.g. 300h, 30Ch, 310h (WIO mode only)
WR	0011	data	data	undef	undef	word access to odd word address, e.g. 302h, 30Eh, 312h (WIO mode only)

EEPROM Microwire Access

The PCnet-PCI controller contains a built-in capability for reading and writing to an external EEPROM. This built-in capability consists of a Microwire interface for direct connection to a Microwire compatible EEPROM, an automatic EEPROM read feature, and a user-programmable register that allows direct access to the Microwire interface pins.

Automatic EEPROM Read Operation

Shortly after the deassertion of the $\overline{RST}$ pin, the PCnet-PCI controller will read the contents of the EEPROM that is attached to the Microwire interface. Because of this automatic-read capability of the PCnet-PCI controller, an EEPROM can be used to program many of the features of the PCnet-PCI controller at power-up, allowing system-dependent configuration information to be stored in the hardware, instead of inside of operating code.

If an EEPROM exists on the Microwire interface, the PCnet-PCI controller will read the EEPROM contents at the end of the H_RESET operation. The EEPROM contents will be serially shifted into a temporary register and then sent to various register locations on board the PCnet-PCI controller. The host can access the PCI Configuration Space during the EEPROM read operations. Access to the PCnet-PCI I/O resources, however, is not possible during the EEPROM read operation. The PCnet-PCI controller will terminate these I/O accesses with the assertion of $\overline{DEVSEL}$ and $\overline{STOP}$ while $\overline{TRDY}$ is not asserted, signaling to the initiator to retry the access at a later time.

A checksum verification is performed on the data that is read from the EEPROM. If the checksum verification of the EEPROM data fails, then at the end of the EEPROM read sequence, the PCnet-PCI controller will force all EEPROM-programmable BCR registers back to their

H_RESET default values. The content of the APROM locations (offsets 0h – Fh from the I/O base address), however, will not be cleared. The 8-bit checksum for the entire 36 bytes of the EEPROM should be FFh.

If no EEPROM is present at the time of the automatic read operation, then the PCnet-PCI controller will recognize this condition and will abort the automatic read operation and reset both the PREAD and PVALID bits in BCR19. All EEPROM-programmable BCR registers will be assigned their default values after H_RESET. The content of the Address PROM locations (offsets 0h – Fh from the I/O base address) will be undefined.

If the user wishes to modify any of the configuration bits that are contained in the EEPROM, then the seven command, data and status bits of BCR19 can be used to write to the EEPROM. After writing to the EEPROM, the host should set the PREAD bit of BCR19. This action forces a PCnet-PCI controller re-read of the EEPROM so that the new EEPROM contents will be loaded into the EEPROM-programmable registers on board the PCnet-PCI controller. (The EEPROM-programmable registers may also be reprogrammed directly, but only information that is stored in the EEPROM will be preserved at system power-down.) When the PREAD bit of BCR19 is set, it will cause the PCnet-PCI controller to terminate further accesses to internal I/O resources with the PCI retry cycle. Accesses to the PCI configuration space is still possible.

EEPROM Auto-Detection

The PCnet-PCI controller uses the EESK/$\overline{\text{LED1}}$ pin to determine if an EEPROM is present in the system. At all rising CLK edges during the assertion of the $\overline{\text{RST}}$ pin, the PCnet-PCI controller will sample the value of the EESK/$\overline{\text{LED1}}$ pin. If the sampled value is a ONE, then the PCnet-PCI controller assumes that an EEPROM is present, and the EEPROM read operation begins shortly after the $\overline{\text{RST}}$ pin is deasserted. If the sampled value of EESK/$\overline{\text{LED1}}$ is a ZERO, then the PCnet-PCI controller assumes that an external pulldown device is holding the EESK/$\overline{\text{LED1}}$ pin low, and therefore, there is no EEPROM in the system. Note that if the designer creates a system that contains an LED circuit on the EESK/$\overline{\text{LED1}}$ pin but has no EEPROM present, then the EEPROM auto-detection function will incorrectly conclude that an EEPROM is present in the system. However, this will not pose a problem for the PCnet-PCI controller, since it will recognize the lack of an EEPROM at the end of the read operation, when the checksum verification fails.

Systems Without an EEPROM

Some systems may be able to save the cost of an EEPROM by storing the ISO 8802-3 (IEEE/ANSI 802.3) station address and other configuration information somewhere else in the system. There are several design choices:

- If the LED1 is not needed in the system, then the system designer may connect the EESK/$\overline{\text{LED1}}$ pin to a resistive pulldown device. This will indicate to the EEPROM auto-detection function that no EEPROM is present.
- If the LED1 function is needed in the system, then the system designer will connect the EESK/$\overline{\text{LED1}}$ pin to a resistive pullup device and the EEPROM auto-detection function will incorrectly conclude that an EEPROM is present in the system. However, this will not pose a problem for the PCnet-PCI controller, since it will recognize the lack of an EEPROM at the end of the read operation, when the checksum verification fails.

In either case, following the PCI configuration, additional information, including the ISO 8802-3 (IEEE/ANSI 802.3) station address, may be loaded into the PCnet-PCI controller. Note that the IESRWE bit (bit 8 of BCR2) must be set before the PCnet-PCI controller will accept writes to the APROM offsets within the PCnet-PCI I/O resources map. Startup code in the system BIOS can perform the PCI configuration accesses, the IESRWE bit write, and the APROM writes.

Direct Access to the Microwire Interface

The user may directly access the Microwire port through the EEPROM register, BCR19. This register contains bits that can be used to control the Microwire interface pins. By performing an appropriate sequence of I/O accesses to BCR19, the user can effectively write to and read from the EEPROM. This feature may be used by a system configuration utility to program hardware configuration information into the EEPROM.

EEPROM-Programmable Registers

The following registers contain configuration information that will be programmed automatically during the EEPROM read operation:

1. I/O offsets 0h – Fh	APROM locations
2. BCR2	Miscellaneous Configuration register
3. BCR16	not used in the PCnet-PCI controller
4. BCR17	not used in the PCnet-PCI controller
5. BCR18	Burst Size and Bus Control Register
6. BCR21	Not Used

If the PREAD bit (BCR19) is reset to ZERO and the PVALID bit (BCR19) is reset to ZERO, then the EEPROM read has experienced a failure and the contents of the EEPROM programmable BCR register will be set to default H_RESET values. The content of the APROM locations, however, will not be cleared.

Note that accesses to the APROM I/O locations do not directly access the Address EEPROM itself. Instead, these accesses are routed to a set of shadow registers on board the PCnet-PCI controller that are loaded with a copy of the EEPROM contents during the automatic read operation that immediately follows the H_RESET operation.

EEPROM MAP

The automatic EEPROM read operation will access 18 words (i.e. 36 bytes) of the EEPROM. The format of the EEPROM contents is shown in Table 8, beginning with the byte that resides at the lowest EEPROM address:

Table 8. EEPROM Contents

EEPROM WORD Address	EEPROM Contents			
	Byte Addr.	Most Significant Byte	Byte Addr.	Least Significant Byte
00h (lowest EEPROM address)	01h	2nd byte of the ISO 8802-3 (IEEE/ANSI 802.3) station physical address for this node	00h	first byte of the ISO 8802-3 (IEEE/ANSI 802.3) station physical address for this node, where "first byte" refers to the first byte to appear on the 802.3 medium
01h	03h	4th byte of the node address	02h	3rd byte of the node address
02h	05h	6th byte of the node address	04h	5th byte of the node address
03h	07h	reserved location: must be 00h	06h	reserved location must be 00h
04h	09h	Hardware ID: must be 11h if compatibility to AMD drivers is desired	08h	reserved location must be 00h
05h	0Bh	user programmable space	0Ah	user programmable space
06h	0Dh	MSByte of two-byte checksum, which is the sum of bytes 00h–0Bh and bytes 0Eh and 0Fh	0Ch	LSByte of two-byte checksum, which is the sum of bytes 00h–0Bh and bytes 0Eh and 0Fh
07h	0Fh	must be ASCII "W" (57h) if compatibility to AMD driver software is desired	0Eh	must be ASCII "W" (57h) if compatibility to AMD driver software is desired
08h	11h	BCR16[15:8] (not used)	10h	BCR16[7:0] (not used)
09h	13h	BCR17[15:8] (not used)	12h	BCR17[7:0] (not used)
0Ah	15h	BCR18[15:8] (Burst Size and Bus Control)	14h	BCR18[7:0] (Burst Size and Bus Control)
0Bh	17h	BCR2[15:8] (Misc. configuration)	16h	BCR2[7:0] (Misc. configuration)
0Ch	19h	BCR21[15:8] (Not Used)	18h	BCR21[7:0] (Not Used)
0Dh	1Bh	reserved location must be 00h	1Ah	reserved location must be 00h
0Eh	1Dh	reserved location must be 00h	1Ch	reserved location must be 00h
0Fh	1Fh	checksum adjust byte for the first 36 bytes of the EEPROM contents; checksum of the first 36 bytes of the EEPROM should total to FFh	1Eh	reserved location must be 00h
10h	21h	reserved location must be 00h	20h	reserved location must be 00h
11h	23h	user programmable byte locations	22h	user programmable byte locations

Note that the first bit out of any WORD location in the EEPROM is treated as the MSB of the register that is being programmed. For example, the first bit out of EEPROM WORD location 08h will be written into BCR16[15], the second bit out of EEPROM WORD location 08h will be written into BCR16[14], etc.

There are two checksum locations within the EEPROM. The first is required for the EEPROM address. This checksum will be used by AMD driver software to verify that the ISO 8802-3 (IEEE/ANSI 802.3) station address has not been corrupted. The value of bytes 0Ch and 0Dh should match the sum of bytes 00h through 0Bh and 0Eh and 0Fh. The second checksum location – byte 1Fh – is not a checksum total, but is, instead, a checksum adjustment. The value of this byte should be such that the total checksum for the entire 36 bytes of EEPROM data equals the value FFh. The checksum adjust byte is needed by the PCnet-PCI controller in order to verify that the EEPROM contents have not been corrupted.

Transmit Operation

The transmit operation and features of the PCnet-PCI controller are controlled by programmable options. The PCnet-PCI controller offers a136-byte Transmit FIFO to provide frame buffering for increased system latency, automatic retransmission with no FIFO reload, and automatic transmit padding.

Transmit Function Programming

Automatic transmit features such as retry on collision, FCS generation/transmission, and pad field insertion can all be programmed to provide flexibility in the (re-)transmission of messages.

Disable retry on collision (DRTY) is controlled by the DRTY bit of the Mode register (CSR15) in the initialization block.

Automatic pad field insertion is controlled by the APAD_XMT bit in CSR4. If APAD_XMT is set, automatic pad field insertion is enabled, the DXMTFCS feature is over-ridden, and the 4-byte FCS will be added to the transmitted frame unconditionally. If APAD_XMT is clear, no pad field insertion will take place and runt packet transmission is possible.

The disable FCS generation/transmission feature can be programmed dynamically on a frame by frame basis. See the ADD_FCS description of TMD1.

Transmit FIFO Watermark (XMTFW) in CSR80 sets the point at which the BMU requests more data from the transmit buffers for the FIFO. A minimum of XMTFW empty spaces must be available in the transmit FIFO before the BMU will request the system bus in order to transfer transmit packet data into the transmit FIFO.

Transmit Start Point (XMTSP) in CSR80 sets the point when the transmitter actually attempts to transmit a frame onto the media. A minimum of XMTSP bytes must be written to the transmit FIFO for the current frame before transmission of the current frame will begin. (When automatically padded packets are being sent, it is conceivable that the XMTSP is not reached when all of the data has been transferred to the FIFO. In this case, the transmission will begin when all of the packet data has been placed into the transmit FIFO.)

When the entire frame is in the FIFO, attempts at transmission of preamble will commence regardless of the value in XMTSP. The default value of XMTSP is 10b, meaning there has to be 64 bytes in the Transmit FIFO to start a transmission.

Automatic Pad Generation

Transmit frames can be automatically padded to extend them to 64 data bytes (excluding preamble). This allows the minimum frame size of 64 bytes (512 bits) for 802.3/Ethernet to be guaranteed with no software intervention from the host/controlling process.

Setting the APAD_XMT bit in CSR4 enables the automatic padding feature. The pad is placed between the LLC data field and FCS field in the 802.3 frame.FCS is always added if the frame is padded, regardless of the state of DXMTFCS. The transmit frame will be padded by bytes with the value of 00h. The default value of APAD_XMT is 0; this will disable auto pad generation after H_RESET.

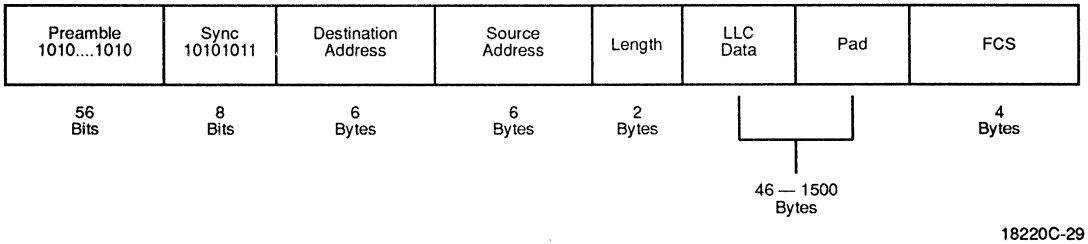


Figure 27. ISO 8802-3(IEEE/ANSI 802.3) Data Frame

It is the responsibility of upper layer software to correctly define the actual length field contained in the message to correspond to the total number of LLC Data bytes encapsulated in the packet (length field as defined in the ISO 8802-3 (IEEE/ANSI 802.3) standard). The length value contained in the message is not used by the PCnet-PCI controller to compute the actual number of pad bytes to be inserted. The PCnet-PCI controller will append pad bytes dependent on the actual number of bits transmitted onto the network. Once the last data byte of the frame has completed, prior to appending the FCS, the PCnet-PCI controller will check to ensure that 544 bits have been transmitted. If not, pad bytes are added to extend the frame size to this value, and the FCS is then added.

The 544 bit count is derived from the following:

Minimum frame size (excluding preamble, including FCS)	64	bytes	512	bits
Preamble/SFD size	8	bytes	64	bits
FCS size	4	bytes	32	bits

To be classed as a minimum size frame at the receiver, the transmitted frame must contain:

Preamble + (Min Frame Size + FCS) bits

At the point that FCS is to be appended, the transmitted frame should contain:

Preamble + (Min Frame Size − FCS) bits

64 + (512 − 32) bits

A minimum length transmit frame from the PCnet-PCI controller will therefore be 576 bits, after the FCS is appended.

The Ethernet specification assumes that minimum length messages will be at least 64 bytes in length.

Transmit FCS Generation

Automatic generation and transmission of FCS for a transmit frame depends on the value of DXMTFCS bit in CSR15. When DXMTFCS = 0 the transmitter will generate and append the FCS to the transmitted frame. If the automatic padding feature is invoked (APAD_XMT is set in CSR4), the FCS will be appended by the PCnet-PCI controller regardless of the state of DXMTFCS. Note that the calculated FCS is transmitted most significant bit first. The default value of DXMTFCS is 0 after H_RESET.

Transmit Exception Conditions

Exception conditions for frame transmission fall into two distinct categories. Those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-PCI controller include collisions within the slot time with automatic retry. The PCnet-PCI controller will ensure that collisions which occur within 512 bit times from the start of transmission (including preamble) will be automatically retried with no host intervention. The transmit FIFO ensures this by guaranteeing that data contained within the FIFO will not be overwritten until at least 64 bytes (512 bits) of preamble plus address, length and data fields have been transmitted onto the network without encountering a collision.

If 16 total attempts (initial attempt plus 15 retries) fail, the PCnet-PCI controller sets the RTRY bit in the current transmit TDTE in host memory (TMD2), gives up ownership (resets the OWN bit to ZERO) for this frame, and processes the next frame in the transmit ring for transmission.

Abnormal network conditions include:

- Loss of carrier.
- Late collision.
- SQE Test Error. (does not apply to 10BASE-T port)

These should not occur on a correctly configured 802.3 network, and will be reported if they do.

When an error occurs in the middle of a multi-buffer frame transmission, the error status will be written in the current descriptor. The OWN bit(s) in the subsequent descriptor(s) will be reset until the STP (the next frame) is found.

Loss of Carrier

A loss of carrier condition will be reported if the PCnet-PCI controller cannot observe receive activity whilst it is transmitting on the AUI port. After the PCnet-PCI controller initiates a transmission it will expect to see data "looped-back" on the DI± pair. This will internally generate a "carrier sense", indicating that the integrity of the data path to and from the MAU is intact, and that the MAU is operating correctly. This "carrier sense" signal must be asserted about 6 bit times before the last transmitted bit on DO±. If "carrier sense" does not become active in response to the data transmission, or becomes inactive before the end of transmission, the loss of carrier (LCAR) error bit will be set in TMD2 after the frame has been transmitted. The frame will not be re-tried on the basis of an LCAR error.

When the 10BASE-T port is selected, LCAR will be reported for every packet transmitted during the Link fail condition.

Late Collision

A late collision will be reported if a collision condition occurs after one slot time (512 bit times) after the transmit process was initiated (first bit of preamble commenced). The PCnet-PCI controller will abandon the transmit process for the particular frame, set Late Collision (LCOL) in the associated TMD2, and process the next transmit frame in the ring. Frames experiencing a late collision will not be re-tried. Recovery from this condition must be performed by upper layer software.

SQE Test Error

During the inter packet gap time following the completion of a transmitted message, the AUI CI± pair is asserted by some transceivers as a self-test. The integral Manchester Encoder/Decoder will expect the SQE Test Message (nominal 10 MHz sequence) to be returned via the CI± pair, within a 40 network bit time period after DI±

goes inactive (this does not apply if the 10BASE-T port is selected). If the CI± input is not asserted within the 40 network bit time period following the completion of transmission, then the PCnet-PCI controller will set the CERR bit in CSR0. CERR will be asserted in 10BASE-T mode after transmit if T-MAU is in Link Fail state. CERR will never cause $\overline{\text{INTA}}$ to be activated. It will, however, set the ERR bit CSR0.

Receive Operation

The receive operation and features of the PCnet-PCI controller are controlled by programmable options.

Address Matching

The PCnet-PCI controller supports three types of address matching: unicast, multicast, and broadcast. The normal address matching procedure can be modified by programming three bits in the MODE register (PROM, DRCVBC, and DRCBC).

If the first bit received after the start of frame delimiter (the least significant bit of the first byte of the destination address field) is 0, the frame is unicast, which indicates that the frame is meant to be received by a single node. If the first bit received is 1, the frame is multicast, which indicates that the frame is meant to be received by a group of nodes. If the destination address field contains all ones, the frame is broadcast, which is a special type of multicast. Frames with the broadcast address in the destination address field are meant to be received by all nodes on the local area network.

When a unicast frame arrives at the PCnet-PCI controller, the controller will accept the frame if the destination address field of the incoming frame exactly matches the 6-byte station address stored in the PADR registers (CSR12, CSR13, and CSR14). The byte ordering is such that the first byte received from the network (after the SFD) must match the least significant byte of CSR12 (PADR[7:0]), and the sixth byte received must match the most significant byte of CSR14 (PADR[47:40]).

If DRCVPA (bit 13 in the MODE register) is set. the PCnet-PCI controller will not accept unicast frames.

If the incoming frame is multicast the PCnet-PCI controller performs a calculation on the contents of the destination address field to determine whether or not to accept the frame. This calculation is explained in the section that describes the Logical Address Filter (LADRF).

If all bits of the LADRF registers are 0 no multicast frames are accepted, except for broadcast frames.

Although broadcast frames are classified as special multicast frames, they are treated differently by the PCnet-PCI controller hardware. Broadcast frames are always accepted, except when DRCVBC (bit 14 in the MODE register) is set.

None of the address filtering described above applies when the PCnet-PCI controller is operating in the promiscuous mode. In the promiscuous mode, all properly formed packets are received, regardless of the contents of their destination address fields. The promiscuous mode overrides the Disable Receive Broadcast bit (DRCVBC bit l4 in the MODE register) and the Disable Receive Physical Address bit (DRCVPA, bit 13 MODE register).

The PCnet-PCI controller operates in promiscuous mode when PROM (bit 15 in the MODE register) is set.

Receive Function Programming

Automatic pad field stripping is enabled by setting the ASTRP_RCV bit in CSR4. This can provide flexibility in the reception of messages using the 802.3 frame format.

All receive frames can be accepted by setting the PROM bit in CSR15. When PROM is set, the PCnet-PCI controller will attempt to receive all messages, subject to minimum frame enforcement. Promiscuous mode over rides the effect of the Disable Receive Broadcast bit on receiving broadcast frames.

The point at which the BMU will start to transfer data from the receive FIFO to buffer memory is controlled by the RCVFW bits in CSR80. The default established during H_RESET is 10b which sets the threshold flag at 64 bytes empty.

Automatic Pad Stripping

During reception of an 802.3 frame the pad field can be stripped automatically. ASTRP_RCV (CSR4, bit 0) = 1 enables the automatic pad stripping feature. The pad field will be stripped before the frame is passed to the FIFO, thus preserving FIFO space for additional frames. The FCS field will also be stripped, since it is computed at the transmitting station based on the data and pad field characters, and will be invalid for a receive frame that has had the pad characters stripped.

The number of bytes to be stripped is calculated from the embedded length field (as defined in the ISO 8802-3 (IEEE/ANSI 802.3) definition) contained in the frame. The length indicates the actual number of LLC data bytes contained in the message. Any received frame which contains a length field less than 46 bytes will have the pad field stripped (if ASTRP_RCV is set). Receive frames which have a length field of 46 bytes or greater will be passed to the host unmodified.

Since any valid Ethernet Type field value will always be greater than a normal 802.3 Length field (≥46), the PCnet-PCI controller will not attempt to strip valid Ethernet frames.

Note that for some network protocols, the value passed in the Ethernet Type and/or 802.3 Length field is not compliant with either standard and may cause problems.

Figure 28 shows the byte/bit ordering of the received length field for an 802.3 compatible frame format.

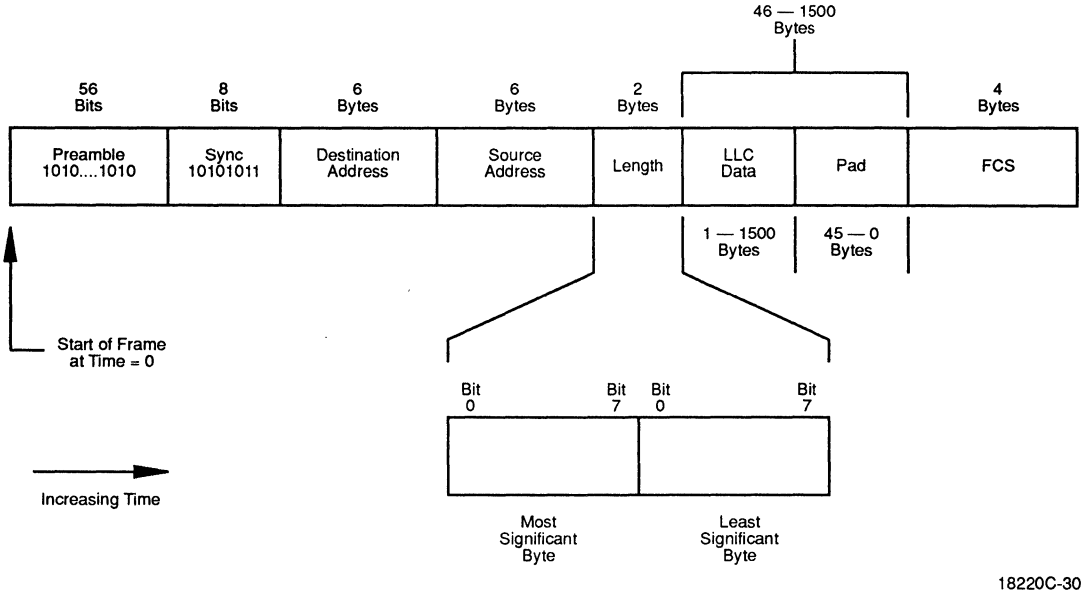


Figure 28. 802.3 Frame and Length Field Transmission Order

Receive FCS Checking

Reception and checking of the received FCS is performed automatically by the PCnet-PCI controller. Note that if the Automatic Pad Stripping feature is enabled, the FCS for padded frames will be verified against the value computed for the incoming bit stream including pad characters, but the FCS value for a padded frame will not be passed to the host. If an FCS error is detected in any frame, the error will be reported in the CRC bit in RMD1.

Receive Exception Conditions

Exception conditions for frame reception fall into two distinct categories; those which are the result of normal network operation, and those which occur due to abnormal network and/or host related events.

Normal events which may occur and which are handled autonomously by the PCnet-PCI controller are basically collisions within the slot time and automatic runt packet rejection. The PCnet-PCI controller will ensure that collisions which occur within 512 bit times from the start of reception (excluding preamble) will be automatically deleted from the receive FIFO with no host intervention. The receive FIFO will delete any frame which is composed of fewer than 64 bytes provided that the Runt Packet Accept (RPA bit in CSR124) feature has not been enabled. This criterion will be met regardless of whether the receive frame was the first (or only) frame in the FIFO or if the receive frame was queued behind a previously received message.

Abnormal network conditions include:

- FCS errors
- Late Collision

Host related receive exception conditions include MISS, BUFF, and OFLO. These are described in the BMU section.

Loopback Operation

Loopback is a mode of operation intended for system diagnostics. In this mode, the transmitter and receiver are both operating at the same time so that the controller receives its own transmissions. The controller provides two types of internal loopback and one type of external loopback. In internal loopback mode, the transmitted data can be looped back to the receiver at one of two places inside the controller without actually transmitting any data to the external network. The receiver will move the received data to the next receive buffer, where it can be examined by software. Alternatively, in external loop-

back mode, data can be transmitted to and received from the external network.

There are restrictions on loopback operation. The PCnet-PCI controller has only one FCS generator circuit. The FCS generator can be used by the transmitter to generate the FCS to append to the frame, or it can be used by the receiver to verify the FCS of the received frame. It can not be used by the receiver and transmitter simultaneously.

If the FCS generator is connected to the receiver, the transmitter will not append an FCS to the frame, but the receiver will check for one. The user can, however, calculate the FCS value for a frame and include this four-byte number in the transmit buffer.

If the FCS generator is connected to the transmitter, the transmitter will append an FCS to the frame, but the receiver will not check for the FCS. However, the user can verify the FCS by software.

During loopback, the FCS logic can be allocated to the receiver by setting DXMTFCS = 1 in CSR15.

If DXMTFCS=0, the MAC Engine will calculate and append the FCS to the transmitted message. The receive message passed to the host will therefore contain an additional 4 bytes of FCS. In this loopback configuration, the receive circuitry cannot detect FCS errors if they occur.

If DXMTFCS=1, the last four bytes of the transmit message must contain the (software generated) FCS computed for the transmit data preceding it. The MAC Engine will transmit the data without addition of an FCS field, and the FCS will be calculated and verified at the receiver.

The loopback facilities of the MAC Engine allow full operation to be verified without disturbance to the network. Loopback operation is also affected by the state of the Loopback Control bits (LOOP, MENDECL, and INTL) in CSR15. This affects whether the internal MENDEC is considered part of the internal or external loopback path.

The multicast address detection logic uses the FCS generator circuit. Therefore, in the loopback mode(s), the multicast address detection feature of the MAC Engine, programmed by the contents of the Logical Address Filter (LADRF [63:0] in CSRs 8–11) can only be tested when DXMTFCS=1, allocating the FCS generator to the receiver. All other features operate identically in loopback as in normal operation, such as automatic transmit padding and receive pad stripping.

When performing an internal loopback, no frame will be transmitted to the network. However, when the PCnet-PCI controller is configured for internal loopback the receiver will not be able to detect network traffic. External loopback tests will transmit frames onto the network if the AUI port is selected, and the PCnet-PCI controller will receive network traffic while configured for external loopback when the AUI port is selected. Runt Packet Accept is automatically enabled when any loopback mode is invoked.

Loopback mode can be performed with any frame size. Runt Packet Accept is internally enabled (RPA bit in CSR124 is not affected) when any loopback mode is invoked. This is to be backwards compatible to the LANCE (Am7990) software.

When the 10BASE-T MAU is selected in external loopback mode, the collision detection is disabled. This is necessary, because a collision in a 10BASE-T system is defined as activity on the transmitter outputs and receiver inputs at the same time, which is exactly what occurs during external loopback.

Since a 10BASE-T hub does not normally feed the station's transmitter outputs back into the station's receiver inputs, the use of external loopback in a 10BASE-T system usually requires some sort of external hardware that connects the outputs of the 10BASE-T MAU to its inputs.

LED Support

The PCnet-PCI controller can support up to 3 LEDs.

LED outputs $\overline{\text{LNKST}}$ and $\overline{\text{LED1}}$ allow for direct connection of an LED and its supporting pullup device. LED output $\overline{\text{LED3}}$ may require an additional buffer between the PCnet-PCI controller output pin and the LED and its supporting pullup device.

Because the $\overline{\text{LED3}}$ output is multiplexed with other PCnet-PCI controller functions, it may not always be possible to connect an LED circuit directly to the $\overline{\text{LED3}}$ pin. For example, when an LED circuit is directly connected to the EEDO/$\overline{\text{LED3}}$ pin, then it is not possible for most serial EEPROM devices to sink enough I_{OL} to maintain a valid low level on the EEDO input to the PCnet-PCI controller. Therefore, in applications that require both an EEPROM and a third LED, then it is necessary to buffer the $\overline{\text{LED3}}$ circuit from the EEPROM-PCnet-PCI connection. The LED registers in the BCR resource space allow each LED output to be programmed for either active high or active low operation, so that both inverting and non-inverting buffering choices are possible.

In applications where an EEPROM is not needed, the $\overline{\text{LED3}}$ pin may be directly connected to an LED circuit. The PCnet-PCI $\overline{\text{LED3}}$ pin driver will be able to sink enough current to properly drive the LED circuit.

By default, after H_RESET, the 3 LED outputs are configured in the following manner:

LED output	Default Interpretation	Default Drive Enable	Default Output Polarity
$\overline{\text{LNKST}}$	Link Status	Enabled	Active LOW
$\overline{\text{LED1}}$	Receive	Enabled	Active LOW
$\overline{\text{LED3}}$	Transmit	Enabled	Active LOW

For each LED register, each of the status signals is ANDed with its enable signal, and these signals are all ORed together to form a combined status signal. Each LED pins combined status signal runs to a pulse stretcher, which consists of a 3-bit shift register clocked at 38 Hz (26 ms). The data input of each shift register is normally at logic 0. The OR gate output for each LED register asynchronously sets all three bits of its shift register when the output becomes asserted. The inverted output of each shift register is used to control an LED pin. Thus the pulse stretcher provides 2–3 clocks of stretched LED output, or 52 ms to 78 ms.

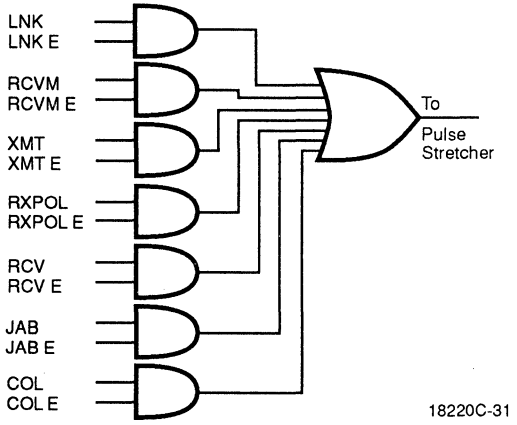


Figure 29. LED Control Logic

The diagram above shows the LED signal circuit that exists for each LED pin within the PCnet-PCI controller.

H_RESET, S_RESET, and STOP

There are three different types of RESET operations that may be performed on the PCnet-PCI device, H_RESET, S_RESET and STOP. These names have been used throughout the document. The following is a description of each type of RESET operation:

H_RESET

H_RESET= HARDWARE_RESET is a PCnet-PCI RESET operation that has been created by the proper assertion of the $\overline{\text{RST}}$ PIN of the PCnet-PCI device. When the minimum pulse width timing as specified in the $\overline{\text{RST}}$ pin description has been satisfied, then an internal RESET operation will be performed.

H_RESET will RESET all of or some portions of CSR0, 3, 4, 15, 58, 80, 82, 100, 112, 114, 122, 124 and 126 to default values. H_RESET will RESET all of or some portions of BCR 2, 4, 5, 6, 7, 18, 19, 20, 21 to default values. H_RESET will reset the Command register in the PCI configuration space. H_RESET will cause the microcode program to jump to its RESET state. Following the end of the H_RESET operation, the PCnet-PCI controller will attempt to read the EEPROM device through the EEPROM Microwire interface. H_RESET resets the T-MAU into the link fail state.

S_RESET

S_RESET = SOFTWARE_RESET is a PCnet-PCI RESET operation that has been created by a read access to the RESET REGISTER which is located at offset 14hex from the PCnet-PCI I/O base address.

S_RESET will RESET all of or some portions of CSR0, 3, 4, 15, 80, 100 and 124 to default values. S_RESET will not affect any of the BCR and PCI configuration space locations. S_RESET will cause the microcode program to jump to its RESET state. Following the end of the S_RESET operation, the PCnet-PCI controller will NOT attempt to read the EEPROM device. S_RESET sets the T-MAU into the link fail state.

Note that S_RESET will not cause a deassertion of the $\overline{\text{REQ}}$ signal, if it happens to be active at the time of the read to the reset register. The $\overline{\text{REQ}}$ signal will remain active until the $\overline{\text{GNT}}$ signal is asserted. Following the read of the RESET register, on the next clock cycle after the $\overline{\text{GNT}}$ signal is asserted, the PCnet-PCI controller will deassert the $\overline{\text{REQ}}$ signal. No bus master accesses will have been performed during this brief bus ownership period.

STOP

STOP is a PCnet-PCI RESET operation that has been created by the ASSERTION of the STOP bit in CSR0. That is, a STOP RESET is generated by writing a ONE to the STOP bit of CSR0 when the STOP bit currently has a value of ZERO. If the STOP bit value is currently a ONE and a ONE is rewritten to the STOP bit, then NO STOP RESET will be generated.

STOP will RESET all or some portions of CSR0, 3, and 4 to default values. STOP will not affect any of the BCR and PCI configuration space locations. STOP will cause the microcode program to jump to its RESET state. Following the end of the STOP operation, the PCnet-PCI controller will NOT attempt to read the EEPROM device. For the identity of individual CSRs and bit locations that are affected by STOP, see the individual CSR register descriptions. Setting the STOP bit does not affect the T-MAU.

NAND Tree Testing

The PCnet-PCI controller provides a NAND tree test mode to allow checking connectivity to the device on a printed circuit board. The NAND tree is built on all PCI bus signals.

NAND tree testing is enabled by asserting $\overline{RST}$. All PCI bus signals will become inputs on the asserting of $\overline{RST}$. The result of the NAND tree test can be observed on the NOUT pin.

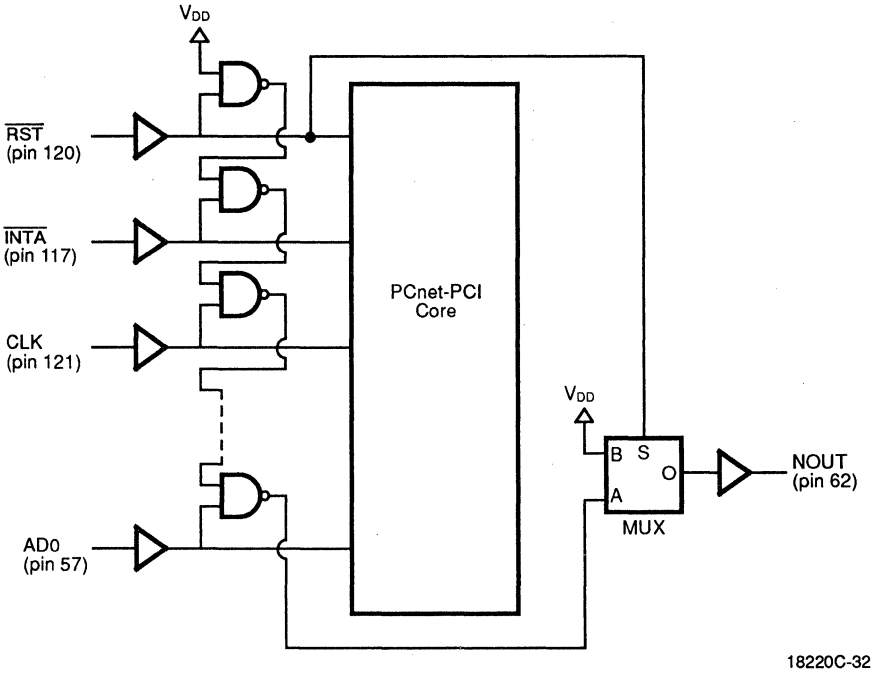


Figure 30. NAND Tree

Pin 120 ($\overline{RST}$) is the first input to the NAND tree. Pin 117 ($\overline{INTA}$) is the second input to the NAND tree, followed by pin 121 (CLK). All other PCI bus signals follow, counter-clockwise, with pin 57 (AD0) being the last. Pins labeled NC and all power supply pins are not part of the NAND tree. Table 9 shows the complete list of pins connected to the NAND tree.

Table 9. NAND Tree Configuration

NAND Tree Input #	Pin #	Name	NAND Tree Input #	Pin #	Name	NAND Tree Input #	Pin #	Name
1	120	$\overline{RST}$	18	15	AD21	35	36	AD15
2	117	$\overline{INTA}$	19	16	AD20	36	38	AD14
3	121	CLK	20	18	AD19	37	39	AD13
4	123	$\overline{GNT}$	21	19	AD18	38	40	AD12
5	126	$\overline{REQ}$	22	21	AD17	39	41	AD11
6	128	AD31	23	22	AD16	40	42	AD10
7	129	AD30	24	23	C/$\overline{BE2}$	41	44	AD9
8	131	AD29	25	24	$\overline{FRAME}$	42	45	AD8
9	132	AD28	26	25	$\overline{IRDY}$	43	47	C/$\overline{BE0}$
10	2	AD27	27	26	$\overline{TRDY}$	44	48	AD7
11	3	AD26	28	27	$\overline{DEVSEL}$	45	49	AD6
12	5	AD25	29	28	$\overline{STOP}$	46	51	AD5
13	6	AD24	30	29	$\overline{LOCK}$	47	52	AD4
14	7	C/$\overline{BE3}$	31	31	$\overline{PERR}$	48	53	AD3
15	10	IDSEL	32	32	$\overline{SERR}$	49	54	AD2
16	12	AD23	33	34	PAR	50	56	AD1
17	13	AD22	34	35	C/$\overline{BE1}$	51	57	AD0

$\overline{RST}$ must be asserted low to start a NAND tree test sequence. Initially, all NAND tree inputs except $\overline{RST}$ should be driven high. This will result in a high output at the NOUT pin. If the NAND tree inputs are driven from high to low in the same order as they are connected to build the NAND tree, NOUT will toggle every time an additional input is driven low. NOUT will change to a ZERO, when $\overline{INTA}$ is driven low and all other NAND tree inputs stay high. NOUT will toggle back to high, when CLK is additionally driven low. The square wave will continue until all NAND tree inputs are driven low. NOUT will be high, when all NAND tree inputs are driven low.

Note that some of the pins connected to the NAND tree are outputs in normal mode of operation. They must not be driven from an external source until the PCnet-PCI controller is configured for NAND tree testing.

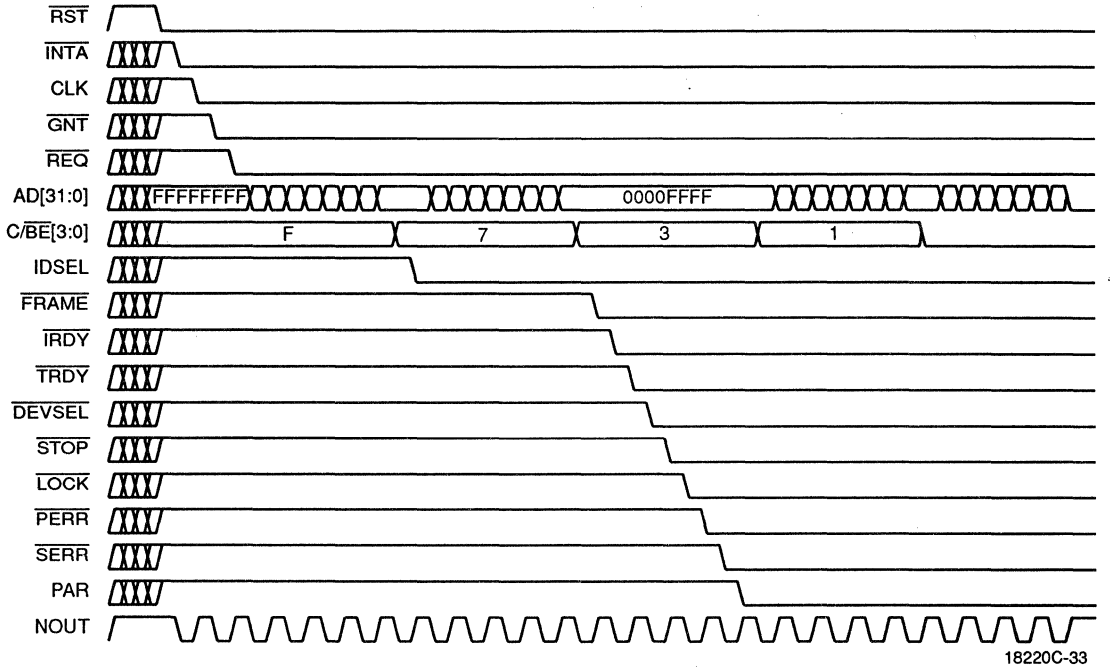


Figure 31. NAND Tree Waveform

USER ACCESSIBLE REGISTERS

The PCnet-PCI controller has three types of user registers: the PCI configuration registers, the Control and Status registers (CSR) and the Bus Control registers (BCR).

The PCnet-PCI controller implements all PCnet-ISA (Am79C960) registers all LANCE (Am7990) registers, all ILACC (Am79C900) registers, plus a number of additional registers. The PCnet-PCI controller CSRs are compatible with both the PCnet-ISA (Am79C960) CSRs and all of the LANCE (Am7990) CSRs upon power up. Compatibility to the ILACC set of CSRs requires one access to the Software Style register (BCR20, bits 7–0) to be performed. By setting an appropriate value of the Software Style register (BCR20, bits 7–0) the user can select a set of CSRs that are compatible with the ILACC set of CSRs.

The PCI configuration registers can be accessed in any data width. All other registers must be accessed according to the IO mode that is currently selected. When WIO mode is selected, all other register locations are defined to be 16 bits in width. When DWIO mode is selected, all these register locations are defined to be 32 bits in width, with the upper 16 bits of most register locations marked as reserved locations with undefined values. When performing register write operations in DWIO mode, the upper 16 bits should always be written as zeros, except for CSR88. When performing register read operations in DWIO mode, the upper 16 bits of I/O resources should always be regarded as having undefined values, except for CSR88.

PCnet-PCI registers can be divided into four groups:

PCI Configuration Registers:

Registers that are intended to be initialized by the system initialization procedure (e.g. BIOS device initialization routine) to program the operation of the PCnet-PCI controller PCI bus interface.

Setup Registers:

Registers that are intended to be initialized by the device driver to program the operation of various PCnet-PCI controller features.

Running Registers:

Registers that are intended to be used by the device driver software once the PCnet-PCI controller is running to access status information and to pass control information.

Test Registers:

Registers that are intended to be used only for testing and diagnostic purposes.

Below is a list of the registers that fall into each of the first three categories. Those registers that are not included in either of these lists can be assumed to be intended for diagnostic purposes.

PCI Configuration Registers

The following is a list of those registers that would typically need to be programmed once during the initialization of the PCnet-PCI controller within a system

- Base Address register
- Interrupt Line register
- Status register
- Command register

Setup Registers

The following is a list of those registers that would typically need to be programmed once during the setup of the PCnet-PCI controller within a system. The control bits in each of these registers typically do not need to be modified once they have been written. However, there are no restrictions as to how many times these registers may actually be accessed. Note that if the default power up values of any of these registers is acceptable to the application, then such registers need never be accessed at all. Also note that some of these registers may be programmable through the EEPROM read operation, and therefore do not necessarily need to be written to by the system initialization procedure or by the driver software.

CSR1	Initialization Address[15:0]
CSR2	Initialization Address[31:16]
CSR3	Interrupt Masks and Deferral Control
CSR4	Test and Features Control
CSR8	Logical Address Filter[15:0]
CSR9	Logical Address Filter[31:16]
CSR10	Logical Address Filter[47:32]
CSR11	Logical Address Filter[63:48]
CSR12	Physical Address Filter[15:0]
CSR13	Physical Address Filter[31:16]
CSR14	Physical Address Filter[47:32]
CSR15	Mode Register
CSR24	Base Address of Receive Ring Lower
CSR25	Base Address of Receive Ring Upper
CSR30	Base Address of Transmit Ring Lower
CSR31	Base Address of Transmit Ring Upper
CSR47	Polling Interval
CSR76	Receive Ring Length
CSR78	Transmit Ring Length
CSR80	DMA Transfer Counter and FIFO Threshold Control
CSR82	Bus Activity Timer
CSR100	Memory Error Timeout Register

CSR122	Receiver Packet Alignment Control
BCR2	Misc. configuration
BCR18	Bus Size and Burst Control Register
BCR20	Software Style

Running Registers

The following is a list of those registers that would typically need to be periodically read and perhaps written during the normal running operation of the PCnet-PCI controller within a system. Each of these registers contains control bits or status bits or both.

RAP	Register Address Port Register
CSR0	PCnet-PCI controller Status Register
CSR4	Test and Features Control
CSR112	Missed Frame Count
CSR114	Receive Collision Count

PCI Configuration Registers

The PCnet-PCI controller supports the 64-byte header portion of the configuration space as defined by the PCI specification revision 2.0. None of the device specific registers in locations 64 – 255 are used. The layout of the configuration registers in the header region is shown in the table below. All registers required to identify the PCnet-PCI controller and its function are implemented. Additional registers are used to setup the configuration of the PCnet-PCI controller in a system.

The configuration registers are accessible only by PCI configuration cycles. They can be accessed right after the PCnet-PCI controller is powered-on, even if the read operation of the serial EEPROM is still on-going. All multi-byte numeric fields follow little endian byte ordering. The Command register is the only register cleared by H_RESET. S_RESET as well as asserting SLEEP have no effect on the value of the PCI configuration registers. All write accesses to Reserved locations have no affect, reads from these locations will return a data value of ZERO.

Vendor ID (Offset 00h)

The Vendor ID register is a 16-bit register that identifies the manufacturer of the PCnet-PCI controller. Advanced Micro Devices, Inc.'s (AMD) Vendor ID is 1022h. Note that this vendor ID is not the same as the Manufacturer ID in CSR88 and CSR89. The vendor ID is assigned by the PCI Special Interest Group.

The Vendor ID register is located at offset 00h in the PCI Configuration Space. It is read only.

Device ID Register (Offset 02h)

The Device ID register is a 16-bit register that uniquely identifies the PCnet-PCI controller within AMD's product line. The PCnet-PCI Device ID is 2000h. Note that this Device ID is not the same as the Part number in CSR88 and CSR89. The Device ID is assigned by Advanced Micro Devices, Inc.

The Device ID register is located at offset 02h in the PCI Configuration Space. It is read only.

31 24	23 16	15 8	7 0	Offset
Device ID		Vendor ID		00h
Status		Command		04h
Base-Class	Sub-Class	Programming IF	Revision ID	08h
Reserved	Header Type	Latency Timer	Reserved	0Ch
Base Address				10h
Reserved				14h
Reserved				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
Reserved				28h
Reserved				2Ch
Reserved				30h
Reserved				34h
Reserved				38h
Reserved	Reserved	Interrupt Pin	Interrupt Line	3Ch

Command Register (offset 04h)

The Command register is a 16-bit register used to control the gross functionality of the PCnet-PCI controller. It controls the PCnet-PCI controller's ability to generate and respond to PCI bus cycles. To logically disconnect the PCnet-PCI device from all PCI bus cycles except Configuration cycles, a value of ZERO should be written to this register.

The Command register is located at offset 04h in the PCI Configuration Space. It is read and written by the host.

15-10 RES — Reserved locations. Read as ZERO, write operations have no effect.

9 FBTBEN — Fast Back-to-Back enable. Read as ZERO, write operations have no effect. The PCnet-PCI controller will not generate Fast Back-to-Back cycles.

8 SERREN — SERR enable. Controls the assertion of the $\overline{SERR}$ pin. $\overline{SERR}$ is disabled when SERREN is cleared. $\overline{SERR}$ will be asserted on detection of an address parity error and if both, SERREN and PERREN (bit 6 of this register) are set.

SERREN is cleared by H_RESET and is not effected by S_RESET or asserting the $\overline{SLEEP}$ pin.

7 ADSTEP — Address/data stepping. Read as ONE, write operations have no effect. The PCnet-PCI controller uses address stepping for the first address phase of each bus master period. $\overline{FRAME}$ will be asserted on the second CLK following the assertion of $\overline{GNT}$ indicating a valid address on the AD bus.

6 PERREN — Parity Error Response enable. Enables the parity error response functions. When PERREN is '0' and the PCnet-PCI controller detects a parity error, it only sets the Detected Parity Error bit in the Status register. When PERREN is '1', the PCnet-PCI controller asserts $\overline{PERR}$ on the detection of a data parity error. It also sets the DATAPERR bit (bit 8 in the Status register), when the data parity error occurred during a master cycle. PERREN also enables reporting address parity errors through the $\overline{SERR}$ pin and the $\overline{SERR}$ bit in the Status register.

PERREN is cleared by H_RESET and is not effected by S_RESET or asserting the $\overline{SLEEP}$ pin.

5 VGASNOOP — VGA palette snoop. Read as ZERO, write operations have no effect.

4 MWIEN — Memory Write and Invalidate Cycle enable. Read as ZERO, write operations have no effect. The PCnet-PCI controller only generates Memory Write cycles.

3 SCYCEN — Special Cycle enable. Read as ZERO, write operations have no effect. The PCnet-PCI controller ignores all Special Cycle operations.

2 BMEN — Bus Master enable. Setting BMEN enables the PCnet-PCI controller to become a bus master on the PCI bus. The host must set BMEN before setting the INIT bit in CSR0 of the PCnet-PCI controller. (Setting INIT causes the PCnet-PCI controller to start its first bus master operation, which is reading in the initialization block.)

BMEN is cleared by H_RESET and is not effected by S_RESET or asserting the $\overline{SLEEP}$ pin.

1 MEMEN — Memory Space access enable. Read as ZERO, write operations have no effect. The PCnet-PCI controller has no memory mapped resources.

0 IOEN — I/O Space access enable. The PCnet-PCI controller will ignore all I/O accesses when IOEN is cleared. The host must set IOEN before the first I/O access to the device. The Base Address register at offset 10h must be programmed with a valid I/O address before setting IOEN.

IOEN is cleared by H_RESET and is not effected by S_RESET or asserting the $\overline{SLEEP}$ pin.

Status Register (Offset 06h)

The Status register is a 16-bit register that contains status information for the PCI bus related events. It is located at offset 06h in the PCI Configuration Space.

15 PERR — Parity Error. PERR is set when the PCnet-PCI controller detects a parity error.

The PCnet-PCI controller samples the AD[31:00], C/$\overline{BE}$[3:0]

and the PAR lines for a parity error at the following times:

- In slave mode, during the address phase of any PCI bus command.
- In slave mode, during the data phase of all I/O and Configuration Write commands that select the PCnet-PCI controller.
- In master mode, during the data phase of all Memory Read and Memory Read Line commands.

During the data phase of the memory write command, the PCnet-PCI controller sets the PERR bit if the target reports a data parity error by asserting the PERR signal.

PERR is not effected by the state of the Parity Error Response enable bit (bit 6 in the Control register).

PERR is set by the PCnet-PCI controller and cleared by writing a ONE. Writing a ZERO has no effect. PERR is not affected by H_RESET or S_RESET or asserting the SLEEP pin.

14 SERR — Signaled SERR. SERR is set when the PCnet-PCI controller detects an address parity error, and both, SERREN and PEREN (bits 8 and 6 of the Command register) are set.

SERR is set by the PCnet-PCI controller and cleared by writing a "1". Writing a "0" has no effect. SERR is not affected by H_RESET or S_RESET or asserting the SLEEP pin.

13 RMABORT — Received Master Abort. RMABORT is set when the PCnet-PCI controller terminates a master cycle with a master abort sequence.

RMABORT is set by the PCnet-PCI controller and cleared by writing a "1". Writing a "0" has no effect. RMABORT is not affected by H_RESET or S_RESET or asserting the SLEEP pin.

12 RTABORT — Received Target Abort. RTABORT is set when a target terminates a PCnet-PCI master cycle with a target abort sequence.

RTABORT is set by the PCnet-PCI controller and cleared by writing a "1". Writing a "0" has no effect. RTABORT is not affected by H_RESET or S_RESET or asserting the SLEEP pin.

11 STABORT — Send Target Abort. STABORT is set when the PCnet-PCI controller terminates a slave access with a target abort sequence.

STABORT is set by the PCnet-PCI controller and cleared by writing a "1". Writing a "0" has no effect. STABORT is not affected by H_RESET or S_RESET or asserting the SLEEP pin.

10–9 DEVSEL — DEVSEL timing. DEVSEL is set to 01b (medium), indicating the PCnet-PCI controller will assert DEVSEL two CLK periods after FRAME is asserted.

DEVSEL is read only.

8 DATAPERR — Data Parity Error detected. DATAPERR is set when the PCnet-PCI controller detects a data parity error during master mode and the Parity Error Response enable bit (bit 6 in the Control register) is set.

During the data phase of all Memory Read and Memory Read Line commands, the PCnet-PCI controller checks for parity error by sampling the AD[31:00] and C/BE[3:0] and the PAR lines. During the data phase of all Memory Write commands, the PCnet-PCI controller checks the PERR input to detect whether the target has reported a parity error.

DATAPERR is set by the PCnet-PCI controller and cleared by writing a ONE. Writing a ZERO has no effect. DATAPERR is not affected by H_RESET or S_RESET or asserting the SLEEP pin.

7–0 RES — Reserved locations. Read as ZERO, write operations have no effect.

Revision ID Register (Offset 08h)

The Revision ID register is an 8-bit register that specifies the PCnet-PCI controller revision number. The current value of this register is 00h.

The Revision ID register is located at offset 08h in the PCI Configuration Space. It is read only.

Programming Interface Register (Offset 09h)

The Programming Interface register is an 8-bit register that identifies the programming interface of PCnet-PCI controller. PCI does not define any specific register-level programming interfaces for network devices. The value of this register is 00h.

The Programming Interface register is located at address 09h in the PCI Configuration Space. It is read only.

Sub-Class Register (Offset 0Ah)

The Sub-Class register is an 8-bit register that identifies specifically the function of the PCnet-PCI controller. The value of this register is 00h which identifies the PCnet-PCI device as an Ethernet controller.

The Sub-Class register is located at offset 0Ah in the PCI Configuration Space. It is read only.

Base-Class Register (Offset 0Bh)

The Base-Class register is an 8-bit register that broadly classifies the function of the PCnet-PCI controller. The value of this register is 02h which classifies the PCnet-PCI device as a network controller.

The Base-Class register is located at offset 0Bh in the PCI Configuration Space. It is read only.

Latency Timer Register (Offset 0Dh)

The Latency Timer register is an 8-bit register that specifies the maximum time the PCnet-PCI controller can continue with bus master transfers after the system arbiter has removed $\overline{GNT}$. The time is measured in CLK cycles. The working copy of the timer will start counting down when the PCnet-PCI controller asserts $\overline{FRAME}$ for the first time during a bus mastership period. The counter will freeze at ZERO. When the counter is ZERO and $\overline{GNT}$ is deasserted by the system arbiter, the PCnet-PCI controller will finish the current data phase and then immediately release the bus.

The value for the PCnet-PCI controller Latency Timer register is 00h, which indicates that, when the PCnet-PCI controller is preempted, it will always release the bus immediately after finishing the current data phase.

The Latency Timer register is located at offset 0Dh in the PCI Configuration Space. It is read only.

Header Type Register (Offset 0Eh)

The Header Type register is an 8-bit register that describes the format of the PCI Configuration Space locations 10h to 3Ch and that identifies a device to be single or multi function. The Header Type register is located at offset 0Eh in the PCI Configuration Space. It is read only.

7	FUNCT	Single function/multi function device. Read as ZERO, write operations have no effect. The PCnet-PCI controller is a single function device.
6–0	LAYOUT	PCI configuration space layout. Read as ZERO, write operations have no effect. The layout of the PCI configuration space locations 10h to 3Ch is as show in the table at the beginning of this section.

Base Address Register (Offset 10h)

The Base Address register is a 32-bit register that determines the location of the PCnet-PCI controller in all of I/O space. It is located at offset 10h in the PCI Configuration Space.

31–5	IOBASE	I/O base address significant 27 bits. These bits are written by the host to specify the location of the PCnet-PCI controller in all of I/O space. IOBASE must be written with a valid address before the PCnet-PCI controller slave I/O mode is turned on with setting the IOEN bit (bit 0 in the Command register). When the PCnet-PCI controller is enabled for I/O mode (IOEN is set), it monitors the PCI bus for a valid I/O command. If the value on AD[31:05] during the address phase of the cycles matches the value of IOBASE, the PCnet-PCI controller will drive $\overline{DEVSEL}$ indicating it will respond to the access. IOBASE is read and written by the host. IOBASE is not effected by H_RESET or S_RESET or asserting the $\overline{SLEEP}$ pin.
4–2	IOSIZE	I/O size requirements. Read as ZERO, write operations have no effect. IOSIZE indicates the size of the I/O space the PCnet-PCI controller requires. When the host writes a value of FFFF FFFFh to the Base Address register, it will read back a value of "0" in bits 4–2. That indicates a PCnet-PCI I/O space requirement of 32 bytes.

1	RES	Reserved location. Read as ZERO, write operations have no effect.
0	IOSPACE	I/O space indicator. Read as ONE, write operations have no effect. Indicating that this Base Address register describes an I/O base address.

Interrupt Line Register (Offset 3Ch)

The Interrupt Line register is an 8-bit register that is used to communicate the routing of the interrupt. This register is written by the POST software as it initialized the PCnet-PCI controller in the system. The register is read by the network driver to determine the interrupt channel which the POST software has assigned to the PCnet-PCI controller. The Interrupt Line register is not modified by the PCnet-PCI controller. It has no effect on the operation of the device.

The Interrupt Line register is located at offset 3Ch in the PCI Configuration Space. It is read an written by the host. It is not effected by H_RESET or S_RESET or asserting the $\overline{\text{SLEEP}}$ pin.

Interrupt Pin Register (Offset 3Dh)

This Interrupt Pin register is an 8-bit register indicating the interrupt pin the PCnet-PCI controller is using. The value for the PCnet-PCI Interrupt Pin register is 01h, which corresponds to INTA.

The Interrupt Pin register is located at offset 3Dh in the PCI Configuration Space. It is read only.

RAP Register

The RAP (Register Address Pointer) register is used to gain access to CSR and BCR registers on board the PCnet-PCI controller. The value of the RAP indicates the address of a CSR or BCR whenever an RDP or BDP access is performed. That is to say, RAP serves as a pointer to CSR and BDP space.

As an example of RAP use, consider a read access to CSR4. In order to access this register, it is necessary to first load the value 0004h into the RAP by performing a write access to the RAP offset of 12h (12h when WIO mode has been selected, 14h when DWIO mode has been selected). Then a second access is performed PCnet-PCI controller, this time to the RDP offset of 10h (for either WIO or DWIO mode). The RDP access is a read access, and since RAP has just been loaded with the value of 0004h, the RDP read will yield the contents of CSR4. A read of the BDP at this time (offset of 16h when WIO mode has been selected, 1Ch when DWIO mode has been selected) will yield the contents of BCR4, since the RAP is used as the pointer into both BDP and RDP space.

RAP: Register Address Port

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	RES	Reserved locations. Read and written as ZEROs.
7–0	RAP	Register Address Port. The value of these 8 bits determines which CSR or BCR will be accessed when an I/O access to the RDP or BDP port, respectively, is performed. A write access to undefined CSR or BCR locations may cause unexpected reprogramming of the PCnet-PCI control registers. A read access will yield undefined values. RAP is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.

Control and Status Registers

The CSR space is accessible by performing accesses to the RDP (Register Data Port). The particular CSR that is read or written during an RDP access will depend upon the current setting of the RAP. RAP serves as a pointer into the CSR space. RAP also serves as the pointer to BCR space, which is described in a later section.

CSR0: PCnet-PCI Controller Status Register

Bit	Name	Description
		Certain bits in CSR0 indicate the cause of an interrupt. The register is designed so that these indicator bits are cleared by writing ONEs to those bit locations. This means that the software can read CSR0 and write back the value just read to clear the interrupt condition.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	ERR	Error is set by the ORing of BABL, CERR, MISS, and MERR. ERR remains set as long as any of the error flags are true. ERR is read only. Write operations are ignored.
14	BABL	Babble is a transmitter time-out error. It indicates that the transmitter has been on the channel

longer than the time required to send the maximum length frame. BABL will be set if 1519 bytes or greater are transmitted.

When BABL is set, $\overline{INTA}$ is asserted if IENA = 1 and the mask bit BABLM in CSR3 is clear. BABL assertion will set the ERR bit.

BABL is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. BABL is cleared by H_RESET or S_RESET or by setting the STOP bit.

13 CERR — Collision Error indicates that the collision inputs to the AUI port failed to activate within 20 network bit times after the chip terminated transmission (SQE Test). This feature is a transceiver test feature.

In 10BASE-T mode, CERR will be set after a transmission if the T-MAU is in link fail state.

CERR assertion will not result in an interrupt being generated. CERR assertion will set the ERR bit.

CERR is set by the MAC layer and cleared by writing a "1". Writing a "0" has no effect. CERR is cleared by H_RESET or S_RESET or by setting the STOP bit.

12 MISS — Missed Frame is set when PCnet-PCI controller has lost an incoming receive frame resulting from a Receive Descriptor not being available. This bit is the only immediate indication that receive data has been lost since there is no current receive descriptor. Missed Frame Counter (CSR112) also increments each time a receive frame is missed.

When MISS is set, $\overline{INTA}$ is asserted if IENA = 1 and the mask bit MISSM in CSR3 is clear. MISS assertion will set the ERR bit.

MISS is set by the Buffer Management Unit and cleared by writing a "1". Writing a "0" has no effect. MISS is cleared by H_RESET or S_RESET or by setting the STOP bit.

11 MERR — Memory Error is set when PCnet-PCI controller requests the use of the system interface bus by asserting $\overline{REQ}$ and has not received $\overline{GNT}$ assertion after a programmable length of time. The length of time in microseconds before MERR is asserted will depend upon the setting of the Bus Timeout Register (CSR100). The default setting of CSR100 will give a MERR after 51.2 microseconds of bus latency.

When MERR is set, $\overline{INTA}$ is asserted if IENA = 1 and the mask bit MERRM in CSR3 is clear. MERR assertion will set the ERR bit, regardless of the settings of IENA and MERRM.

MERR is set by the Bus Interface Unit and cleared by writing a "1". Writing a "0" has no effect. MERR is cleared by H_RESET, S_RESET or by setting the STOP bit.

10 RINT — Receive Interrupt. RINT is set by the Buffer Management Unit of the PCnet-PCI controller after the last descriptor of a receive packet has been updated by writing a ZERO to the ownership bit. RINT may also be set when the first descriptor of a receive packet has been updated by writing a ZERO to the ownership bit if the SPRINTEN bit of CSR3 has been set to a ONE.

When RINT is set, $\overline{INTA}$ is asserted if IENA = 1 and the mask bit RINTM in CSR3 is clear.

RINT is cleared by the host by writing a "1". Writing a "0" has no effect. RINT is cleared by H_RESET, S_RESET or by setting the STOP bit.

9 TINT — Transmit Interrupt is set after the OWN bit in the last descriptor of a transmit frame has been cleared to indicate the frame has been sent or an error occurred in the transmission.

When TINT is set, $\overline{INTA}$ is asserted if IENA = 1 and the mask bit TINTM in CSR3 is clear.

TINT is set by the Buffer Management Unit and cleared by writing a "1". Writing a "0" has no effect. TINT is cleared by H_RESET or S_RESET or by setting the STOP bit.

8 IDON — Initialization Done indicates that the initialization sequence has

completed. When IDON is set, PCnet-PCI controller has read the Initialization block from memory.

When IDON is set, $\overline{INTA}$ is asserted if IENA = 1 and the mask bit IDONM in CSR3 is clear.

IDON is set by the Buffer Management Unit after the initialization block has been read from memory and cleared by writing a "1". Writing a "0" has no effect. IDON is cleared by H_RESET or S_RESET or by setting the STOP bit.

7 INTR — Interrupt Flag indicates that one or more following interrupt causing conditions has occurred: BABL, MISS, MERR, MPCO, RCVCCO, RINT, RPCO, TINT, IDON, JAB or TXSTRT; and its associated mask bit is clear. If IENA = 1 and INTR is set, $\overline{INTA}$ will be active.

INTR is read only. INTR is cleared by H_RESET, S_RESET, setting the STOP bit or by clearing all of the active individual interrupt bits that have not been masked out.

6 IENA — Interrupt Enable allows $\overline{INTA}$ to be active if the Interrupt Flag is set. If IENA = 0 then $\overline{INTA}$ will be disabled regardless of the state of INTR.

IENA is set by writing a "1" and cleared by writing a "0". IENA is cleared by H_RESET or S_RESET or by setting the STOP bit.

5 RXON — Receive On indicates that the Receive function is enabled. RXON is set if DRX = 0 in CSR15[1] after the START bit is set. If INIT and START are set together, RXON will not be set until after the initialization block has been read in.

RXON is read only. RXON is cleared by H_RESET or S_RESET or by setting the STOP bit.

4 TXON — Transmit On indicates that the Transmit function is enabled. TXON is set if DTX = 0 in CSR15[1] after the START bit is set. If INIT and START are set together, TXON will not be set until after the initialization block has been read in.

TXON is read only. TXON is cleared by H_RESET or S_RESET or by setting the STOP bit.

3 TDMD — Transmit Demand, when set, causes the Buffer Management Unit to access the Transmit Descriptor Ring without waiting for the poll-time counter to elapse. If TXON is not enabled, TDMD bit will be reset and no Transmit Descriptor Ring access will occur.

TDMD is required to be set if the DPOLL bit in CSR4 is set; setting TDMD while DPOLL = 0 merely hastens the PCnet-PCI controller's response to a Transmit Descriptor Ring Entry.

TDMD is set by writing a "1". Writing a "0" has no effect. TDMD will be cleared by the Buffer Management Unit when it fetches a Transmit Descriptor. TDMD is cleared by H_RESET or S_RESET and setting the STOP bit.

2 STOP — STOP assertion disables the chip from all DMA activity. The chip remains inactive until either STRT or INIT are set. If STOP, STRT and INIT are all set together, STOP will override STRT and INIT.

STOP is set by writing a "1", by H_RESET or S_RESET. Writing a "0" has no effect. STOP is cleared by setting either STRT or INIT.

1 STRT — STRT assertion enables PCnet-PCI controller to send and receive frames, and perform buffer management operations. Setting STRT clears the STOP bit. If STRT and INIT are set together, PCnet-PCI controller initialization will be performed first.

STRT is set by writing a "1". Writing a "0" has no effect. STRT is cleared by H_RESET, S_RESET or by setting the STOP bit.

0 INIT — INIT assertion enables PCnet-PCI controller to begin the initialization procedure which reads in the initialization block from memory. Setting INIT clears the STOP bit. If STRT and INIT are set together, PCnet-PCI controller initialization will be performed first. INIT is not cleared when the

initialization sequence has completed.

INIT is set by writing a "1". Writing a "0" has no effect. INIT is cleared by H_RESET, S_RESET or by setting the STOP bit.

CSR1: IADR[15:0]

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	IADR[15:0]	Lower 16 bits of the address of the Initialization Block. Bit locations 1 and 0 must both be ZERO to align the initialization block to a double-word boundary, regardless of the value of SSIZE32 (BCR20/CSR58, bit 8). This register is aliased with CSR16. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by H_RESET or S_RESET or by setting the STOP bit.

CSR2: IADR[31:16]

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	IADR[31:24]	If SSIZE32 is set (BCR20, bit 8), then the IADR[31:24] bits will be used strictly as the upper 8 bits of the initialization block address. However, if SSIZE32 is reset (BCR20, bit 8), then the IADR[31:24] bits will be used to generate the upper 8 bits of all bus mastering addresses, as required for a 32 bit address bus. Note that the 16-bit software structures specified by the SSIZE32=0 setting will yield only 24 bits of address for PCnet-PCI bus master accesses, while the 32-bit hardware for which the PCnet-PCI controller is intended will require 32 bits of address. Therefore, whenever SSIZE32=0, the IADR[31:24] bits will be appended to the 24-bit initialization address, to each 24-bit descriptor base address and to each beginning 24-bit buffer address in order to form complete 32-bit addresses. The upper 8 bits that exist in the descriptor address registers and the buffer address registers which are stored on board the PCnet-PCI controller will be overwritten with the IADR[31:24] value, so that CSR accesses to these registers will show the 32 bit address that includes the appended field. If SSIZE32=1, then software will provide 32-bit pointer values for all of the shared software structures – i.e. descriptor bases and buffer addresses, and therefore, IADR[31:24] will not be written to the upper 8 bits of any of these resources, but it will be used as the upper 8 bits of the initialization address. This register is aliased with CSR17. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by H_RESET, S_RESET or by setting the STOP bit.
7–0	IADR[23:16]	Bits 23 through 16 of the address of the Initialization Block. Whenever this register is written, CSR17 is updated with CSR2's contents. Read/Write accessible only when the STOP bit in CSR0 is set. Unaffected by H_RESET, S_RESET or by setting the STOP bit.

CSR3: Interrupt Masks and Deferral Control

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	RES	Reserved location. Read and written as ZERO.
14	BABLM	Babble Mask. If BABLM is set, the BABL bit in CSR0 will be masked and unable to set INTR flag in CSR0. Read/Write accessible always. BABLM is cleared by H_RESET or S_RESET and is not affected by STOP.
13	RES	Reserved location. Read and written as ZERO.
12	MISSM	Missed Frame Mask. If MISSM is set, the MISS bit in CSR0 will be masked and unable to set INTR flag in CSR0. Read/Write accessible always. MISSM is cleared by H_RESET

or S_RESET and is not affected by STOP.

11 MERRM — Memory Error Mask. If MERRM is set, the MERR bit in CSR0 will be masked and unable to set INTR flag in CSR0.

Read/Write accessible always. MERRM is cleared by H_RESET or S_RESET and is not affected by STOP.

10 RINTM — Receive Interrupt Mask. If RINTM is set, the RINT bit in CSR0 will be masked and unable to set INTR flag in CSR0.

Read/Write accessible always. RINTM is cleared by H_RESET or S_RESET and is not affected by STOP.

9 TINTM — Transmit Interrupt Mask. If TINTM is set, the TINT bit in CSR0 will be masked and unable to set INTR flag in CSR0.

Read/Write accessible always. TINTM is cleared by H_RESET or S_RESET and is not affected by STOP.

8 IDONM — Initialization Done Mask. If IDONM is set, the IDON bit in CSR0 will be masked and unable to set INTR flag in CSR0.

Read/Write accessible always. IDONM is cleared by H_RESET or S_RESET and is not affected by STOP.

7–6 RES — Reserved locations. Read and written as ZEROs.

5 LAPPEN — Look-Ahead Packet Processing Enable. When set to a ONE, the LAPPEN bit will cause the PCnet-PCI controller to generate an interrupt following the descriptor write operation to the first buffer of a receive packet. This interrupt will be generated in addition to the interrupt that is generated following the descriptor write operation to the last buffer of a receive packet. The interrupt will be signaled through the RINT bit of CSR0.

Setting LAPPEN to a ONE also enables the PCnet-PCI controller to read the STP bit of receive descriptors. PCnet-PCI controller will use the STP information to determine where it should begin writing a receive packets data. Note that while in this mode, the PCnet-PCI controller can write intermediate packet data to buffers whose descriptors do not contain STP bits set to ONE. Following the write to the last descriptor used by a packet, the PCnet-PCI controller will scan through the next descriptor entries to locate the next STP bit that is set to a ONE. The PCnet-PCI controller will begin writing the next packets data to the buffer pointed to by that descriptor.

Note that because several descriptors may be allocated by the host for each packet, and not all messages may need all of the descriptors that are allocated between descriptors that contain STP = ONE, then some descriptors/buffers may be skipped in the ring. While performing the search for the next STP bit that is set to ONE, the PCnet-PCI controller will advance through the receive descriptor ring regardless of the state of ownership bits. If any of the entries that are examined during this search indicate PCnet-PCI controller ownership of the descriptor but also indicate STP = 0, then the PCnet-PCI controller will reset the OWN bit to ZERO in these entries. If a scanned entry indicates host ownership with STP = 0, then the PCnet-PCI controller will not alter the entry, but will advance to the next entry.

When the STP bit is found to be true, but the descriptor that contains this setting is not owned by the PCnet-PCI controller, then the PCnet-PCI controller will stop advancing through the ring entries and begin periodic polling of this entry. When the STP bit is found to be true, and the descriptor that contains this setting is owned by the PCnet-PCI controller, then the PCnet-PCI controller will stop advancing through the ring entries, store the descriptor information that it has just read, and wait for the next receive to arrive.

This behavior allows the host software to pre-assign buffer space in such a manner that the "header" portion of a receive packet will always be written to a particular memory area, and the "data" portion of a receive packet

		will always be written to a separate memory area. The interrupt is generated when the "header" bytes have been written to the "header" memory area. Read/Write accessible always. The LAPPEN bit will be reset to ZERO by H_RESET or S_RESET and will be unaffected by STOP. See Appendix D for more information on the LAPP concept.
4	DXMT2PD	Disable Transmit Two Part Deferral (see Medium Allocation section in Media Access Management fro more details). If DXMT2PD is set, Transmit Two Part Deferral will be disabled. Read/Write accessible always. DXMT2PD is cleared by H_RESET or S_RESET and is not affected by STOP.
3	EMBA	Enable Modified Back-off Algorithm (see Contention Resolution section in Media Access Management for more details). If EMBA is set, a modified back-off algorithm is implemented. Read/Write accessible always. EMBA is cleared by H_RESET or S_RESET and is not affected by STOP.
2	BSWP	Byte Swap. This bit is used to choose between big and little Endian modes of operation. When BSWP is set to a ONE, big Endian mode is selected. When BSWP is set to ZERO, little Endian mode is selected. When big Endian mode is selected, the PCnet-PCI controller will swap the order of bytes on the AD bus during a data phase on accesses to the FIFOs only. Specifically, AD[31:24] becomes BYTE0, AD[23:16] becomes BYTE1, AD[15:8] becomes BYTE2 and AD[7:0] becomes BYTE3 when big Endian mode is selected. When little Endian mode is selected, the order of bytes on the AD bus during a data phase is: AD[31:24] is BYTE3, AD[23:16] is BYTE2, AD[15:8] is BYTE1 and AD[7:0] is BYTE0. Byte swap only affects data transfers that involve the FIFOs. Initialization block transfers are not affected by the setting of the BSWP bit. Descriptor transfers are not affected by the setting of the BSWP bit. RDP, RAP and BDP accesses are not affected by the setting of the BSWP bit. APROM transfers are not affected by the setting of the BSWP bit. Note that the byte ordering of the PCI bus is defined to be little endian. BSWP must not be set to ONE when the PCnet-PCI controller operates in a PCI system. BSWP is write/readable regardless of the state of the STOP bit. BSWP is cleared by H_RESET or S_RESET and is not affected by STOP bit.
1	RES	Reserved location. The default value of this bit is a ZERO. Writing a ONE to this bit has no effect on device function; If a ONE is written to this bit, then a ONE will be read back. Existing drivers may write a ONE to this bit for compatibility, but new drivers should write a ZERO to this bit and should treat the read value as undefined.
0	RES	Reserved location. The default value of this bit is a ZERO. Writing a ONE to this bit has no effect on device function. If a ONE is written to this bit, then a ONE will be read back. Existing drivers may write a ONE to this bit for compatibility, but new drivers should write a ZERO to this bit and should treat the read value as undefined.

CSR4: Test and Features Control

Bit	Name	Description
		Certain bits in CSR4 indicate the cause of an interrupt. The register is designed so that these indicator bits are cleared by writing ONEs to those bit locations. This means that the software can read CSR4 and write back the value just read to clear the interrupt condition.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	ENTST	Enable Test Mode operation. Setting ENTST to ONE enables internal test functions which are useful only for stand alone integrated circuit testing. In addition,

the Runt Packet Accept (RPA) bit (CSR124, bit 3) may be changed only when ENTST is set to ONE.

To enable RPA, the user must first write a ONE to the ENTST bit. Next, the user must first write a ONE to the RPA bit (CSR124, bit 3). Finally, the user must write a ZERO to the ENTST bit to take the device out of test mode operation. Once, the RPA bit has been set to ONE, the device will remain in the Runt Packet Accept mode until the RPA bit is cleared to ZERO.

Read/Write accessible. ENTST is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.

14 DMAPLUS

When DMAPLUS = "1", DMA Burst Counter in CSR80 is disabled. If DMAPLUS = "0", the counter is enabled.

Read/Write accessible. DMAPLUS is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.

13 TIMER

Timer Enable Register. If TIMER is set, the Bus Timer Register, CSR82 is enabled. If TIMER is cleared, the Bus Timer Register is disabled.

Read/Write accessible. TIMER is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.

12 DPOLL

Disable Transmit Polling. If DPOLL is set, the Buffer Management Unit will disable transmit polling. Likewise, if DPOLL is cleared, automatic transmit polling is enabled. If DPOLL is set, TDMD bit in CSR0 must be set in order to initiate a manual poll of a transmit descriptor. Transmit descriptor polling will not take place if TXON is reset.

Read/Write accessible. DPOLL is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.

11 APAD_XMT

Auto Pad Transmit. When set, APAD_XMT enables the automatic padding feature. Transmit frames will be padded to extend them to 64 bytes including FCS. The FCS is calculated for the entire frame including pad, and appended after the pad field. APAD_XMT will override the programming of the DXMTFCS bit.

Read/Write accessible. APAD_XMT is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.

10 ASTRP_RCV

Auto Strip Receive. When set, ASTRP_RCV enables the automatic pad stripping feature. The pad and FCS fields will be stripped from receive frames and not placed in the FIFO.

Read/Write accessible. ASTRP_RCV is cleared by H_RESET or S_RESET and is unaffected by the STOP bit.

9 MFCO

Missed Frame Counter Overflow Interrupt.

Indicates the MFC (CSR112) wrapped around. Can be cleared by writing a 1 to this bit. Also cleared by H_RESET, S_RESET or by asserting the STOP bit. Writing a 0 has no effect.

When MFCO is set, $\overline{\text{INTA}}$ is asserted if IENA is ONE and the mask bit MFCOM is ZERO.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI controller will never set the value of this bit to ONE.

8 MFCOM

Missed Frame Counter Overflow Mask.

If MFCOM is set, MFCO will be unable to set INTR in CSR0.

Set to a ONE by H_RESET or S_RESET, unaffected by the STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI controller will set the value of this bit to ZERO.

7 RES

Reserved location. Written as ZERO and read as ZERO.

6 RES

Reserved location. This bit may be written to as either a ONE or a ZERO, but will always be read as a ZERO. This bit has no effect on PCnet-PCI controller operation.

5 RCVCCO

Receive Collision Counter Overflow.

Indicates the Receive Collision Counter (CSR114) wrapped around. Can be cleared by

writing a 1 to this bit. Also cleared by H_RESET, S_RESET or by setting the STOP bit. Writing a 0 has no effect.

When RCVCCO is set, $\overline{INTA}$ is asserted if IENA is ONE and the mask bit RCVCCOM is ZERO.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI controller will never set the value of this bit to ONE.

4 RCVCCOM — Receive Collision Counter Overflow Mask.

If RCVCCOM is set, RCVCCO will be unable to set INTR in CSR0.

RCVCCOM is set by H_RESET or S_RESET and is not affected by STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI controller will set the value of this bit to ZERO.

3 TXSTRT — Transmit Start status is set whenever PCnet-PCI controller begins transmission of a frame.

When TXSTRT is set, $\overline{INTA}$ is asserted if IENA = 1 and the mask bit TXSTRTM (CSR4 bit 2) is clear.

TXSTRT is set by the MAC Unit and cleared by writing a "1", by H_RESET, S_RESET or by asserting the STOP bit. Writing a "0" has no effect.

2 TXSTRTM — Transmit Start Mask. If TXSTRTM is set, the TXSTRT bit in CSR4 will be masked and unable to set INTR flag in CSR0.

Read/Write accessible. TXSTRTM is set by H_RESET or S_RESET and is not affected by the STOP bit.

1 JAB — Jabber Error is set when the PCnet-PCI controller Twisted-pair MAU function exceeds an allowed transmission limit. Jabber is set by the TMAU cell and can only be asserted in10BASE-T mode.

When JAB is set, $\overline{INTA}$ is asserted if IENA = 1 and the mask bit JABM (CSR4 bit 0) is clear.

JAB is set by the TMAU circuit and cleared by writing a "1". Writing a "0" has no effect. JAB is also cleared by H_RESET, S_RESET or by asserting the STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI controller will never set the value of this bit to ONE.

0 JABM — Jabber Error Mask. If JABM is set, the JAB bit in CSR4 will be masked and unable to set INTR flag in CSR0.

Read/Write accessible. JABM is set by H_RESET or S_RESET and is not affected by the STOP bit.

When the value 01h has been programmed into the SWSTYLE register (BCR20, bits 7–0) for ILACC (Am79C900) compatibility, then this bit has no meaning and PCnet-PCI controller will set the value of this bit to ZERO.

CSR6: RX/TX Descriptor Table Length

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	TLEN	Contains a copy of the transmit encoded ring length (TLEN) field read from the initialization block during PCnet-PCI controller initialization. This field is written during the PCnet-PCI controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. TLEN is only defined after initialization. These bits are unaffected by H_RESET, S_RESET or STOP.
11–8	RLEN	Contains a copy of the receive encoded ring length (RLEN) read from the initialization block during PCnet-PCI controller initialization. This field is written during

		the PCnet-PCI controller initialization routine. Read accessible only when STOP bit is set. Write operations have no effect and should not be performed. RLEN is only defined after initialization. These bits are unaffected by H_RESET, S_RESET or STOP.
7–0	RES	Reserved locations. Read as ZERO. Write operations should not be performed.

CSR8: Logical Address Filter, LADRF[15:0]

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	LADRF[15:0]	Logical Address Filter, LADRF[15:0]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR9: Logical Address Filter LADRF[31:16]

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	LADRF[31:16]	Logical Address Filter, LADRF[31:16]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR10: Logical Address Filter, LADRF[47:32]

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	LADRF[47:32]	Logical Address Filter, LADRF[47:32]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR11: Logical Address Filter, LADRF[63:48]

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	LADRF[63:48]	Logical Address Filter, LADRF[63:48]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR12: Physical Address Register, PADR[15:0]

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PADR[15:0]	Physical Address Register, PADR[15:0]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR13: Physical Address Register, PADR[31:16]

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PADR[31:16]	Physical Address Register, PADR[31:16]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR14: Physical Address Register, PADR[47:32]

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PADR[47:32]	Physical Address Register, PADR[47:32]. The content of this register is undefined until loaded from the initialization block after the INIT bit in CSR0 has been set or a direct I/O write has been performed on this register. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR15: Mode Register

Bit	Name	Description
		This register's fields are loaded during the PCnet-PCI controller initialization routine with the corresponding Initialization Block values or a direct I/O write has been performed on this register.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	PROM	Promiscuous Mode. When PROM = "1", all incoming receive frames are accepted. Read/write accessible only when STOP bit is set.
14	DRCVBC	Disable Receive Broadcast. When set, disables the PCnet-PCI controller from receiving broadcast messages. Used for protocols that do not support broadcast addressing, except as a function of multicast. DRCVBC is cleared by activation of H_RESET or S_RESET (broadcast messages will be received) and is unaffected by STOP. Read/write accessible only when STOP bit is set.
13	DRCVPA	Disable Receive Physical Address. When set, the physical address detection (Station or node ID) of the PCnet-PCI controller will be disabled. Frames addressed to the nodes individual physical address will not be recognized. Read/write accessible only when STOP bit is set.
12	DLNKTST	Disable Link Status. When DLNKTST = "1", monitoring of Link Pulses is disabled. When DLNKTST = "0", monitoring of Link Pulses is enabled. This pin only has meaning when the 10BASE-T network interface is selected. Read/write accessible only when STOP bit is set.
11	DAPC	Disable Automatic Polarity Correction. When DAPC = "1", the 10BASE-T receive polarity reversal algorithm is disabled. Likewise, when DAPC = "0", the polarity reversal algorithm is enabled. This bit only has meaning when the 10BASE-T network interface is selected. Read/write accessible only when STOP bit is set.
10	MENDECL	MENDEC Loopback Mode. See the description of the LOOP bit in CSR15. Read/write accessible only when STOP bit is set.
9	LRT	Low Receive Threshold (T-MAU Mode only)
	TSEL	Transmit Mode Select (AUI Mode only)
	LRT	Low Receive Threshold. When LRT = "1", the internal twisted pair receive thresholds are reduced by 4.5 dB below the standard 10BASE-T value (approximately 3/5) and the unsquelch threshold for the RXD circuit will be 180 mV – 312 mV peak. When LRT = "0", the unsquelch threshold for the RXD circuit will be the standard 10BASE-T value, 300 – 520 mV peak. In either case, the RXD circuit post squelch threshold will be one half of the unsquelch threshold. This bit only has meaning when the 10BASE-T network interface is selected. Read/write accessible only when STOP bit is set. Cleared by H_RESET or S_RESET and is unaffected by STOP.
	TSEL	Transmit Mode Select. TSEL controls the levels at which the

AUI drivers rest when the AUI transmit port is idle. When TSEL = 0, DO+ and DO– yield "zero" differential to operate transformer coupled loads (Ethernet 2 and 802.3). When TSEL = 1, the DO+ idles at a higher value with respect to DO–, yielding a logical HIGH state (Ethernet 1).

This bit only has meaning when the AUI network interface is selected.

Read/write accessible only when STOP bit is set. Cleared by H_RESET or S_RESET.

8–7 PORTSEL[1:0] Port Select bits allow for software controlled selection of the network medium.

PORTSEL settings of AUI and 10BASE-T are ignored when the ASEL bit of BCR2 (bit 1) has been set to ONE.

The network port configurations are as follows:

PORTSEL CSR15[1:0]	ASEL (BCR2 [1])	Link Status (of 10BASE-T)	Network Port
0 X	1	Fail	AUI
0 X	1	Pass	10BASE-T
0 0	0	X	AUI
0 1	0	X	10BASE-T
1 0	X	X	Reserved
1 1	X	X	Reserved

Read/write accessible only when STOP bit is set. Cleared by H_RESET or S_RESET and is unaffected by STOP.

6 INTL Internal Loopback. See the description of LOOP, CSR15[2].

Read/write accessible only when STOP bit is set.

5 DRTY Disable Retry. When DRTY = "1", PCnet-PCI controller will attempt only one transmission. If DRTY = "0", PCnet-PCI controller will attempt 16 transmissions before signaling a retry error.

Read/write accessible only when STOP bit is set.

4 FCOLL Force Collision. This bit allows the collision logic to be tested. PCnet-PCI controller must be in internal loopback for FCOLL to be valid. If FCOLL = "1", a collision will be forced during loopback transmission attempts; a Retry Error will ultimately result. If FCOLL = "0", the Force Collision logic will be disabled. FCOLL is defined after the Initialization Block is read.

Read/write accessible only when STOP bit is set.

3 DXMTFCS Disable Transmit CRC (FCS). When DXMTFCS = 0, the transmitter will generate and append a FCS to the transmitted frame. When DXMTFCS = 1, the FCS logic is allocated to the receiver and no FCS is generated or sent with the transmitted frame. DXMTFCS is overridden when ADD_FCS is set in TMD1.

See also the ADD_FCS bit in TMD1. If DXMTFCS is set and ADD_FCS is clear for a particular frame, no FCS will be generated. The value of ADD_FCS is valid only when STP is set in TMD1. If ADD_FCS is set for a particular frame, the state of DXMTFCS is ignored and a FCS will be appended on that frame by the transmit circuitry.

In loopback mode, this bit determines if the transmitter appends FCS or if the receiver checks the FCS.

This bit was called DTCR in the LANCE (Am7990).

Read/write accessible only when STOP bit is set.

2 LOOP Loopback Enable allows PCnet-PCI controller to operate in full duplex mode for test purposes. When LOOP = "1", loopback is enabled. In combination with INTL and MENDECL, various loopback modes are defined as follows:

LOOP	INTL	MENDECL	Loopback Mode
0	X	X	Non-loopback
1	0	X	External Loopback
1	1	0	Internal Loopback Include MENDEC
1	1	1	Internal Loopback Exclude MENDEC

Read/write accessible only when STOP bit is set. LOOP is cleared by H_RESET or S_RESET and is unaffected by STOP.

1 DTX Disable Transmit results in PCnet-PCI controller not accessing the Transmit Descriptor Ring

		and therefore no transmissions are attempted. DTX = "0", will set TXON bit (CSR0 bit 4) if STRT (CSR0 bit 1) is asserted. Read/write accessible only when STOP bit is set.
0	DRX	Disable Receiver results in PCnet-PCI controller not accessing the Receive Descriptor Ring and therefore all receive frame data are ignored. DRX = "0", will set RXON bit (CSR0 bit 5) if STRT (CSR0 bit 1) is asserted. Read/write accessible only when STOP bit is set.

CSR16: Initialization Block Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	IADR[15:0]	This register is an alias of CSR1.

CSR17: Initialization Block Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	IADR[31:16]	This register is an alias of CSR2.

CSR18: Current Receive Buffer Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CRBAL	Contains the lower 16 bits of the current receive buffer address at which the PCnet-PCI controller will store incoming frame data. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR19: Current Receive Buffer Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CRBAU	Contains the upper 16 bits of the current receive buffer address at which the PCnet-PCI controller will store incoming frame data. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR20: Current Transmit Buffer Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CXBAL	Contains the lower 16 bits of the current transmit buffer address from which the PCnet-PCI controller is transmitting. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR21: Current Transmit Buffer Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CXBAU	Contains the upper 16 bits of the current transmit buffer address from which the PCnet-PCI controller is transmitting. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR22: Next Receive Buffer Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NRBAL	Contains the lower 16 bits of the next receive buffer address to which the PCnet-PCI controller will store incoming frame data. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR23: Next Receive Buffer Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NRBAU	Contains the upper 16 bits of the next receive buffer address to which the PCnet-PCI controller will store incoming frame data.

		Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR24: Base Address of Receive Ring Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	BADRL	Contains the lower 16 bits of the base address of the Receive Ring. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR25: Base Address of Receive Ring Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	BADRU	Contains the upper 16 bits of the base address of the Receive Ring. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR26: Next Receive Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NRDAL	Contains the lower 16 bits of the next RDRE address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR27: Next Receive Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NRDAU	Contains the upper 16 bits of the next RDRE address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR28: Current Receive Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CRDAL	Contains the lower 16 bits of the current RDRE address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR29: Current Receive Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CRDAU	Contains the upper 16 bits of the current RDRE address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR30: Base Address of Transmit Ring Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	BADXL	Contains the lower 16 bits of the base address of the Transmit Ring. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR31: Base Address of Transmit Ring Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	BADXU	Contains the upper 16 bits of the base address of the Transmit Ring. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR32: Next Transmit Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NXDAL	Contains the lower 16 bits of the next TDRE address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR33: Next Transmit Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NXDAU	Contains the upper 16 bits of the next TDRE address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR34: Current Transmit Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CXDAL	Contains the lower 16 bits of the current TDRE address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR35: Current Transmit Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	CXDAU	Contains the upper 16 bits of the current TDRE address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR36: Next Next Receive Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NNRDAL	Contains the lower 16 bits of the next next receive descriptor address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR37: Next Next Receive Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NNRDAU	Contains the upper 16 bits of the next next receive descriptor address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR38: Next Next Transmit Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NNXDAL	Contains the lower 16 bits of the next next transmit descriptor address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR39: Next Next Transmit Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.

15–0	NNXDAU	Contains the upper 16 bits of the next next transmit descriptor address pointer. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR40: Current Receive Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written as ZERO.
11–0	CRBC	Current Receive Byte Count. This field is a copy of the BCNT field of RMD1 of the current receive descriptor. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR41: Current Receive Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	CRST	Current Receive Status. This field is a copy of bits 31–24 of RMD1 of the current receive descriptor. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.
7–0	RES	Reserved locations. Read and written as ZERO.

CSR42: Current Transmit Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written as ZERO.
11–0	CXBC	Current Transmit Byte Count. This field is a copy of the BCNT field of TMD1 of the current transmit descriptor. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR43: Current Transmit Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	CXST	Current Transmit Status. This field is a copy of bits 31–24 of TMD1 of the current transmit descriptor. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.
7–0	RES	Reserved locations. Read and written as ZERO.

CSR44: Next Receive Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written as ZERO.
11–0	NRBC	Next Receive Byte Count. This field is a copy of the BCNT field of RMD1 of the next receive descriptor. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR45: Next Receive Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	NRST	Next Receive Status. This field is a copy of bits 31–24 of RMD1 of the next receive descriptor. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.
7–0	RES	Reserved locations. Read and written as ZERO.

CSR46: Poll Time Counter

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	POLL	Poll Time Counter. This counter is incremented by the PCnet-PCI controller microcode and is used to trigger the descriptor ring polling operation of the PCnet-PCI controller. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR47: Polling Interval

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	POLLINT	Polling Interval. This register contains the time that the PCnet-PCI controller will wait between successive polling operations. The POLLINT value is expressed as the twos complement of the desired interval, where each bit of POLLINT represents 1 CLK period of time. POLLINT[3:0] are ignored. (POLLINT[16] is implied to be a one, so POLLINT[15] is significant, and does not represent the sign of the twos complement POLLINT value.) The default value of this register is 0000b. This corresponds to a polling interval of 65,536 clock periods (1.966 ms when CLK = 33 MHz). The POLLINT value of 0000b is created during the microcode initialization routine, and therefore might not be seen when reading CSR47 after H_RESET or S_RESET. If the user desires to program a value for POLLINT other than the default, then the correct procedure is to first set INIT only in CSR0. Then, when the initialization sequence is complete, the user must set STOP in CSR0. Then the user may write to CSR47 and then set STRT in CSR0. In this way, the default value of 0000b in CSR47 will be overwritten with the desired user value. If the user does NOT use the standard initialization procedure (standard implies use of an initialization block in memory and setting the INIT bit of CSR0), but instead, chooses to write directly to each of the registers that are involved in the INIT operation, then it is imperative that the user also write 0000 0000 to CSR47 as part of the alternative initialization sequence. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR58: Software Style

Bit	Name	Description
		This register is an alias of the location BCR20. Accesses to/from this register are equivalent to accesses to BCR20.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–10	RES	Reserved locations. Written as ZEROs and read as undefined.
9	CSRPCNET	CSR PCnet-ISA configuration bit. When set, this bit indicates that the PCnet-PCI controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the PCnet–ISA (Am79C960) device. When cleared, this bit indicates that PCnet-PCI controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the ILACC (Am79C900) device. The value of CSRPCNET is determined by the PCnet-PCI controller. CSRPCNET is read only by the host. The PCnet-PCI controller uses the setting of the Software Style register (BCR20 bits7-0/CSR58 bits 7–0) to determine the value for this bit. CSRPCNET is set by H_RESET and is not affected by S_RESET or STOP.
8	SSIZE32	Software Size 32 bits. When set, this bit indicates that the PCnet-PCI controller utilizes Am79C900 (ILACC) software structures. In particular,

Initialization Block and Transmit and Receive descriptor bit maps are affected. When cleared, this bit indicates that the PCnet-PCI controller utilizes Am79C960 (PCnet-ISA) software structures. ***Note:*** Regardless of the setting of SSIZE32, the Initialization Block must always begin on a double-word boundary.

The value of SSIZE32 is determined by the PCnet-PCI controller. SSIZE32 is read only by the host.

The PCnet-PCI controller uses the setting of the Software Style register (BCR20, bits 7-0/CSR58 bits 7-0) to determine the value for this bit. SSIZE32 is cleared by H_RESET and is not affected by S_RESET or STOP.

If SSIZE32 is reset, then bits IADR[31–24] of CSR2 will be used to generate values for the upper 8 bits of the 32 bit address bus during master accesses initiated by the PCnet-PCI controller. This action is required, since the 16-bit software structures specified by the SSIZE32=0 setting will yield only 24 bits of address for PCnet-PCI controller bus master accesses.

If SSIZE32 is set, then the software structures that are common to the PCnet-PCI controller and the host system will supply a full 32 bits for each address pointer that is needed by the PCnet-PCI controller for performing master accesses.

The value of the SSIZE32 bit has no effect on the drive of the upper 8 address bits. The upper 8 address pins are always driven, regardless of the state of the SSIZE32 bit.

Note that the setting of the SSIZE32 bit has no effect on the defined width for I/O resources. I/O resource width is determined by the state of the DWIO bit.

7–0 SWSTYLE

Software Style register. The value in this register determines the style of I/O and memory resources that are used by the PCnet-PCI controller. The Software Style selection will affect the interpretation of a few bits within the CSR space and the width of the descriptors and initialization block. Specifically: All PCnet-PCI controller CSR bits and BCR bits and all descriptor, buffer and initialization block entries not cited in the table above are unaffected by the Software Style selection and are therefore always fully functional as specified in the CSR and BCR sections.

SWSTYLE [7:0]	Style Name	CSR-PCNET	SSIZE32	Altered Bit Interpretations
00h	LANCE/ PCnet-ISA	1	0	ALL CSR4 bits will function as defined in the CSR4 section. TMD1[29] functions as ADD_FCS
01h	ILACC	0	1	CSR4[9:8], CSR4[5:4] and CSR4[1:0] will have no function, but will be writeable and readable. CSR4[15:10], CSR4[7:6] and CSR4[3:2] will function as defined in the CSR4 section. TMD1[29] becomes NO_FCS.
02h	PCnet-PCI	1	1	ALL CSR4 bits will function as defined in the CSR4 section. TMD1[29] functions as ADD_FCS
All other combs.	Res.	Undef.	Undef.	Undef.

Read/write accessible only when STOP bit is set.

The SWSTLYE register will contain the value 00h following H_RESET or S_RESET and will be unaffected by STOP.

CSR59: IR Register

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	IRREG	Reserved locations. After H_RESET, the value in this register will be 0105h. The settings of this register will have no effect on any PCnet-PCI controller function. This register always contains the same value. It is not writeable. Read accessible only when STOP bit is set.

CSR60: Previous Transmit Descriptor Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PXDAL	Contains the lower 16 bits of the previous TDRE address pointer. PCnet-PCI controller has the capability to stack multiple transmit frames. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR61: Previous Transmit Descriptor Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	PXDAU	Contains the upper 16 bits of the previous TDRE address pointer. PCnet-PCI controller has the capability to stack multiple transmit frames. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR62: Previous Transmit Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written as ZERO. Accessible only when STOP bit is set.
11–0	PXBC	Previous Transmit Byte Count. This field is a copy of the BCNT field of TMD1 of the previous transmit descriptor. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR63: Previous Transmit Status Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	PXST	Previous Transmit Status. This field is a copy of bits 31–24 of TMD1 of the previous transmit descriptor. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.
7–0	RES	Reserved locations. Read and written as ZERO. Accessible only when STOP bit is set.

CSR64: Next Transmit Buffer Address Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NXBAL	Contains the lower 16 bits of the next transmit buffer address from which the PCnet-PCI controller will transmit an outgoing frame. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR65: Next Transmit Buffer Address Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	NXBAU	Contains the upper 16 bits of the next transmit buffer address from which the PCnet-PCI controller will transmit an outgoing frame. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR66: Next Transmit Byte Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved locations. Read and written as ZERO. Accessible only when STOP bit is set.
11–0	NXBC	Next Transmit Byte Count. This field is a copy of the BCNT field of TMD1 of the next transmit descriptor.

		Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR67: Next Transmit Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–8	NXST	Next Transmit Status. This field is a copy of bits 31–24 of TMD1 of the next transmit descriptor. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.
7–0	RES	Reserved locations. Read and written as ZERO. Accessible only when STOP bit is set.

CSR72: Receive Ring Counter

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	RCVRC	Receive Ring Counter location. Contains a Two's complement binary number used to number the current receive descriptor. This counter interprets the value in CSR76 as pointing to the first descriptor. A counter value of ZERO corresponds to the last descriptor in the ring. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR74: Transmit Ring Counter

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	XMTRC	Transmit Ring Counter location. Contains a Two's complement binary number used to number the current transmit descriptor. This counter interprets the value in CSR78 as pointing to the first descriptor. A counter value of ZERO corresponds to the last descriptor in the ring. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR76: Receive Ring Length

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	RCVRL	Receive Ring Length. Contains the two's complement of the receive descriptor ring length. This register is initialized during the PCnet-PCI initialization routine based on the value in the RLEN field of the initialization block. However, this register can be manually altered. The actual receive ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR78: Transmit Ring Length

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	XMTRL	Transmit Ring Length. Contains the two's complement of the transmit descriptor ring length. This register is initialized during the PCnet-PCI initialization routine based on the value in the TLEN field of the initialization block. However, this register can be manually altered. The actual transmit ring length is defined by the current value in this register. The ring length can be defined as any value from 1 to 65535. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR80: DMA Transfer Counter and FIFO Threshold Control

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–14	RES	Reserved locations. Read as ones and written as ZERO.

13–12 RCVFW[1:0] Receive FIFO Watermark. RCVFW controls the point at which receive DMA is requested in relation to the number of received bytes in the receive FIFO. RCVFW specifies the number of bytes which must be present (once the frame has been verified as a non-runt) before receive DMA is requested. Note however that in order for receive DMA to be performed for a new frame, at least 64 bytes must have been received. This effectively avoids having to react to receive frames which are runts or suffer a collision during the slot time (512 bit times). If the Runt Packet Accept feature is enabled, receive DMA will be requested as soon as either the RCVFW threshold is reached, or a complete valid receive frame is detected (regardless of length). RCVFW is set to a value of 10b (64 bytes) after H_RESET or S_RESET and is unaffected by STOP.

RCVFW[1:0]	Bytes Received
00	16
01	32
10	64
11	Reserved

Read/write accessible only when STOP bit is set.

Certain combinations of watermark programming and LINBC (BCR18, bits 2–0) programming may create situations where no linear bursting is possible, or where the FIFO may be excessively read or excessively written. Such combinations are declared as illegal.

Combinations of watermark settings and LINBC (BCR 18, bits 2–0) settings must obey the following relationship:

watermark (in bytes) ≥ LINBC (in bytes)

Combinations of watermark and LINBC settings that violate this rule may cause unexpected behavior.

11–10 XMTSP[1:0] Transmit Start Point. XMTSP controls the point at which preamble transmission attempts commence in relation to the number of bytes written to the transmit FIFO for the current transmit frame. When the entire frame is in the FIFO, transmission will start regardless of the value in XMTSP. XMTSP is given a value of 10b (64 bytes) after H_RESET or S_RESET and is unaffected by STOP. Regardless of XMTSP, the FIFO will not internally over write its data until at least 64 bytes (or the entire frame if < 64 bytes) have been transmitted onto the network. This ensures that for collisions within the slot time window, transmit data need not be re-written to the transmit FIFO, and re-tries will be handled autonomously by the MAC. This bit is read/write accessible only when the STOP bit is set.

XMTSP[1:0]	Bytes Written
00	4
01	16
10	64
11	112

9–8 XMTFW[1:0] Transmit FIFO Watermark. XMTFW specifies the point at which transmit DMA stops, based upon the number of write cycles that could be performed to the transmit FIFO without FIFO overflow. Transmit DMA is allowed at any time when the number of write cycles specified by XMTFW could be executed without causing transmit FIFO overflow. XMTFW is set to a value of 00b (8 cycles) after H_RESET or S_RESET and is unaffected by STOP. Read/write accessible only when STOP bit is set.

XMTFW[1:0]	Bytes Written
00	16
01	32
10	64
11	Reserved

Certain combinations of watermark programming and LINBC (BCR18, bits 2–0) programming may create situations where no linear bursting is possible, or where the FIFO may be excessively read or excessively written. Such combinations are declared as illegal.

Combinations of watermark settings and LINBC (BCR18, bits 2–0) settings must obey the following relationship:

watermark (in bytes) ≥ LINBC (in bytes)

Combinations of watermark and LINBC settings that violate this rule may cause unexpected behavior.

7–0 DMATC[7:0] DMA Transfer Counter. This counter contains the maximum allowable number of transfers to system memory that the Bus Interface Unit will perform during a single bus mastership period. The DMA Transfer Counter is not used to limit the number of transfers during Descriptor transfers. A value of ZERO will be interpreted as one transfer. During H_RESET or S_RESET a value of 16 is loaded in the DMA Transfer Counter. The value of DMATC is unaffected by the assertion of the STOP bit. If the DMAPLUS bit in CSR4 is set the DMA Transfer Counter is disabled.

When the DMA Transfer Counter times out in the middle of a linear burst, the linear burst will continue until a legal starting address is reached, and then the PCnet-PCI controller will relinquish the bus.

Therefore, if linear bursting is enabled, and the user wishes the PCnet-PCI controller to limit bus activity to desired_max transfers, then the DMA Transfer Counter should be programmed to a value of:

DMA Transfer Counter = (desired_max DIV (length of burst in transfers)) x length of burst in transfers

where DIV is the operation that yields the INTEGER portion of the ÷ operation.

Read/write accessible only when the STOP bit is set.

CSR82: Bus Activity Timer

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	DMABAT	Bus Activity Timer Register. If the TIMER bit in CSR4 is set, this register contains the maximum allowable time that PCnet-PCI controller will take up on the system bus during FIFO data transfers for a single DMA cycle. The Bus Activity Timer Register does not limit the number of transfers during Descriptor transfers.

The DMABAT value is interpreted as an unsigned number with a resolution of 0.1 µs. For instance, a value of 51 µs would be programmed with a value of 510. If the TIMER bit in CSR4 is set, DMABAT is enabled and must be initialized by the user. The DMABAT register is undefined until written.

If the user has NOT enabled the Linear Burst function and wishes the PCnet-PCI controller to limit bus activity to MAX_TIME microseconds, then the Burst Timer should be programmed to a value of:

MAX_TIME– ((11 + 4w) x (CLK period))

where w = wait states

If the user has enabled the Linear Burst function and wishes the PCnet-PCI controller to limit bus activity to MAX_TIME microseconds, then the Burst Timer should be programmed to a value of:

MAX_TIME– (((3+lbs) x w + 10 + lbs) x (CLK period))

where w = wait states and lbs = linear burst size in number of transfers per sequence

This is because the PCnet-PCI controller may use as much as one linear burst size plus three transfers in order to complete the

linear burst before releasing the bus.

As an example, if the linear burst size is 4 transfers, and the number of wait states for the system memory is 2, and the CLK period is 30ns and the MAX time allowed on the bus is 3 µs, then the Burst Timer should be programmed for:

MAX_TIME– (((3+lbs) x w + 10 + lbs) x (CLK period))

3 µs – (((3 + 4) x 2 +10 + 4) x (30 ns)) = 3 µs – (28 x 30 ns) = 3 – 0.84 µs = 2.16 µs.

Then, if the PCnet-PCIs Bus Activity Timer times out after 2.16µs when the PCnet-PCI controller has completed all but the last three transfers of a linear burst, the PCnet-PCI controller may take as much as 0.84 µs to complete the bursts and release the bus. The bus release will occur at 2.16 + 0.84 = 3 µs.

A value of ZERO in the DMABAT register with the TIMER bit in CSR4 set to ONE will produce single linear burst sequences per bus master period when programmed for linear burst mode, and will yield sets of 3 transfers when not programmed for linear burst mode.

The Bus Timer register is set to a value of 00h after H_RESET or S_RESET and is unaffected by STOP.

Read/write accessible only when STOP bit is set.

CSR84: DMA Address Register Lower

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	DMABAL	DMA Address Register. This register contains the lower 16 bits of the address of system memory for the current DMA cycle. The Bus Interface Unit controls the Address Register by issuing increment commands to increment the memory address for sequential operations. The DMABA register is undefined until the first PCnet-PCI controller DMA operation. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR85: DMA Address Register Upper

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	DMABAU	DMA Address Register. This register contains the upper 16 bits of the address of system memory for the current DMA cycle. The Bus Interface Unit controls the Address Register by issuing increment commands to increment the memory address for sequential operations. The DMABA register is undefined until the first PCnet-PCI controller DMA operation. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR86: Buffer Byte Counter

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–12	RES	Reserved, Read and written with ones.
11–0	DMABC	DMA Byte Count Register. Contains the two's complement of the current size of the remaining transmit or receive buffer in bytes. This register is incremented by the Bus Interface Unit. The DMABC register is undefined until written. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR88: Chip ID Register Lower

Bit	Name	Description
31 – 28		Version. This 4-bit pattern is silicon-revision dependent.
27 – 12		Part number. The 16-bit code for the PCnet-PCI controller is 0010 0100 0011 0000b.

	Note that this code is not the same as the Device ID in the PCI configuration space.
11 – 1	Manufacturer ID. The 11-bit manufacturer code for AMD is 00000000001. This code is per the JEDEC Publication 106-A. Note that this code is not the same as the Vendor ID in the PCI configuration space.
0	Always a logic 1.

CSR89: Chip ID Register Upper

Bit	Name	Description
31 – 16		Reserved locations; Read as undefined.
15 – 12		Version. This 4-bit pattern is silicon-revision dependent.
11 – 0		Upper 12 bits of the PCnet-PCI controller part number. i.e. 0010 0100 0011b.

CSR92: Ring Length Conversion

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	RCON	Ring Length Conversion Register. This register performs a ring length conversion from an encoded value as found in the initialization block to a two's complement value used for internal counting. By writing bits 15–12 with an encoded ring length, a Two's complemented value is read. The RCON register is undefined until written. Read/write accessible only when STOP bit is set. These bits are unaffected by H_RESET, S_RESET or STOP.

CSR94: Transmit Time Domain Reflectometry Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–10	RES	Reserved locations. Read and written as ZERO.
9–0	XMTTDR	Time Domain Reflectometry reflects the state of an internal counter that counts from the start of transmission to the occurrence of loss of carrier. TDR is incremented at a rate of 10 MHz. Read accessible only when STOP bit is set. Write operations are ignored. XMTTDR is cleared by H_RESET or S_RESET.

CSR100: Bus Timeout

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	MERRTO	This register contains the value of the longest allowable bus latency (interval between assertion of $\overline{REQ}$ and assertion of $\overline{GNT}$) that a system may insert into a PCnet-PCI controller master transfer. If this value of bus latency is exceeded, then a MERR will be indicated in CSR0, bit 11, and an interrupt may be generated, depending upon the setting of the MERRM bit (CSR3, bit 11) and the IENA bit (CSR0, bit 6). The value in this register is interpreted as the unsigned number of XTAL1 clock periods divided 2. (i.e., the value in this register is given in 0.1 µsecond increments.) For example, the value 0200h (512 decimal) will cause a MERR to be indicated after 51.2 microseconds of bus latency. A value of ZERO will allow an infinitely long bus latency; i.e. a value of ZERO will never give a bus timeout error. This register is set to 0200h by H_RESET or S_RESET and is unaffected by STOP. Read/write accessible only when STOP bit is set.

CSR112: Missed Frame Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	MFC	Missed Frame Count. Indicates the number of missed frames. MFC will roll over to a count of ZERO from the value 65535. The MPCO bit of CSR4 (bit 8) will be set each time that this occurs.

This register is always readable and is cleared by H_RESET or S_RESET or STOP.

A write to this register performs an increment when the ENTST bit in CSR4 is set.

CSR114: Receive Collision Count

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	RCC	Receive Collision Count. Indicates the total number of collisions encountered by the receiver since the last reset of the counter. RCC will roll over to a count of ZERO from the value 65535. The RCVCCO bit of CSR4 (bit 5) will be set each time that this occurs. The RCC value is read accessible at all times, regardless of the value of the STOP bit. Write operations are ignored. RCC is cleared by H_RESET or S_RESET or by setting the STOP bit. A write to this register performs an increment when the ENTST bit in CSR4 is set.

CSR122: Receive Frame Alignment Control

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–1	RES	Reserved locations, written as ZEROs and read as undefined.
0	RCVALGN	Receive Packet Align. When set, this bit forces the data field of ISO 8802-3 (IEEE/ANSI 802.3) packets to align to 0 MOD 4 address boundaries (i.e. DWORD (double-word) aligned addresses). It is important to note that this feature will only function correctly if all receive buffer boundaries are DWORD aligned and all receive buffers have 0 MOD 4 lengths. In order to accomplish the data alignment, the PCnet-PCI controller simply inserts two bytes of random data at the beginning of the receive packet (i.e. before the ISO 8802-3 (IEEE/ANSI 802.3) destination address field). The MCNT field reported to the receive descriptor will not include the extra two bytes. RCVALGN is cleared by H_RESET or S_RESET and is not affected by STOP. Read/write accessible only when STOP bit is set.

CSR124: Test Register 1

Bit	Name	Description
		This register is used to place the PCnet-PCI into various test modes. Only the Runt Packet Accept is an user accessible test mode. All other test modes are for AMD internal use only. ENTST must set before programming CSR124. ENTST must be reset after writing to CSR124 before writing to any other register. All bits in this register are cleared by H_RESET or S_RESET and are not affected by STOP.
31–4	RES	Reserved locations. Written as ZEROs and read as undefined.
3	RPA	Runt Packet Accept. This bit forces the PCnet-PCI controller receive logic to accept runt packets (packets shorter than 64 bytes). The state of the RPA bit can be changed only when the device is in the test mode (when ENTST bit in CSR4 is set to ONE). To enable RPA, the user must first write a ONE to the ENTST bit. Next, the user must first write a ONE to the RPA bit (CSR124, bit 3). Finally, the user must write a ZERO to the ENTST bit to take the device out of test mode operation. Once, the RPA bit has been set to ONE, the device will remain in the Runt Packet Accept mode until the RPA bit is cleared to ZERO.
2–0	RES	Reserved locations. Written as ZEROs and read as undefined.

Bus Configuration Registers

The Bus Configuration Registers (BCR) are used to program the configuration of the bus interface and other special features of the PCnet-PCI controller that are not related to the IEEE 8802-3 MAC functions. The BCRs are accessed by first setting the appropriate RAP value, and then by performing a slave access to the BDP.

All BCR registers are 16 bits in width in WIO mode and 32 bits in width in DWIO mode. The upper 16 bits of all BCR registers is undefined when in DWIO mode. These bits should be written as ZEROs and should be treated as undefined when read. The Default value given for any BCR is the value in the register after H_RESET, and is hexadecimal unless otherwise stated. BCR register values are unaffected by S_RESET and are unaffected by the assertion of the STOP bit.

Note that several registers have no default value. BCR3 and BCR8-15 are reserved and have undefined values. BCR2, BCR16, BCR17 and BCR21 are not observable without first being programmed through the EEPROM read operation or a user register write operation. Therefore the only observable values for these registers are those that have been programmed and a default value is not applicable.

BCR0, BCR1, BCR6, BCR16, BCR17, and BCR21 are reserved in the PCnet-PCI controller. These registers are used by other devices in the PCnet family. Writing to these registers have no effect on the operation of the PCnet-PCI controller.

Writes to those registers marked as "Reserved" will have no effect. Reads from these locations will produce undefined values.

BCR	MNEMONIC	Default	Description	User	EEPROM
0	MSRDA	0005h	Reserved	No	No
1	MSWRA	0005h	Reserved	No	No
2	MC	N/A*	Miscellaneous Configuration	Yes	Yes
3	Reserved	N/A	Reserved	No	No
4	LNKST	00C0h	Link Status Status (Default)	Yes	No
5	LED1	0084h	Receive Status (Default)	Yes	No
6	LED2	0088h	Reserved	Yes	No
7	LED3	0090h	Transmit Status (Default)	Yes	No
8–15	Reserved	N/A	Reserved	No	No
16	IOBASEL	N/A*	Reserved	Yes	Yes
17	IOBASEU	N/A*	Reserved	Yes	Yes
18	BSBC	2101h	Burst Size and Bus Control	Yes	Yes
19	EECAS	0002h	EEPROM Control and Status	Yes	No
20	SWS	0000h	Software Style	Yes	No
21	INTCON	N/A*	Reserved	Yes	Yes

** Registers marked with an "*" have no default value, since they are not observable without first being programmed through the EEPROM read operation or a user register write operation. Therefore, the only observable values for these registers are those that have been programmed and a default value is not applicable.*

BCR0: Master Mode Read Active

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	MSRDA	Reserved locations. After H_RESET, the value in this register will be 0005h. The settings of this register will have no effect on any PCnet-PCI controller function. Writes to this register have no effect on the operation of the PCnet-PCI controller and will not alter the value that is read.

BCR1: Master Mode Write Active

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	MSWRA	Reserved locations. After H_RESET, the value in this register will be 0005h. The settings of this register will have no effect on any PCnet-PCI controller function. Writes to this register have no effect on the operation of the PCnet-PCI controller and will not alter the value that is read.

BCR2: Miscellaneous Configuration

Bit	Name	Description
		Note that all bits in this register are programmable through the EEPROM PREAD operation.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	RES	Reserved location. Written and read as ZERO.
14	TMAULOOP	When set, this bit allows external loopback packets to pass onto the network through the TMAU interface, if the TMAU interface has been selected. If the TMAU interface has not been selected, then this bit has no effect. This bit is reset to ZERO by H_RESET and is unaffected by S_RESET or STOP.
13–9	RES	Reserved locations. Written and read as ZERO.
8	IESRWE	IEEE Shadow Ram Write Enable. The PCnet-PCI controller contains a shadow RAM on board for storage of the IEEE address following the serial EEPROM read operation. Accesses to APROM I/O Resources will be directed toward this RAM. When IESRWE is set to a ONE, then write access to the shadow RAM will be enabled. This bit is reset to ZERO by H_RESET and is unaffected by S_RESET or STOP.
7	RES	Reserved location. The default of this bit is zero. Writing a ONE to this bit has no effect on the operation of the PCnet-PCI controller. This reserved location is cleared by H_RESET and is unaffected by S_RESET or STOP.
6–3	RES	Reserved locations. Written and read as ZERO.
2	AWAKE	This bit selects one of two different sleep modes. If AWAKE=1 and the $\overline{\text{SLEEP}}$ pin is asserted, the PCnet-PCI controller goes into snooze mode. If AWAKE=0 and the $\overline{\text{SLEEP}}$ pin is asserted, the PCnet-PCI controller goes into coma mode. See Power Saving Modes section for more details. This bit only has meaning when the 10BASE-T network interface is selected. This bit is reset to ZERO by H_RESET and is unaffected by S_RESET or STOP.
1	ASEL	Auto Select. When set, the PCnet-PCI controller will automatically select the operating media interface port. If ASEL has been set to a ONE, then when the 10BASE-T transceiver is in the link pass state (due to receiving valid frame data and/or Link Test pulses or the DLNKTST bit is set), the 10BASE-T port will be used. If ASEL has been set to a ONE, then when the 10BASE-T port is in the link fail state, the AUI port will be used. Switching between the ports will not occur during transmission, to avoid any type of fragment generation. When ASEL is set to ONE, Link Beat Pulses will be transmitted on the 10BASE-T port, regardless of the state of Link Status. When ASEL is reset to ZERO, Link Beat Pulses will only be transmitted on the 10BASE-T

port when the PORTSEL bits of the Mode Register (CSR15) have selected 10BASE-T as the active port.

When ASEL is set to a ZERO, then the selected network port will be determined by the settings of the PORTSEL bits of CSR15.

The ASEL bit is reset to ONE by H_RESET and is unaffected by S_RESET or STOP.

The network port configurations are as follows:

PORTSEL CSR15[1:0]	ASEL (BCR2[1])	LINK Status (of 10BASE-T)	Network Port
0 X	1	Fail	AUI
0 X	1	Pass	10BASE-T
0 0	0	X	AUI
0 1	0	X	10BASE-T
1 0	X	X	Reserved
1 1	X	X	Reserved

0 XMAUSEL — Reserved location. The default value of this bit is a ZERO. Writing a ONE to this bit has no effect on the operation of the device. Existing drivers may write a ONE to this bit, but new drivers should write a ZERO to this bit.

BCR4: Link Status LED (LNKST)

Bit	Name	Description
		BCR4 controls the function(s) that the $\overline{\text{LNKST}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR4 defaults to Link Status (LNKST) with pulse stretcher enabled (PSE = 1) and is fully programmable. The default setting after H_RESET for the LNKST register is 00C0h. The LNKST register value is unaffected by S_RESET or STOP.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (Bits 6–0). This bit is READ only by the host, and is unaffected by H_RESET, S_RESET or STOP.
14	LEDPOL	LED Polarity. When this bit has the value ZERO, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false. (i.e. the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit.) When this bit has the value ONE, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false. (i.e. the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit.) The setting of this bit will not affect the polarity of the LEDOUT bit for this register.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value ONE, then the LED output will always be floated. When LEDDIS has the value ZERO, then the LED output value will be governed by the LEDOUT and LEDPOL values.
12–8	RES	Reserved locations. Write as ZEROS, read as undefined.
7	PSE	A value of 0 disables the signal. A value of 1 enables the signal. Pulse Stretcher Enable. Extends the LED illumination time for each new occurrence of the enabled function for this LED output. A value of 0 disables the function. A value of 1 enables the function.
6	LNKSTE	Link Status Enable. Indicates the current link status on the Twisted Pair interface. When this bit is set to one, a value of ONE will be passed to the LEDOUT signal to indicate that the link status state is PASS. A value of ZERO will be passed to the LEDOUT signal to indicate that the link status state is FAIL.

		A value of 0 disables the signal. A value of 1 enables the signal.
5	RCVME	Receive Match status Enable. Indicates receive activity on the network that has passed the address match function for this node. All address matching modes are included: Physical, Logical filtering, Promiscuous and Broadcast. A value of 0 disables the signal. A value of 1 enables the signal.
4	XMTE	Transmit status Enable. Indicates PCnet-PCI controller transmit activity. A value of 0 disables the signal. A value of 1 enables the signal.
3	RXPOLE	Receive Polarity status Enable. Indicates the current Receive Polarity condition on the Twisted Pair interface. A value of ONE indicates that the polarity of the RXD± pair has been reversed. A value of ZERO indicates that the polarity of the RXD± pair has *not* been reversed. Receive polarity indication is valid only if the T-MAU is in Link Pass state. A value of 0 disables the signal. A value of 1 enables the signal.
2	RCVE	Receive status Enable. Indicates receive activity on the network. A value of 0 disables the signal. A value of 1 enables the signal.
1	JABE	Jabber status Enable. Indicates that the PCnet-PCI controller is jabbering on the network. A value of 0 disables the signal. A value of 1 enables the signal.
0	COLE	Collision status Enable. Indicates collision activity on the network. A value of 0 disables the signal. A value of 1 enables the signal.

BCR5: LED1 Status

Bit	Name	Description
		BCR5 controls the function(s) that the $\overline{\text{LED1}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR1 defaults to Receive Status (RCV) with pulse stretcher enabled (PSE = 1) and is fully programmable.
		The default setting after H_RESET for the LED1 register is 0084h. The LED1 register value is unaffected by S_RESET or STOP.
31–16	RES	Reserved locations. Written as ZERO and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (Bits 6–0). This bit is READ only by the host, and is unaffected by H_RESET, S_RESET or STOP.
14	LEDPOL	LED Polarity. When this bit has the value ZERO, then the LED pin will be driven to a LOW level whenever the OR of the enabled signals is true and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false. (i.e. the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit.) When this bit has the value ONE, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false. (i.e. the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit.) The setting of this bit will not affect the polarity of the LEDOUT bit for this register.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value ONE, then the LED output will always be tristated. When LEDDIS has the value ZERO, then the LED output value will be governed by the LEDOUT and LEDPOL values.
12–8	RES	Reserved locations. Write as ZEROS, read as undefined. A value of 0 disables the signal. A value of 1 enables the signal.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination time for

		each new occurrence of the enabled function for this LED output. A value of 0 disables the signal. A value of 1 enables the signal.
6	LNKSTE	Link Status Enable. Indicates the current link status on the Twisted Pair interface. When this bit is enabled, a value of ONE will be passed to the LEDOUT signal to indicate that the link status state is PASS. A value of ZERO will be passed to the LEDOUT signal to indicate that the link status state is FAIL. A value of 0 disables the signal. A value of 1 enables the signal.
5	RCVME	Receive Match status Enable. Indicates receive activity on the network that has passed the address match function for this node. All address matching modes are included: Physical, Logical filtering and Promiscuous. A value of 0 disables the signal. A value of 1 enables the signal.
4	XMTE	Transmit status Enable. Indicates PCnet-PCI controller transmit activity. A value of 0 disables the signal. A value of 1 enables the signal.
3	RXPOLE	Receive Polarity status Enable. Indicates the current Receive Polarity condition on the Twisted Pair interface. A value of ONE indicates that the polarity of the RXD± pair has been reversed. A value of ZERO indicates that the polarity of the RXD± pair has *not* been reversed. Receive polarity indication is valid only if the T-MAU is in the Link Pass state A value of 0 disables the signal. A value of 1 enables the signal.
2	RCVE	Receive status Enable. Indicates receive activity on the network. A value of 0 disables the signal. A value of 1 enables the signal.
1	JABE	Jabber status Enable. Indicates that the PCnet-PCI controller is jabbering on the network. A value of 0 disables the signal. A value of 1 enables the signal.
0	COLE	Collision status Enable. Indicates collision activity on the network. A value of 0 disables the signal. A value of 1 enables the signal.

BCR6: LED2 Status

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	LED2	Reserved locations. After H_RESET, the value in this register will be 0x0088h. The settings of this register will have no effect on any PCnet-PCI controller function. Writes to this register have no effect on the operation of the PCnet-PCI controller.

BCR7: LED3 Status

Bit	Name	Description
		BCR7 controls the function(s) that the $\overline{\text{LED3}}$ pin displays. Multiple functions can be simultaneously enabled on this LED pin. The LED display will indicate the logical OR of the enabled functions. BCR7 defaults to Transmit Status (XMT) with pulse stretcher enabled (PSE = 1) and is fully programmable. The default setting after H_RESET for the LED3 register is 0090h. The LED3 register value is unaffected by S_RESET or STOP.
31–16	RES	Reserved location. Written as ZEROs and read as undefined.
15	LEDOUT	This bit indicates the current (non-stretched) value of the LED output pin. A value of ONE in this bit indicates that the OR of the enabled signals is true. The logical value of the LEDOUT status signal is determined by the settings of the individual Status Enable bits of the LED register (Bits 6–0). This bit is READ only by the host, and is unaffected by H_RESET, S_RESET or STOP.
14	LEDPOL	LED Polarity. When this bit has the value ZERO, then the LED

		pin will be driven to a LOW level whenever the OR of the enabled signals is true and the LED pin will be disabled and allowed to float high whenever the OR of the enabled signals is false. (i.e. the LED output will be an Open Drain output and the output value will be the inverse of the LEDOUT status bit.) When this bit has the value ONE, then the LED pin will be driven to a HIGH level whenever the OR of the enabled signals is true and the LED pin will be driven to a LOW level whenever the OR of the enabled signals is false. (i.e. the LED output will be a Totem Pole output and the output value will be the same polarity as the LEDOUT status bit.) The setting of this bit will not affect the polarity of the LEDOUT bit for this register.
13	LEDDIS	LED Disable. This bit is used to disable the LED output. When LEDDIS has the value ONE, then the LED output will always be tristated. When LEDDIS has the value ZERO, then the LED output value will be governed by the LEDOUT and LEDPOL values.
12–8	RES	Reserved locations. Write as ZEROS, read as undefined. A value of 0 disables the signal. A value of 1 enables the signal.
7	PSE	Pulse Stretcher Enable. Extends the LED illumination time for each new occurrence of the enabled function for this LED output. A value of 0 disables the signal. A value of 1 enables the signal.
6	LNKSTE	Link Status Enable. Indicates the current link status on the Twisted Pair interface. When this bit is enabled, a value of ONE will be passed to the LEDOUT signal to indicate that the link status state is PASS. A value of ZERO will be passed to the LEDOUT signal to indicate that the link status state is FAIL. A value of 0 disables the signal. A value of 1 enables the signal.
5	RCVME	Receive Match status Enable. Indicates receive activity on the network that has passed the address match function for this node. All address matching modes are included: Physical, Logical filtering and Promiscuous. A value of 0 disables the signal. A value of 1 enables the signal.
4	XMTE	Transmit status Enable. Indicates PCnet-PCI controller transmit activity. A value of 0 disables the signal. A value of 1 enables the signal.
3	RXPOLE	Receive Polarity status Enable. Indicates the current Receive Polarity condition on the Twisted Pair interface. A value of ONE indicates that the polarity of the RXD± pair has been reversed. A value of ZERO indicates that the polarity of the RXD± pair has *not* been reversed. Receive polarity indication is valid only if the T-MAU is in the Link Pass state. A value of 0 disables the signal. A value of 1 enables the signal.
2	RCVE	Receive status Enable. Indicates receive activity on the network. A value of 0 disables the signal. A value of 1 enables the signal.
1	JABE	Jabber status Enable. Indicates that the PCnet-PCI controller is jabbering on the network. A value of 0 disables the signal. A value of 1 enables the signal.
0		COLE Collision status Enable. Indicates collision activity on the network. A value of 0 disables the signal. A value of 1 enables the signal.

BCR16: I/O Base Address Lower

Bit	Name	Description
		Note that all bits in this register are programmable through the EEPROM PREAD operation.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–5	IOBASEL	Reserved locations. After H_RESET, the value of these bits will be undefined. The settings of these bits will have no affect on any PCnet-PCI controller function. IOBASEL is not affected by S_RESET or STOP.

4–0	RES	Reserved locations. Written as ZEROS, read as undefined.

BCR17: I/O Base Address Upper

Bit	Name	Description
		Note that all bits in this register are programmable through the EEPROM PREAD operation.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	IOBASEU	Reserved locations. After H_RESET, the value in this register will be undefined. The settings of this register will have no affect on any PCnet-PCI controller function. IOBASEU is not affected by S_RESET or STOP.

BCR18: Burst Size and Bus Control Register

Bit	Name	Description
		Note that all bits, except bit 7, in this register are programmable through the EEPROM PREAD operation.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–11	RES	Reserved locations. After H_RESET, these five bits will read 00100b. The settings of these bits will have no affect on any PCnet-PCI controller function. Writes to these bits have no affect on the operation of PCnet-PCI controller. RES is set to 00100b by H_RESET and is not affected by S_RESET or STOP.
10	RES	Reserved location. Must be written as ZERO. Writing a one to this bit may cause the PCnet-PCI controller to malfunction in a system. This reserved location is cleared by H_RESET and is not affected by S_RESET or STOP.
9	RES	Reserved location. Written as ZERO and read as undefined. This reserved location is cleared by H_RESET and is not affected by S_RESET or STOP.
8	RES	Reserved bit. Must be written as a ONE. Will be read as a ONE. This reserved location is SET by H_RESET and is not affected by S_RESET or STOP.
7	DWIO	Double Word I/O. When set, this bit indicates that the PCnet-PCI controller is programmed for DWIO mode. When cleared, this bit indicates that the PCnet-PCI controller is programmed for Word I/O mode. This bit affects the I/O Resource Offset map and it affects the defined width of the PCnet-PCI controller's I/O resources. See the DWIO and WIO sections for more details. The PCnet-PCI controller will set DWIO if it detects a DWORD *write* access to offset 10h from the PCnet-PCI controller I/O base address (corresponding to the RDP resource). A double-word write access to offset 10h is the only way that the DWIO bit can be set. DWIO cannot be set by a direct write to BCR18. Once the DWIO bit has been set to a ONE, only a H_RESET can reset it to a ZERO. DWIO is read only by the host. DWIO is cleared by H_RESET and is not affected by S_RESET or STOP.
6	BREADE	Burst Read Enable. When set, this bit enables Linear Bursting during memory read accesses, where Linear Bursting is defined to mean that only the first transfer in the current bus arbitration will contain an address phase. Subsequent transfers will consist of data phases only. When cleared, this bit prevents the part from performing linear bursting during read accesses. In no case will the part linearly burst a descriptor access or an initialization access. BREADE should be set to ONE when the PCnet-PCI controller is used in a PCI bus application. The use of burst transfers guarantees maximum performance during memory read operations. BREADE is cleared by H_RESET and is not affected by S_RESET or STOP.

5 BWRITE

Burst Write Enable. When set, this bit enables Linear Bursting during memory write accesses, where Linear Bursting is defined to mean that only the first transfer in the current bus arbitration will contain an address phase. Subsequent transfers will consist of data phases only. When cleared, this bit prevents the part from performing linear bursting during write accesses. In no case will the part linearly burst a descriptor access or an initialization access.

BWRITE should be set to ONE when the PCnet-PCI controller is used in a PCI bus application. The use of burst transfers guarantees maximum performance during memory write operations.

BWRITE is cleared by H_RESET and is not affected by S_RESET or STOP.

4–3 RES

Reserved location. Written as ZEROs and read as undefined.

2–0 LINBC[2:0]

Linear Burst Count. The 3 bit value in this register sets the upper limit for the number of transfer cycles in a Linear Burst. This limit determines how often the PCnet-PCI controller will assert a new FRAME signal during linear burst transfers. Each time that the interpreted value of LINBC transfers is reached, the PCnet-PCI controller will assert a new FRAME signal with a new valid address. The LINBC value should contain only one active bit. LINBC values with more than one active bit may produce predictable results, but such values will not be compatible with future AMD network controllers.The LINBC entry is shifted by two bits before being used by the PCnet-PCI controller. For example, the value LINBC[2:0] = 010b is understood by the PCnet-PCI controller to mean 01000b = 8. Therefore, the value LINBC[2:0] = 010b will cause the PCnet-PCI controller to issue a new FRAME every 01000b = 8 transfers. The PCnet-PCI controller may linearly burst fewer than the value represented by LINBC, due to other conditions that cause the burst to end prematurely. Therefore, LINBC should be regarded as an upper limit to the length of linear burst.

Note that linear burst operation will only begin on certain addresses. The general rule for linear burst starting addresses is:

AD[31:00] MOD (LINBC x 16) = 0,

The following table illustrates all possible starting address values for all legal LINBC values. Note that AD[31:06] are don't care values for all addresses. Also note that while AD[1:0] do not physically exist within a 32 bit system (the PCnet-PCI controller always drives AD[1:0] to ZERO during the address phase to indicate a linear burst order), they are valid bits within the buffer pointer field of descriptor word 0.

LINBC[2:0]	LBS = Linear Burst Size (number of transfers)	Size of Burst (bytes)	Linear Burst Beginning Addresses AD[31:6] = (don't care) (AD[5:0] = (Hex)
1	4	16	00, 10, 20, 30
2	8	32	00, 20
4	16	64	00

There are several events which may cause early termination of linear burst. Among those events are: no more data available for transfer in either a buffer or in the FIFO or if either the DMA Transfer Counter (CSR80) or the Bus Timer Register (CSR82) times out.

Certain combinations of watermark programming and LINBC programming may create situations where no linear bursting is possible, or where the FIFO may be excessively read or excessively written. Such combinations are declared as illegal.

Combinations of watermark settings and LINBC settings must obey the following relationship:

watermark (in bytes) ≥ LINBC (in bytes)

Combinations of watermark and LINBC settings that violate this rule may cause unexpected behavior.

LINBC is set to the value of 001b by H_RESET and is not affected by S_RESET or STOP. This

gives a default linear burst length of 4 transfers = 001b x 4.

BCR19: EEPROM Control and Status Register

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15	PVALID	EEPROM Valid status bit. This bit is read only by the host. A value of ONE in this bit indicates that a PREAD operation has occurred, and that 1) there is an EEPROM connected to the PCnet-PCI controller Microwire interface pins and 2) the contents read from the EEPROM have passed the checksum verification operation. A value of ZERO in this bit indicates that the contents of the EEPROM are different from the contents of the applicable PCnet-PCI controller on-board registers and/or that the checksum for the entire 36 bytes of EEPROM is incorrect *or* that no EEPROM is connected to the Microwire interface pins. PVALID is set to ZERO during H_RESET and is unaffected by S_RESET or the STOP bit. However, following the H_RESET operation, an automatic read of the EEPROM will be performed. Just as is true for the normal PREAD command, at the end of this automatic read operation, the PVALID bit may be set to ONE. Therefore, H_RESET will set the PVALID bit to ZERO at first, but the automatic EEPROM read operation may later set PVALID to a ONE. If PVALID becomes ZERO following an EEPROM read operation (either automatically generated after H_RESET, or requested through PREAD), then all EEPROM-programmable BCR locations will be reset to their H_RESET values. The content of the APROM locations, however, will not be cleared. If no EEPROM is present at the EESK, EEDI and EEDO pins, then all attempted PREAD commands will terminate early and PVALID will NOT be set. This applies to the automatic read of the EEPROM after H_RESET as well as to host–initiated PREAD commands.
14	PREAD	EEPROM Read command bit. When this bit is set to a ONE by the host, the PVALID bit (BCR19, bit 15) will immediately be reset to a ZERO and then the PCnet-PCI controller will perform a read operation of 36 bytes from the EEPROM through the Microwire interface. The EEPROM data that is fetched during the read will be stored in the appropriate internal registers on board the PCnet-PCI controller. Upon completion of the EEPROM read operation, the PCnet-PCI controller will assert the PVALID bit. EEPROM contents will be indirectly accessible to the host through I/O read accesses to the APROM (offsets 0h through Fh) and through I/O read accesses to other EEPROM programmable registers. Note that I/O read accesses from these locations will not actually access the EEPROM itself, but instead will access the PCnet-PCI controllers internal copy of the EEPROM contents. I/O write accesses to these locations may change the PCnet-PCI controller register contents, but the EEPROM locations will not be affected. EEPROM locations may be accessed directly through BCR19. At the end of the read operation, the PREAD bit will automatically be reset to a ZERO by the PCnet-PCI controller and PVALID will bet set, provided that an EEPROM existed on the Microwire interface pins and that the checksum for the entire 36 bytes of EEPROM was correct. Note that when PREAD is set to a ONE, then the PCnet-PCI controller will no longer respond to I/O accesses directed toward it, until the PREAD operation has completed successfully. The PCnet-PCI controller will terminate these I/O accesses with the assertion of $\overline{\text{DEVSEL}}$ and $\overline{\text{STOP}}$ while $\overline{\text{TRDY}}$ is not asserted, signaling to the initiator to retry the access at a later time. If a PREAD command is given to the PCnet-PCI controller but no EEPROM is attached to the Microwire interface pins, then the

PREAD command will terminate early, the PREAD bit will be cleared to a ZERO and the PVALID bit will remain reset with a value of ZERO. This applies to the automatic read of the EEPROM after H_RESET as well as to host initiated PREAD commands. EEPROM programmable locations on board the PCnet-PCI controller will be set to their default values by such an aborted PREAD operation. For example, if the aborted PREAD operation immediately followed the H_RESET operation, then the final state of the EEPROM programmable locations will be equal to the H_RESET programming for those locations.

If a PREAD command is given to the PCnet-PCI controller and the auto-detection pin (EESK/$\overline{\text{LED1}}$) indicates that no EEPROM is present, then the EEPROM read operation will still be attempted.

Note that at the end of the H_RESET operation, a read of the EEPROM will be performed automatically. This H_RESET–generated EEPROM read function will not proceed if the auto-detection pin (EESK/$\overline{\text{LED1}}$) indicates that no EEPROM is present.

PREAD is set to ZERO during H_RESET and is unaffected by S_RESET or the STOP bit.

PREAD is only writeable when the STOP bit is set to ONE.

13 EEDET

EEPROM Detect. This bit indicates the sampled value of the EESK/$\overline{\text{LED1}}$ pin at the end of H_RESET. This value indicates whether or not an EEPROM is present at the EEPROM interface. If this bit is a ONE, it indicates that an EEPROM is present. If this bit is a ZERO, it indicates that an EEPROM is not present.

The value of this bit is determined at the end of the H_RESET operation. It is unaffected by S_RESET or the STOP bit.

This bit is not writeable. It is read only.

The following table indicates the possible combinations of EEDET and the existence of an EEPROM and the resulting operations that are possible on the EEPROM Microwire interface:

Table 10. EEDET Combinations

EEDET Value (BCR19[3])	EEPROM Connected?	Result if PREAD is Set to ONE	Result of Automatic EEPROM Read Operation Following H_RESET
0	No	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to ZERO.	First TWO EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to ZERO.
0	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.	First TWO EESK clock cycles are generated, then EEPROM read operation is aborted and PVALID is reset to ZERO.
1	No	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to ZERO.	EEPROM read operation is attempted. Entire read sequence will occur, checksum failure will result, PVALID is reset to ZERO.
1	Yes	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.	EEPROM read operation is attempted. Entire read sequence will occur, checksum operation will pass, PVALID is set to ONE.

12–5 RES Reserved locations. Written as ZERO, read as undefined.

4 EEN EEPROM port enable. When this bit is set to a one, it causes the values of ECS, ESK and EDI to be driven onto the EECS, EESK and EEDI pins, respectively. If EEN=0 and no EEPROM read function is currently active, then EECS will be driven LOW. When EEN=0 and no EEPROM read function is currently active, EESK and EEDI pins will be driven by the LED registers BCR5 and BCR4, respectively.

EEN is set to ZERO by H_RESET and is unaffected by the S_RESET or STOP bit.

3 RES Reserved location. Written as ZERO and read as undefined.

2 ECS EEPROM Chip Select. This bit is used to control the value of the EECS pin of the Microwire interface when the EEN bit is set to ONE and the PREAD bit is set to ZERO. If EEN = 1 and PREAD = 0 and ECS is set to a ONE, then the EECS pin will be forced to a HIGH level at the rising edge of the next CLK following bit programming. If EEN = 1 and PREAD = 0 and ECS is set to a ZERO, then the EECS pin will be forced to a LOW level at the rising edge of the next CLK following bit programming. ECS has no effect on the output value of the EECS pin unless the PREAD bit is set to ZERO and the EEN bit is set to ONE.

ECS is set to ZERO by H_RESET and is not affected by S_RESET or STOP.

1 ESK EEPROM Serial Clock. This bit and the EDI/EDO bit are used to control host access to the EEPROM. Values programmed to this bit are placed onto the EESK pin at the rising edge of the next CLK following bit programming, except when the PREAD bit is set to ONE or the EEN bit is set to ZERO. If both the ESK bit and the EDI/EDO bit values are changed during one BCR19 write operation, while EEN = 1, then setup and hold times of the EEDI pin value with respect to the EESK signal edge are not guaranteed.

ESK has no effect on the EESK pin unless the PREAD bit is set to ZERO and the EEN bit is set to ONE.

ESK is reset to ONE by H_RESET and is not affected by S_RESET or STOP.

Table 11. EEPROM Port Enable

$\overline{\text{RST}}$ Pin	PREAD or Auto Read in Progress	EEN	EECS	EESK	EEDI
Low	X	X	0	Z	Z
High	1	X	Active	Active	Active
High	0	1	From ECS Bit of BCR19	From ESK Bit of BCR19	From EDI Bit of BCR19
High	0	0	0	$\overline{\text{LED1}}$	$\overline{\text{LNKST}}$

0	EDI/EDO	EEPROM Data In / EEPROM Data Out. Data that is written to this bit will appear on the EEDI output of the Microwire interface, except when the PREAD bit is set to ONE or the EEN bit is set to ZERO. Data that is read from this bit reflects the value of the EEDO input of the Microwire interface. EDI/EDO has no effect on the EEDI pin unless the PREAD bit is set to ZERO and the EEN bit is set to ONE. EDI/EDO is reset to ZERO by H_RESET and is not affected by S_RESET or STOP.

BCR20: Software Style

Bit	Name	Description
		This register is an alias of the location CSR58. Accesses to/from this register are equivalent to accesses to CSR58.
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–10	RES	Reserved locations. Written as ZEROs and read as undefined.
9	CSRPCNET	CSR PCnet-ISA configuration bit. When set, this bit indicates that the PCnet-PCI controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the PCnet-ISA (Am79C960) device. When cleared, this bit indicates that PCnet-PCI controller register bits of CSR4 and CSR3 will map directly to the CSR4 and CSR3 bits of the ILACC (Am79C900) device. The value of CSRPCNET is determined by the PCnet-PCI controller. CSRPCNET is read only by the host. The PCnet-PCI controller uses the setting of the Software Style register (BCR20 bits7-0/CSR58 bits 7-0) to determine the value for this bit. CSRPCNET is set by H_RESET and is not affected by S_RESET or STOP.
8	SSIZE32	Software Size 32 bits. When set, this bit indicates that the PCnet-PCI controller utilizes Am79C900 (ILACC) software structures. In particular, Initialization Block and Transmit and Receive descriptor bit maps are affected. When cleared, this bit indicates that the PCnet-PCI controller utilizes Am79C960 (PCnet-ISA) software structures. ***Note:*** Regardless of the setting of SSIZE32, the Initialization Block must always begin on a double-word boundary. The value of SSIZE32 is determined by the PCnet-PCI controller. SSIZE32 is read only by the host. The PCnet-PCI controller uses the setting of the Software Style register (BCR20, bits 7-0/CSR58 bits 7-0) to determine the value for this bit. SSIZE32 is cleared by H_RESET and is not affected by S_RESET or STOP. If SSIZE32 is reset, then bits IADR[31–24] of CSR2 will be used to generate values for the upper 8 bits of the 32 bit address bus during master accesses initiated by the PCnet-PCI controller. This action is required, since the 16-bit software structures specified by the SSIZE32=0 setting will yield only 24 bits of address for PCnet-PCI controller bus master accesses. If SSIZE32 is set, then the software structures that are common to the PCnet-PCI controller and the host system will supply a full 32 bits for each address pointer that is needed by the PCnet-PCI controller for performing master accesses. The value of the SSIZE32 bit has no effect on the drive of the upper 8 address pins. The upper 8 address pins are always driven, regardless of the state of the SSIZE32 bit. Note that the setting of the SSIZE32 bit has no effect on the defined width for I/O resources. I/O resource width is determined by the state of the DWIO bit.

7–0 SWSTYLE Software Style register. The value in this register determines the style of I/O resources that shall be used by the PCnet-PCI controller. The Software Style selection will affect the interpretation of a few bits within the CSR space and the width of the descriptors and initialization block. Specifically:

Table 12. SWSTYLE Values

SWSTYLE [7:0]	Style Name	CSR-PCNET	SSIZE32	Altered Bit Interpretations
00h	LANCE/ PCnet-ISA	1	0	ALL CSR4 bits will function as defined in the CSR4 section. TMD1[29] functions as ADD_FCS
01h	ILACC	0	1	CSR4[9:8], CSR4[5:4] and CSR4[1:0] will have no function, but will be writeable and readable. CSR4[15:10], CSR4[7:6] and CSR4[3:2] will function as defined in the CSR4 section. TMD1[29] becomes NO_FCS.
02h	PCnet-PCI	1	1	ALL CSR4 bits will function as defined in the CSR4 section. TMD1[29] functions as ADD_FCS
All other combs.	Res.	Undef.	Undef.	Undef.

All PCnet-PCI controller CSR bits and BCR bits and all descriptor, buffer and initialization block entries not cited in the table above are unaffected by the Software Style selection and are therefore always fully functional as specified in the CSR and BCR sections.

Read/write accessible only when STOP bit is set.

The SWSTLYE register will contain the value 00h following H_RESET or S_RESET and will be unaffected by STOP.

BCR21: Interrupt Control

Bit	Name	Description
31–16	RES	Reserved locations. Written as ZEROs and read as undefined.
15–0	INTCON	Reserved locations. Writes to this register will have no effect on the operation of the PCnet-PCI controller.

Initialization Block

When SSIZE32=0 (BCR20, bit 8), then the software structures are defined to be 16 bits wide. The base address of the Initialization block in this mode must be aligned to a WORD boundary, i.e. CSR1, bit 0 and CSR16, bit 0 must be set to ZERO. When SSIZE32 = 0, the initialization block looks like Table 13.

Note that the PCnet-PCI device performs DWORD accesses to read the initialization block. This statement is always true, regardless of the setting of the SSIZE32 bit.

When SSIZE32=1 (BCR20, bit 8), then the software structures are defined to be 32 bits wide. The base address of the Initialization block in this mode must be aligned to a DOUBLE WORD boundary, i.e., CSR1, bits 0 and 1 and CSR16, bits 0 and 1 must be set to ZERO. When SSIZE32 = 1, the initialization block looks like Table 14.

Table 13. 16-Bit Data Structure Initialization Block

Address	Bits 15–13	Bit 12	Bits 11–8	Bits 7–4	Bits 3–0
IADR+00h	MODE 15–00				
IADR+02h	PADR 15–00				
IADR+04h	PADR 31–16				
IADR+06h	PADR 47–32				
IADR+08h	LADRF 15–00				
IADR+0Ah	LADRF 31–16				
IADR+0Ch	LADRF 47–32				
IADR+0Eh	LADRF 63–48				
IADR+10h	RDRA 15–00				
IADR+12h	RLEN	0	RES	RDRA 23–16	
IADR+14h	TDRA 15–00				
IADR+16h	TLEN	0	RES	TDRA 23–16	

Table 14. 32-Bit Data Structure Initialization Block

Address	Bits 31–28	Bits 27–24	Bits 23–20	Bits 19–16	Bits 15–12	Bits 11–8	Bits 7–4	Bits 3–0
IADR+00h	TLEN	RES	RLEN	RES	MODE			
IADR+04h	PADR 31–00							
IADR+08h	RES				PADR 47–32			
IADR+0Ch	LADR 31–00							
IADR+10h	LADR 63–32							
IADR+14h	RDRA 31–00							
IADR+18h	TDRA 31–00							

RLEN and TLEN

When SSIZE32=0 (BCR20, bit 8), then the software structures are defined to be 16 bits wide, and the RLEN and TLEN fields in the initialization block are 3 bits wide, occupying bits 15, 14, and 13, and the value in these fields determines the number of Receive and Transmit Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is as follows:

R/TLEN	# of DREs
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

If a value other than those listed in the above table is desired, CSR76 and CSR78 can be written after initialization is complete. See the description of the appropriate CSRs.

When SSIZE32=1 (BCR20, bit 8), then the software structures are defined to be 32 bits wide, and the RLEN and TLEN fields in the initialization block are 4 bits wide, occupying bits 23–20 (RLEN) and 31–28 (TLEN) and the value in these fields determines the number of Receive and Transmit Descriptor Ring Entries (DRE) which are used in the descriptor rings. Their meaning is as follows:

R/TLEN	# of DREs
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
11XX	512
1X1X	512

If a value other than those listed in the above table is desired, CSR76 and CSR78 can be written after initialization is complete. See the description of the appropriate CSRs.

RDRA and TDRA

TDRA and RDRA indicate where the transmit and receive descriptor rings, respectively, begin. When SSIZE32=1 (BCR20, bit 8), each DRE must be aligned to a 16-byte boundary (TDRA [3:0]=0, RDRA [3:0]=0). When SSIZE32=0 (BCR20, bit 8), each DRE must be aligned to an 8-byte boundary (TDRA [2:0]=0, RDRA [2:0]=0).

LADRF

The Logical Address Filter (LADRF) is a 64-bit mask that is used to accept incoming Logical Addresses. If the first bit in the destination address of the incoming frame (as received from the wire) is a "1", the address is deemed logical. If the first bit is a "0", it is a physical address and is compared against the physical address that was loaded through the initialization block.

A logical address is passed through the CRC generator, producing a 32-bit result. The high order 6 bits of the CRC is used to select one of the 64-bit positions in the Logical Address Filter. If the selected filter bit is set, the address is accepted and the frame is placed into memory.

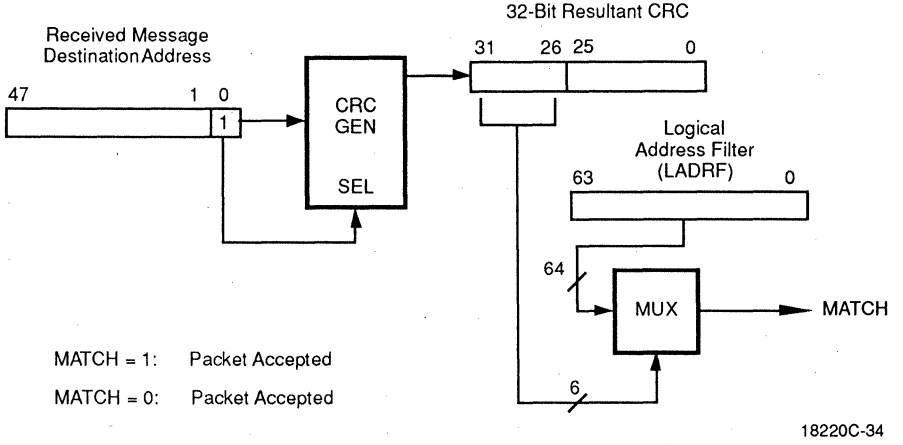


Figure 32. Address Match Logic

The Logical Address Filter is used in multicast addressing schemes. The acceptance of the incoming frame based on the filter value indicates that the message may be intended for the node. It is the node's responsibility to determine if the message is actually intended for the node by comparing the destination address of the stored message with a list of acceptable logical addresses.

If the Logical Address Filter is loaded with all ZEROs and promiscuous mode is disabled, all incoming logical addresses except broadcast will be rejected.

The Broadcast address, which is all ones, does not go through the Logical Address Filter and is handled as follows:

- If the Disable Broadcast Bit is cleared, the broadcast address is accepted.
- If the Disable Broadcast Bit is set and promiscuous mode is enabled, the broadcast address is accepted.
- If the Disable Broadcast Bit is set and promiscuous mode is disabled, the broadcast address is rejected.

If external loopback is used, the FCS logic must be allocated to the receiver (by setting the DXMTFCS bit in CSR15, and clearing the ADD_FCS bit in TMD1) when using multicast addressing.

PADR

This 48-bit value represents the unique node address assigned by the ISO 8802-3 (IEEE/ANSI 802.3) and used for internal address comparison. PADR[0] is compared with the first bit in the destination address set the incoming frame (as received from the wire) the first address bit transmitted on the wire, and must be ZERO. The six hex-digit nomenclature used by the ISO 8802-3 (IEEE/ANSI 802.3) maps to the PCnet-PCI PADR register as follows: the first byte is PADR[7:0], with PADR[0] being the least significant bit of the byte. The second ISO 8802-3 (IEEE/ANSI 802.3) byte is compared with PADR[15:8], again from LS bit to MS bit, and so on. The sixth byte is compared with PADR[47:40], the LS bit being PADR[40].

MODE

The mode register in the initialization block is copied into CSR15 and interpreted according to the description of CSR15.

Receive Descriptors

When SSIZE32=0 (BCR20, bit 8), then the software structures are defined to be 16 bits wide, and receive descriptors look like this (CRDA = Current Receive Descriptor Address):

Table 15. 16-Bit Data Structure Receive Descriptor

Address	15	14	13	12	11	10	9	8	7–0
CRDA+00h	RBADR[15:0]								
CRDA+02h	OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	RBADR[23:16]
CRDA+04h	1	1	1	1	BCNT				
CRDA+06h	0	0	0	0	MCNT				

When SSIZE32=1 (BCR 20, bit 8), then the software structures are defined to be 32 bits wide, and receive descriptors look like this (CRDA = Current Receive Descriptor Address):

Table 16. 32-Bit Data Structure Receive Descriptor

Address	31	30	29	28	27	26	25	24	23–16	15–12	11–0
CRDA+00h	RBADR[31:0]										
CRDA+04h	OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	RES	1111	BCNT
CRDA+08h	RCC								RPC	0000	MCNT
CRDA+0Ch	RESERVED										

RMD0

Bit	Name	Description
31–0	RBADR	Receive Buffer address. This field contains the address of the receive buffer that is associated with this descriptor.

RMD1

Bit	Name	Description
31	OWN	This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-PCI controller (OWN=1). The PCnet-PCI controller clears the OWN bit after filling the buffer pointed to by the descriptor entry. The host sets the OWN bit after emptying the buffer. Once the PCnet-PCI controller or host has relinquished ownership of a buffer, it must not change any field in the descriptor entry.
30	ERR	ERR is the OR of FRAM, OFLO, CRC, or BUFF. ERR is set by the PCnet-PCI controller and cleared by the host.
29	FRAM	FRAMING ERROR indicates that the incoming frame contained a non-integer multiple of eight bits and there was an FCS error. If there was no FCS error on the incoming frame, then FRAM will not be set even if there was a non integer multiple of eight bits in the frame. FRAM is not valid in internal loopback mode. FRAM is valid only when ENP is set and OFLO is not. FRAM is set by the PCnet-PCI controller and cleared by the host.
28	OFLO	OVERFLOW error indicates that the receiver has lost all or part of the incoming frame, due to an inability to store the frame in a memory buffer before the internal FIFO overflowed. OFLO is valid only when ENP is not set. OFLO is set by the PCnet-PCI controller and cleared by the host.
27	CRC	CRC indicates that the receiver has detected a CRC (FCS) error on the incoming frame. CRC is valid only when ENP is set and OFLO is not. CRC is set by the PCnet-PCI controller and cleared by the host.
26	BUFF	BUFFER ERROR is set any time the PCnet-PCI controller does not own the next buffer while data chaining a received frame. This can occur in either of two ways: 1. The OWN bit of the next buffer is ZERO. 2. FIFO overflow occurred before the PCnet-PCI controller received the STATUS byte (RMD1[31:24] of the next descriptor). If a Buffer Error occurs, an Overflow Error may also occur internally in the FIFO, but will not be reported in the descriptor status entry unless both BUFF and OFLO errors occur at the same time . BUFF is set by the PCnet-PCI controller and cleared by the host.
25	STP	START OF PACKET indicates that this is the first buffer used by the PCnet-PCI controller for this frame. It is used for data chaining buffers. When SPRINTEN=0 (CSR3, bit 5), STP is set by the PCnet-PCI controller and cleared by the host. When SPRINTEN=1 (CSR3, bit 5), STP must be set by the host.
24	ENP	END OF PACKET indicates that this is the last buffer used by the PCnet-PCI controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the PCnet-PCI controller and cleared by the host.
23–16	RES	Reserved locations. These locations should be read and written as ZEROs.
15–12	ONES	These four bits must be written as ONES. They are written by the host and unchanged by the PCnet-PCI controller.
11–00	BCNT	BUFFER BYTE COUNT is the length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This field is written by the host and unchanged by the PCnet-PCI controller.

RMD2

Bit	Name	Description
31–24	RCC	Receive Collision Count. Indicates the accumulated number of collisions on the network since the last packet was received, excluding collisions that occurred during transmissions from this node. The PCnet-PCI implementation of this counter may not be compatible with the ILACC RCC definition. If network statistics are to be monitored, then CSR114 should be used for the purpose of monitoring Receive collisions instead of these bits.
23–16	RPC	Runt Packet Count. Indicates the accumulated number of runts that were addressed to this node since the last time that a receive packet was successfully received and its corresponding RMD2 ring entry was written to by the PCnet-PCI controller. In order to be included in the RPC value, a runt must be long enough to meet the minimum requirement of the internal address matching logic. The minimum requirement for a runt to pass the internal address matching mechanism is: 18 bits of valid preamble plus a valid SFD detected, followed by 7 bytes of frame data. This requirement is unvarying, regardless of the address matching mechanisms in force at the time of reception. (i.e. physical, logical, broadcast or promiscuous). The PCnet-PCI implementation of this counter may not be compatible with the ILACC RPC definition.
15–12	ZEROS	This field is reserved. PCnet-PCI controller will write ZEROs to these locations.
11–0	MCNT	MESSAGE BYTE COUNT is the length in bytes of the received message, expressed as an unsigned binary integer. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the PCnet-PCI controller and cleared by the host.

RMD3

Bit	Name	Description
31–0	RES	Reserved locations.

Transmit Descriptors

When SSIZE32=0 (BCR 20, bit 8), then the software structures are defined to be 16 bits wide, and transmit descriptors look like this (CXDA = Current Transmit Descriptor Address):

Table 17. 16-Bit Data Structure Transmit Descriptor

Address	15	14	13	12	11	10	9	8	7–0
CXDA+00h	TBADR[15:0]								
CXDA+02h	OWN	ERR	ADD_/ NO_ FCS	MORE	ONE	DEF	STP	ENP	TBADR[23:16]
CXDA+04h	1	1	1	1	BCNT				
CXDA+06h	BUFF	UFLO	EX DEF	LCOL	LCAR	RTRY	TDR		

When SSIZE32=1 (BCR 20, bit 8), then the software structures are defined to be 32 bits wide, and transmit descriptors look like this (CXDA = Current Transmit Descriptor Address):

Table 18. 32-Bit Data Structure Transmit Descriptor

Address	31	30	29	28	27	26	25	24	23–16	15–4	3–0
CXDA+00h	TBADR[31:0]										
CXDA+04h	OWN	ERR	ADD_/ NO_ FCS	MORE	ONE	DEF	STP	ENP	RES	1111	BCNT
CXDA+08h	BUFF	UFLO	EX DEF	LCOL	LCAR	RTRY	TDR			RES	TRC
CXDA+0Ch	RESERVED										

TMD0

Bit	Name	Description
31–0	TBADR	Transmit Buffer address. This field contains the address of the transmit buffer that is associated with this descriptor.

TMD1

Bit	Name	Description
31	OWN	This bit indicates that the descriptor entry is owned by the host (OWN=0) or by the PCnet-PCI controller (OWN=1). The host sets the OWN bit after filling the buffer pointed to by the descriptor entry. The PCnet-PCI controller clears the OWN bit after transmitting the contents of the buffer. Both the PCnet-PCI controller and the host must not alter a descriptor entry after it has relinquished ownership.
30	ERR	ERR is the OR of UFLO, LCOL, LCAR, or RTRY. ERR is set by the PCnet-PCI controller and cleared by the host. This bit is set in the current descriptor when the error occurs, and therefore may be set in any descriptor of a chained buffer transmission.
29	ADD_FCS_ NO_FCS	Bit 29 functions as ADD_FCS when programmed for the default I/O style of PCnet-ISA and when programmed for the I/O style PCnet-PCI controller. Bit 29 functions as NO_FCS when programmed for the I/O style ILACC.
	ADD_FCS	ADD_FCS dynamically controls the generation of FCS on a frame by frame basis. It is valid only if the STP bit is set. When ADD_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is activated. When ADD_FCS = 0, FCS generation is controlled by DXMTFCS. ADD_FCS is set by the host, and unchanged by the PCnet-PCI controller. This was a reserved bit in the LANCE

(Am7990). This function differs from the ILACC function for this bit.

NO_FCS — NO_FCS dynamically controls the generation of FCS on a frame by frame basis. It is valid only if the ENP bit is set. When NO_FCS is set, the state of DXMTFCS is ignored and transmitter FCS generation is deactivated. When NO_FCS = 0, FCS generation is controlled by DXMTFCS. NO_FCS is set by the host, and unchanged by the PCnet-PCI controller. This was a reserved bit in the LANCE (Am7990). This function is identical to the ILACC function for this bit

28 MORE — MORE indicates that more than one re-try was needed to transmit a frame. The value of MORE is written by the PCnet-PCI controller. This bit has meaning only if the ENP bit is set. ONE, MORE, and RTRY are mutually exclusive.

27 ONE — ONE indicates that exactly one re-try was needed to transmit a frame. ONE flag is not valid when LCOL is set. The value of the ONE bit is written by the PCnet-PCI controller. This bit has meaning only if the ENP bit is set. ONE, MORE, and RTRY are mutually exclusive.

26 DEF — DEFERRED indicates that the PCnet-PCI controller had to defer while trying to transmit a frame. This condition occurs if the channel is busy when the PCnet-PCI controller is ready to transmit. DEF is set by the PCnet-PCI controller and cleared by the host.

25 STP — START OF PACKET indicates that this is the first buffer to be used by the PCnet-PCI controller for this frame. It is used for data chaining buffers. The STP bit must be set in the first buffer of the frame, or the PCnet-PCI controller will skip over the descriptor and poll the next descriptor(s) until the OWN and STP bits are set.

STP is set by the host and unchanged by the PCnet-PCI controller.

24 ENP — END OF PACKET indicates that this is the last buffer to be used by the PCnet-PCI controller for this frame. It is used for data chaining buffers. If both STP and ENP are set, the frame fits into one buffer and there is no data chaining. ENP is set by the host and unchanged by the PCnet-PCI controller.

23–16 RES — Reserved locations.

15–12 ONES — MUST BE ONES. This field is written by the host and unchanged by the PCnet-PCI controller.

11–00 BCNT — BUFFER BYTE COUNT is the usable length of the buffer pointed to by this descriptor, expressed as the two's complement of the length of the buffer. This is the number of bytes from this buffer that will be transmitted by the PCnet-PCI controller. This field is written by the host and unchanged by the PCnet-PCI controller. There are no minimum buffer size restrictions.

TMD2

Bit	Name	Description
31	BUFF	BUFFER ERROR is set by the PCnet-PCI controller during transmission when the PCnet-32 controller does not find the ENP flag in the current buffer and does not own the next buffer. This can occur in either of two ways: 1. The OWN bit of the next buffer is ZERO. 2. FIFO underflow occurred before the PCnet-PCI controller obtained the STATUS byte (TMD1[31:24]) of the next descriptor. BUFF is set by the PCnet-PCI controller and cleared by the host. BUFF error will turn off the transmitter (CSR0, TXON = 0). If a Buffer Error occurs, an Underflow Error will also occur. BUFF is not valid when LCOL or RTRY error is set during transmit data chaining. BUFF is set by the PCnet-PCI controller and cleared by the host.
30	UFLO	UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that the FIFO has emptied

		before the end of the frame was reached. Upon UFLO error, the transmitter is turned off (CSR0, TXON = 0). UFLO is set by the PCnet-PCI controller and cleared by the host.
29	EXDEF	Excessive Deferral. Indicates that the transmitter has experienced Excessive Deferral on this transmit frame, where Excessive Deferral is defined in ISO 8802-3 (IEEE/ANSI 802.3).
28	LCOL	LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The PCnet-PCI controller does not re-try on late collisions. LCOL is set by the PCnet-PCI controller and cleared by the host.
27	LCAR	LOSS OF CARRIER is set when the carrier is lost during an PCnet-PCI controller-initiated transmission when in AUI mode. The PCnet-PCI controller does not re-try upon loss of carrier. It will continue to transmit the whole frame until done. LCAR is not valid in Internal Loopback Mode. LCAR is set by the PCnet-PCI controller and cleared by the host. In 10BASE-T mode, LCAR will be set when T-MAU is in link fail state.
26	RTRY	RETRY ERROR indicates that the transmitter has failed after 16 attempts to successfully transmit a message, due to repeated collisions on the medium. If DRTY = 1 in the MODE register, RTRY will set after 1 failed transmission attempt. RTRY is set by the PCnet-PCI controller and cleared by the host. ONE, MORE, and RTRY are mutually exclusive.
25–16	TDR	TIME DOMAIN REFLECTOMETRY reflects the state of an internal PCnet-PCI controller counter that counts at a 10 MHz rate from the start of a transmission to the occurrence of a collision or loss of carrier. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the PCnet-PCI controller and is valid only if RTRY is set. Note that 10 MHz gives very low resolution and in general has not been found to be particularly useful. This feature is here primarily to maintain full compatibility with the LANCE device (Am7990).
15–4	RES	Reserved locations.
3–0	TRC	Transmit Retry Count. Indicates the number of transmit retries of the associated packet. The maximum count is 15. However, if a RETRY error occurs, the count will roll over to ZERO. In this case only, the Transmit Retry Count value of ZERO should be interpreted as meaning 16. TRC is written by the PCnet-PCI controller into the last transmit descriptor of a frame, or when an error terminates a frame. Valid only when OWN = 0.

TMD3

Bit	Name	Description
31–0	RES	Reserved locations.

Register Summary

PCI Configuration Registers

Note: RO = read only, RW = read/write, U = undefined value

Offset	Name	Width in Bit	Access Mode	Default Value
00h	Vendor ID	16	RO	1022h
02h	Device ID	16	RO	2000h
04h	Command	16	RW	uuuu
06h	Status	16	RW	uuuu
08h	Revision ID	8	RO	00h
09h	Programming IF	8	RO	00h
0Ah	Sub-Class	8	RO	00h
0Bh	Base-Class	8	RO	02h
0Dh	Latency Timer	8	RO	00h
0Eh	Header Type	8	RO	00h
10h	Base Address	32	RW	uuuu uuuu
3Ch	Interrupt Line	8	RW	uu
3Dh	Interrupt Pin	8	RO	01h

Control and Status Registers

Note: u = undefined value, R = Running register, S = Setup register, T = Test register

RAP Addr	Symbol	Default Value After H_RESET	Comments	Use
00	CSR0	uuuu 0004	PCnet-PCI Status Register	R
01	CSR1	uuuu uuuu	IADR[15:0]: Base Address of INIT Block Lower	S
02	CSR2	uuuu uuuu	IADR[31:16]: Base Address of INIT Block Upper	S
03	CSR3	uuuu 0000	Interrupt Masks and Deferral Control	S
04	CSR4	uuuu 0115	Test and Features Control	R
05	CSR5	uuuu 0000	Reserved	T
06	CSR6	uuuu uuuu	RXTX: RX/TX Descriptor Table Lengths	T
07	CSR7	uuuu 0000	Reserved	T
08	CSR8	uuuu uuuu	LADR0: Logical Address Filter — LADRF[15:0]	T
09	CSR9	uuuu uuuu	LADR1: Logical Address Filter — LADRF[31:16]	T
10	CSR10	uuuu uuuu	LADR2: Logical Address Filter — LADRF[47:32]	T
11	CSR11	uuuu uuuu	LADR3: Logical Address Filter — LADRF[63:48]	T
12	CSR12	uuuu uuuu	PADR0: Physical Address Register — PADR[15:0]	T
13	CSR13	uuuu uuuu	PADR1: Physical Address Register — PADR[31:16]	T
14	CSR14	uuuu uuuu	PADR2: Physical Address Register — PADR[47:32]	T
15	CSR15	see reg. desc.	MODE: Mode Register	S
16	CSR16	uuuu uuuu	IADR[15:0]: Alias of CSR1	T
17	CSR17	uuuu uuuu	IADR[31:16]: Alias of CSR2	T
18	CSR18	uuuu uuuu	CRBAL: Current RCV Buffer Address Lower	T
19	CSR22	uuuu uuuu	CRBAU: Current RCV Buffer Address Upper	T
20	CSR20	uuuu uuuu	CXBAL: Current XMT Buffer Address Lower	T
21	CSR21	uuuu uuuu	CXBAU: Current XMT Buffer Address Upper	T

Control and Status Registers (continued)

RAP Addr	Symbol	Default Value After H_RESET	Comments	Use
22	CSR22	uuuu uuuu	NRBAL: Next RCV Buffer Address Lower	T
23	CSR23	uuuu uuuu	NRBAU: Next RCV Buffer Address Upper	T
24	CSR24	uuuu uuuu	BADRL: Base Address of RCV Ring Lower	S
25	CSR25	uuuu uuuu	BADRU: Base Address of RCV Ring Upper	S
26	CSR26	uuuu uuuu	NRDAL: Next RCV Descriptor Address Lower	T
27	CSR27	uuuu uuuu	NRDAU: Next RCV Descriptor Address Upper	T
28	CSR28	uuuu uuuu	CRDAL: Current RCV Descriptor Address Lower	T
29	CSR29	uuuu uuuu	CRDAU: Current RCV Descriptor Address Upper	T
30	CSR30	uuuu uuuu	BADXL: Base Address of XMT Ring Lower	S
31	CSR31	uuuu uuuu	BADXU: Base Address of XMT Ring Upper	S
32	CSR32	uuuu uuuu	NXDAL: Next XMT Descriptor Address Lower	T
33	CSR33	uuuu uuuu	NXDAU: Next XMT Descriptor Address Upper	T
34	CSR34	uuuu uuuu	CXDAL: Current XMT Descriptor Address Lower	T
35	CSR35	uuuu uuuu	CXDAU: Current XMT Descriptor Address Upper	T
36	CSR36	uuuu uuuu	NNRDAL: Next Next Receive Descriptor Address Lower	T
37	CSR37	uuuu uuuu	NNRDAU: Next Next Receive Descriptor Address Upper	T
38	CSR38	uuuu uuuu	NNXDAL: Next Next Transmit Descriptor Address Lower	T
39	CSR39	uuuu uuuu	NNXDAU: Next Next Transmit Descriptor Address Upper	T
40	CSR40	uuuu uuuu	CRBC: Current RCV Byte Count	T
41	CSR41	uuuu uuuu	CRST: Current RCV Status	T
42	CSR42	uuuu uuuu	CXBC: Current XMT Byte Count	T
43	CSR43	uuuu uuuu	CXST: Current XMT Status	T
44	CSR44	uuuu uuuu	NRBC: Next RCV Byte Count	T
45	CSR45	uuuu uuuu	NRST: Next RCV Status	T
46	CSR46	uuuu uuuu	POLL: Poll Time Counter	T
47	CSR47	uuuu uuuu	POLLINT: Polling Interval	S
48	CSR48	uuuu uuuu	Reserved	T
49	CSR49	uuuu uuuu	Reserved	T
50	CSR50	uuuu uuuu	Reserved	T
51	CSR51	uuuu uuuu	Reserved	T
52	CSR52	uuuu uuuu	Reserved	T
53	CSR53	uuuu uuuu	Reserved	T
54	CSR54	uuuu uuuu	Reserved	T
55	CSR55	uuuu uuuu	Reserved	T
56	CSR56	uuuu uuuu	Reserved	T
57	CSR57	uuuu uuuu	Reserved	T
58	CSR58	see reg. desc.	SWS: Software Style	S
59	CSR59	uuuu 0105	IR: IR Register	T
60	CSR60	uuuu uuuu	PXDAL: Previous XMT Descriptor Address Lower	T
61	CSR61	uuuu uuuu	PXDAU: Previous XMT Descriptor Address Upper	T
62	CSR62	uuuu uuuu	PXBC: Previous XMT Byte Count	T
63	CSR63	uuuu uuuu	PXST: Previous XMT Status	T
64	CSR64	uuuu uuuu	NXBA: Next XMT Buffer Address Lower	T

Control and Status Registers (continued)

RAP Addr	Symbol	Default Value After H_RESET	Comments	Use
65	CSR65	uuuu uuuu	NXBAU: Next XMT Buffer Address Upper	T
66	CSR66	uuuu uuuu	NXBC: Next XMT Byte Count	T
67	CSR67	uuuu uuuu	NXST: Next XMT Status	T
68	CSR68	uuuu uuuu	Reserved	T
69	CSR69	uuuu uuuu	Reserved	T
70	CSR70	uuuu uuuu	Reserved	T
71	CSR71	uuuu uuuu	Reserved	T
72	CSR72	uuuu uuuu	RCVRC: RCV Ring Counter	T
73	CSR73	uuuu uuuu	Reserved	T
74	CSR74	uuuu uuuu	XMTRC: XMT Ring Counter	T
75	CSR75	uuuu uuuu	Reserved	T
76	CSR76	uuuu uuuu	RCVRL: RCV Ring Length	S
77	CSR77	uuuu uuuu	Reserved	T
78	CSR78	uuuu uuuu	XMTRL: XMT Ring Length	S
79	CSR79	uuuu uuuu	Reserved	T
80	CSR80	uuuu E810	DMATCFW: DMA Transfer Counter and FIFO Threshold	S
81	CSR81	uuuu uuuu	Reserved	T
82	CSR82	uuuu 0000	DMABAT: Bus Activity Timer	S
83	CSR83	uuuu uuuu	Reserved	T
84	CSR84	uuuu uuuu	DMABAL: DMA Address Register Lower	T
85	CSR85	uuuu uuuu	DMABAU: DMA Address Register Upper	T
86	CSR86	uuuu uuuu	DMABC: Buffer Byte Counter	T
87	CSR87	uuuu uuuu	Reserved	T
88	CSR88	0242 0003	Chip ID Register Lower	T
89	CSR89	uuuu 0242	Chip ID Register Upper	T
91	CSR91	uuuu uuuu	Reserved	T
92	CSR92	uuuu uuuu	RCON: Ring Length Conversion	T
93	CSR93	uuuu uuuu	Reserved	T
94	CSR94	uuuu 0000	XMTTDR: Transmit Time Domain Reflectometry Count	T
95	CSR95	uuuu uuuu	Reserved	T
96	CSR96	uuuu uuuu	Reserved	T
97	CSR97	uuuu uuuu	Reserved	T
98	CSR98	uuuu uuuu	Reserved	T
99	CSR99	uuuu uuuu	Reserved	T
100	CSR100	uuuu 0200	MERRTO: Bus Time-Out	S
101	CSR101	uuuu uuuu	Reserved	T
102	CSR102	uuuu uuuu	Reserved	T
103	CSR103	uuuu 0105	Reserved	T
104	CSR104	uuuu uuuu	Reserved	T
105	CSR105	uuuu uuuu	Reserved	T
106	CSR106	uuuu uuuu	Reserved	T
107	CSR107	uuuu uuuu	Reserved	T

Control and Status Registers (continued)

RAP Addr	Symbol	Default Value After H_RESET	Comments	Use
108	CSR108	uuuu uuuu	Reserved	T
109	CSR109	uuuu uuuu	Reserved	T
110	CSR110	uuuu uuuu	Reserved	T
111	CSR111	uuuu uuuu	Reserved	T
112	CSR112	uuuu 0000	MFC: Missed Frame Count	R
113	CSR113	uuuu uuuu	Reserved	T
114	CSR114	uuuu 0000	RCC: Receive Collision Count	R
115	CSR115	uuuu uuuu	Reserved	T
116	CSR116	uuuu uuuu	Reserved	T
117	CSR117	uuuu uuuu	Reserved	T
118	CSR118	uuuu uuuu	Reserved	T
119	CSR119	uuuu uuuu	Reserved	T
120	CSR120	uuuu uuuu	Reserved	T
121	CSR121	uuuu uuuu	Reserved	T
122	CSR122	see reg. desc.	Receive Frame Alignment Control	S
123	CSR123	uuuu uuuu	Reserved	T
124	CSR124	see reg. desc.	Test Register 1	T
125	CSR125	uuuu uuuu	Reserved	T
126	CSR126	uuuu uuuu	Reserved	T
127	CSR127	uuuu uuuu	Reserved	T

BCR—Bus Configuration Registers

Writes to those registers marked as "Reserved" will have no effect. Reads from these locations will produce undefined values.

BCR	MNEMONIC	Default	Description	Programmability	
				User	EEPROM
0	MSRDA	0005h	Reserved	No	No
1	MSWRA	0005h	Reserved	No	No
2	MC	N/A*	Miscellaneous Configuration	Yes	Yes
3	Reserved	N/A	Reserved	No	No
4	LNKST	00C0h	Link Status (Default)	Yes	No
5	LED1	0084h	Receive Status (Default)	Yes	No
6	LED2	0088h	Reserved	Yes	No
7	LED3	0090h	Transmit Status (Default)	Yes	No
8–15	Reserved	N/A	Reserved	No	No
16	IOBASEL	N/A*	Reserved	Yes	Yes
17	IOBASEU	N/A*	Reserved	Yes	Yes
18	BSBC	2101h	Burst Size and Bus Control	Yes	Yes
19	EECAS	0002h	EEPROM Control and Status	Yes	No
20	SWS	0000h	Software Style	Yes	No
21	INTCON	N/A*	Reserved	Yes	Yes

** Registers marked with an "*" have no default value, since they are not observable without first being programmed through the EEPROM read operation. Therefore, the only observable values for these registers are those that have been programmed and a default value is not applicable.*

ABSOLUTE MAXIMUM RATINGS

Storage Temperature . . . . . −65°C to +150°C

Ambient Temperature
Under Bias . . . . . −65°C to +125°C

Supply Voltage to AV_{SS} or V_{SSB}
(AV_{DD}, V_{DD}, V_{DDB}) . . . . . −0.3 V to +6.0 V

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Temperature (T_A) . . . . . 0°C to +70°C

Supply Voltages
(AV_{DD}, V_{DD}, V_{DDB}) . . . . . +5 V ± 5%

All Inputs within the Range:
$AV_{SS} - 0.5\ V \le V_{IN} \le AV_{DD} + 0.5\ V$,
or $V_{SS} - 0.5\ V \le V_{IN} \le V_{DD} + 0.5\ V$,
or $V_{SSB} - 0.5\ V \le V_{IN} \le V_{DDB} + 0.5\ V$

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified

Parameter Symbol	Parameter Description	Test Conditions		Min	Max	Unit
Digital Input Voltage						
V_{IL}	Input LOW Voltage				0.8	V
V_{IH}	Input HIGH Voltage			2.0		V
Digital Output Voltage						
V_{OL}	Output LOW Voltage	I_{OL1} = 3 mA I_{OL2} = 6 mA			0.55	V
		I_{OL3} = 12 mA (Note 1)			0.4	V
V_{OH}	Output HIGH Voltage (Note 2)	I_{OH} = −2 mA (Note 5)		2.4		V
Digital Input Leakage Current						
I_{IL}	Input Low Leakage Current (Note 3)	V_{IN} = 0			10	µA
I_{IH}	Input High Leakage Current (Note 3)	$V_{IN} = V_{DD}, V_{DDB}$			−10	µA
Digital Output Leakage Current						
I_{OZL}	Output Low Leakage Current (Note 4)	V_{OUT} = 0.4 V			−10	µA
I_{OZH}	Output High Leakage Current (Note 4)	$V_{OUT} = V_{DD}, V_{DDB}$			10	µA
Crystal Input Current						
V_{ILX}	XTAL1 Input LOW Voltage Threshold	V_{IN} = External Clock		−0.5	0.8	V
V_{IHX}	XTAL1 Input HIGH Voltage Threshold	V_{IN} = External Clock		V_{DD} − 0.8	V_{DD} + 0.5	V
I_{ILX}	XTAL1 Input LOW Current	V_{IN} = External Clock	Active	−120	0	µA
		$V_{IN} = V_{SS}$	Sleep	−10	+10	µA
I_{IHX}	XTAL1 Input HIGH Current	V_{IN} = External Clock	Active	0	120	µA
		$V_{IN} = V_{DD}$	Sleep		400	µA
Attachment Unit Interface (AUI)						
I_{IAXD}	Input Current at DI+ and DI−	$-1\ V < V_{IN} < AV_{DD}+0.5V$		−500	+500	µA
I_{IAXC}	Input Current at CI+ and CI−	$-1\ V < V_{IN} < AV_{DD}+0.5V$		−500	+500	µA
V_{AOD}	Differential Output Voltage |(DO+)−(DO−)|	R_L = 78 Ω		630	1200	mV
V_{AODOFF}	Transmit Differential Output Idle Voltage	R_L = 78 Ω (Note 9)		−40	40	mV
I_{AODOFF}	Transmit Differential Output Idle Current	R_L = 78 Ω (Note 8)		−1	1	mA
V_{CMT}	Transmit Output Common Mode Voltage	R_L = 78 Ω		2.5	A_{VDD}	V
V_{ODI}	DO± Transmit Differential Output Voltage Imbalance	R_L = 78 Ω (Note 7)			25	mV
V_{ATH}	Receive Data Differential Input Threshold			−35	35	mV

DC CHARACTERISTICS (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
V_{ASQ}	DI± and CI± Differential Input Threshold (Squelch)		–275	–160	mV
V_{IRDVD}	DI± and CI± Differential Mode Input Voltage Range		–1.5	1.5	V
V_{ICM}	DI± and CI± Input Bias Voltage	I_{IN} = 0 mA	AV_{DD}–3.0	AV_{DD}–1.0	V
V_{OPD}	DO± Undershoot Voltage at ZERO Differential on Transmit Return to ZERO (ETD)	(Note 9)		–100	mV
Twisted Pair Interface (10BASE–T)					
I_{IRXD}	Input Current at RXD±	$AV_{DD} < V_{IN} < AV_{DD}$	–500	500	µA
R_{RXD}	RXD± Differential Input Resistance		10		KΩ
V_{TIVB}	RXD±, RXD– Open Circuit Input Voltage (Bias)	I_{IN} = 0 mA	AV_{DD}–3.0	AV_{DD}–1.5	V
V_{TIDV}	Differential Mode Input Voltage Range (RXD±)	AV_{DD} = 5.0 V	–3.1	3.1	V
V_{TSQ+}	RXD Positive Squelch Threshold (peak)	Sinusoid, 5 MHz ≤ f ≤ 10 MHz	300	520	mV
V_{TSQ-}	RXD Negative Squelch Threshold (peak)	Sinusoid, 5 MHz ≤ f ≤ 10 MHz	–520	–300	mV
V_{THS+}	RXD Post–squelch Positive Threshold (peak)	Sinusoid, 5 MHz ≤ f ≤ 10 MHz	150	293	mV
V_{THS-}	RXD Post-Squelch Negative Threshold (peak)	Sinusoid, 5 MHz ≤ f ≤ 10 MHz	–293	–150	mV
V_{LTSQ+}	RXD Positive Squelch Threshold (peak)	LRT = LOW	180	312	mV
V_{LTSQ-}	RXD Negative Squelch Threshold (peak)	LRT = LOW	–312	–180	mV
V_{LTHS+}	RXD Post-Squelch Positive Threshold (peak)	LRT = LOW	90	176	mV
V_{LTHS-}	RXD Post-Squelch Negative Threshold (peak)	LRT = LOW	–176	–90	mV
V_{RXDTH}	RXD Switching Threshold	(Note 4)	–35	35	mV
V_{TXH}	TXD± and TXP± Output HIGH Voltage	V_{SS} = 0 V	V_{DD}–0.6	V_{DD}	V
V_{TXL}	TXD± and TXP± Output LOW Voltage	V_{DD} = 5 V	V_{SS}	V_{SS}+0.6	V
V_{TXI}	TXD± and TXP± Differential Output Voltage Imbalance		–40	40	mV
V_{TX}OFF	TXD± and TXP± Idle Output Voltage			40	mV
R_{TX}	TXD± , TXP± Differential Driver Output Impedance	(Note 4)	40	80	Ω

DC CHARACTERISTICS (continued)

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Power Supply Current					
I_{DD}	Active Power Supply Current	XTAL1 = 20 MHz, CLK = 33 MHz		150	mA
I_{DDCOMA}	Sleep Mode Power Supply Current	$\overline{SLEEP}$ Active		TBD	µA
$I_{DDSNOOZE}$	Auto Wake Mode Power Supply Current	Awake Bit Set Active		TBD	µA
Pin Capacitance					
C_{IN}	Input Pin Capacitance	FC = 1 MHz (Notes 6 & 10)		10	pF
C_O	I/O or Output Pin Capacitance	FC = 1 MHz (Note 6)		10	pF
C_{CLK}	CLK Pin Capacitance	FC = 1 MHz (Note 6)		12	pF

Notes:

1. *I_{OL1} applies to AD[31:00], C/$\overline{BE}$[3:0], $\overline{REQ}$, and PAR*
   I_{OL2} applies to $\overline{FRAME}$, $\overline{IRDY}$, $\overline{TRDY}$, $\overline{DEVSEL}$, $\overline{STOP}$, $\overline{SERR}$, $\overline{PERR}$, and $\overline{LOCK}$
   I_{OL3} applies to EESK/$\overline{LED1}$, EEDO/$\overline{LED3}$, and EEDI/$\overline{LNKST}$
2. *V_{OH} does not apply to open–drain output pins.*
3. *I_{IL} and I_{IH} apply to all input pins except XTAL1.*
4. *I_{OZL} and I_{OZH} apply to all three-state output pins and bi-directional pins.*
5. *Outputs are CMOS and will be driven to rail if the load is not resistive.*
6. *Parameter not tested. Value determined by characterization.*
7. *Tested, but to values in excess of limits. Test accuracy not sufficient to allow screening guard bands.*
8. *Correlated to other tested parameters – not tested directly.*
9. *Test not implemented to data sheet specification.*
10. *C_{IN} = 8 pF for IDSEL input.*

SWITCHING CHARACTERISTICS: Bus Interface

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Clock Timing					
	CLK Frequency		0	33	MHz
t_{CYC}	CLK Period	@ 1.5 V	30	∞	ns
t_{HIGH}	CLK High Time	@ 2.0 V	12		ns
t_{LOW}	CLK Low Time	@ 0.8 V	12		ns
t_{FALL}	CLK Fall Time	over 2 V p-p	1	4	V/ns
t_{RISE}	CLK Rise Time	over 2 V p-p	1	4	V/ns
Output and Float Delay Timing					
t_{VAL}	AD[31:00], C/$\overline{BE}$[3:0], PAR, $\overline{FRAME}$, $\overline{IRDY}$, $\overline{TRDY}$, $\overline{STOP}$, $\overline{LOCK}$, $\overline{DEVSEL}$, $\overline{PERR}$, $\overline{SERR}$ Valid Delay		2	11	ns
t_{VAL} ($\overline{REQ}$)	$\overline{REQ}$ Valid Delay		1	12	ns
f_{EESK}	EESK Frequency	(See note below)		650	KHz
t_{VAL} (EEDI)	EEDI Valid Output Delay from EESK	(See note below)	100	400	ns
t_{VAL} (EESK)	EECS Valid Output Delay from EESK	(See note below)	0	400	ns
t_{ON}	AD[31:00], C/$\overline{BE}$[3:0], PAR, $\overline{FRAME}$, $\overline{IRDY}$, $\overline{TRDY}$, $\overline{STOP}$, $\overline{LOCK}$, $\overline{DEVSEL}$ Active Delay		2	11	ns
t_{OFF}	AD[31:00], C/$\overline{BE}$[3:0], PAR, $\overline{FRAME}$, $\overline{IRDY}$, $\overline{TRDY}$, $\overline{STOP}$, $\overline{LOCK}$, $\overline{DEVSEL}$ Float Delay			28	ns
Setup and Hold Timing					
t_{SU}	AD[31:00], C/$\overline{BE}$[3:0], PAR, $\overline{FRAME}$, $\overline{IRDY}$, $\overline{TRDY}$, $\overline{STOP}$, $\overline{LOCK}$, $\overline{DEVSEL}$, IDSEL Setup Time		7		ns
t_H	AD[31:00], C/$\overline{BE}$[3:0], PAR, $\overline{FRAME}$, $\overline{IRDY}$, $\overline{TRDY}$, $\overline{STOP}$, $\overline{LOCK}$, $\overline{DEVSEL}$, IDSEL Hold Time		0		ns
t_{SU} ($\overline{GNT}$)	$\overline{GNT}$ Setup Time		10		ns
t_H ($\overline{GNT}$)	$\overline{GNT}$ Hold Time		0		ns
t_{SU} (EEDO)	EEDO Setup Time to EESK	(See note below)	50		ns
t_H (EEDO)	EEDO Hold Time from EESK	(See note below)	0		ns

Note:

Parameter value is given for automatic EEPROM read operation. When EEPROM port (BCR19) is used to access the EEPROM, software is responsible for meeting EEPROM timing requirements.

SWITCHING CHARACTERISTICS: 10BASE-T Interface

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
Transmit Timing					
t_{TETD}	Transmit Start of Idle		250	350	ns
t_{TR}	Transmitter rise time	(10% to 90%)		5.5	ns
t_{TF}	Transmitter fall time	(90% to 10%)		5.5	ns
t_{TM}	Transmitter rise and fall time mismatch	($t_{TM} = \lvert t_{TR} - t_{TF} \rvert$)		1	ns
t_{PERLP}	Idle Signal Period		8	24	ms
t_{PWLP}	Idle Link Pulse Width	(See note below)	75	120	ns
t_{PWPLP}	Predistortion Idle Link Pulse Width	(See note below)	45	55	ns
t_{JA}	Transmit jabber activation time		20	150	ms
t_{JR}	Transmit jabber reset time		250	750	ms
t_{JREC}	Transmit jabber recovery time (minimum time gap between transmitted frames to prevent jabber activation)		1.0		µs
Receiving Timing					
t_{PWNRD}	RXD pulse width not to turn off internal carrier sense	$V_{IN} > V_{THS}$ (min)	136		ns
t_{PWROFF}	RXD pulse width to turn off	$V_{IN} > V_{THS}$ (min)		200	ns
t_{RETD}	Receive Start of Idle		200		ns

Note:

Not tested; parameter guaranteed by characterization.

SWITCHING CHARACTERISTICS: Attachment Unit Interface

Parameter Symbol	Parameter Description	Test Conditions	Min	Max	Unit
AUI Port					
t_{DOTR}	DO+, DO– Rise Time (10% to 90%)		2.5	5.0	ns
t_{DOTF}	DO+, DO– Fall Time (10% to 90%)		2.5	5.0	ns
t_{DORM}	DO+, DO– Rise and Fall Time Mismatch			1.0	ns
t_{DOETD}	DO± End of Transmission		200	375	ns
t_{PWODI}	DI Pulse Width Accept/Reject Threshold	$\|V_{IN}\| > \|VASQ\|$ (Note 1)	15	45	ns
t_{PWKDI}	DI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|VASQ\|$ (Note 2)	136	200	ns
t_{PWOCI}	CI Pulse Width Accepth/Reject Threshold	$\|V_{IN}\| > \|VASQ\|$ (Note 3)	10	26	ns
t_{PWKCI}	CI Pulse Width Maintain/Turn-Off Threshold	$\|V_{IN}\| > \|VASQ\|$ (Note 4)	90	160	ns
Internal MENDEC Clock Timing					
tx1	XTAL1 Period	V_{IN} = External Clock	49.995	50.001	ns
tx1H	XTAL1 HIGH Pulse Width	V_{IN} = External Clock	20		ns
tx1L	XTAL1 LOW Pulse Width	V_{IN} = External Clock	20		ns
tx1R	XTAL1 Rise Time	V_{IN} = External Clock		5	ns
tx1F	XTAL1 Fall Time	V_{IN} = External Clock		5	ns

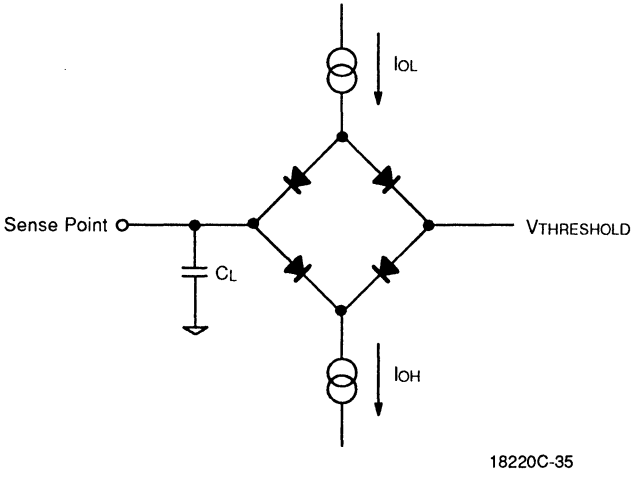
Notes:

1. *DI pulses narrower than t_{PWODI} (min) will be rejected; pulses wider than t_{PWODI} (max) will turn internal DI carrier sense on.*
2. *DI pulses narrower than t_{PWKDI} (min) will maintain internal DI carrier sense on; pulses wider than t_{PWKDI} (max) will turn internal DI carrier sense off.*
3. *CI pulses narrower than t_{PWOCI} (min) will be rejected; pulses wider than t_{PWOCI} (max) will turn internal CI carrier sense on.*
4. *CI pulses narrower than t_{PWKCI} (min) will maintain internal CI carrier sense on; pulses wider than t_{PWKCI} (max) will turn internal CI carrier sense off.*

KEY TO SWITCHING WAVEFORMS

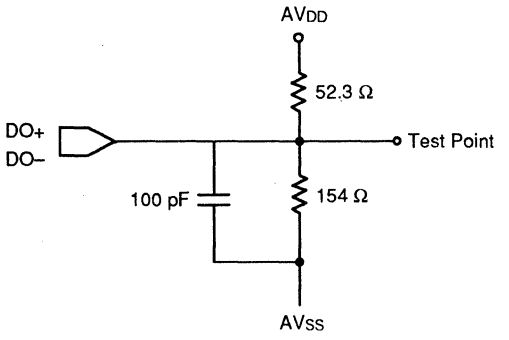
WAVEFORM	INPUTS	OUTPUTS
	Must be Steady	Will be Steady
	May Change from H to L	Will be Changing from H to L
	May Change from L to H	Will be Changing from L to H
	Don't Care, Any Change Permitted	Changing, State Unknown
	Does Not Apply	Center Line is High-Impedance "Off" State

SWITCHING TEST CIRCUITS

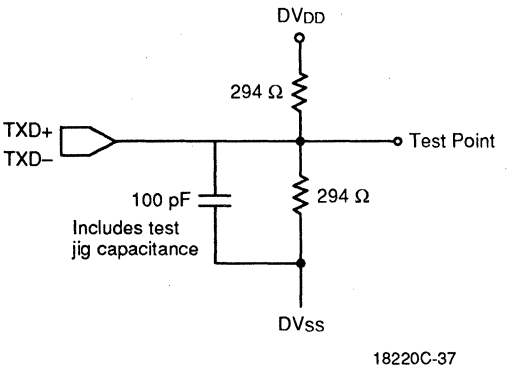


Normal and Three-State Outputs

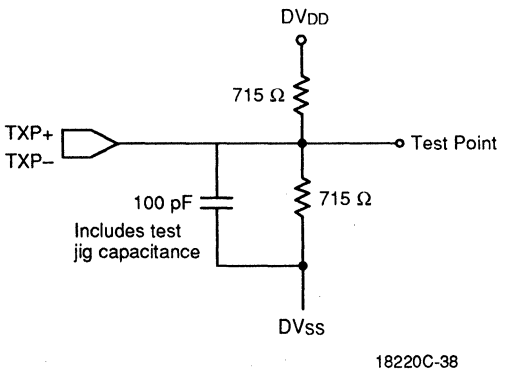
SWITCHING TEST CIRCUITS



AUI DO Switching Test Circuit

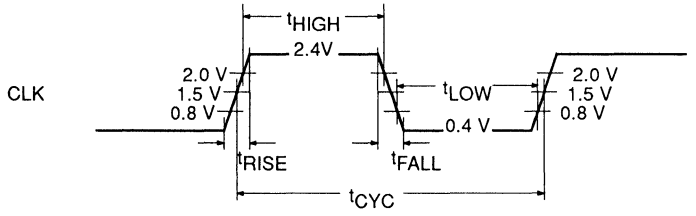


TXD Switching Test Circuit

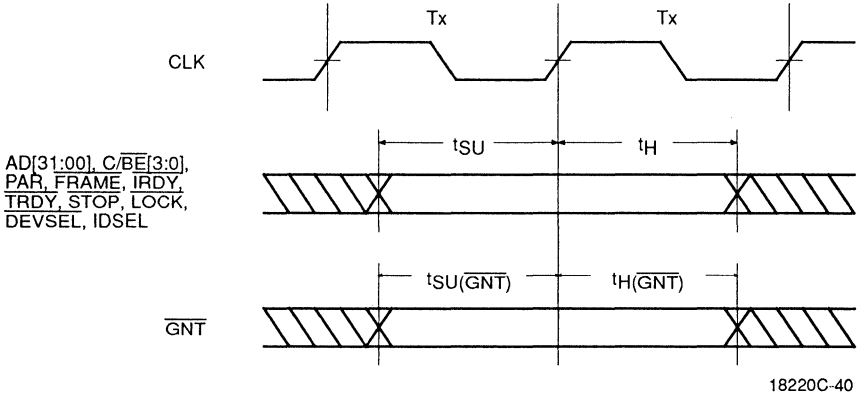


TXP Outputs Test Circuit

SWITCHING WAVEFORMS: System Bus Interface

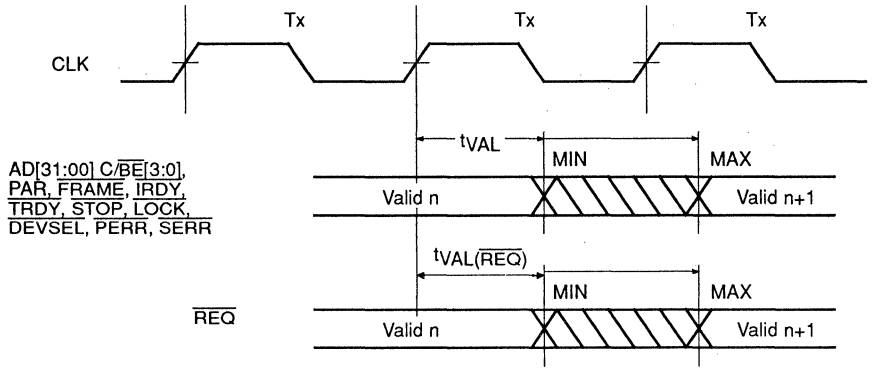


CLK Waveform



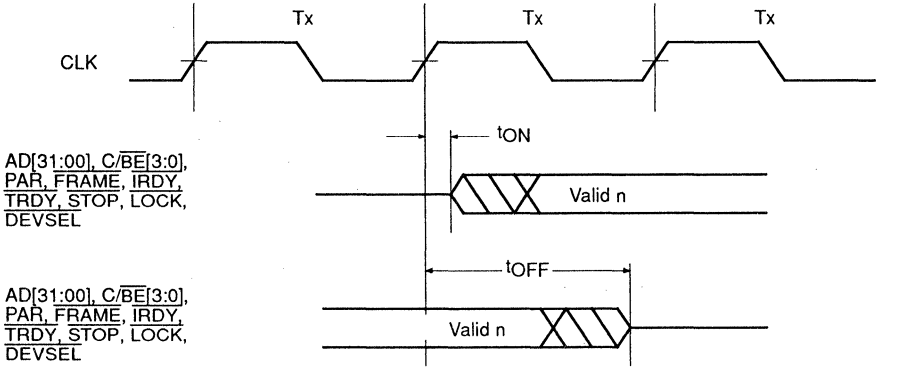
Input Setup and Hold Timing

SWITCHING WAVEFORMS: System Bus Interface



18220C-41

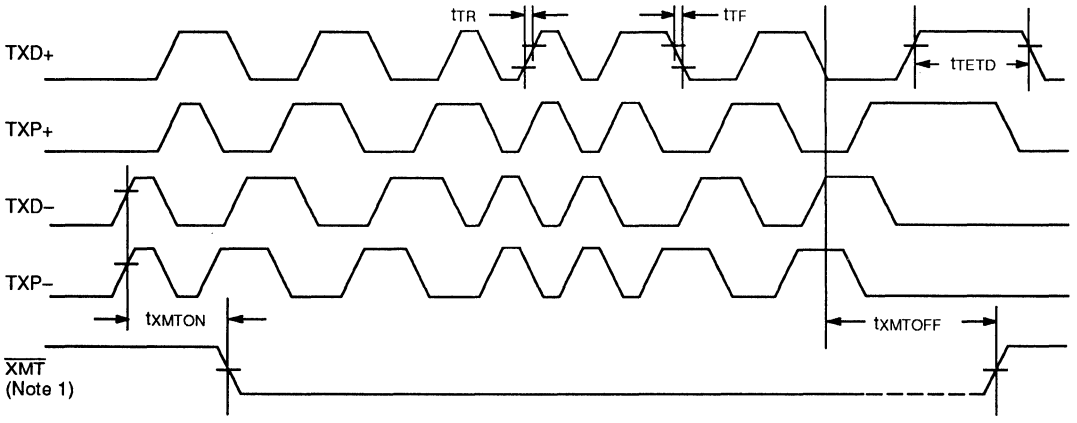
Output Valid Delay Timing



18220C-42

Output Tri–State Delay Timing

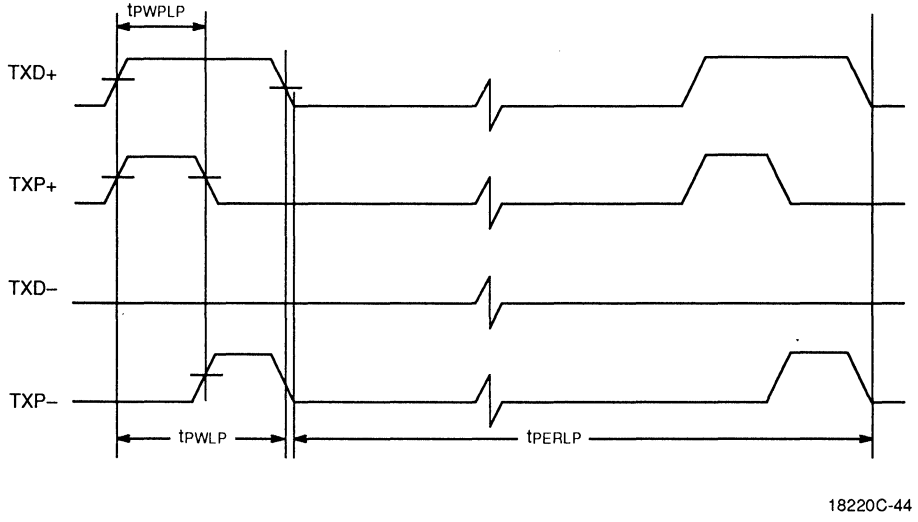
SWITCHING WAVEFORMS: 10BASE-T Interface



Note:

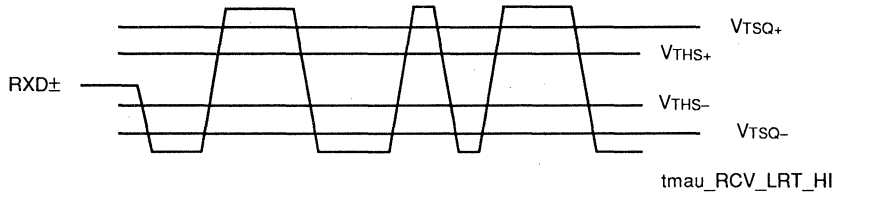
1. Internal signal and is shown for clarification only.

Transmit Timing

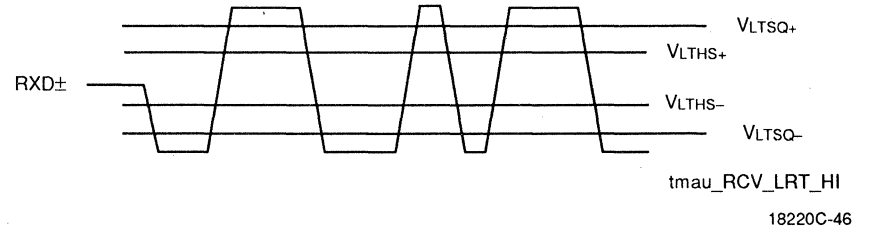


Idle Link Test Pulse

SWITCHING WAVEFORMS: 10BASE-T Interface

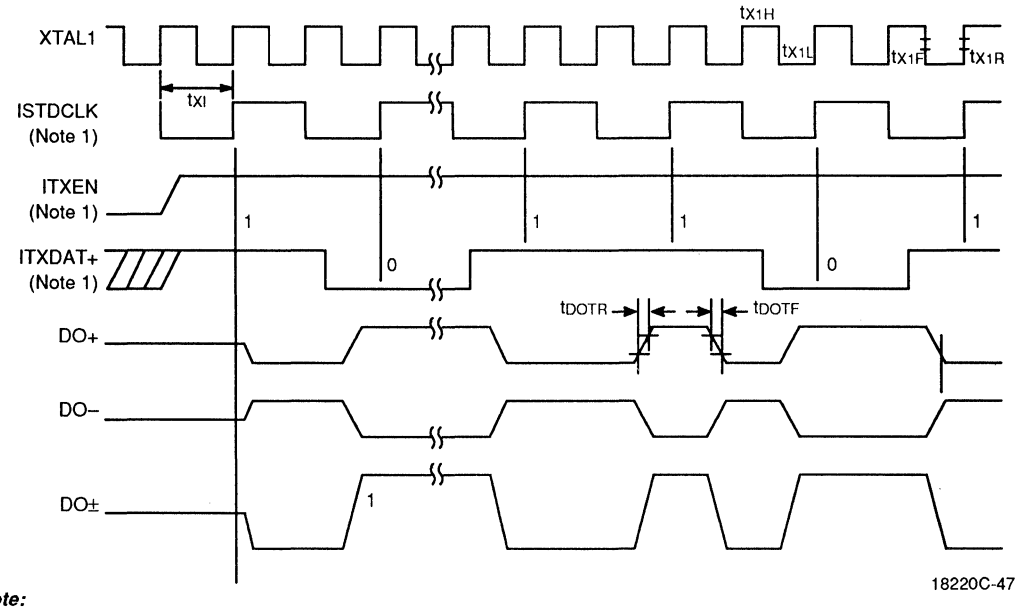


Receive Thresholds (LRT=0)



Receive Thresholds (LRT=1)

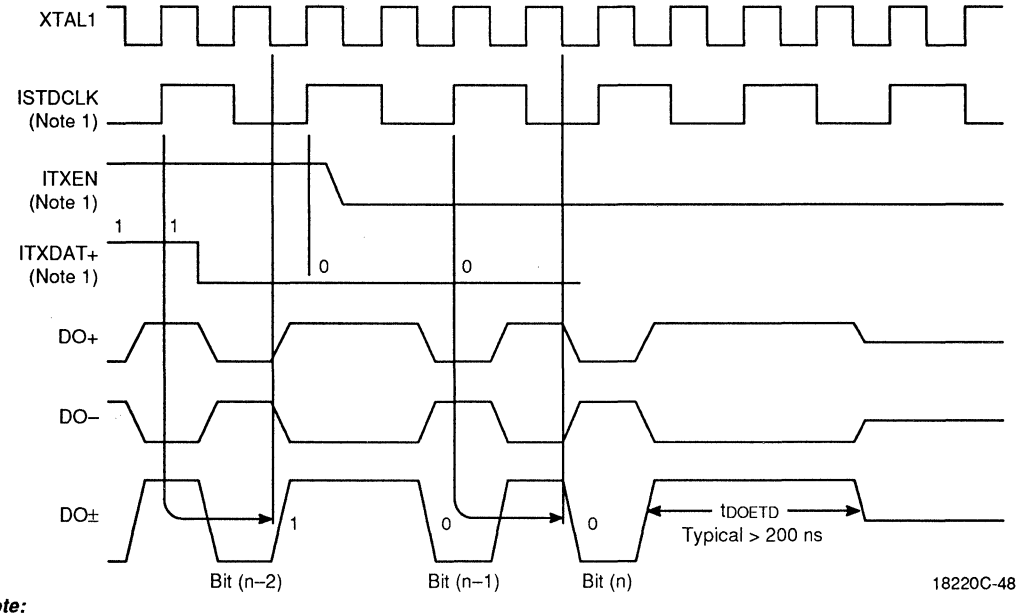
SWITCHING WAVEFORMS: Attachment Unit Interface



Note:

1. Internal signal and is shown for clarification only.

Transmit Timing – Start of Frame

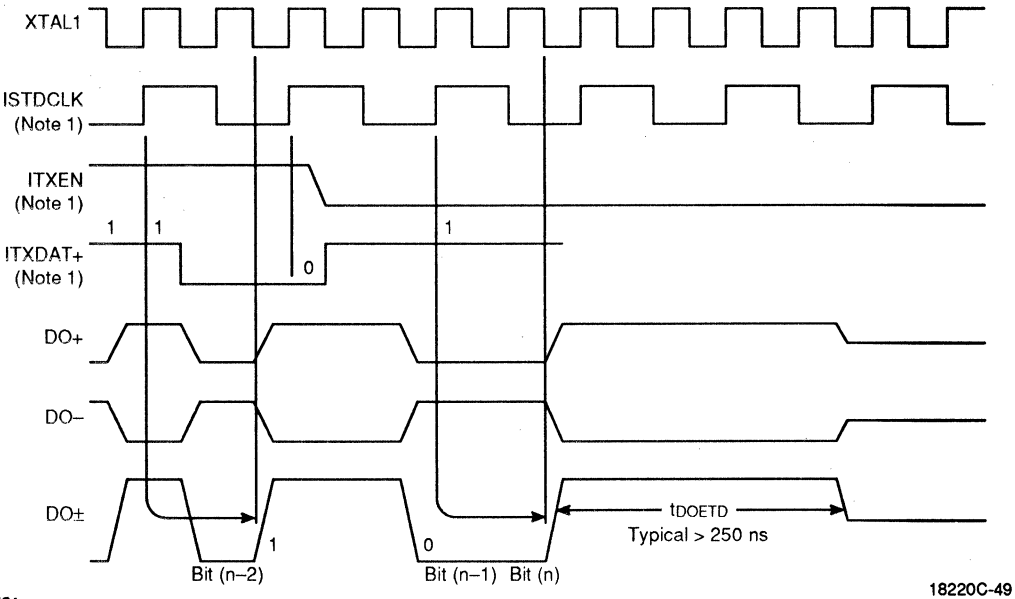


Note:

1. Internal signal and is shown for clarification only.

Transmit Timing – End of Frame (Last Bit = 0)

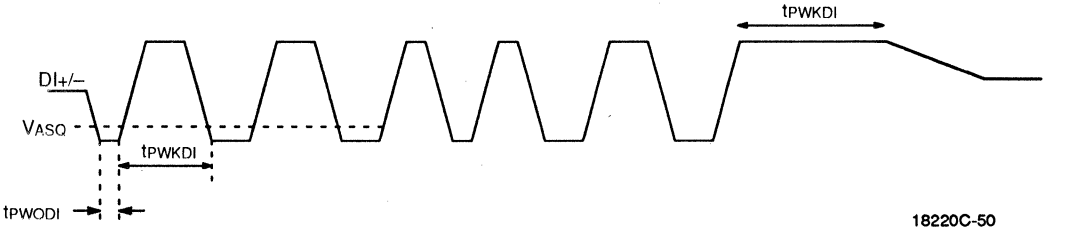
SWITCHING WAVEFORMS: Attachment Unit Interface



Note:

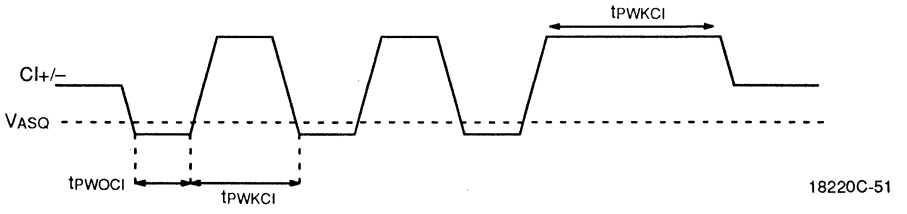
1. Internal signal and is shown for clarification only.

Transmit Timing – End of Frame (Last Bit = 1)

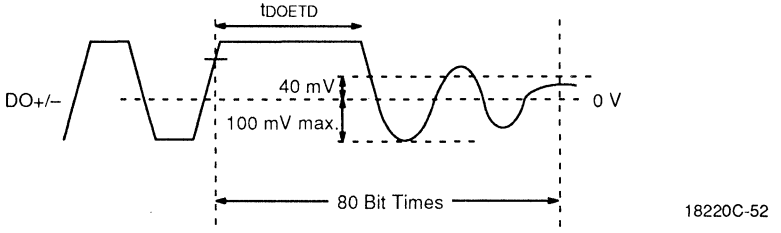


Receive Timing

SWITCHING WAVEFORMS: Attachment Unit Interface



Collision Timing



Port DO ETD Waveform

APPENDIX A

PCnet-PCI Compatible Media Interface Modules

PCnet-PCI Compatible 10BASE-T Filters and Transformers

The table below provides a sample list of PCnet-PCI compatible 10BASE-T filter and transformer modules available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part #	Package	Filters and Transformers	Filters Transformers and Choke	Filters Transformers Dual Chokes	Filters Transformers Resistors Dual Chokes
Bel Fuse	A556-2006-DE	16-pin 0.3 DIL	√			
Bel Fuse	0556-2006-00	14-pin SIP	√			
Bel Fuse	0556-2006-01	14-pin SIP			√	
Bel Fuse	0556-6392-00	16-pin 0.5 DIL			√	
Halo Electronics	FD02-101G	16-pin 0.3 DIL	√			
Halo Electronics	FD12-101G	16-pin 0.3 DIL		√		
Halo Electronics	FD22-101G	16-pin 0.3 DIL			√	
PCA Electronics	EPA1990A	16-pin 0.3 DIL	√			
PCA Electronics	EPA2013D	16-pin 0.3 DIL		√		
PCA Electronics	EPA2162	16-pin 0.3 SIP			√	
Pulse Engineering	PE-65421	16-pin 0.3 DIL	√			
Pulse Engineering	PE-65434	16-pin 0.3 SIL			√	
Pulse Engineering	PE-65445	16-pin 0.3 DIL			√	
Pulse Engineering	PE-65467	12-pin 0.5 SMT				√
Valor Electronics	PT3877	16-pin 0.3 DIL	√			
Valor Electronics	FL1043	16-pin 0.3 DIL			√	

PCnet-PCI Compatible AUI Isolation Transformers

The table below provides a sample list of PCnet-PCI compatible AUI isolation transformers available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part #	Package	Description
Bel Fuse	A553-0506-AB	16-pin 0.3 DIL	50 µH
Bel Fuse	S553-0756-AE	16-pin 0.3 SMD	75 µH
Halo Electronics	TD01-0756K	16-pin 0.3 DIL	75 µH
Halo Electronics	TG01-0756W	16-pin 0.3 SMD	75 µH
PCA Electronics	EP9531-4	16-pin 0.3 DIL	50 µH
Pulse Engineering	PE64106	16-pin 0.3 DIL	50 µH
Pulse Engineering	PE65723	16-pin 0.3 SMT	75 µH
Valor Electronics	LT6032	16-pin 0.3 DIL	75 µH
Valor Electronics	ST7032	16-pin 0.3 SMD	75 µH

PCnet-PCI Compatible DC/DC Converters

The table below provides a sample list of PCnet-PCI compatible DC/DC converters available from various vendors. Contact the respective manufacturer for a complete and updated listing of components.

Manufacturer	Part #	Package	Voltage	Remote On/Off
Halo Electronics	DCU0-0509D	24-pin DIP	5/-9	No
Halo Electronics	DCU0-0509E	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1007P	24-pin DIP	5/-9	No
PCA Electronics	EPC1054P	24-pin DIP	5/-9	Yes
PCA Electronics	EPC1078	24-pin DIP	5/-9	Yes
Valor Electronics	PM7202	24-pin DIP	5/-9	No
Valor Electronics	PM7222	24-pin DIP	5/-9	Yes

MANUFACTURER CONTACT INFORMATION

Contact the following companies for further information on their products.

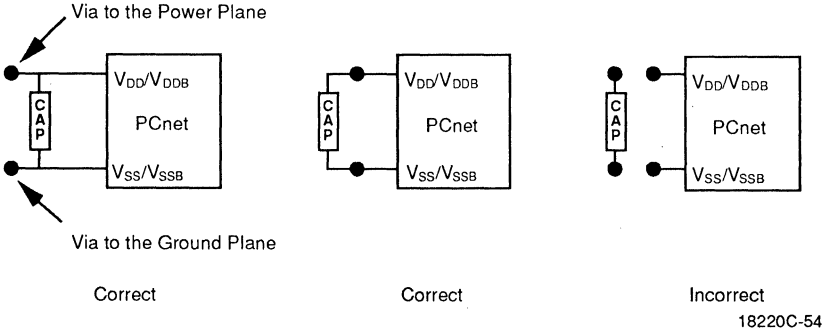
Company	U.S. and Domestic		Asia	Europe
Bel Fuse	Phone:	(201) 432-0463	852-328-5515	33-1-69410402
	FAX:	(201) 432-9542	852-352-3706	33-1-69413320
Halo Electronics	Phone:	(415) 969-7313	65-285-1566	
	FAX:	(415) 367-7158	65-284-9466	
PCA Electronics	Phone:	(818) 892-0761	852-553-0165	33-1-44894800
(HPC in Hong Kong)	FAX:	(818) 894-5791	852-873-1550	33-1-42051579
Pulse Engineering	Phone:	(619) 674-8100	852-425-1651	353-093-24107
	FAX:	(619) 675-8262	852-480-5974	353-093-24459
Valor Electronics	Phone:	(619) 537-2500	852-513-8210	49-89-6923122
	FAX:	(619) 537-2525	852-513-8214	49-89-6926542

APPENDIX B

Recommendation for Power and Ground Decoupling

The mixed analog/digital circuitry in the PCnet-PCI make it imperative to provide noise-free power and ground connections to the device. Without clean power and ground connections, a design may suffer from high bit error rates or may not function at all. Hence, it is highly recommended that the guidelines presented here are followed to ensure a reliable design.

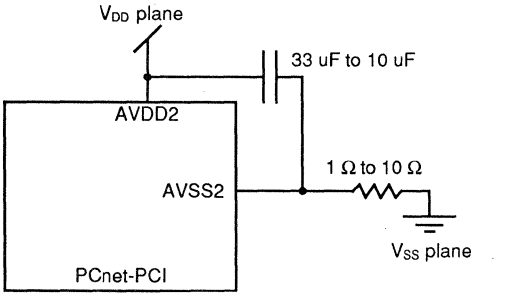
Decoupling/Bypass Capacitors: Adequate decoupling of the power and ground pins and planes is required by all PCnet-PCI designs. This includes both low-frequency bulk capacitors and high frequency capacitors. It is recommended that **at least one** low-frequency bulk (e.g. 22 µF) decoupling capacitor be used in the area of the PCnet-PCI device. The bulk capacitor(s) should be connected directly to the power and ground planes. In addition, **at least 8** high frequency decoupling capacitors (e.g. 0.1 µF multilayer ceramic capacitors) should be used around the periphery of the PCnet-PCI device to prevent power and ground bounce from affecting device operation. To reduce the inductance between the power and ground pins and the capacitors, the pins should be connected directly to the capacitors, rather than through the planes to the capacitors. The suggested connection scheme for the capacitors is shown in the figure below. Note also that the traces connecting these pins to the capacitors should be as wide as possible to reduce inductance (15 mils is desirable).



The most critical pins in the layout of a PCnet-PCI design are the 4 analog power and 2 analog ground pins, AVDD[1–4] and AVSS[1–2], respectively. All of these pins are located in one corner of the device, the "analog corner." Specific functions and layout requirements of the analog power and ground pins are given below.

AVSS1 and AVDD3: These pins provide the power and ground for the Twisted Pair and AUI drivers. In addition AVSS1 serves as the ground for the logic interfaces in the 20 MHz Crystal Oscillator. Hence, these pins can be very noisy. A dedicated 0.1 µF capacitor between these pins is recommended.

AVSS2 and AVDD2: These pins are the **most critical** pins on the PCnet-PCI device because they provide the power and ground for the phase-lock loop (PLL) portion of the chip. The voltage-controlled oscillator (VCO) portion of the PLL is sensitive to noise in the 60 kHz – 200 kHz. range. To prevent noise in this frequency range from disrupting the VCO, it is **strongly recommended** that the low-pass filter shown below be implemented on these pins.



18220C-53

To determine the value for the resistor and capacitor, the formula is:

$$R * C \geq 88$$

Where R is in Ohms and C is in microfarads. Some possible combinations are given below. To minimize the voltage drop across the resistor, the R value should not be more than 10 Ω.

R	C
2.7 Ω	33 µF
4.3 Ω	22 µF
6.8 Ω	15 µF
10 Ω	10 µF

AVSS2 and AVDD2/AVDD4: These pins provide power and ground for the AUI and twisted pair receive circuitry. In addition, as mentioned earlier, AVSS2 and AVDD2 provide power and ground for the phase-lock loop portion of the chip. Except for the filter circuit already mentioned, no specific decoupling is necessary on these pins.

AVDD1: AVDD1 provides power for the control and interface logic in the PLL. Ground for this logic is provided by digital ground pins. No specific decoupling is necessary on this pin.

Special Note for Adapter Cards: In adapter card designs, **it is important to utilize all available power and ground pins available on the bus edge connector**. In addition, the connection from the bus edge connector to the power or ground plane should be made through more than one via and with wide traces (15 mils desirable) wherever possible. Following these recommendations results in minimal inductance in the power and ground paths. By minimizing this inductance, ground bounce is minimized.

APPENDIX C

Alternative Method for Initialization

The PCnet-PCI controller may be initialized by performing I/O writes only. That is, data can be written directly to the appropriate control and status registers (CSR instead of reading from the initialization Block in memory).

The registers that must be written are shown in the table below. These register writes are followed by writing the START bit in CSR0.

Control and Status Register	Comment
CSR8	LADRF[15:0]
CSR9	LADRF[31:16]
CSR10	LADRF[47:32]
CSR11	LADRF[63:48]
CSR12	PADR[15:0]
CSR13	PADR[31:16]
CSR14	PADR[47:32]
CSR15	Mode
CSR24-25	BADR
CSR30-31	BADX
CSR47	POLLINT
CSR76	RCVRL
CSR78	XMTRL

Note:

The INIT bit must not be set or the initialization block will be accessed instead.

APPENDIX D

Look-Ahead Packet Processing (LAPP) Concept

Introduction of the LAPP Concept

A driver for the PCnet-PCI controller would normally require that the CPU copy receive frame data from the controllers buffer space to the applications buffer space after the entire frame has been received by the controller. For applications that use a ping-pong windowing style, the traffic on the network will be halted until the current frame has been completely processed by the entire application stack. This means that the time between last byte of a receive frame arriving at the clients Ethernet controller and the clients transmission of the first byte of the next outgoing frame will be separated by:

1. The time that it takes the clients CPUs interrupt procedure to pass software control from the current task to the driver
2. plus the time that it takes the client driver to pass the header data to the application and request an application buffer
3. plus the time that it takes the application to generate the buffer pointer and then return the buffer pointer to the driver
4. plus the time that it takes the client driver to transfer all of the frame data from the controllers buffer space into the applications buffer space and then call the application again to process the complete frame
5. plus the time that it takes the application to process the frame and generate the next outgoing frame
6. plus the time that it takes the client driver to set up the descriptor for the controller and then write a TDMD bit to CSR0

The sum of these times can often be about the same as the time taken to actually transmit the frames on the wire, thereby yielding a network utilization rate of less than 50%.

An important thing to note is that the PCnet-PCI controllers data transfers to its buffer space are such that the system bus is needed by the PCnet-PCI controller for approximately 4% of the time. This leaves 96% of the system bus bandwidth for the CPU to perform some of the inter-frame operations in advance of the completion of network receive activity, if possible. The question then becomes: how much of the tasks that need to be performed between reception of a frame and transmission of the next frame can be performed before the reception of the frame actually ends at the network, and how can the CPU be instructed to perform these tasks during the network reception time?

The answer depends upon exactly what is happening in the driver and application code, but the steps that can be performed at the same time as the receive data are arriving include as much as the first 3 steps and part of the 4th step shown in the sequence above. By performing these steps before the entire frame has arrived, the frame throughput can be substantially increased.

A good increase in performance can be expected when the first 3 steps are performed before the end of the network receive operation. A much more significant performance increase could be realized if the PCnet-PCI controller could place the frame data directly into the applications buffer space; (i.e., eliminate the need for step 4.) In order to make this work, it is necessary that the application buffer pointer be determined before the frame has completely arrived, then the buffer pointer in the next descriptor for the receive frame would need to be modified in order to direct the PCnet-PCI controller to write directly to the application buffer. More details on this operation will be given later.

An alternative modification to the existing system can gain a smaller, but still significant improvement in performance. This alternative leaves step 4 unchanged in that the CPU is still required to perform the copy operation, but it allows a large portion of the copy operation to be done before the frame has been completely received by the controller; i.e., the CPU can perform the copy operation of the receive data from the PCnet-PCI controllers buffer space into the application buffer space before the frame data has completely arrived from the network. This allows the copy operation of step 4 to be performed concurrently with the arrival of network data, rather than sequentially, following the end of network receive activity.

Outline of the LAPP Flow

This section gives a suggested outline for a driver that utilizes the LAPP feature of the PCnet-PCI controller.

Note: *The labels in the following text are used as references in the timeline diagram that follows.*

SETUP:

The driver should set up descriptors in groups of 3, with the OWN and STP bits of each set of three descriptors to read as follows: 11b, 10b, 00b.

An option bit (LAPPEN) exists in CSR3, bit position 5; the software should set this bit; When set, the LAPPEN bit directs the PCnet-PCI controller to generate an INTERRUPT when STP has been written to a receive descriptor by the PCnet-PCI controller.

FLOW:

The PCnet-PCI controller polls the current receive descriptor at some point in time before a message arrives. The PCnet-PCI controller determines that this receive buffer is OWNed by the PCnet-PCI controller and it stores the descriptor information to be used when a message does arrive.

N0: Frame preamble appears on the wire, followed by SFD and destination address.

N1: The 64th byte of frame data arrives from the wire. This causes the PCnet-PCI controller to begin frame data DMA operations to the first buffer.

C0: When the 64th byte of the message arrives, the PCnet-PCI controller performs a lookahead operation to the next receive descriptor. This descriptor should be owned by the PCnet-PCI controller .

C1: The PCnet-PCI controller intermittently requests the bus to transfer frame data to the first buffer as it arrives on the wire.

S1: The driver remains idle.

C2: When the PCnet-PCI controller has completely filled the first buffer, it writes status to the first descriptor.

C3: When the first descriptor for the frame has been written, changing ownership from the PCnet-PCI controller to the CPU, the PCnet-PCI controller will generate an SRP INTERRUPT. (This interrupt appears as a RINT interrupt in CSR0.)

S1: The SRP INTERRUPT causes the CPU to switch tasks to allow the PCnet-PCI controllers driver to run.

C4: During the CPU interrupt-generated task switching, the PCnet-PCI controller is performing a lookahead operation to the third descriptor. At this point in time, the third descriptor is owned by the CPU.

Note: *Even though the third buffer is not owned by the PCnet-PCI controller, existing AMD Ethernet controllers will continue to perform data DMA into the buffer space that the controller already owns (i.e., buffer number 2). The controller does not know if buffer space in buffer number 2 will be sufficient or not, for this frame, but it has no way to tell except by trying to move the entire message into that space. Only when the message does not fit will it signal a buffer error condition – there is no need to panic at the point that it discovers that it does not yet own descriptor number 3.*

S2: The first task of the drivers interrupt service routine is to collect the header information from the PCnet-PCI controllers first buffer and pass it to the application.

S3: The application will return an application buffer pointer to the driver. The driver will add an offset to the application data buffer pointer, since the PCnet-PCI controller will be placing the first portion of the message into the first and second buffers. (The modified application data buffer pointer will only be directly used by the PCnet-PCI controller when it reaches the third buffer.) The driver will place the modified data buffer pointer into the final descriptor of the group (#3) and will grant ownership of this descriptor to the PCnet-PCI controller.

C5: Interleaved with S2, S3 and S4 driver activity, the PCnet-PCI controller will write frame data to buffer number 2.

S4: The driver will next proceed to copy the contents of the PCnet-PCI controllers first buffer to the beginning of the application space. This copy will be to the exact (unmodified) buffer pointer that was passed by the application.

S5: After copying all of the data from the first buffer into the beginning of the application data buffer, the driver will begin to poll the ownership bit of the second descriptor. The driver is waiting for the PCnet-PCI controller to finish filling the second buffer.

C6: At this point, knowing that it had not previously owned the third descriptor, and knowing that the current message has not ended (there is more data in the FIFO), the PCnet-PCI controller will make a last ditch lookahead to the final (third) descriptor. This time, the ownership will be TRUE (i.e. the descriptor belongs to the controller), because the driver wrote the application pointer into this descriptor and then changed the ownership to give the descriptor to the PCnet-PCI controller back at S3. Note that if steps S1, S2 and S3 have not completed at this time, a BUFF error will result.

C7: After filling the second buffer and performing the last chance lookahead to the next descriptor, the PCnet-PCI controller will write the status and change the ownership bit of descriptor number 2.

S6: After the ownership of descriptor number 2 has been changed by the PCnet-PCI controller, the next driver poll of the 2nd descriptor will show ownership granted to the CPU. The driver now copies the data from buffer number 2 into the middle section of the application buffer space. This operation is interleaved with the C7 and C8 operations.

C8: The PCnet-PCI controller will perform data DMA to the last buffer, whose pointer is pointing to application space. Data entering the last buffer will not need the infamous double copy that is required by existing drivers, since it is being placed directly into the application buffer space.

N2: The message on the wire ends.

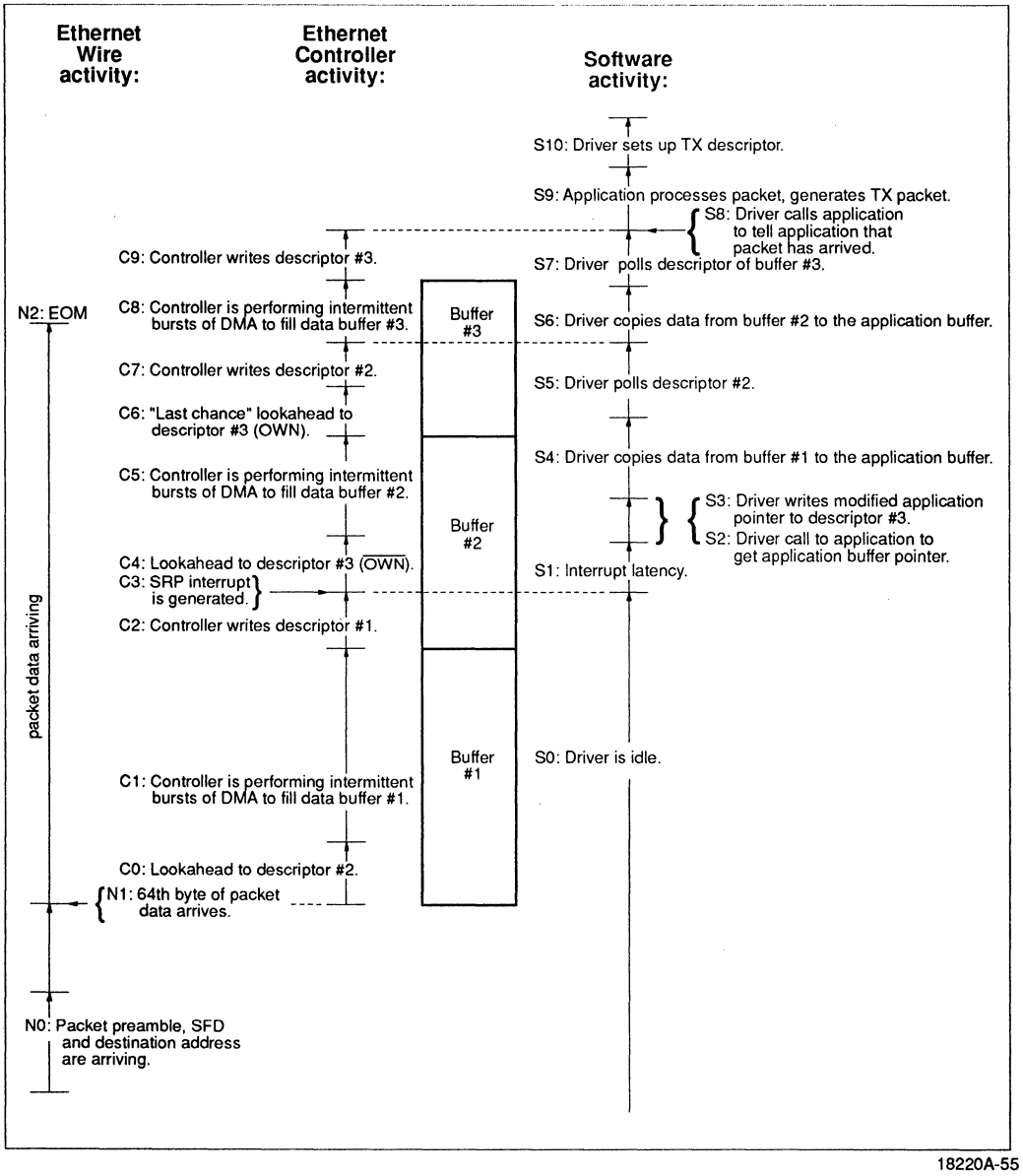
S7: When the driver completes the copy of buffer number 2 data to the application buffer space, it begins polling descriptor number 3.

C9: When the PCnet-PCI controller has finished all data DMA operations, it writes status and changes ownership of descriptor number 3.

S8: The driver sees that the ownership of descriptor number 3 has changed, and it calls the application to tell the application that a frame has arrived.

S9: The application processes the received frame and generates the next TX frame, placing it into a TX buffer.

S10: The driver sets up the TX descriptor for the PCnet-PCI controller.



18220A-55

Figure D1. LAPP Timeline

LAPP Software Requirements:

Software needs to set up a receive ring with descriptors formed into groups of 3. The first descriptor of each group should have OWN = 1 and STP = 1, the second descriptor of each group should have OWN = 1 and STP = 0. The third descriptor of each group should have OWN = 0 and STP = 0. The size of the first buffer (as indicated in the first descriptor), should be at least equal to the largest expected header size; however, for maximum efficiency of CPU utilization, the first buffer size should be larger than the header size. It should be equal to the expected number of message bytes, minus the time needed for Interrupt latency and minus the applica-

tion call latency, minus the time needed for the driver to write to the third descriptor, minus the time needed for the driver to copy data from buffer #1 to the application buffer space, and minus the time needed for the driver to copy data from buffer #2 to the application buffer space. Note that the time needed for the copies performed by the driver depends upon the sizes of the 2nd and 3rd buffers, and that the sizes of the second and third buffers need to be set according to the time needed for the data copy operations! This means that an iterative self-adjusting mechanism needs to be placed into the software to determine the correct buffer sizing for optimal operation. Fixed values for buffer sizes may be used; in such a case, the LAPP method will still provide a significant performance increase, but the performance increase will not be maximized.

The following diagram illustrates this setup for a receive ring size of 9:

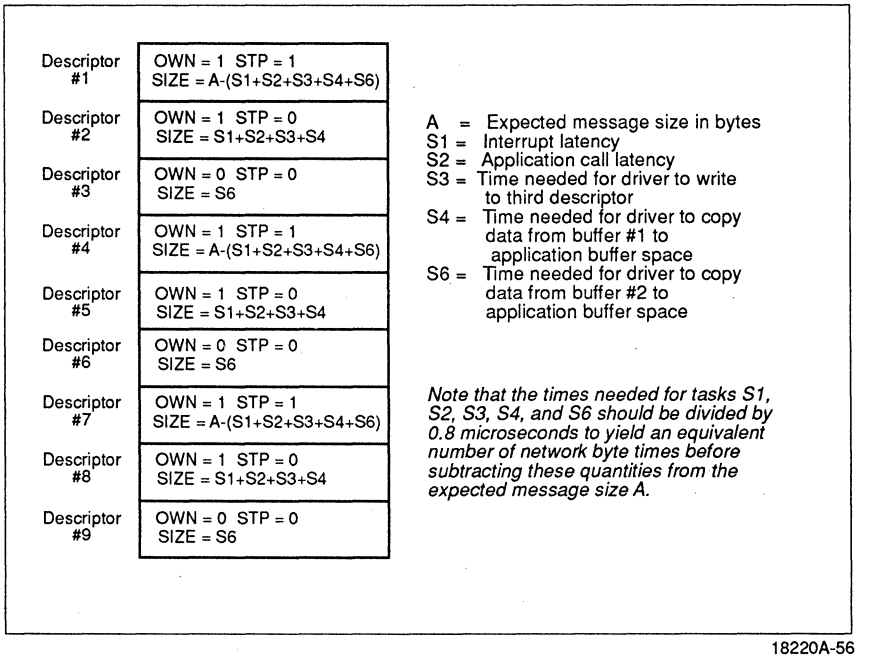


Figure D2. LAPP 3 Buffer Grouping

LAPP Rules for Parsing of Descriptors

When using the LAPP method, software must use a modified form of descriptor *parsing* as follows:

- Software will examine OWN and STP to determine where a RCV frame begins. RCV frames will only begin in buffers that have OWN = 0 and STP = 1.
- Software shall assume that a frame continues until it finds either ENP = 1 or ERR= 1.
- Software must discard all descriptors with OWN = 0 and STP = 0 and move to the next descriptor when searching for the beginning of a new frame; ENP and ERR should be ignored by software during this search.
- Software cannot change an STP value in the receive descriptor ring after the initial setup of the ring is complete, even if software has ownership of the STP descriptor unless the previous STP descriptor in the ring is also OWNED by the software.

When LAPPEN = 1, then hardware will use a modified form of descriptor *parsing* as follows:

- The controller will examine OWN and STP to determine where to begin placing a RCV frame. A new RCV frame will only begin in a buffer that has OWN = 1 and STP = 1.
- The controller will always obey the OWN bit for determining whether or not it may use the next buffer for a chain.
- The controller will always mark the end of a frame with either ENP = 1 or ERR= 1.

The controller will discard all descriptors with OWN = 1 and STP = 0 and move to the next descriptor when searching for a place to begin a new frame. It discards these descriptors by simply changing the ownership bit from OWN=1 to OWN = 0. Such a descriptor is unused for receive purposes by the controller, and the driver must recognize this. (The driver will recognize this if it follows the software rules).

The controller will ignore all descriptors with OWN = 0 and STP = 0 and move to the next descriptor when searching for a place to begin a new frame. In other words, the controller is allowed to skip entries in the ring that it does not own, but only when it is looking for a place to begin a new frame.

Some Examples of LAPP Descriptor Interaction

Choose an expected frame size of 1060 bytes. Choose buffer sizes of 800, 200 and 200 bytes.

- Assume that a 1060 byte frame arrives correctly, and that the timing of the early interrupt and the software is smooth. The descriptors will have changed from:

Descriptor Number	Before the Frame Arrives			After the Frame has Arrived			Comments
	OWN	STP	ENP	OWN	STP	ENP†	(after frame arrival)
1	1	1	X	0	1	0	bytes 1–800
2	1	0	X	0	0	0	bytes 801–1000
3	0	0	X	0	0	1	bytes 1001–1060
4	1	1	X	1	1	X	controller's current location
5	1	0	X	1	0	X	not yet used
6	0	0	X	0	0	X	not yet used
etc.	1	1	X	1	1	X	not yet used

† *ENP or ERR*

- Assume that instead of the expected 1060 byte frame, a 900 byte frame arrives, either because there was an error in the network, or because this is the last frame in a file transmission sequence.

Descriptor Number	Before the Frame Arrives			After the Frame has Arrived			Comments
	OWN	STP	ENP	OWN	STP	ENP†	(after frame arrival)
1	1	1	X	0	1	0	bytes 1–800
2	1	0	X	0	0	1	bytes 801–900
3	0	0	X	0	0	?*	discarded buffer
4	1	1	X	1	1	X	controller's current location
5	1	0	X	1	0	X	not yet used
6	0	0	X	0	0	X	not yet used
etc.	1	1	X	1	1	X	not yet used

† *ENP or ERR*

Note that the PCnet-PCI controller might write a ZERO to ENP location in the 3rd descriptor. Here are the two possibilities:

1. If the controller finishes the data transfers into buffer number 2 after the driver writes the applications modified buffer pointer into the third descriptor, then the controller will write a ZERO to ENP for this buffer and will write a ZERO to OWN and STP.
2. If the controller finishes the data transfers into buffer number 2 before the driver writes the applications modified buffer pointer into the third descriptor, then the controller will complete the frame in buffer number two and then skip the then unowned third buffer. In this case, the PCnet-PCI controller will not have had the opportunity to RESET the ENP bit in this descriptor, and it is possible that the software left this bit as ENP=1 from the last time through the ring. Therefore, the software must treat the location as a don't care; The rule is, after finding ENP=1 (or ERR=1) in descriptor number 2, the software must ignore ENP bits until it finds the next STP=1.

   Assume that instead of the expected 1060 byte frame, a 100 byte frame arrives, because there was an error in the network, or because this is the last frame in a file transmission sequence, or perhaps because it is an acknowledge frame.

* Same as note in case 2 above, except that in this case, it is very unlikely that the driver can respond to the interrupt and get the pointer from the application before the PCnet-PCI controller has completed its poll of the next descriptors. This means that for almost all occurrences of this case, the PCnet-PCI controller will not find the OWN bit set for this descriptor and therefore, the ENP bit will almost always contain the old value, since the PCnet-PCI controller will not have had an opportunity to modify it.

** Note that even though the PCnet-PCI controller will write a ZERO to this ENP location, the software *should* treat the location as a don't care, since after finding the ENP=1 in descriptor number 2, the software should ignore ENP bits until it finds the next STP=1.

Descriptor Number	Before the Frame Arrives			After the Frame has Arrived			Comments
	OWN	STP	ENP	OWN	STP	ENP†	(after frame arrival)
1	1	1	X	0	1	1	bytes 1–100
2	1	0	X	0	0	0**	discarded buffer
3	0	0	X	0	0	?*	discarded buffer
4	1	1	X	1	1	X	controller's current location
5	1	0	X	1	0	X	not yet used
6	0	0	X	0	0	X	not yet used
etc.	1	1	X	1	1	X	not yet used

† *ENP or ERR*

Buffer Size Tuning

For maximum performance, buffer sizes should be adjusted depending upon the expected frame size and the values of the interrupt latency and application call latency. The best driver code will minimize the CPU utilization while also minimizing the latency from frame end on the network to frame sent to application from driver (frame latency). These objectives are aimed at increasing throughput on the network while decreasing CPU utilization.

Note that the buffer sizes in the ring may be altered at any time that the CPU has ownership of the corresponding descriptor. The best choice for buffer sizes will maximize the time that the driver is swapped out, while minimizing the time from the last byte written by the PCnet-PCI controller to the time that the data is passed from the driver to the application. In the diagram, this corresponds to maximizing S0, while minimizing the time between C9 and S8. (The timeline happens to show a minimal time from C9 to S8.)

Note that by increasing the size of buffer number 1, we increase the value of S0. However, when we increase the size of buffer number 1, we also increase the value of S4. If the size of buffer number 1 is too large, then the driver will not have enough time to perform tasks S2, S3, S4, S5 and S6. The result is that there will be delay from the execution of task C9 until the execution of task S8. A *perfectly timed system will have the values for S5 and S7 at a minimum.*

An average increase in performance can be achieved if the general guidelines of buffer sizes in figure 2 is followed. However, as was noted earlier, the correct sizing for buffers will depend upon the expected message size. There are two problems with relating expected message size with the correct buffer sizing:

1. Message sizes cannot always be accurately predicted, since a single application may expect different message sizes at different times, therefore, the buffer sizes chosen will not always maximize throughput.
2. Within a single application, message sizes might be somewhat predictable, but when the same driver is to be shared with multiple applications, there may not be a common predictable message size.

Additional problems occur when trying to define the correct sizing because the correct size also depends upon the interrupt latency, which may vary from system to system, depending upon both the hardware and the software installed in each system.

In order to deal with the unpredictable nature of the message size, the driver can implement a self tuning mechanism that examines the amount of time spent in tasks S5 and S7 as such: while the driver is polling for each descriptor, it could count the number of poll operations performed and then adjust the number 1 buffer size to a larger value, by adding "t" bytes to the buffer count, if the number of poll operations was greater than "x". If fewer than "x" poll operations were needed for each of S5 and S7, then the software should adjust the buffer size to a smaller value by, subtracting "y" bytes from the buffer count. Experiments with such a tuning mechanism must be performed to determine the best values for "X" and "y".

Note whenever the size of buffer number 1 is adjusted, buffer sizes for buffer number 2 and buffer 3 should also be adjusted.

In some systems, the typical mix of receive frames on a network for a client application consists mostly of large data frames, with very few small frames. In this case, for maximum efficiency of buffer sizing, when a frame arrives under a certain size limit, the driver should not adjust the buffer sizes in response to the short frame.

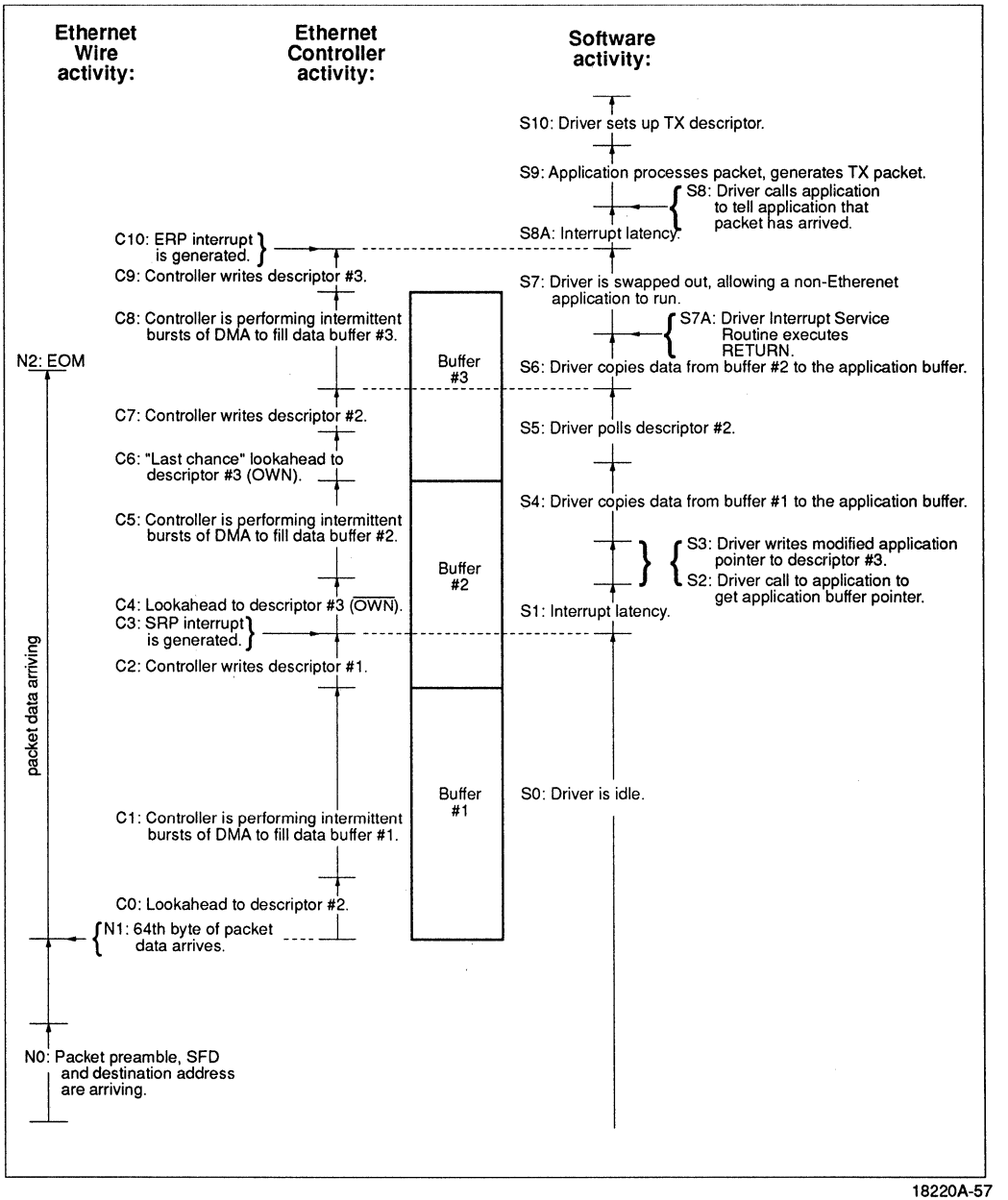
An Alternative LAPP Flow – the TWO Interrupt Method

An alternative to the above suggested flow is to use two interrupts, one at the start of the receive frame and the other at the end of the receive frame, instead of just looking for the SRP interrupt as was described above. This alternative attempts to reduce the amount of time that the software wastes while polling for descriptor own bits. This time would then be available for other CPU tasks. It also minimizes the amount of time the CPU needs for data copying. This savings can be applied to other CPU tasks.

The time from the end of frame arrival on the wire to delivery of the frame to the application is labeled as frame latency. For the one-interrupt method, frame latency is minimized, while CPU utilization increases. For the two-interrupt method, frame latency becomes greater, while CPU utilization decreases.

Note that some of the CPU time that can be applied to non-Ethernet tasks is used for task switching in the CPU. One task switch is required to swap a non-Ethernet task into the CPU (after S7A) and a second task switch is needed to swap the Ethernet driver back in again (at S8A). If the time needed to perform these task switches exceeds the time saved by not polling descriptors, then there is a net loss in performance with this method. Therefore, the LAPP method implemented should be carefully chosen.

Figure D3 shows the event flow for the two-interrupt method:



18220A-57

Figure D3. LAPP Timeline for TWO-INTERRUPT Method

Figure D4 shows the buffer sizing for the two-interrupt method. Note that the second buffer size will be about the same for each method.

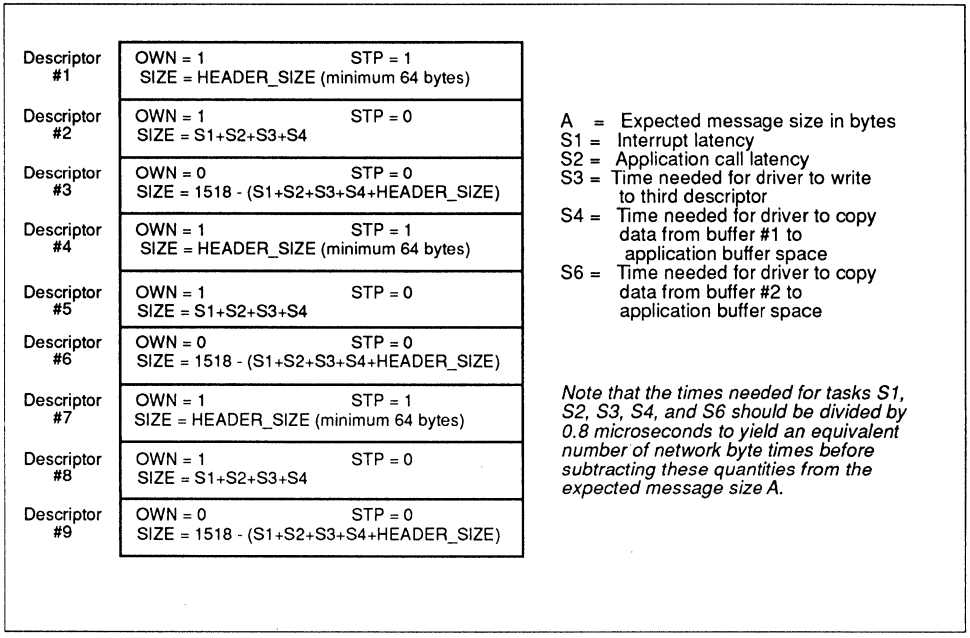


Figure D4. LAPP 3 Buffer Grouping for TWO-INTERRUPT Method

There is another alternative which is a marriage of the two previous methods. This third possibility would use the buffer sizes set by the two-interrupt method, but would use the polling method of determining frame end. This will give good frame latency but at the price of very high CPU utilization.

And still, there are even more compromise positions that use various fixed buffer sizes and effectively, the flow of the one-interrupt method. All of these compromises will reduce the complexity of the one-interrupt method by removing the heuristic buffer sizing code, but they all become less efficient than heuristic code would allow.

DATA SHEET REVISION SUMMARY

The following list represents the key differences between revision B (May 1994) and revision C (June 1994).

Global Change

DWIO mode cannot be set by reading the EEPROM or writing directly to BCR18.

The Initialization Block must be on a double-word boundary.

Detailed Functions

Page 1-888:

The section on PCnet-PCI controller I/O Resource Mapping is rewritten to reflect changes in methods of setting DWIO mode.

User Accessible Registers

Page 1-955:

CSR1
The description for Initialization Block boundary requirement is rewritten for clarity.

Page 1-967:

CSR58
The description for bit 8 (SSIZE32) is rewritten for clarity.

Page 1-982:

BCR18
The description for bit 7 (DWIO) is rewritten for clarity.

The following list represents the key differences between revision A (October 1993) and revision B (April 1994).

Page 1-987:

BCR20
The description for bit 8 (SSIZE32) is rewritten for clarity.

Global Change

Look-Ahead Packet Processing (LAPP) is the name for the early protocol analysis.

Block Diagram

Page 1-869:

The $\overline{\text{LOCK}}$ pin is changed from bidirectional to unidirectional. It is now an input only. The EESK/$\overline{\text{LED1}}$ pin is now changed to bidirectional for typographical correction.

Connection Diagram

Page 1-877:

Various pin names have been changed for either enhanced clarity or to correct typographical errors.

The pins that are affected are 9, 58, 91, 94, 96, 97, 98, 100, 103, 108, 116, and 127.

Pin Designations

Page 1-879:

Various pin names have been changed for either enhanced clarity or to correct typographical errors.

The pins that are affected are 9, 58, 91, 94, 96, 97, 98, 100, 103, 108, 116, and 127.

Page 1-880:

The $\overline{\text{LOCK}}$ pin is changed from an I/O to an input. No driver type is available.

Page 1-881:

The I_{OH} value for TS3 and TS6 are now –2. Driver type O6 is removed. Driver type O8 is added.

Pin Description

Page 1-882:

Pin descriptions for various pins were rewritten for clarity. The pins are $\overline{\text{GNT}}$, $\overline{\text{LOCK}}$, and $\overline{\text{PERR}}$.

Detailed Functions
Page 1-937:

Table 8

EEPROM Contents—Corrected the Hardware ID (byte address 09h) value to 11h.

Page 1-944:
Figure 30

NAND Tree—Typographical errors were corrected.

User Accessible Regisers

Page 1-956:

CSR3
The bit name for bit 5 is changed to LAPPEN (Look-Ahead Packet Processing ENable).

Page 1-958:

CSR4
The bit name for bit 9 is changed to MFCO (Missed Frame Counter Overflow), and the bit name for bit 8 is changed to MFCOM (Missed Frame Counter Overflow Mask).

Page 1-971:

CSR80—The description for bits 9–8 is rewritten for clarity.

Page 1-974:

CSR89—The upper 12 bits of the PCnet–PCI controller part number contained in bits 11–0 are corrected. The 12 bits now read 0010 0100 0011b.

CSR90—The description for this register is removed, because it is now a reserved register.

Page 1-975:

CSR124—The description for bit 3 is rewritten for clarity.

Page 1-982:

BCR18—The descriptions for bits 5 and 6 are rewritten for clarity.

Page 1-994:

Tabel 18—The table is cleaned up for clarity.

DC Characteristics

Page 1-1001:

V_{OL}
The maximum value is 0.55 V if I_{OL} is 3 or 6 mA. The maximum value is 0.4 V if I_{OL} is 12 mA.

Page 1-1003:

CO
The maximum value is now 10 pF.

Appendices

Appendix A

This section is updated with the latest information.

Appendix B

This section is rewritten for clarity.

Appendix D

This is a new section on the LAPP Concept.

SECTION

2 APPLICATIONS

Am7996 IEEE-802.3 (Ethernet/Cheapernet) Transceiver Application Note

Am7996 IEEE-802.3 (ETHERNET/CHEAPERNET) Transceiver

INTRODUCTION

The Am7996 IEEE-802.3 (Ethernet/Cheapernet) Transceiver integrates all the transceiver functions required for 10 Mbps CSMA/CD (Carrier Sense Multiple Access with Collision Detection) LANs. These functions include transmit receiver, collision detect, iptional signal Quality Error (SQE) test, Jabber timer (including the hooks for an external redundant jabber) and noise rejection filters (see Figure 1.) It highly integrates the Ethernet/Cheapernet systems when used with the AM7990 LANCE (data link controller) and Am7992B Serial Interface Adapter (SIA) (Manchester encoder/decoder).

The Am7996 is a CSMA/CD transceiver whose main structure consists of three functional blocks; transmit including jabber control, receiver, and collision detection. Each section op-

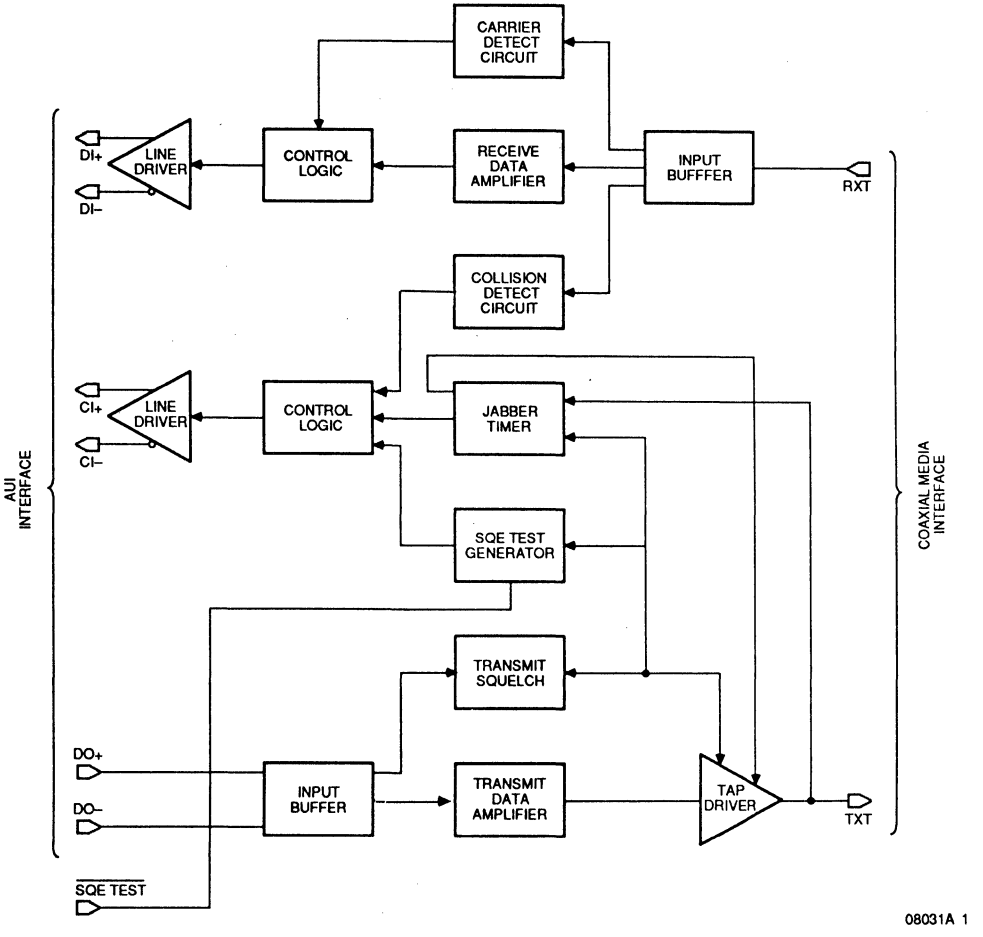


Figure 1. Am7996 Block Diagram

erates independently. In the transmit section, data is received differently from the Data Terminal Equipment (DTE) and transmitted out, single ended, to the medium (coax cable). The Jabber function guards the medium from node transmissions that are excessive in length. The receive section listens to data differentially to the DTE. The collision detection section monitors the medium for simultaneous transmissions, and when that occurs it reports it to the DTE via a 10 MHz differential signal.

This application note first describes briefly the Ethernet/Cheapernet standards. The use of Am7996 with extended cable lengths is then explained. The three functional blocks (transmit, receive, and collision detection) mentioned above are then discussed in detail. This is followed by practical guidelines regarding the external components required. Measurement techniques are also discussed. Finally, application examples are given.

IEEE-802.3 Standard (Ethernet/Cheapernet)

The IEEE-802.3 is the existing standard for the bottom two layers of the 7 layer Open System Interconnection (OSI) which was formulated and adopted by the International Standards Organizations (ISO). The main structure of the specification comes from Ethernet which was jointly developed by XEROX, Digital Equipment Corporation and Intel.

Another standard, known in the industry as "Cheapernet", was developed by the same committee at a faster pace than the 802.3 Ethernet standard. Cheapernet is an extension to the existing and proven standard, IEEE-802.3 Ethernet. It is a CSMA/CD network at 10 Mbps. Its network architecture is the same as Ethernet except it incorporates cheaper cable, connectors, and maintenance. Its installation consists of simply connecting an RG-58 cable to a BNC connector.

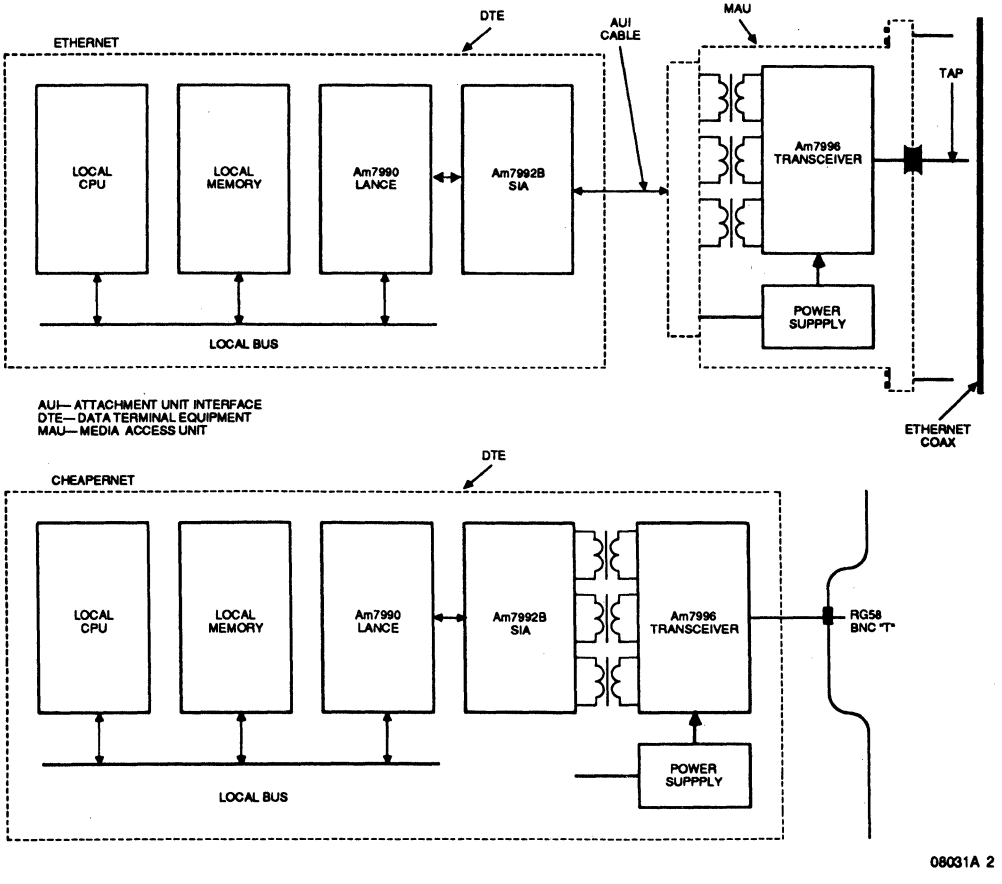


Figure 2. Ethernet and Cheapernet Configurations

Figure 2 shows a block diagram of an Ethernet and a Cheapernet configuration.

IEEE-802.3 refers to the original standard, Ethernet, as 10BASE5 or Type A applications, and refers to the second one, Cheapernet, as 10BASE2 or Type B applications. In the IEEE-802.3 terminology, 10BASE refers to 10 MHz baseband and the suffix 5 or 2 refers to 500 or 200 meter cable segment, respectively. Note that the actual length of the cable segment is 185 meters in the Cheapernet specification.

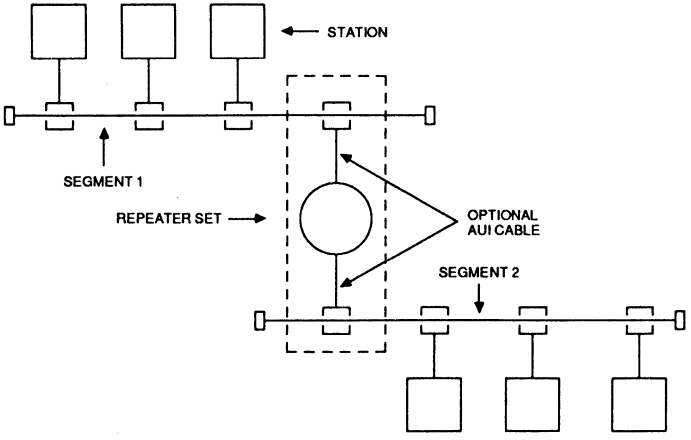
In an Ethernet installation, up to 100 Media Access Units (MAU) may be connected to one cable segment of 500 meters. In a Cheapernet installation, up to 30 MAUs may be connected to one cable segment of 185 meters. In either Ethernet or Cheapernet, repeaters may be used to connect up to five segments together into one network. Refer to Figure 3.

The Am7996 has been targeted for both applications, Ethernet and Cheapernet. It offers the flexibility and the engineering hooks for some of the tight parameters, and for network protection which is required in Ethernet applications. This gives the user the flexibilty of applying the same chip for both applications. Most OEM boards are now designed to include both options, an onboard transceiver for cheapernet application, and an optional 15 pin D connector for the AUI cable to access the Ethernet transceiver box.

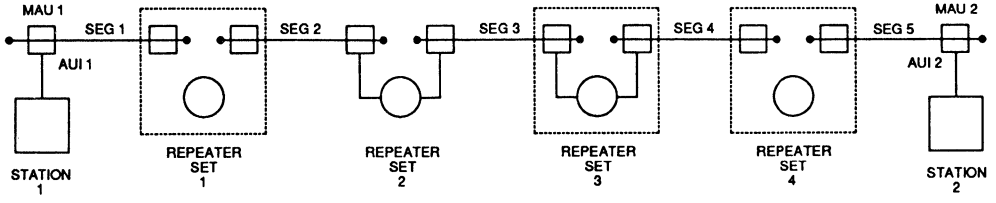
The salient features of Ethernet and Cheapernet are shown in Table 1.

Am7996 Application in Extended Cable Lengths

The Am7996 has been designed for Transmit Mode collision detection. (Collision detection methods are discussed later in this manual.) As a result, the Am7996 can be used in longer cable segments than specified in the IEEE-802.3. Table 2 below shows the extended cable length feature of the Am7996.



MINIMAL SYSTEM CONFIGURATION REQUIRING A REPEATER SET



MAXIMUM IEEE-802.3 TRANSMISSION PATH

08031A 3

Figure 3. Ethernet/Cheapernet Network Configuration

The extended cable segment feature can eliminate the need for repeaters. This reduces the cost and adds flexibility in installing more nodes. Note that the extended cable segment support does not violate the IEEE-802.3 standard. It is the Transmit Mode collision detection scheme, in Am7996, which allows this extended feature.

TRANSMIT FUNCTION

The Am7996 receives differential signals from the DTE (in the case of Am7990 family applications, from the Am7992B–Serial Interface Adapter–SIA). For IEEE-802.3 Type A (Ethernet) applications, this signal is received through the AUI cable and isolation transformer. In IEEE-802.3 Type B (Cheapernet) applications, the AUI cable (but not the isolation transformer) is optional.

Data is received through an internal noise rejection filter that rejects signals with pulse widths less than 7 ns (negative going), or greater than 160 ns (positive going) with levels less than –175 mV peak. Only signals greater than –275 mV peak from the DTE are accepted. This minimizes false starts due to noise and ensures no valid packets are missed.

The Am7996's Tap driver provides the driving capability to ensure adequate signal level at the end of the maximum length network segment (500

Table 1. IEEE-802.3 10BASE5 (Ethernet) and 10BASE2 (Cheapernet) Comparison

Parameter	IEEE-802.3 10BASE5 (Ethernet)	IEEE-802.3 10BASE2 (Cheapernet)
Data Rate	10 Mbps	10 Mbps
Segment Length	500 meter	185 meter
Network Length	2500 meter	925 meter
Nodes Per Segment	100	30
Node Spacing (Min)	2.5 meter	0.5 meter
Cable/Connector	0.4 in diameter, 50 Ohm Double shielded shielded N-series connectors Rugged	0.2 diameter, 50 Ohm (RG-58 A/U type) Single BNC connectors Flexible (thin)
Transceiver Cable (AUI interface)	0.38 in. diameter multiway cable with 15 pin D connectors (Length up to 50 meter)	Optional
Capacitance/Node	4 pF	8 pF
Installation	Installer required (costly)	Mainly by user (low cost)

Table 2. Extended Cable Lengths with Am7996

Applications	CABLE LENGTH	
	IEEE-802.3	*Am7996
10BASE5, Ethernet (Ethernet Thick cable)	500 meter	1000 meter
10BASE2, Cheapernet (RG58 Thin cable)	185 meter	300 meter

* Transmit mode collision implementation

meters Ethernet, or 185 meters Cheapernet) under the worst case number of connections (100 nodes Ethernet, or 30 nodes Cheapernet). Required rise and fall times of data transmitted on the network are maintained by the Am7996 Tap driver. The Tap driver's output is connected to the media through external isolating diodes. To safeguard network integrity, the driver is disabled whenever the operating voltage falls below the minimum.

Signal Quality Error (SQE) Test (Heartbeat)

A diagnostic feature has been specified for the MAU in the IEEE-802.3. The Signal Quality Error (SQE) Test is a self test feature in the MAU which is invoked after the end of each transmission by the DTE. The SQE test starts eight bit times after the last transition of the transmitted signal and lasts for a duration of eight bit times. The Am7996 sends a 10 MHz differential signal through $CI_{\pm}$ to the DTE when $\overline{SQE\ TEST}$ pin (pin 7) is tied to V_{EE}. This test is an indication to the DTE that the MAU has recognized the end of the transmission and the collision pair, $CI_{\pm}$, is intact and operational.

Pin-strappable $\overline{SQE\ Test}$ Option: The SQE test is selectable via the $\overline{SQE\ TEST}$ pin (pin 7). It can be tied to V_{EE} for SQE test or to ground (V_{CC1} or V_{CC2}) for a non-SQE test MAU. The optional feature allows the use of the Am7996 in both repeater and non-repeater applications.

Jabber Function

Another means of protecting the network (medium) is to ensure that no node, under any circumstances, hangs the network. In an IEEE-802.3 network, the maximum packet size is limited to 1518 octets which is equivalent to about 1.2 ms (including the 64 bit preamble). The Jabber timer monitors the activity on the DO pair and senses TXT (pin 12) faults for excessive continuous transmission. The Jabber goes active and inhibits transmission if the Tap driver is active for longer than the Jab time. The Jab time specified by IEEE-802.3 is from 20 to 150 ms. In the Am7996, it is from 20 to 35 ms. When the Jabber goes active, it isolates the output drivers at the Tap from the coax and enables an SQE message (10 MHz differential collision signal) on the CI pair for the fault duration.

IEEE 802.3 states that in a self-powered MAU, the Jabber timer and collision presence on the CI pair are cleared after the fault condition goes away for a period of 500 ms ± 50% (250 to 750 ms). The Jabber reset time in the Am7996 is between 340 and 500 ms.

Redundant Jabber: The hooks for optional redundant protection specified by the IEEE-802.3 have been implemented in the Am7996. A redundant Jabber sits outside the Am7996 and typically duplicates the Jabber internal to the Am7996. When the external Jabber is implemented, the VTX-pin is directly controlled by the external Jabber circuitry. To externally disable TXT (and enable an SQE message on CI pair), the voltage at VTX-pin should be brought to a value more positive than V_{EE} + 2V.

Jabber Recovery Time: One of the parameters in the Jabber function is the Jabber recovery time which is significant in terms of the correct operation of the Jabber timer. The Jabber timer always starts counting from the start of each transmission and is reset at the end of the transmission. The time required for the timer to reset is called the Jabber recovery time. This parameter is 1 µs (max) in the Am7996. It is important that this time be as short as possible.

Consider an IEEE-802.3 network in maximum configuration. There are four repeaters in such a configuration. Because of the nature of such regenerative repeaters, the InterPacket Gap time (IPG) can shrink from 9.6 µsec (9.6 µsec is the IPG spec in IEEE-802.3 for the gap between two consecutive transmissions) to about 5.0 µsec in a normal packet transfer. The IPG can even shrink to less than 5.0 µsec under excessive collisions. Therefore, if the Jabber recovery is not short enough, the Jabber timer will not be cleared and will continue to count after the start of the next packet. Under the worst case condition, when back-to-back packets are in progress with short IPG, the Jabber goes falsely active. The Jabber recovery time is 1 µsec (max) in the Am7996 insuring proper operation under the worst case conditions.

Inhibit Internal Jabber: It may be desired in some non IEEE-802.3 applications to disable the Jabber function. Consider a point-to-point application where a continuous transmission of more than 20 ms is desired. For such applications, the internal Jabber can be disabled by removing the external collision oscillator circuitry (R_4, C_1) and connecting COLL OSC (pin 19) to V_{CC2} (pin 20). Note that this will also inhibit the SQE test and any CI pair message for collision presence.

RECEIVE AND CARRIER DETECT

The signal is acquired from the Tap through a high impedance (100 kOhm) resistive divider. A high input-impedance (low capacitance, high bandwidth, low noise) DC-coupled input amplifier in the Am7996 receives the signal. The received signal passes through a high-pass filter to minimize intersymbol distortion, and then through a data slicer. The Am7996 carrier detect compares received sig-

nals to a reference. Signals meeting carrier squelch criteria enable data to the differential line driver within five bit times from the start of the packet.

Received data is transmitted from the DI pair through an isolation transformer to the Ethernet AUI cable (IEEE-802.3–Type A). In IEEE-802.3 Type B (Cheapernet), the AUI cable is optional. Following the last transition of the packet, the DI pair is held high for two bit times and then decreases to idle level within twenty bit times.

COLLISION

Collision occurs when two or more transceivers attempt simultaneous transmissions on the medium. In a CSMA/CD network, a mechanism is needed to resolve the contention. All the intelligence for collision back-offs, and the retry process resides in the controller (Am7990). The Am7996 detects a collision when the DC average of the signals on the coax crosses the collision detect threshold. The collision threshold window has been based on the worst case conditions in the IEEE-802.3 cable segment (500 meter Ethernet cable, or 185 meters of RG-58 cable) when two nodes transmit at the same time.

Collision Reporting

When the Am7996 detects a collision, it generates a 10 MHz differential signal at CI± which continues as long as there is a collision in progress. The 10 MHz differential signal is normally detected by the Manchester Encoder/Decoder at the DTE (SIA, Am7992B) which translates to a TTL signal for the LAN controller (LANCE, Am7990).

Collision Detection Methods

There are two types of collision detection specified by the IEEE-802.3 standard: Transmit Mode and Receive Mode collision detection. The Am7996 has been designed to support Transmit Mode collision detection. The collision threshold window for Transmit Mode collision allows longer cable segment applications than what IEEE-802.3 has specified. For the Am7996 in repeater applications, Receive Mode collision detection can also be accomplished by adding two resistors, R_9 and R_{10}, external to the chip. Receive Mode collision detection is optional when the Media Access Unit (MAU) is used in non-repeater applications.

Transmit Mode Collision Detection: While transmitting, the MAU must detect a collision if one or more other nodes are also transmitting, and may detect collision, while not transmitting, if two other nodes are transmitting. This is called Transmit Mode collision detection. As a result of this type of collision detection, longer cable segments than what IEEE-802.3 has specified can be used. This type of collision detection is normally used in non-repeater applications.

Receive Mode Collision Detection: Regardless of whether a MAU is transmitting or not transmitting, the MAU must detect collision if two or more modes (perhaps including itself) are transmitting. This scheme requires a tighter threshold than Transmit Mode collsion detection. The Receive Mode collision detection limits the cable length to what the IEEE-802.3 has specified whereas transmit mode collision detection, due to its wider threshold window, allows for the extended cable segment.

Receive Mode collision detection is not necessary in non-repeater applications, but, it is a must in repeater applications since the carrier has to be sensed by both sides of the repeater. Figure 4 shows the external component configuration for Cheapernet Receive Mode collision detection.

The Am7996 meets the IEEE-802.3 collision detect requirements (see Table 3).

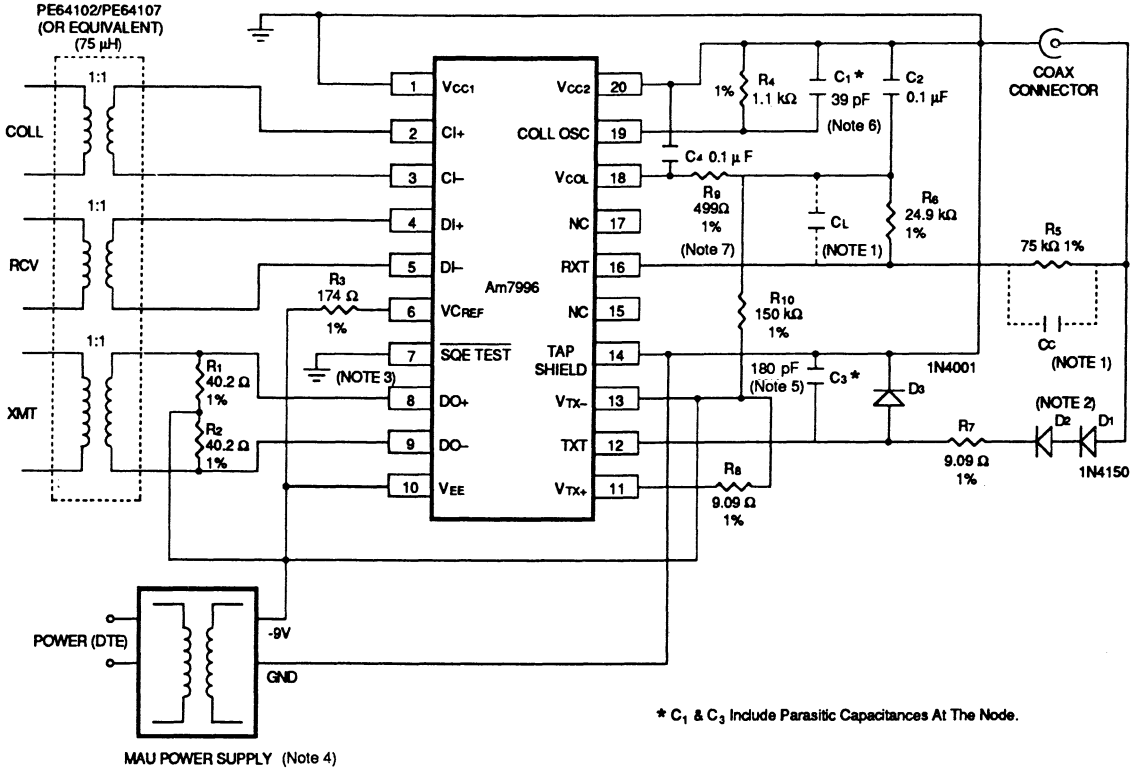
Collision Detection in Non-Repeater Applications: Receive Mode collision detection is normally not necessary when designing a MAU in a non-repeater application. This is because the received packet, the packet participating in collision, usually ends up as a runt packet (a packet less than 64 bytes long) which is normally discarded by Data Link layer controllers. In other cases such as late collision (collision occurring after 64 bytes), CRC error can be an indication that there might have been a collision in progress while receiving the packet.

Collision Detection in Repeater Applications: Receive Mode collision detect is strictly required in Repeater design applications. A repeater must report all the activities of either side of the network to the other side. In the case of a collision at one side of the network, the repeater must create the collision environment on the other side. The repeater must detect a collision caused by two nodes that occur from the far end of a segment. Since cable attenuation results in a lower level seen by the repeater, the Receive Mode Collision Detection specs must be tighter. If a collision occurs on one segment, the repeater sends a collision jam signal to the other segment to report such activity.

The Am7996 meets the Receive Mode collision detect as well, when R_9, R_{10}, and C_4 are integrated into the Am7996 external component diagram (see Figure 4).

EXTERNAL COMPONENT DESIGN GUIDELINES

The design and layout choices of the compo-



Notes: 1. C_L is the effective load capacitance across R_6; C_C is the compensation capacitance (C_C = 1/3 C_L).

2. D_2 can eliminated in Cheapernet (IEEE-802.3, 10Base2) applications.

3. Shown with SQE Test disabled.

4. Discrete Power Supply or Hybrid – Hybrid DC-DC Converter Manufacturers include:

Ethernet (IEEE 802.3, 10BASE5)	**Cheapernet (IEEE 802.3,10BASE2)**
Reliability Inc. 2E12R9	Reliability Inc: 2VP5U9
Valor Electronics: PM1001	Valor Electronics: PM7102

5. The capacitance of C_3, Am7996 package, D_3 and the printed circuit board should add up to 180 pF ± 20%.

6. The capacitance of C_1, Am7996 package and the printed circuit board should add up to 39 pF.

7. R_9, R_{10}, and C_4 are for Receive Mode Collision detection only.

08031A 4

Figure 4. Am7996 External Component Diagram for Receive Mode Collision Detection

nents external to the chip in the Am7996 adds flexibility, network protection, and hooks for achieving the tight parameters specified in the IEEE-802.3 and Ethernet specifications. The following describes the design considerations to be aware of in choosing those external components around the Am7996. Figure 5, the external components diagram, should be used in reference to the discussion below.

Layout Considerations

The Am7996 should be mounted as close as possible to the Tap for minimum capacitive loading. To minimize the capacitance at RXT (pin 16) between its adjacent pins (pins 15 and 17) and the capacitance introduced by TXT (pin12) to the Tap through external components, package, and PC trace, carefully layout the PC board as follows:

1. It is recommended that metal feed-throughs are not used at pins 15 and 17. These pins are No Connect pins.

2. Generally, all the PC traces between the chip and external components should be as short as possible. Additional effort should be made to place R_5, R_6, RXT (pin 16) and D_1, D_2, R_7, TXT (pin 12), and Am7996 close to the Tap.

3. To achieve the minimum capacitive loading at the Tap connection, there should be no power, ground, or signal planes in the area of Tap interconnections to 7996 pins (pins 11–18). (See also ground requirements discussed later.

4. The 7996 should be directly soldered to the PC board without a socket to reduce capacitive loading at the Tap connection.

5. Grounding:

   V_{CC1} and V_{CC2} (pin 1 and 20 respectively) must be connected together to the positive return (positive polarity of power to 7996).

   The Tap shield pin (pin 14) should be connected directly, via a single trace, to the shield of the coax connector. There should not be any ground plane connection to the Tap shield which will add to Tap capacitive loading.

   The DTE ground plane should not be extended beyond the pulse transformer (the one at the Am7996 side).

In IEEE-802.3 applications, using the AUI (Attachment Unit Interface) cable, the DTE logic ground can be extracted from any of pins 1, 4, 6, 8, 11, or 14 of the 15 pin D connector. In Ethernet Version 2 applications, Pins 4, 8, 11, and 14 are No Connect (NC) pins. The DTE logic ground can be extracted from Pin 6 only. See appendices A and C for pinout details and PC board layout considerations.

Tap Capacitance Loading Considerations

The goal is to minimize the capacitive loading at the Tap from both the receive path (RXT, pin16) and the transmit path (TXT, pin 12) to achieve the input impedance requirements of IEEE-802.3 specification.

A properly compensated external 4:1 attenuator (75K and 25K in series) reduces any parasitic capacitive loading in the receive path by a factor of 4 and ensures that the resistance presented to the coaxial cable will be at least 100K Ohms.

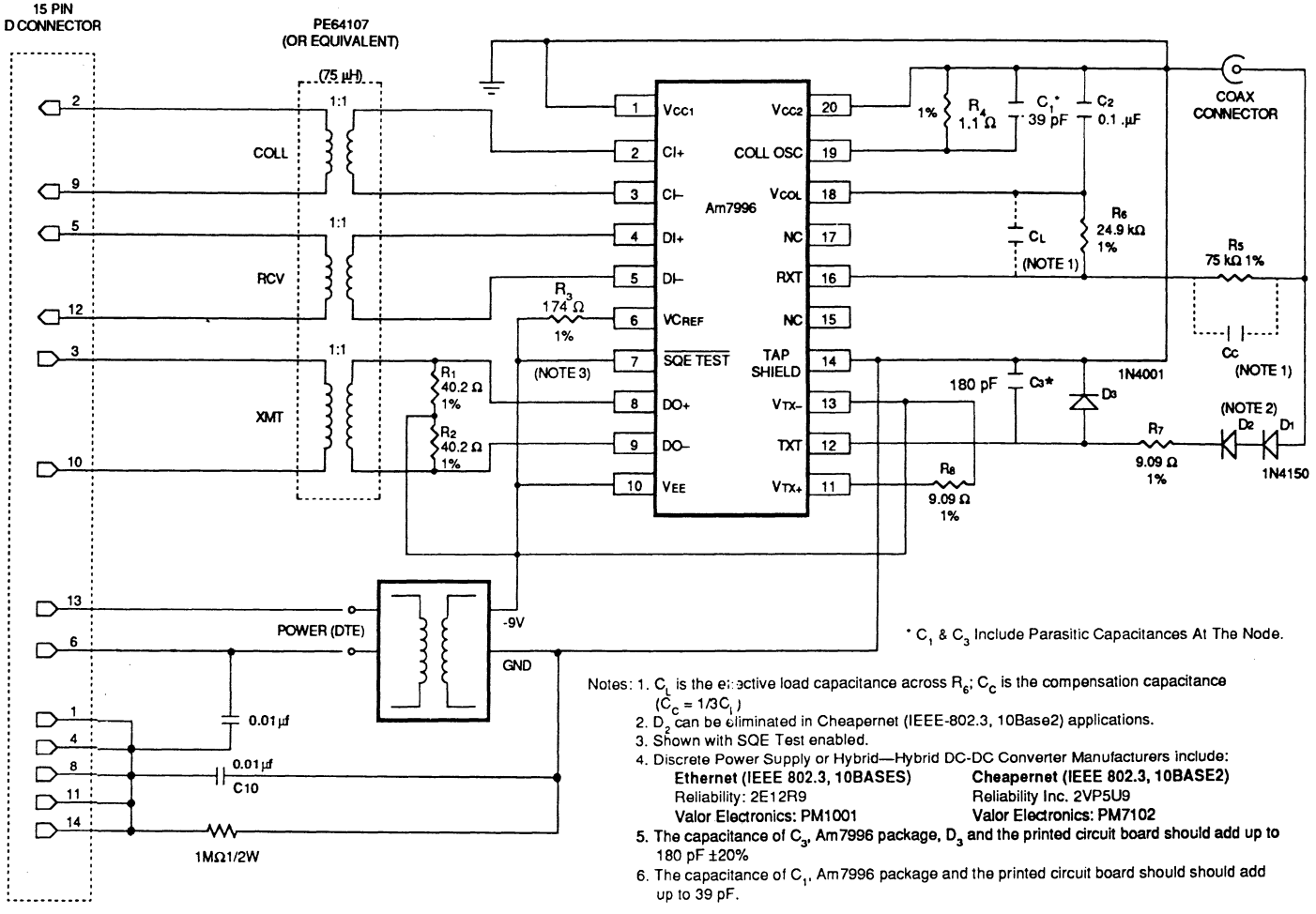
When the chip is not transmitting, the transmit path

Table 3. IEEE-802.3 Transmit Mode and Receive Mode Collision Detection.

	TRANSMIT MODE			*RECEIVE MODE		
MAU	Number of transmitters			Number of transmitters		
	<2	=2	>2	<2	=2	>2
Transmitting	NO	YES	YES	NO	YES	YES
Not transmitting	NO	MAY	YES	NO	YES	YES

NO: Will not generate SQE message
YES: Will generate SQE message
MAY: May generate SQE message

* Receive Mode collision detection is optional per IEEE-802.3 in non-repeater applications



* C_1 & C_3 Include Parasitic Capacitances At The Node.

Notes:
1. C_L is the effective load capacitance across R_6; C_C is the compensation capacitance ($C_C = 1/3C_L$)
2. D_2 can be eliminated in Cheapernet (IEEE-802.3, 10Base2) applications.
3. Shown with SQE Test enabled.
4. Discrete Power Supply or Hybrid—Hybrid DC-DC Converter Manufacturers include:

Ethernet (IEEE 802.3, 10BASES)	**Cheapernet (IEEE 802.3, 10BASE2)**
Reliability: 2E12R9	Reliability Inc. 2VP5U9
Valor Electronics: PM1001	Valor Electronics: PM7102

5. The capacitance of C_3, Am7996 package, D_3 and the printed circuit board should add up to 180 pF ±20%
6. The capacitance of C_1, Am7996 package and the printed circuit board should should add up to 39 pF.

Figure 5. Am7996 External Component Diagram for Transmit Mode Collision Detection

08031A 5

is isolated from the Tap through a low capacitance switching diode. When transmitting, the additional diode in series reduces the capacitance loading at the Tap. IEEE-802.3 10BASE5 (Ethernet) has specified a limit of 2 pF for capacitive loading due to the MAU (Medium Attachment Unit) circuitry (total of 4 pF including the Tap connection). This tight limit is quite difficult to meet without additional circuitry such as the 4:1 attenuator, external to the Am7996.

In the IEEE-802.3 10BASE2, (Cheapernet) applications, the capacitive loading specification, due to the MAU circuitry, has been relaxed to 6 pF. Therefore, it is not as difficult to meet the capacitive loading requirement in Cheapernet applications. The external 4:1 attenuator also isolates the receive input of the integrated circuit from the Tap for safety and protection.

Attachment Unit Interface (AUI) Cable Terminator (R_1, R_2)

The DO± line receiver inputs should be terminated with R_1 and R_2, as shown in the external component diagram, equivalent to AUI cable impedance of 78.0 Ohms nominal. The effective parallel combination of the 80.4 Ohms ($R_1 + R_2$) and the DO± input impedance meets the IEEE-802.3 requirement of 78.0 ± 5 Ohms.

Although AUI cable is not normally used in Cheapernet applications, R_1, R_2, and the terminating resistors at the SIA (Am7992B) side must remain in. The terminating resistors are part of the load seen by the output drivers of the Am7996, DI±, and Transmit± of the Am7992B. Therefore, the removal of the terminating resistors will affect the differential level signals at DI± and Transmit±. Refer to the Cheapernet application example in the application section appearing later in this manual.

Timing Reference Resistor (R_3)

When the resistor, R_3, is connected between VC_{REF} (VC_{REF} is a compensated voltage reference input with respect to V_{EE}) and V_{EE}, the internal transmit and receive squelch timing and SQE oscillator frequency are set. SQE frequency is also determined by components connected between V_{CC2} (pin 20) and COLL OSC (pin 19).

SQE Oscillator Control (R_4, C_1)

In the Am7996, the collision oscillator frequency control is external to the chip. For a 10 MHz nominal SQE oscillator frequency, R_4 should be 1.1K 1%, and C1, 39pF ±5%, (including any parasitic capacitance). This will generate an SQE message with frequency of 10 MHz ± 15% for the following three cases:

1. SQE test
2. Collision for multinode transmission
3. Active Jabber

When VC_{REF} (pin 6) is properly set (it is set by placing $R_3 = 174\Omega$ between VC_{REF} and V_{EE} pin 10), the SQE oscillator period is set at $2.331R_4C_1$.

4:1 Attenuator (R5 and R6, CL and CC)

The chip acquires the signal from the Tap through a high impedance (100K Ohms) 4 to 1 attenuator. For proper reception of 10.0 MHz Manchester bit streams, the input attenuator at the RXT pin should be compensated to maintain the 4:1 ratio. Compensation is acieved by making $75 \times C_C = 25 \times C_L$ ($C_C = 1/3\ C_L$). C_L is the total effective capacitance between RXT (pin16) and V_{COL} due to the package, external components, and PC trace. C_C is the compensation capacitance across the 75 kOhm resistor.

C_C is typically less than 2.0 pF when short PC traces are used around RXT's (pin 16) external components. A possible way of achieving the compensation is by placing a PC trace at one end of R_5 to obtain the equivalent C_C. A properly compensated attenuator will reduce the effective capacitive loading seen at the Tap to 1/4 of that seen at RXT (pin 16).

The ratio of the attenuator does not affect the collision detection threshold (V_{COT} specification in data sheet); it only affects the carrier threshold (V_{CAT} specification in data sheet) at the coax.

Figure 6 shows the attenuator section of the Am7996. At low frequencies (e.g., DC) with R_6 = 24.9K and R_5 = 75K, a 4:1 attenuator is achieved (1/4 V_{COAX} is added to V_{RXT}). With high input impedance at RXT, the series combination of R_5 and R_6 ensures that the resistance to the coaxial cable is at least 100 kOhms as specified by the IEEE 802.3.

At high frequencies (5 or 10 MHz), the parasitic capacitance across R_5 and R_6 determines the attenuator ratio. Therefore, R_5 must have a capacitance (C_C) to compensate for the effective capacitance (C_L) across R_6. For a 4:1 attenuator, $C_C = 1/3\ C_L$ (see appendix C for detail PC board layout considerations).

$$V_{RXT} - V_{COL} = \frac{(V_{COAX} - V_{COL})(1/C_L)}{(1/C_C) + (1/C_L)}$$

$$(V_{RXT} - V_{COL})(1 + C_L/C_C) = V_{COAX} - V_{COL}$$

Let: $y = C_L/C_C$

Then:

Equation 1: $V_{COAX} = V_{RXT}(1 + y) - y(V_{COL})$

For a 4:1 attenuator, y = 3

Note: All voltages are referenced to the Tap shield.

Example: (4:1 attenuator, y = 3)

For: $V_{COAX} = 0$

$V_{COL} = -1600$ mVDC (nominal)

Using Equation 1:

$0 = 4V_{RXT} - 3V_{COL}$

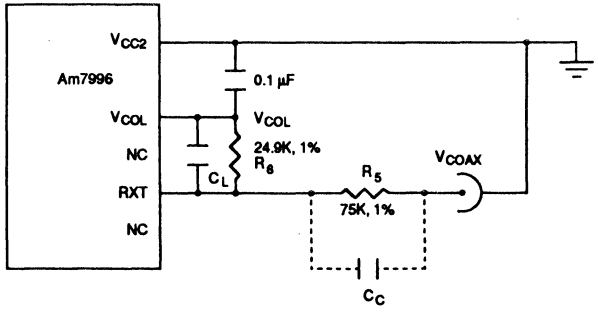
$V_{RXT} = -1200$ mVDC

For: $V_{COAX} = V_{CAT} = -600$ mVDC

Using Equation 1:

$-600 = 4V_{RXT} - 3(-1600)$

$V_{RXT} = -1350$ mVDC



$C_C = (1/y)\,C_L$

FOR A 4:1 ATTENUATOR, y = 3

08031A 6

Figure 6. Am7996 External Attenuator

Attenuator Tolerance

Any deviation from the ideal compensation value for capacitor (C_C) will change the ratio of the attenuator. The attenuator ratio can deviate from the 4:1 ratio as long as the minimum signal level at the coax allowed by the standard can be recognized (see Table 4). Using Ohm's law for Figure 6, the voltage at coax (V_{COAX}) is obtained from equation 1.

The minimum average DC signal at coax can be obtained when two nodes are attached to a maximum length cable segment (500 meters of Ethernet cable or 185 meters of Cheapernet, RG58A/U or RG58C/U). While on node is transmitting from one end of the cable with minimum transmit current, the other node is measuring the attenuated signal received at the other end of the cable.

The external attenuator scheme serves the following two purposes:

1. Reduces the capacitive loading, seen at the coax, by the attenuator ratio (by 1/4 in a 4:1 attenuator). This helps to achieve the tight Tap capacitive loading specification of 2 pF, due to MAU circuitry, per IEEE 802.3, 10BASE5 (Ethernet).

2. Isolates the receive section of the chip (RXT pin) from the medium. It protects the chip which is part of the circuitry of the MAU.

Isolation Diodes D_1, D_2

Another part of the Tap capacitive loading is introduced by the transmit path. The diode D_1, external to the Am7996, is used to isolate the transmit path from the receive path when the chip is not transmitting. The second diode, D_2, protects the first diode, and it further reduces the capacitive loading introduced by the first diode, D_1. D_1 and D_2 are forward biased only when the chip is transmitting. The insertion of the second diode serves two purposes:

1. The capacitance seen at the Tap is reduced to the effective capacitance of the two diodes in series.

2. It provides redundancy in isolating the Tap from TXT pin should one diode get shorted. In Cheapernet applications, the 2nd diode, D_2, may be removed. Cheapernet does not require the redundancy for protection, and the limit for the Tap capacitance loading is not as tight as Ethernet (6 pF versus 2 pF).

   The capacitance introduced by the diode should be as low as possible. Low capacitance switching diodes with adequate current handling capability (80.0 mA nominal) such as the 1N4150 should be used for D_1 and D_2. Am7996 did not integrate the diodes into the chip because of power consideration.

Transmit Signal Wave Shaping (C_3)

C_3 provides wave-shaping for the transmitted signal at TXT (pin12). This 180 pF capacitance between the TXT and TAP SHIELD (pin 12 and 14 includes any parasitic capacitance at the node. A physical capacitor of 150 pF is a nominal value in a typical PC board which takes all the parasitic capacitance into consideration.

The low pass filter at the output stage of the Am7996 is one of the three poles which have been implemented to meet the harmonic content specification of IEEE-802.3 (two poles are internal to the Am7996). The RC components of the low pass filter are outside the Am7996. C_3, the combined resistance of D_1 and D_2, the 25 Ohm load presented by the coax line, and R_7 form the third pole of the TXT output filter.

The time constant for the low pass filter is: T = R x C where:

R is the total resistance seen between the Tap and the shield.

R = Current limiting resistor, (R_7 = 9.09) + Forward biased resistance of the diodes (about 2 Ohms) + 25 Ohms load

C = C_3 = 180 pF (including any parasitic capacitance)

T = 8 ns; Fundamental frequency of the filter is 20 MHz.

C_3 will have some effect on the rise and fall time of the transmit signals at the coax. The rise and fall time values can be improved (reduced) by reducing C_3. There is a limit to how much the capacitor value can be reduced without violating the harmonic content specification. A 150 pF capacitor used for C_3 in the Am7996 evaluation board meets the rise/fall time, and harmonic content specification.

Some attenuation of 10 MHz signals, relative to the 5 MHz signals, is due to the low pass filter implementation. The attenuation does not cause a problem and transmit level signals meet the IEEE-802.3 specifications.

Table 4. IEEE 802.3 Receive Mode Collision Detect Threshold

Application	No Detect (Average DC)	Must Detect (Average DC)
10BASE5 (Ethernet)	–1.492 V	–1.629 V
10BASE2 (Cheapernet)	–1.404 V	–1.581 V

The harmonic content specified by IEEE-802.3 is as follows:

2nd and 3rd harmonics:	at least 20 db below fundamental
4th and 5th harmonics:	at least 30 db below fundamental
6th and 7th harmonics:	at least 40 db below fundamental
All higher harmonics:	at least 50 db below fundamental

Network Protection (D_3 and R_7)

Am7996 offers a solution for network protection (see also the jabber function for protection). It is protected against high voltage surges when the clamp diode (D_3) and the resistor (R_7) are placed external to the chip. The diode protects the chip and the resistor limits the current to protect the diode.

If the Tap is at a positive voltage due to a fault condition, D_3 protects the TXT (pin 12) from sinking high currents from the Tap by shunting high current into ground. Under this condition, R7 (9.09 Ohms, 1/4W) helps limit the current through D_3. D_3 should have a rating of at least 50.0 Volts. A diode such as the 1N4001 can be used for D_3. The capacitive effect of diode D_3 should be taken into consideration as part of the total capacitance (180 pF) between TXT (pin 12) and TAP SHIELD (pin 14).

Set Transmit Current (R_8 = 9.09 Ohm)

This resistor is used to set the transmit output current at TXT (pin 12) nominally at 80.0 mA peak. If a redundant jabber controller is used externally, the supply to the current source comes from the jabber controller. This resistor should be placed as close to the chip as possible, to minimize any parasitic inductance.

Coax Collision Reference Threshold (C_2=0.1 μF)

V_{COL} is a DC reference for incoming signals from the Tap at RXT (pin 16). It is required that Vcol be a good analog signal ground in the presence of 10 Mbps Manchester data streams. In order to achieve that, C_2 is used to by-pass all the RF signals to Tap ground.

Additional Pins

Pins 17 and 15 on the Am7996 have been purposely allocated as No Connect pins on either side of the RXT Pin to give a minimum adjacent pin capacitance. The low RXT input capacitance, combined with any parasitic capacitance due to the resistor and PC trace, is reduced to 1/4 when measured at the Tap. This feature makes it feasible to meet the low input Tap capacitance required by the Ethernet specification. The input Tap capacitance in the Am7996 at RXT (pin 16) is 1.1 pF (typical) for plastic packages, and 1.7 pF (typical) for ceramic packages. Note that the input capacitance at RXT seen at the coax Tap is reduced to 1/4 th through the 4:1 attenuater, external to the chip.

Power (V_{EE}) Requirements

The Am7996 requires a single power supply at –9 ± 10% V. The IEEE-802.3 requires that the power to the MAU must be isolated from DTE. This indicates either the use of a dedicated power supply or isolating the power from DTE through a discrete or commercially available DC–DC converter. In summary, the power requirements are as follows:

- Power to the chip must be isolated from DTE to meet the high voltage isolation required by IEEE-802.3, 10Base5 (2000 VRMS) and 10Base2 (500 VRMS).
- V_{EE} = –9.0 V ± 10%
- Not more than 100 mV (P–P) ripple.
- Ripple frequency less than 100 KHz.

DC–DC Converter Recommendations: Refer to Table 5 for the DC–DC converters (or equivalents) that can be used with the Am7996.)

Isolation Transformers

The AUI transmit, receive, and collision signal pairs

Table 5. DC-DC Converters for Am7996

	APPLICATION	
Manufacturer	IEEE-802.3, 10BASE2 (Cheapernet)	IEEE-802.3, 10BASE5 (Ethernet)
Reliability Inc.	2VP5U9	2E12R9
Pulse Engineering Inc.	PE64381	PE64430

(DO±, DI±, and CI±) must be isolated through transformer coupling from the AUI cable. The isolation at the MAU side is required for two reasons:

1. To eliminate the common mode difference between the signals from the Am7996 and the DTE.

2. To protect the MAU from the fault conditions at the AUI cable.

In the Am7996, the inductance of the transformers for the AUI differential pairs DI±, DO±, and CI± should not be less than 75 µH.

The pin assignments for the transceiver cable are given in Appendix A.

Pulse Transformer Recommendations:

PE64102 (75 µH with 500 VRMS, IEEE-802.3, 10BASE2).
PE64107 (75 µH with 2000 VRMS, IEEE-802.3, 10BASE5) or equivalent.

MEASUREMENT TECHNIQUES

The following are guidelines for measuring some of the key parameters. For the actual test measurements and conditions, refer to the Am7996 data sheet and pertinent test documentation.

4:1 Attenuator Compensation

There are two ways to compensate for the parasitic capacitance across the 24.9 kOhm resistor:

1. By directly measuring the capacitance across the 24.9K resistor (with all the components mounted) and inserting a capacitor whose value is 1/3 of that capacitance across the 75K resistor. The compensation capacitance may also be achieved through the parallel PC traces. The 2nd technique eliminates the need for an additional capacitor across the 75K resistor.
2. Using a *high input impedance (low capacitance)* scope probe. This technique is accomplished through the observation of the signal at the coax Tap and at RXT pin (pin 16) as follows: Apply a square wave signal (0 to –2 Volts) to the coax Tap and observe the signal at the RXT pin. The signal at the RXT pin should be 1/4 of the signal at the coax. The signal at RXT should be somewhat underdamped to compensate for the scope probe 1–2 pF capacitance loading (see Figure 7).

   If a high input impedance (low capacitance) scope probe is not available, compensate for the capacitance of the available probe (the probe used for the RXT pin, pin 16) by adding a capacitor across the 75 kOhm resistor. For example, if the scope probe capacitance is 12 pF, place a 4 pF capacitance across the 75 kOhm resistor. The compensation for the scope probe capacitance ensures that any capacitance added across the 75 kOhm resistor for compensation is independent of scope probe loading.

   Once the scope observation shows the correct 4:1 ratio for the attenuator (see Figure 7), the added capacitance across the 75 kOhm (excluding the 4 pF added capacitance in the above example) is the compensation capacitance that can be added by a physical capacitor or through PC layout.

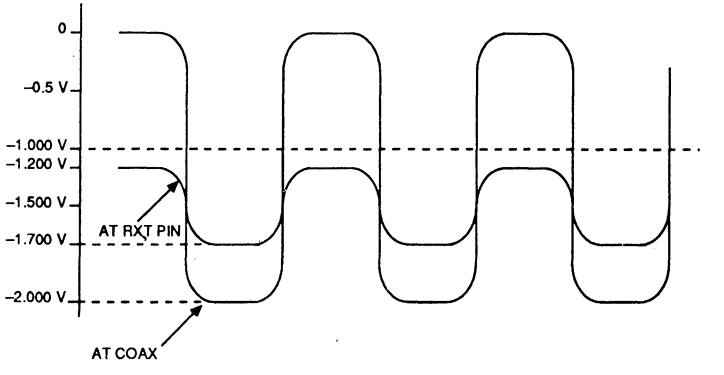


Figure 7. Attenuator Compensation Technique

NOTE.

Once the correct compensation capacitance has been determined for the prototype PC board, no more tuning should be required on the PC board when in production. The RXT input capacitance stays at what has been typically specified (1.7 pF ceramic, 1.1 pF plastic).

Coax Rise and Fall Time Measurements

The rise and fall times are specified at 25 ± 5 ns in the IEEE-802.3 standard, as well as in the Am7996 data sheet. The 10 Mbps Manchester encoded signals carry 10 MHz (all 1's or 0's) and 5 MHz (alternative 1's and 0's) signals due to the nature of the Manchester encoding. Therefore, the rise and fall time measurements should be performed at both 10 and 5 MHz as follows:

Using an IEEE-802.3 controller (e.g., Am7990, an IEEE-802.3 packet generator), send a packet which contains a series of 1's or 0's. Measure the rise and fall time in the data portion (all 1's, or 0's) for 10 MHz signals, and use the preamble portion (1010...) for 5 MHz signals. Adjust the maximum and the minimum peaks of the signal, using the vertical calibrator vernier of the oscilloscope, to form a vertical 0 to 100 grid. Using a small scale time base (e.g., 5 ns/div), measure the rise time from 10% to 90% of the signal. Note that the Am7996 transmit 10 MHz signals are attenuated somewhat relative to the 5 MHz signals due to the output stage low pass filter (the third pole) which is designed to meet the harmonic content specification of the IEEE-802.3 standard. Before measuring the 10 MHz signals, the 0 to 100% levels must be readjusted from the 0 to 100% levels of the 5 MHz signals.

The output characteristic of the low pass filter is such that an improvement can be accomplished in the rise and fall time and the rise/fall time mismatch by reducing the margin for harmonic content specification. The rise and fall time and mismatch can be reduced (improved) in value by reducing the value of C_3. C_3 can also be totally removed if the harmonic content specification is not a concern. In general, the rise and fall time mismatch directly affects the coax transmit jitter budget. The more mismatch, the more jitter will be induced on coax transmit signals.

Transmit Jitter Measurements

Jitter is the displacement of a signal transition relative to where it would ideally be placed as defined by the clock of the encoder. This displacement can be in either direction of the signal transition. The jitter can be measured for two cases:

Bit Cell Center (BCC)
Bit Cell Boundary (BCB)

BCC designates where ideally a clock transition takes place in a Manchester bit cell. The direction of the clock transition represents the value of the data. BCB designates where ideally a signal transition takes place to indicate an end of a bit cell and start of the second bit cell whose value is the complement of the previous one. The transmit jitter is the amount of the jitter which can be introduced to the coaxial cable by the MAU circuitry when transmitting. The IEEE-802.3 and the Am7996 data sheet call for 2 ns (max) (the data sheet parameter is Tskew).

One way of measuring the jitter is to produce an "eye pattern" for a bit cell center and bit cell boundary. An eye pattern can be produced (using the scope trigger control) by transmitting a packet which contains random data. As shown in Figure 8, the jitter can be measured by measuring the time between the 50% crossings of the signals which overlap each other.

Receive Jitter Measurements

The Receive jitter is the amount of jitter introduced differentially at the AUI side of the MAU at the DI± pins. The same method of jitter measurements used for measuring transmit jitter, by producing an "eye pattern" differentially at DI±, can be used for receive jitter measurements.

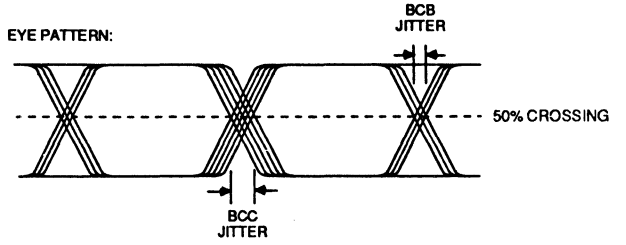


Figure 8. Jitter Measurement

The jitter measurements can be performed under 2 cases: jitter at near end and the jitter at far end. The jitter at near end is the amount of jitter which the MAU under test can produce differentially at DI± when an adjacent MAU (at a distance of 0.5 meter of RG-58, or 2.5 meter of Ethernet cable) is transmitting. The jitter at far end is when a MAU at the far end of the cable (185 meter of RG-58, or 500 meter of Ethernet cable) is transmitting. The latter case is the jitter measurement under the worst case conditions.

Collision Oscillator Frequency

The collision oscillator frequency can be measured at COLL OSC pin (pin 19). R_4 and C_1 are the RC components of the oscillator. C_1 can be adjusted for a 10 MHz nominal frequency. In the Am7996 evaluation board, a 39 pF capacitor was used to obtain a 10 MHz frequency.

APPLICATION EXAMPLES

Ethernet (IEEE-802.3, 10BASE5)

In an Ethernet application, the transceiver module (MAU) resides outside the DTE. The Ethernet coax cables run through the ceiling where the transceiver module is tapped on to it. The transceiver is linked to the DTE through a relatively flexible cable. This cable is a bundle of twisted pair wires, shielded individually, which carry the differential signals to and from the DTE and MAU. The power to the MAU is also carried through this cable. In the standard, this cable is known as Attachment Unit Interface (AUI) cable, and commercially is known as transceiver (or drop) cable. The AUI cable can be up to 50 meters long. In an Ethernet application, the Am7996 resides in the MAU, and Am7990/7992B reside in the DTE.

In Ethernet implementations, where the transceiver section resides outside the board with up to 50 meter AUI cable, there are two pulse transformers: One at the SIA side to protect the SIA, and one at the transceiver side to protect the transceiver against high voltage surges.

Cheapernet (IEEE-802.3, 10BASE2)

In the Cheapernet application, the MAU normally resides within the DTE, and the AUI cable is optional. In a typical Cheapernet controller board, the chip-set (Am7990, Am7992B, and Am7996) resides on the same board within the DTE.

Figure 9 shows a typical Cheapernet implementation using Am7990, Am7992B, and Am7996 (also refer to AMD stand alone Ethernet/Cheapernet evaluation board). For a detailed discussion of the external Ethernet/Cheapernet components to the Am7996, refer to the External Component Design Guidelines presented earlier in this application note. Most of the Cheapernet implementations also make provision for supporting the Ethernet connection. This is done by routing the ECL differential signals to the Ethernet D connector, bypassing the transceiver (Am7996), as shown in Figure 9. Note that only one pulse transformer per twisted pair is required between the Am7992B and Am7996 in Cheapernet implementations. The pulse transformer is needed to isolate the positive common mode levels of the SIA (Am7992B) from the Am7996 (which has negative common mode levels).

The SIA external components at pin 5 should be configured as shown in Figure 6 for half-step signaling when used in IEEE-802.3 applications. Ethernet Version II specification makes it optional, allowing either half-step or full-step signaling for connection with transceivers. Most of the current transceiver modules support the half-step signaling which is recommended by the IEEE-802.3 standard.

The Am7992B SIA generates negative narrow spikes (less than 10 ns, within 200 mV) every time RCLK (pin 4) is running. The spikes are due to the RC circuitry around TSEL (pin 5). The TSEL pin is an open collector output and a sense amplifier input. The gain of the amplifier is about four. The RC circuit controls the decay of the last positive transition (end of the packet) at Transmit± when half-step signaling is used (TSEL is grounded for full-step signaling).

The positive transitions on RCLK couple capacitively with the adjacent pin having the 510 Ohm pull up to V_{CC}. This noise is then amplified and appears as spikes at Transmit±. The problem is significant only when the SIA is receiving a packet (RCLK active) or when TEST is grounded (continuous RCLK). Any false signals at Transmit± meeting the amplitude and pulse width requirement can wake up the transceiver causing a collision to occur. Usually, the spikes are too narrow to wake up the transmitter section of the transceiver. As an extra precaution, a 20 pF or higher (50 pF if spikes are more than 200mV) capacitor across 510 Ohms can reduce the spikes significantly.

Consider the 40.2 Ohm resistors at the SIA side receive± and collision±, and at the transceiver side DO±, (see Figure 9). One may think that there is no need for the terminating resistors since no AUI cable is normally used in Cheapernet applications. True, there is no AUI cable; however, the resistors at DO± also form part of the output load when the SIA is driving transmit±,

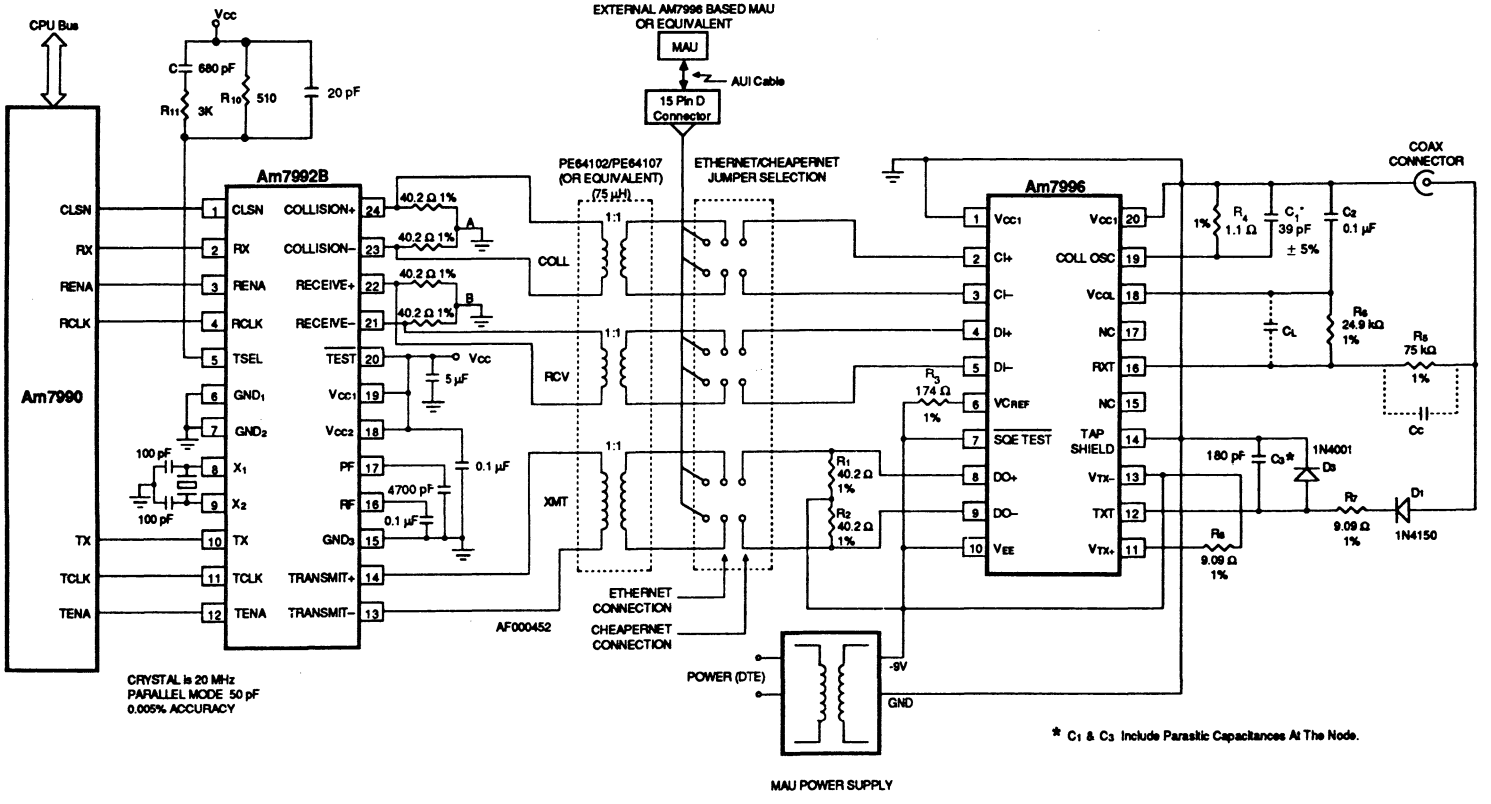


Figure 9. Am7996 Cheapernet Chip-set (Am7990, Am7992B, and Am7996) Interconnection Diagram

and the resistors at collision± and receive± form part of the output load when the transceiver is driving its CI± and DI± outputs. If these resistors are removed, the SIA and transceiver will still be functional; however, distortion on SIA/transceiver interface is more likely to occur. It is recommended that the 40.2Ω resistors be used as specified.

Repeater Design

A transceiver module is designed to drive either one Ethernet or one Cheapernet cable segment. There can be up to five cable segments as specified by the standard. Repeaters are used to link the cable segments together so that all the nodes attached to different cable segments can communicate with each other. The main function of the repeater is make all the isolated cable segments appear as one single cable. The repeater restores the energy of the signal to permit driving another cable segment. Refer to Figure 10 for a block diagram of a repeater and also to Figure 3 for a network configuration using repeaters.

There are two sides to the repeater: One side of the repeater is attached to one cable segment; the other side is attached to another cable segment. A repeater should transfer the messages across regardless of the address or the data contents of the packet. When a collision is detected on any side to which the repeater is transmitting, the repeater transmits a Jam (1010 pattern) to both sides of its connection. This mechanism ensures that the collision is recognized by all the nodes on both cable segments connected via the repeater.

When using the Am7996 in a repeater application, the external component diagram, Figure 4, should be used. In this configuration, the collision detection threshold is adjusted for receive mode collision detection, and the SQE test is inhibited.

Re-Generative Repeaters: There are two types of repeaters: re-generative and non-generative. The re-generative repeaters re-generate the 64 preamble bits (including the sync bits) normally within the frame. When a packet arrives (data carrier is detected), the repeater starts sending preamble bits to the other side while searching for the sync bits from the receiving end. When it finds the sync bits, it stores the data (the bits following the sync bits) until it is finished sending the preamble bits. Then it immediately starts sending the data. The preamble duplication mechanism ensures that any lost preamble bits, within the receiving packet, are restored before the data reaches the other cable segment.

Non-Generative Repeaters: The non-generative repeaters just repeat the signals without adding any preamble bits to the frame. The drawback with this type of repeater is that the lost preamble bits (due to the cables, transceivers, and other repeaters along the path) are not restored. Fewer preamble bits will leave less margin for the PLL decoders to lock into the Manchester data. It may take as many as 12 preamble bits for the PLL to acquire the clock in some of the VLSI decoders. The Am7992B (SIA) PLL acquires the clock in only 4 bit times.

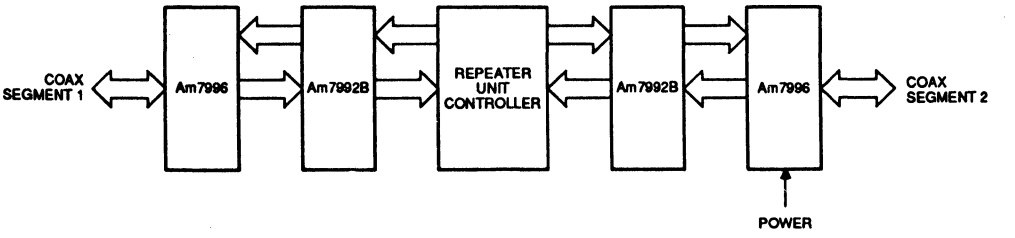


Figure 10. Typical Repeater Unit Block Diagram

08031A 11

APPENDIX A

TRANSCEIVER CABLE PIN ASSIGNMENTS (IEEE 802.3)

Pin	Circuit	Description	Use
1	CI-S	Control In circuit Shield	Coll shield
2	CI-A	Control In circuit A	Coll –
3	DO-A	Data Out circuit A	Xmit (DO) +
4	DI-S	Data In circuit Shield	Rx Shield
5	DI-A	Data In circuit A	Rx (DI) +
6	Vc	Voltage Common	12 V ground
7	CO-A	Control Out circuit A	–option–
8	CO-S	Control Out circuit Shield	–option–
9	CI-B	Control In circuit B	Coll +
10	DO-B	Data Out circuit B	Xmit (DO) –
11	DO-S	Data Out circuit Shield	Xmit shield
12	DI-B	Data In circuit B	Rx (DI) –
13	VP	Voltage Plus	+ 12 V
14	VS	Voltage Shield	DTE ground
15	CO-B	Control Out circuit B	–option–
Shell	PG	Protective Ground	Chassis

ETHERNET AUI CABLE

Pin	Description	Use
1	Shield	Chassis
2	Collision Presence	Coll +
3	Transmit +	Xmit (DO) +
4	Reserved	NC
5	Receive +	Rx (DI) +
6	Power Return	12 V ground
7	Reserved	NC
8	Reserved	NC
9	Collision Presence	Coll –
10	Transmit –	Xmit (DO) –
11	Reserved	NC
12	Receive	Rx (DI) –
13	Power	+ 12 V
14	Reserved	NC
15	Reserved	NC
Shell		Chassis

APPENDIX B

RELATED HARDWARE SUPPORT

There are two different types of evaluation boards which integrate the Am7996. They are:

- Am7996 evaluation board (PN Am7996EVAL)
- Ethernet-Cheapernet Low Lost AT Board (PN ETHNEVAL5)

RELATED DOCUMENTATION

- Am7990 Family Reference Guide #03394
- Local Area Network Controller Am7990 (LANCE) Technical Manual #06363
- Interfacing the Am7990 to 7422 8-bit Microprocessors
- Am7990 (LANCE) Data Sheet #05698
- Am7992B (SIA) Data Sheet #03378
- Am7996 (Tranceiver) Data Sheet #07506

APPENDIX C

PC BOARD LAYOUT CONSIDERATIONS

To protect the transceiver from the environment and to achieve optimum performance, the Am7996 is designed to be used with two sets of external components: the transmitter circuit consisting of components D1, D2, D3, R7, R8, and C3, and the receiver circuit consisting of components R5, R6, C_L, and C_C (C_L is a parasitic capacitance rather than a discrete component). These two circuits are shown in both Figure 4 and in Figure 5. The resistor tolerances for these circuits are specified as 1% for temperature stability.

The only layout restriction for the transmitter circuit is that the longest current path from the TXT pin (pin 12) to the coaxial cable's center conductor must be no longer than 4 inches.

The layout of the receiver circuit, however, is critical. to minimize parasitic capacitance that can degrade the received signal, the external receiver circuit should be isolated from power and ground planes. There must be no power or ground plane under the area of the PC board that includes pins 15 through 20, R5, R6, and the connector for the coaxial cable. If a power or ground plane extends under this area, the receiver will not function properly due to excessive crosstalk and under- or over-compensation of the R5–R6 attenuator. Also, the RXT pin (Pin 16) should be as close to the coaxial cable connector as possible.

Since there are no severe layout restrictions on the transmitter circuit, the layout can be simplified by omittiing power and ground planes from the whole area on the right side of the Am7996 as shown in Figure11.

If the above layout rules are followed, the parasitic capacitance in parallel with R6 will be about 6 pF. This parasitic capacitance is shown in the schematics as C_L. (Note that C_L is a parasitic capacitance. Do not add a discrete capacitor in parallel with R6.) The capacitor labelled C_C in the schematics is the total capacitance in parallel with R5 including parasitic capacitance. The parasitic component of C_C will be about 1 pF. For optimum performance, the ratio of C_L to C_C should be the same as the ratio of R5 to R6, which is 3 to 1. This means that an additional 1 pF of capacitance must be added in parallel with R5.

This extra 1 pF of capacitance can easily be added by building a parallel-plate capacitor from PC traces right under resistor R5. This capacitor can consist of a 0.200 inch by 0.200 inch square of conductor on each side of the board as shown in Figure 12. (These dimensions assume that the PC board is made from 0.060 inch thick G-10 material.) The top plate of the capacitor should be connected to the one lead of R5, and the bottom plate should be connected to the other lead. Figure 13 shows an example of the suggested layout for a 4-layer printed circuit board. Note that the component labeling used in Figure 13 is not intended to correspond with the component labeling used in Figure 4 and Figure 5.

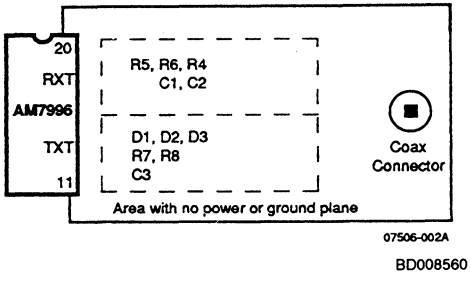


Figure 11. PC Board Outline

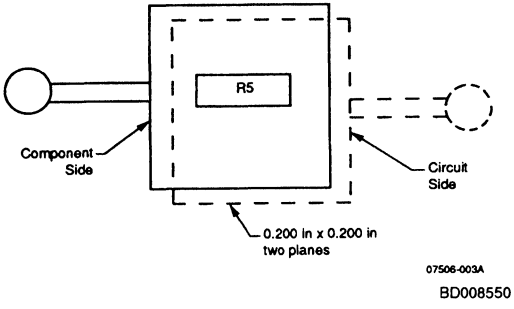
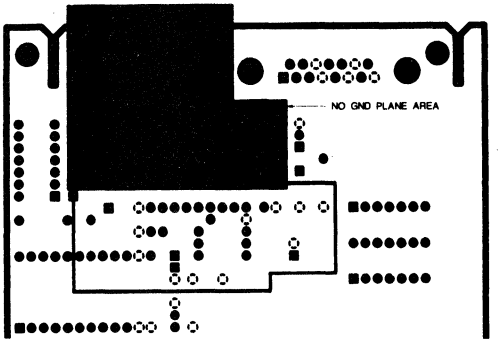
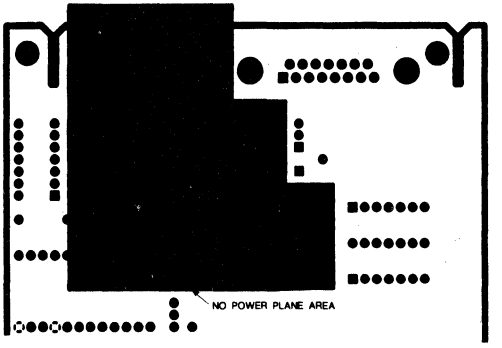


Figure 12 PC Board Trace Capacitor



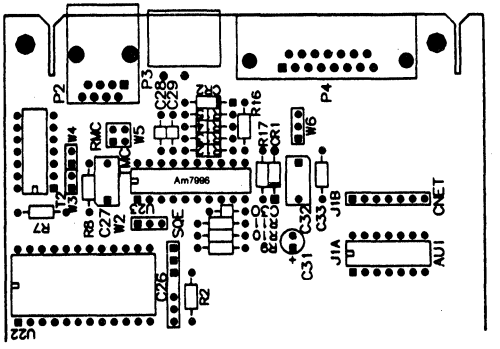
CD012021

GND PLANE

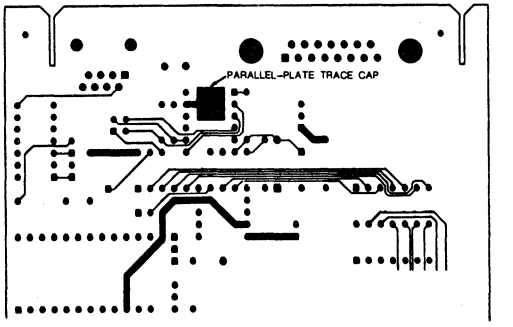


CD012011

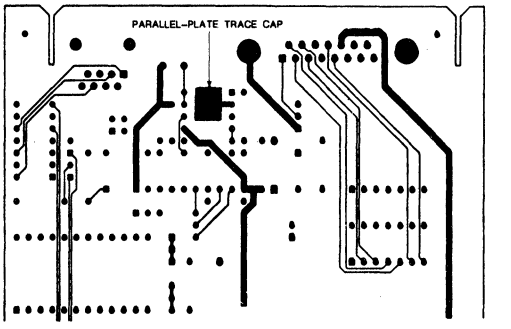
POWER PLANE



CD012001



CD011981



CD011991

C26	CAP-0.01UF
C27,28,29,30	CAP-0.1UF
C31	CAP-4.7UF
C32	CAP-150PF
C33	GAP CAP-0.001UF
CR1	DIODE-1N4150
CR2	DIODE-1N4001
P3	BNC
P4	15-PIN D SHELL
R2	RES-1M
R8	RES-1.1K
R9	RES-40.2
R10	RES-40.2
R11	RES-174
R12	RES-499
R13	RES-150K
R14	RES-24.9K
R15	RES-75K WITH TRACE CAP
R16	RES-9.09
R17	RES-9.09

Figure 13. Suggested Printed Circuit Board Layout for a 4 Layer PCB Application

SECTION

3 DEVELOPMENTS TOOLS

FINAL

C-LANCE-AT-KT

Am79C90 Based Ethernet/Cheapernet/Twisted Pair Half Card Evaluation Kit for PC/AT™

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **C-LANCE-AT-KT/2 integrated with 10BASE2 transceiver and AUI port**
- **C-LANCE-AT-KT/2T integrated with 10BASE-T transceiver and AUI port**
- **Implements a working Ethernet node using an IBM PC/AT or compatible as host**
- **Complete solution utilizes CMOS Local Area Network Controller for Ethernet (C-LANCE) high performance, low-cost bus master architecture**
- **Supports the following types of network interface:**
  - — Standard AUI port for external 10BASE2, 10BASE5, or 10BASE-T MAU connection
  - — On board transceiver for Cheapernet connection (C-LANCE-AT-KT/2 only)
  - — On board transceiver for Twisted Pair connection (C-LANCE-AT-KT/2T only)
- **Evaluation software allows the system designer access to register level functions, thereby facilitating quick development of custom software drivers**
- **100% compatible with Novell NE2100 card**
- **Includes driver object code only for NetWare™ ODI, NDIS 2.01, NDIS 3.0, Artisoft® LANtastic/AI™, Streams (for SCO UNIX), and Packet Driver**

GENERAL DESCRIPTION

The Ethernet/Cheapernet/IEEE 802.3 evaluation kit, called the C-LANCE-AT-KT (Am2100 architecture), is a design evaluation vehicle for AMD's Ethernet chipset. It is intended for use in IBM PC/AT or compatible machines, and represents a low component count, minimum board space, low cost network adapter implementation. There are two versions of the card available for evaluation, each optimized for a particular medium. The C-LANCE-AT-KT/2 card supports the 10BASE2 (Cheapernet) connection, while the C-LANCE-AT-KT/2T card supports the 10BASE-T (Twisted Pair) connection. Both versions have a standard AUI port (fully IEEE 802.3, Section 7, compliant) for connecting to 10BASE5 (Ethernet) medium through an external MAU. The choice of network media operation is jumper selectable.

Despite the card's low cost, it is still a high performance design that takes full advantage of the PC/AT multimaster I/O bus. When installed in a host machine, the system becomes a platform upon which to evaluate network hardware and to develop software for a complete node processor. The software designer can take advantage of the many tools available for the PC/AT-compilers, assemblers, and debuggers.

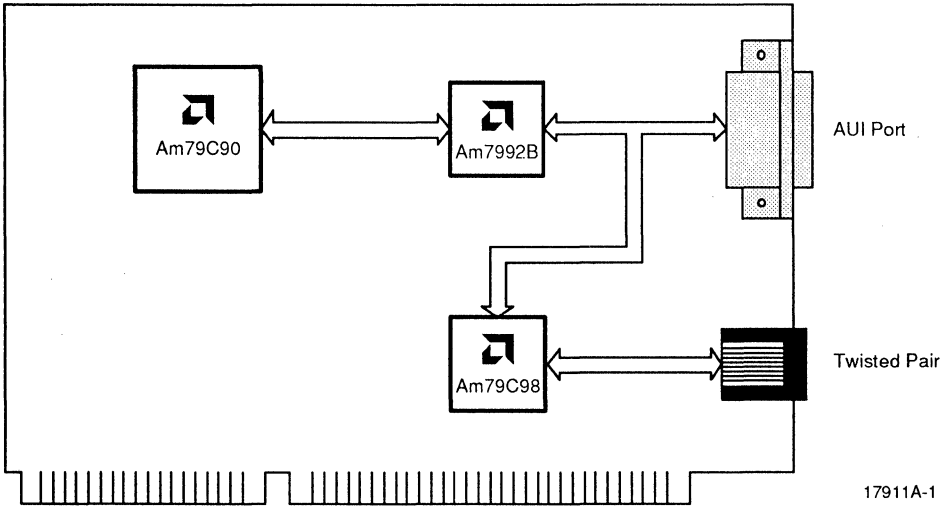
The C-LANCE-AT-KT comes with a floppy diskette of software programs. The software includes a high-level demonstration program, a low-level driver/monitor, Packet Driver (version 10.3), driver object code for NetWare ODI, NDIS 2.01, NDIS 3.0, Artisoft LANtastic/AI, and Streams (for SCO UNIX). The demonstration program contains an ISO data link layer with a menu driven interface which allows the user to assign physical and logical addresses, establish connections, and send and receive messages. The driver/monitor lets the user view and change the contents of the C-LANCE's registers, the memory resident Initialization Block, and the data buffer Descriptor Rings. The program also allows the designer to establish loops for hardware data probing. The Packet Driver (source code and object code) is included as a sample C-LANCE driver in 80x86 assembly language. The Novell NetWare object code, as well as the other drivers, transforms the C-LANCE-AT-KT into a cost effective, competitive, and manufacturing-ready commercial board level product. This allows the user to evaluate the performance of the AMD Ethernet chipset in a real PC LAN system.

In addition to the board and the software diskette, the C-LANCE-AT-KT kit comes with a user's manual including device specifications, device application notes, the Am2100/Am1500T Network Driver Installation Guide, C-LANCE device datasheet, and cable hook-up hardware.

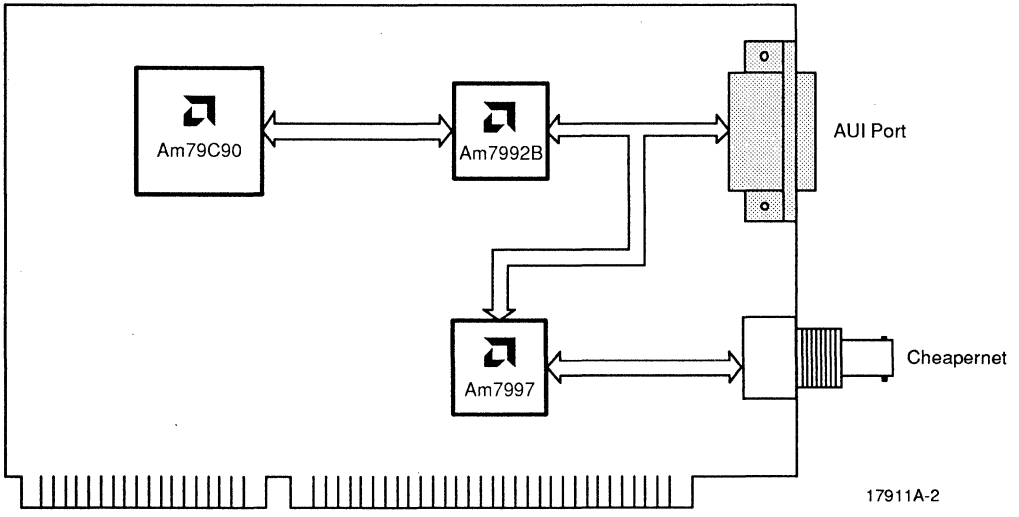
Publication# 17986 Rev. A Amendment/0
Issue Date: May 1993

BLOCK DIAGRAM

C-LANCE-AT-KT/2T



C-LANCE-AT-KT/2



Am7996EVAL-HW

Ethernet/Cheapernet Transceiver Evaluation Kit

Advanced Micro Devices

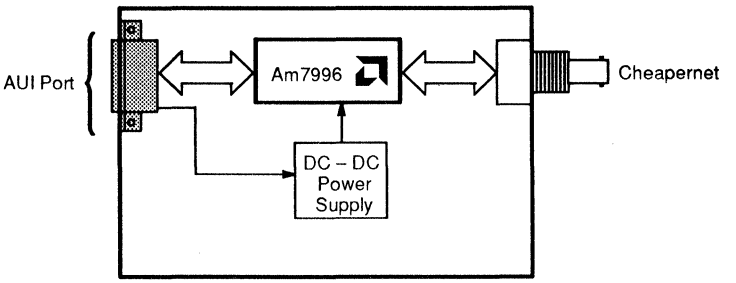
DISTINCTIVE CHARACTERISTICS

- **Connects the standard AUI port interface to 10BASE–2 (Cheapernet) connector.**
- **Jumper selectable Signal Quality Error (SQE) test allows evaluation in both node and repeater applications.**
- **Self contained board facilitates rapid evaluation of the AUI -to-coax system block, and demonstrates optimal analog design criteria.**

GENERAL DESCRIPTION

The Am7996EVAL-HW board is a self contained module that implements all the functions of a Medium Attachment Unit (MAU) for IEEE 802.3 10BASE–2 (Cheapernet). This 3"x3" board can be used to connect any AUI port interface to the 'thin' coaxial cable used in Cheapernet networks. This system permits the network designer to monitor signals on the card to increase his understanding of IEEE 802.3 standards and conventions. A jumper selectable SQE test enable/disable lets the board function in both user node or network repeater applications. In addition, optimal PC board layout is employed. The Am7996EVAL-HW kit comes complete with schematics, application notes, and coaxial cable connection hardware.

BLOCK DIAGRAM



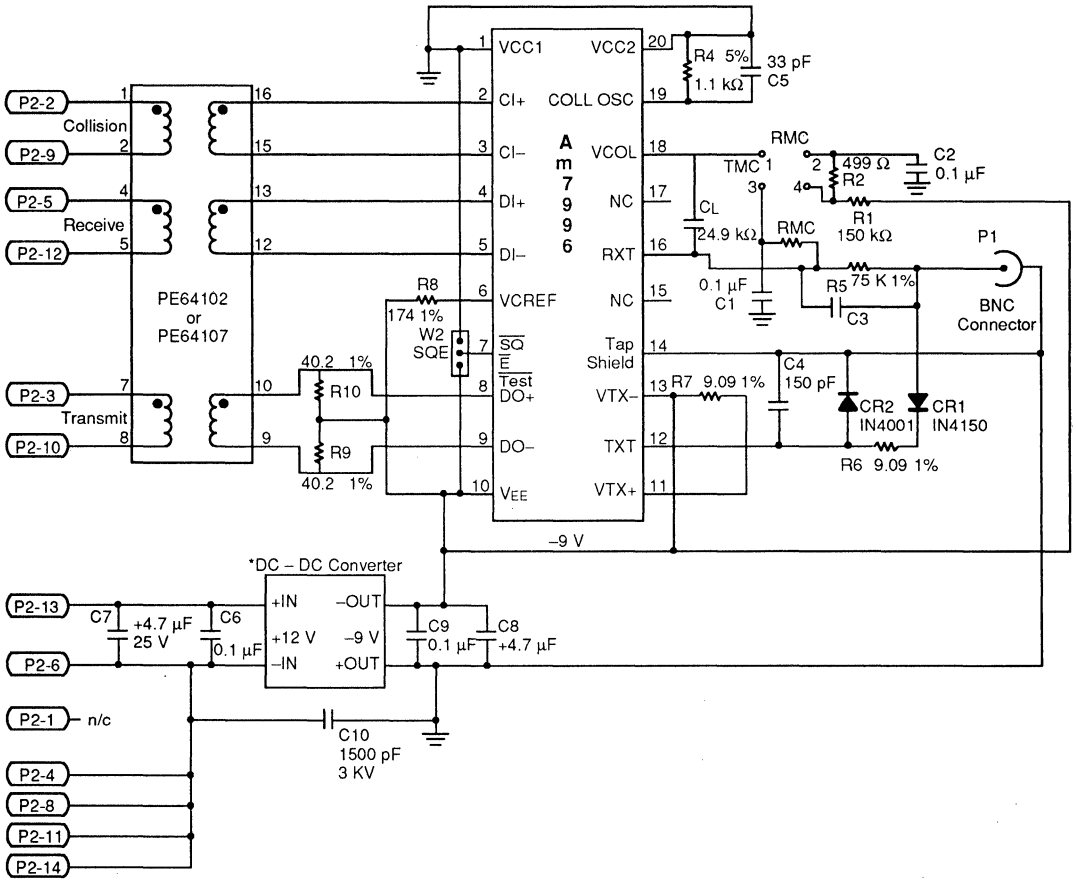
JUMPER SETTINGS

The jumper settings specify the type of collision detect that is used.

Transmit collision detect:	connect 1 & 3
Receive collision detect:	connect 1 & 2 3 & 4

Publication# 14606 Rev. B Amendment/0
Issue Date: January 1992

CIRCUIT DIAGRAM



Notes:

CL = Effective Parasitic Capacitance

C3 = 1/3 CL (PC Trace was Used for C3)

RMC = Receive Mode Collision Detection (R1, R2, and C2 are needed for RMC Function)
Jumper 1 to 2 and 3 to 4

TMC = Transmit Mode Collision Detection
Jumper 1 to 3, open 2 and 4
R8 = 0 (Short)

DC – DC CONVERTER PINOUT

DC/DC Converters	Input = +12 V		Output = –9 V	
	+IN	–IN	+OUT	–OUT
Reliability INC. 2E12R9	1	2	4	5
Pulse Engineering PE64540	1, 2	23, 24	11, 12	9, 10

FINAL

ISA-HUB™-KT

Am79C981 and Am79C987 Based Ethernet Managed Server Hub Card

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Modular, expandable PC ISA/EISA bus compatible server hub card based on AMD's HIMIB™ and IMR+™ devices**
- **Fully compatible with the IEEE 802.3 Layer Management for 10 Mb/s Baseband Repeaters (Repeater Management Standard)**
- **Compatible with the Novell's Hub Management Interface (HMI); HMI driver included**
- **HIMON provides user friendly display and statistics monitoring features, and simple menu-driven diagnostic and configuration capabilities**
- **Designed to be independent of network adapter card used in the file server**
- **Supports eight 10BASE-T ports and one AUI port**
- **Provides Link Status LED for 10BASE-T ports**

GENERAL DESCRIPTION

The ISA-HUB-Kit is designed to serve as a repeater application example as well as an evaluation vehicle for the AMD's IMR+ (Am79C981) and HIMIB (Am79C987) devices. It is a nine port (eight 10BASE-T and one AUI), manageable, expandable 10BASE-T Ethernet Repeater card, and is fully compatible with the IEEE 802.3 Repeater Management Standard. The ISA-HUB card is intended for use in IBM PC ISA and EISA bus compatible machines. The kit includes all the necessary hardware, software and documentation to build a fully managed 9-port repeater. When installed in a file server, the ISA-HUB card, in conjunction with a network adapter card (such as NE2100 or NE1500T), provides direct file server to node connectivity.

The ISA-HUB card is designed to be modular and expandable. Up to eight cards can be cascaded to support 64 10BASE-T ports.

The HIMON (HImib MONitor) program is a menu driven, PC (MSDOS) software program that allows the user to monitor and program various registers in the HIMIB and IMR+ devices.

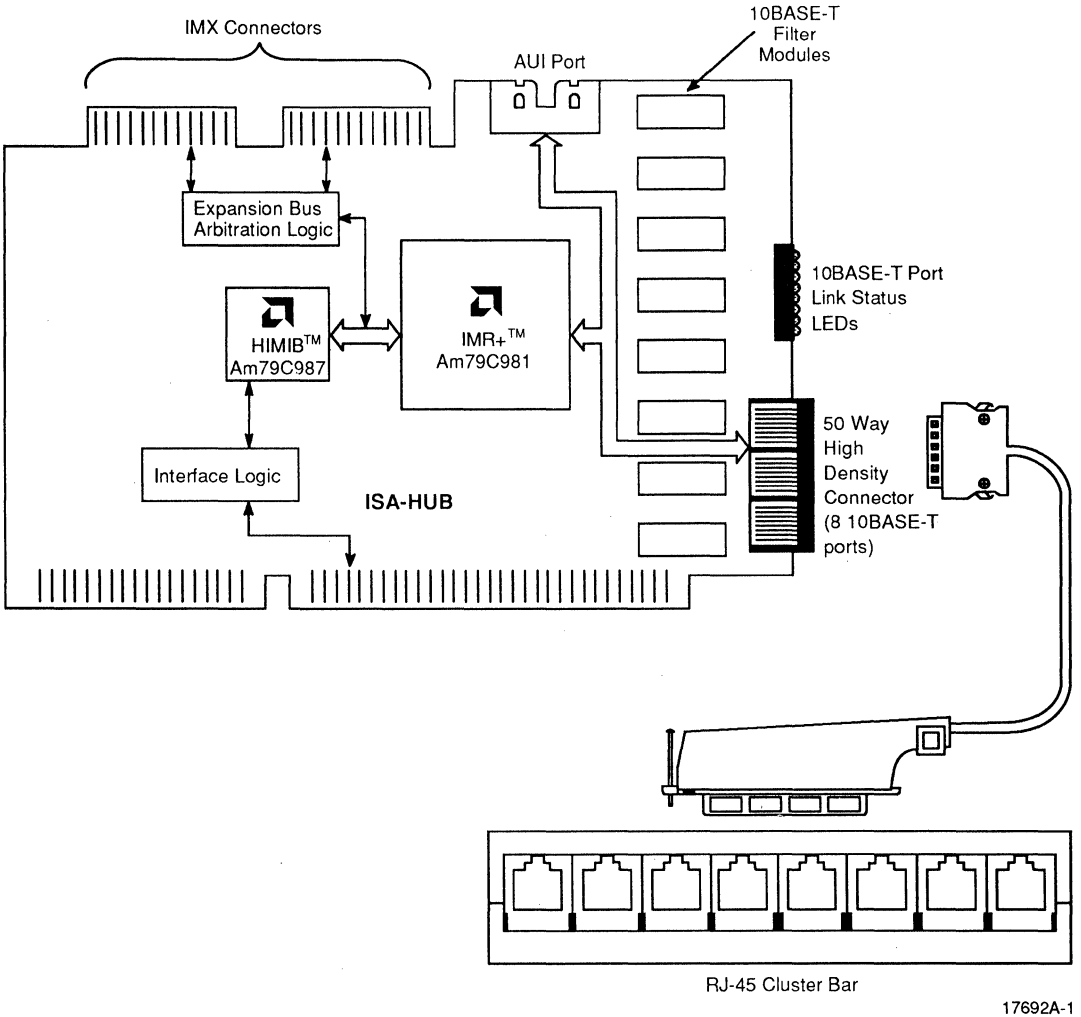
The ISA-HUB card is fully compatible with the Novell Hub Management Interface (HMI) specification. An HMI driver is included as a part of the kit. The card is designed to be independent of the network adapter card used in the file server. File server connection to the network is accomplished by connecting the adapter card to one of the ISA-HUB ports.

The ISA-HUB-Kit is composed of the following items.

- ISA-HUB ISA/EISA bus compatible adapter card
- HIMON evaluation program
- Novell certified HMI compliant driver, object code
- Novell HUBCON Netware Loadable Module (NLM) executable
- User's Guide and Reference Manual
- Telco cable assembly
- RJ45 cluster bar (8 ports)
- AUI connector assembly (includes PC/AT bracket and ribbon cable)
- Inter Module Expansion (IMX) bus ribbon cable and connector assembly
- IMR+ and HIMIB datasheets (PID# 17306A and 17305A)
- AMD's IEEE 802.3 Repeater Technical Manual (PID# 17314A)

Publication# 17692 Rev. A Amendment/0
Issue Date: February 1993

BLOCK DIAGRAM



17692A-1

FINAL

PCnet™-ISA+-KT

Am79C961 Based Evaluation Kits for ISA Systems

Advanced Micro Devices

GENERAL DESCRIPTION

- **Based on the PCnet-ISA+ (Am79C961) jumperless single-chip ethernet controller**
- **Two evaluation kit options available:**
  - PCnet-ISA+-KT/2 with 10BASE-T and 10BASE2 ports
  - PCnet-ISA+-KT with 10BASE-T port
- **Implements a full functional Ethernet node using an IBM PC/AT or compatible as host**
- **Ethernet adapter card utilizing a high performance, low-cost bus master architecture (NE2100)**
- **Supports Microsoft's Plug and Play System configuration for jumperless ISA designs**
- **Supports Look Ahead Packet Processing (LAPP) that allows protocol analysis to begin before end of receive frame for higher performance**
- **Software compatible with all PCnet family members**
- **Software drivers support all popular Network Operating Systems**
- **Includes evaluation board, software driver diskettes and supporting documentation**

DISTINCTIVE CHARACTERISTICS

The PCnet-ISA+ evaluation kit is an evaluation vehicle for the Am79C961 PCnet™-ISA+ single chip Ethernet controller. There are two versions of the kit available for evaluation. The PCnet-ISA+-KT supports the 10BASE-T (Twisted Pair) connection, while the PCnet-ISA+-KT/2 supports both the 10BASE2 (Cheapernet) connection and the 10BASE-T connection on the evaluation card.

The evaluation board, when installed in a PC/AT or compatible host system, provides a platform for demonstrating the high performance of the PCnet-ISA+ device, the low manufacturing cost of a PCnet-ISA+ based solution and overall ease of design. The platform further allows the user to evaluate network hardware and to develop software for an Ethernet node based on the PCnet-ISA+ single-chip Ethernet controller.

In addition to the evaluation board, the kit comes with software diskettes, a hardware user's manual, the *PCnet Family Network Driver Installation Guide*, device datasheet and the *PCnet Family Technical Manual.*

The software includes driver object codes for Novell NetWare ODI DOS and OS/2, NetWare Lite, Microsoft Windows NT, Windows for Workgroups, NDIS 2.0.1, Microsoft LAN Manager, Banyan VINES Client, IBM LAN Server, SCO UNIX, Artisoft LANtastic/AI, DEC Pathworks, and Packet Driver. Also included are three utility programs, one for configuration in Microsoft Plug and Play compatible environments, one for configuration of the network adapter card and software driver installation in non-Plug and Play environments, and one for EEPROM configuration.

In addition, AMD provides a low-level evaluation program to establish connections and to send and receive messages. The evaluation program allows the user to view and change the contents of the PCnet-ISA+ registers, the memory resident Initialization Block, and the data buffer Descriptor Rings. The program also allows the designer to establish loops for hardware probing.

When installed in Microsoft Plug and Play compatible environments, a Plug and Play utility program will automatically configure the board at power up. Automatic configuration of the I/O base address and the interrupt channel occurs upon power up by the utility program. No hardware jumpers are required for configuration. The utility program will also scan and detect for other attached Plug and Play peripherals in the system and configure accordingly.

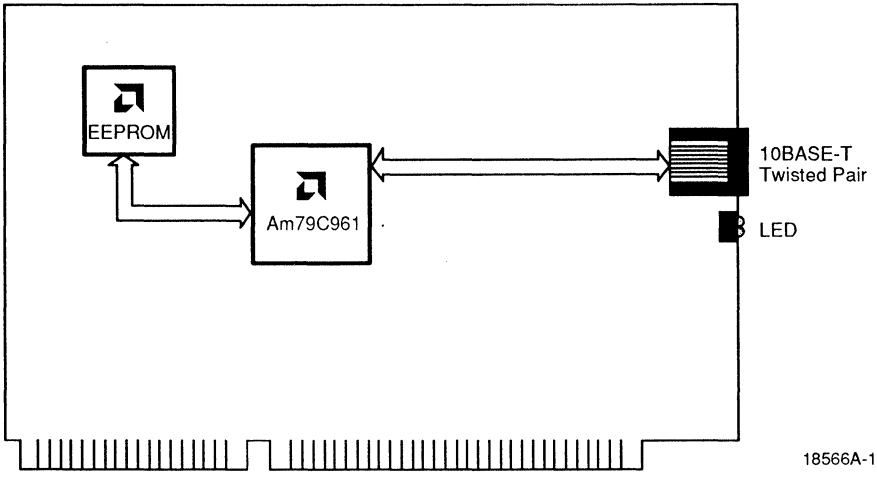
When installed in a non-Plug and Play environment, the PCnet Family Configuration and Installation utility program, AMINSTALL, provides an easy user interface to view the configuration of the PCnet-ISA+ evaluation board. The utility program will automatically scan the system bus(es), which may include ISA, VL, or PCI to find the installed PCnet device. This utility program will identify AMD's PCnet-ISA, PCnet-ISA+, PCnet-32, and PCnet-PCI devices. With the configuration portion of the program, the utility allows the user to select the I/O address, IRQ channel, DMA channel, and Boot ROM address for the PCnet-ISA+ device. After configuration, the user may use the installation portion of the utility to install a selected driver by copying the appropriate driver from the AMD diskette and create or modify the CONFIG.SYS, AUTOEXEC.BAT, and PROTOCOL.INI or NET.CFG files on the user's system.

Publication# 18566 Rev. A Amendment/0
Issue Date: March 1994

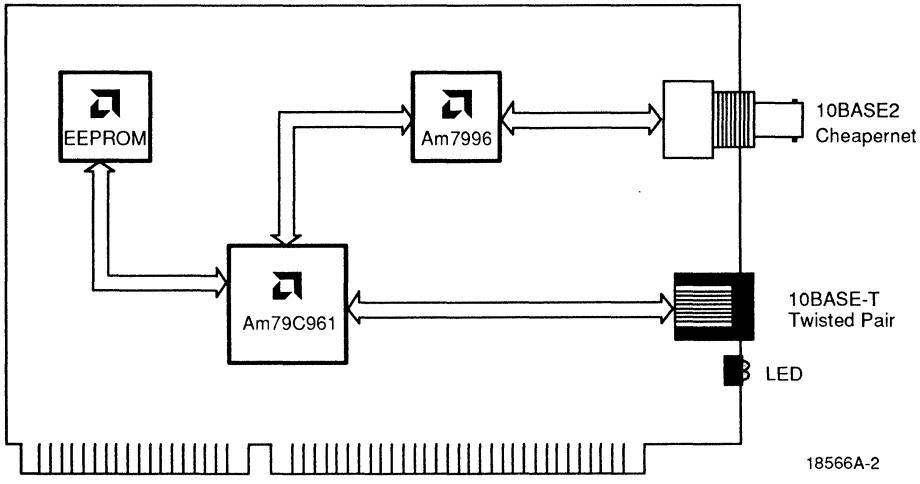
The PCnet-ISA+ evaluation board stores the unique IEEE physical address in the serial EEPROM. Once powered up, the Am79C961 device reads the node's IEEE physical address from the EEPROM through the Microwire interface protocol. The EEPROM utility program allows the user to change the bus configuration options such as burst length, FIFO thresholds, I/O location, etc. For more details about the PCnet-ISA+ single chip Ethernet controller, refer to the *Am79C961 PCnet-ISA+* data sheet (PID# 18183)and *PCnet Family Technical Manual* (PID# 18216).

For more details about other members of the PCnet Family, including PCnet-ISA (Am79C960), PCnet-32 (Am79C965) and PCnet-PCI (Am79C970), refer to their respective datasheets (PID#'s 16907B, 18219B, and 18220B).

BLOCK DIAGRAM
PCnet-ISA+-KT



PCnet-ISA+-KT/2



FINAL

PCnet™-32-KT

Am79C965 Based Evaluation Kits for VESA VL-Bus™ Systems

Advanced Micro Devices

GENERAL DESCRIPTION

- **Based on the PCnet-32 (Am79C965) 32-bit single-chip Ethernet controller**
- **Two evaluation kit options available:**
  - — PCnet-32-KT/2 with 10BASE-T and 10BASE2 ports
  - — PCnet-32-KT with 10BASE-T port
- **Implements a full functional Ethernet node using a VESA VL-Bus system as host**
- **Ethernet adapter card utilizing a high performance, low-cost, bus-master architecture (NE2100)**
- **Software compatible with all PCnet family members**
- **Software drivers support all popular Network Operating Systems**
- **Includes evaluation board, software driver diskettes and supporting documentation**

DISTINCTIVE CHARACTERISTICS

The PCnet-32 evaluation kit is an evaluation vehicle for the Am79C965 PCnet™-32 single chip Ethernet controller. There are two versions of the kit available for evaluation. The PCnet-32-KT supports the 10BASE-T (Twisted Pair) connection, while the PCnet-32-KT/2 supports both the 10BASE2 (Cheapernet) connection and the 10BASE-T connection on the evaluation card.

The evaluation board, when installed in a VESA Local-Bus host system, provides a platform for demonstrating the high performance design of the PCnet-32 device, the low manufacturing cost of a PCnet-32 based solution and the overall ease of design. The platform further allows the user to evaluate network hardware and to develop software for an Ethernet node based on the PCnet-32.

In addition to the evaluation board, the kit comes with software diskettes, a hardware user's manual, the *PCnet Family Network Driver Installation Guide*, and device datasheet and the *PCnet Family Technical Manual.*

The software includes driver object codes for Novell NetWare™ ODI DOS and OS/2, NetWare Lite, Microsoft Windows NT, Windows for Workgroups, LAN Manager, Banyan VINES Client, IBM LAN Server, SCO UNIX, Artisoft LANtastic/AI, DEC Pathworks, and Packet Driver. Also included are two utility programs, one for configuration of the network adapter card and software driver installation, and one for EEPROM configuration.

In addition, AMD provides a low-level evaluation program to establish connections and send and receive messages. The evaluation program allows the user to view and change the contents of the PCnet-32 registers, the memory resident Initialization Block, and the data buffer Descriptor Rings. The program also allows the designer to establish loops for hardware probing.

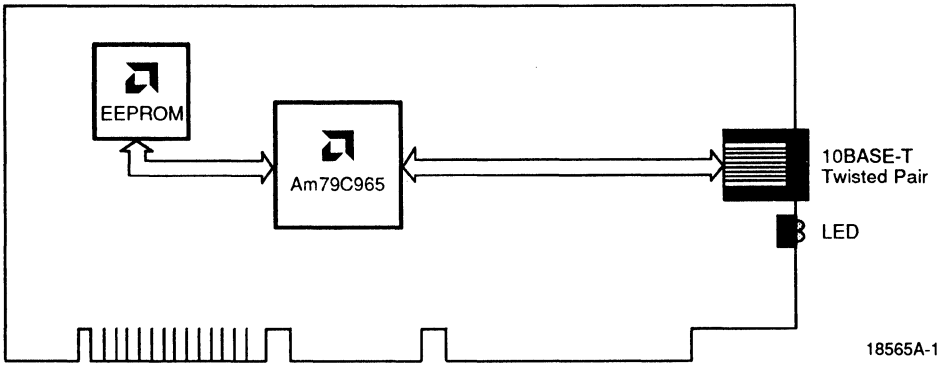
The PCnet Family Configuration and Installation utility program, AmInstall, provides an easy user interface to view the configuration of the PCnet-32 evaluation board. The utility program will automatically scan the system bus(es), which may include ISA, VL, or PCI to find the installed PCnet device. This utility program will identify AMD's PCnet-ISA, PCnet-ISA+, PCnet-32, and PCnet-PCI devices. With the configuration portion of the program, the utility allows the user to select the I/O address, IRQ channel, DMA channel, and Boot ROM address for the PCnet-32 device. After configuration, the user may use the installation portion of the utility to install a selected driver by copying the appropriate driver from the AMD diskette and create or modify the CONFIG.SYS, AUTOEXEC.BAT, and PROTOCOL.INI or NET.CFG files on the user's system.

The PCnet-32 evaulation board stores the unique IEEE physical address in the serial EEPROM. Once powered up , the Am79C965 device reads the node's IEEE physical address from the EEPROM through the Microwire interface protocol. The EEPROM utility program allows the user to change the bus configuration options such as burst length, FIFO thresholds, I/O location, etc. For more details about the PCnet-32 single chip Ethernet controller, refer to the *Am79C965 PCnet-32* data sheet (PID# 18219) and *PCnet Family Technical Manual* (PID# 18216).

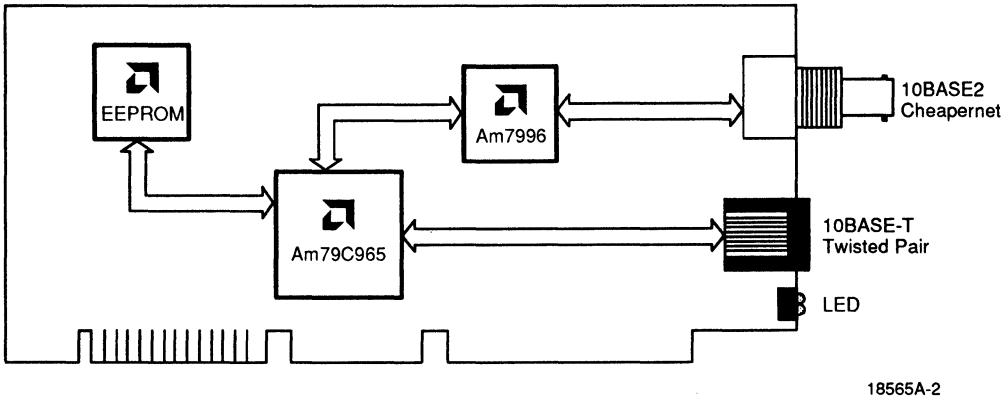
Publication# **18565** Rev. **A** Amendment/**0**
Issue Date: **February 1994**

BLOCK DIAGRAM

PCnet-32-KT



PCnet-32-KT/2



FINAL

PCnet™-PCI-KT

Am79C970 Based Evaulation Kit for PCI Systems

Advanced Micro Devices

GENERAL DESCRIPTION

- **Based on the PCnet-PCI (Am79C970) single-chip Ethernet controller for PCI local bus**
- **PCnet-PCI Ethernet card with on-board 10BASE-T and 10BASE2 connections**
- **Implements a full functional Ethernet node using a PCI (Peripheral Component Interconnect) based system**
- **Ethernet adapter card utilizing a high performance, low-cost bus master architecture (NE2100)**
- **Software configurable**
- **Software compatible with all PCnet family members**
- **Software drivers support all popular Network Operating Systems**
- **Includes evaluation board, software driver diskettes and supporting documentation**

DISTINCTIVE CHARACTERISTICS

The PCnet-PCI evaluation kit is a design evaluation vehicle for the Am79C970 PCnet-PCI single chip Ethernet controller with a PCI bus interface. The kit includes an evaluation board with the Am79C970 single chip Ethernet controller and for 10BASE-T (Twisted Pair) or 10BASE2 (Cheapernet) media.

The evaulation board when installed in a PCI based host system provides a platform for demonstrating the high performance of the PCnet-PCI device, the low manufacturing cost of a PCnet-PCI based solution and the overall ease of design. The platform further allows the user to evaluate network hardware and to develop software for an Ethernet node based on the PCnet-PCI.

In addition to the evaluation board, the kit comes with software diskettes, a hardware user's manual, the *PCnet Family Network Driver Installation Guide*, device datasheet and *PCnet Family Technical Manual*.

The software includes driver object codes for Novell NetWare ODI DOS and OS/2, NetWare Lite, Microsoft Windows NT, Windows for Workgroups, LAN Manager, Banyan VINES Client, IBM LAN Server, SCO UNIX, Artisoft LANtastic/AI, DEC Pathworks, and Packet Driver. Also included are two utility programs, one for configuration of the network adapter card and software driver installation, and one for EEPROM configuration.

No hardware jumpers are required for configuration. The PCI system BIOS automatically configures on power up, the I/O Base Address, interrupt channel, and DMA channel for the PCI Ethernet adapter card.

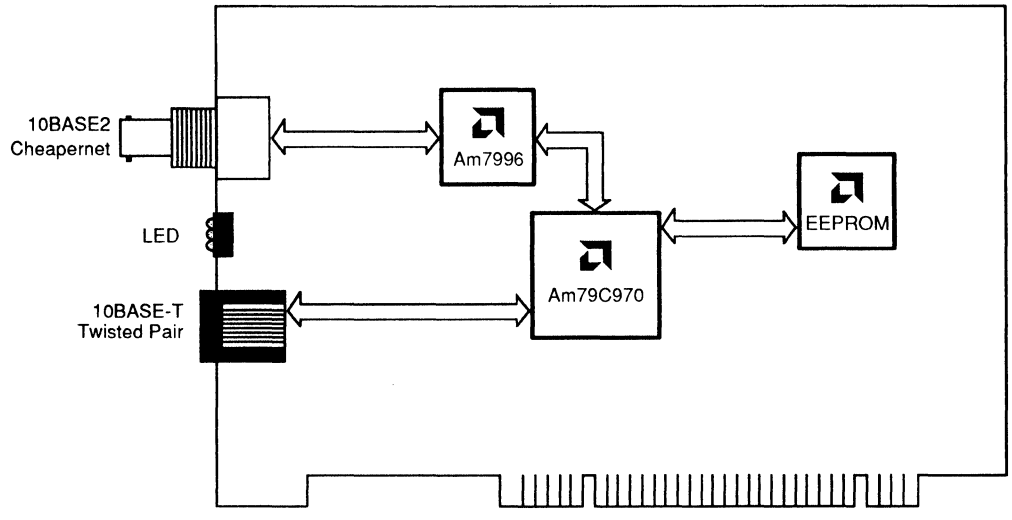
In addition, AMD provides a low-level evaluation program to establish connections and send and receive messages. The evaluation program allows the user to view and change the contents of the PCnet-PCI registers, the memory resident Initialization Block, and the data buffer Descriptor Rings. The program also allows the designer to establish loops for hardware probing.

The PCnet Family Configuration and Installation utility program, AmInstall, provides an easy user interface to view the configuration of the PCnet-PCI evaluation board. The utility program will automatically scan the system bus(es), which may include ISA, VL, and PCI to find the installed PCnet device. This utility program will identify AMD's PCnet-ISA, PCnet-ISA+, PCnet-32, and PCnet-PCI devices. With the configuration portion of the program, the utility will find and report to the user, the I/O address, IRQ channel and DMA channel, assigned to the PCnet-PCI device by the system BIOS. After configuration, the user may use the installation portion of the utility to install a selected driver by copying the appropriate driver, from the AMD diskette and create or modify the CONFIG.SYS, AUTOEXEC.BAT, and PROTOCOL.INI or NET.CFG files on the user's system.

The PCnet-PCI evaulation board stores the unique IEEE physical address in the serial EEPROM. Once powered up, the Am79C970 device reads the node's IEEE physical address from the EEPROM through the Microwire interface protocol. For more details about the PCnet-PCI Ethernet Controller, also refer to the *Am79C970 PCnet-PCI* data sheet (PID# 18220) and *PCnet Family Technical Manual* (PID# 18216).

Publication# **18567** Rev. **A** Amendment/**0**
Issue Date: **February 1994**

PCnet-PCI-KT



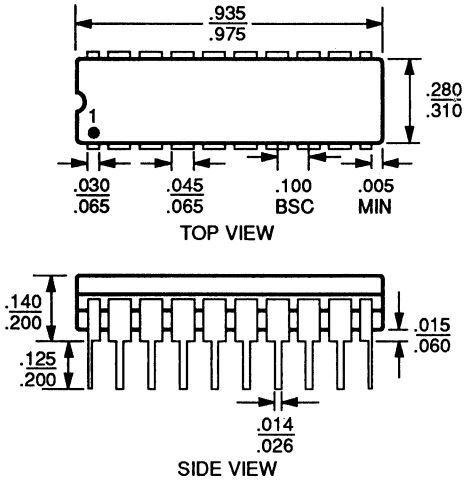
18567A-1

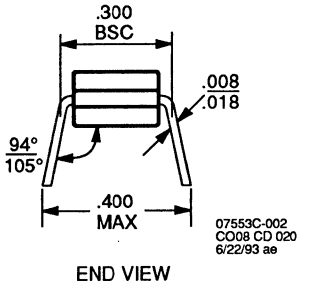
SECTION

4 PHYSICAL DIMENSIONS*

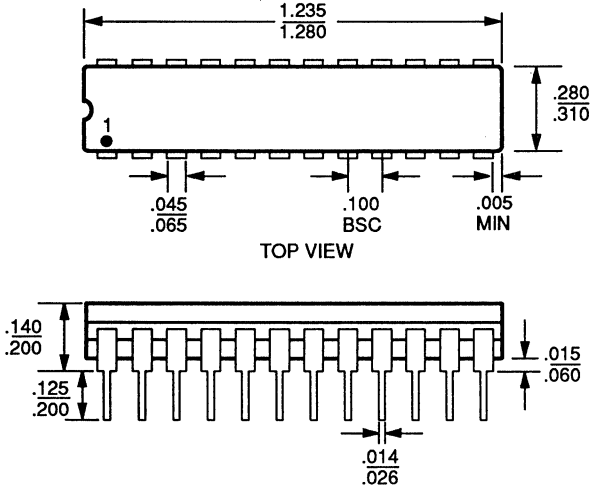
**For Reference only, not to scale. BSC is an ANSI standard for Basic Space Centering.*

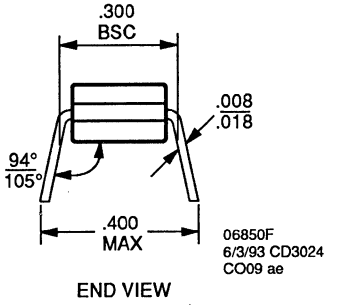
CD 020
20-Pin Ceramic Dual In Package (measured in inches)



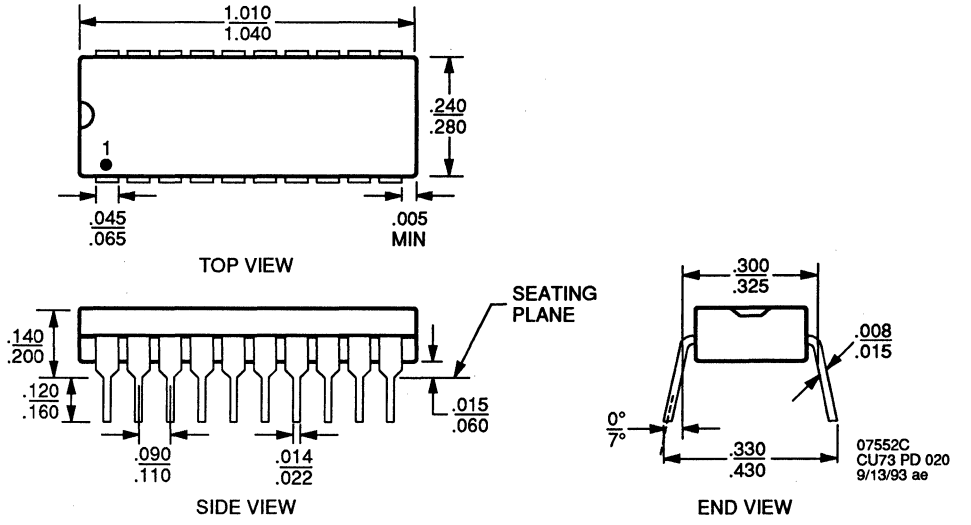


CD3024
24-Pin (300 mil) Ceramic Dual In Line Package (measured in inches)

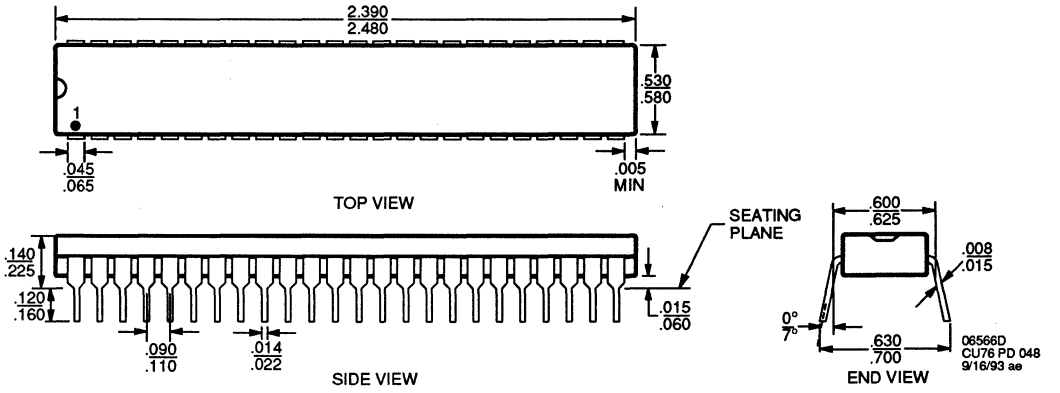




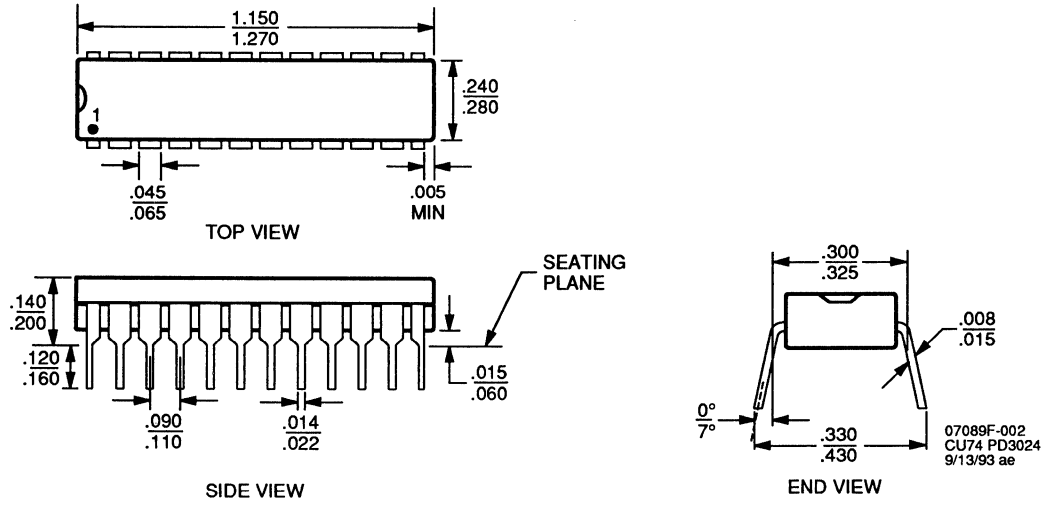
PD 020
20-Pin Plastic Dual In Line Package (measured in inches)



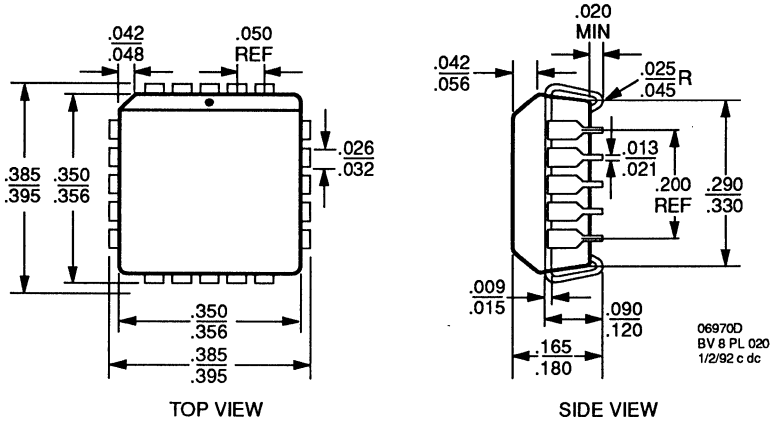
PD 048
48-Pin Plastic Dual In Line Package (measured in inches)



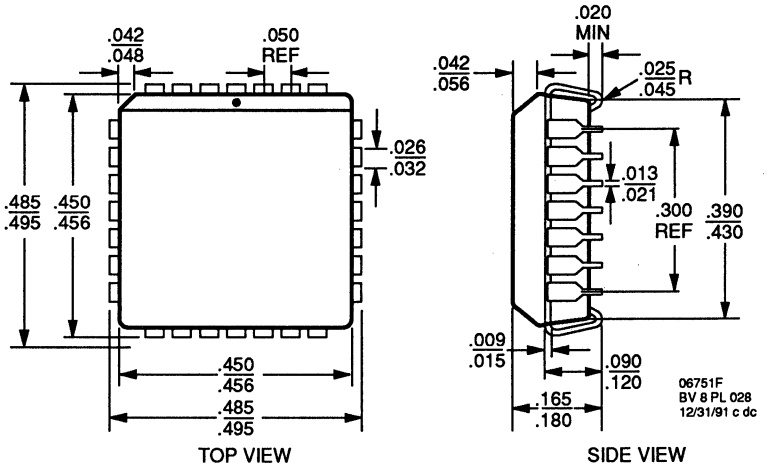
PD3024
24-Pin Plastic Dual In Line Package (measured in inches)



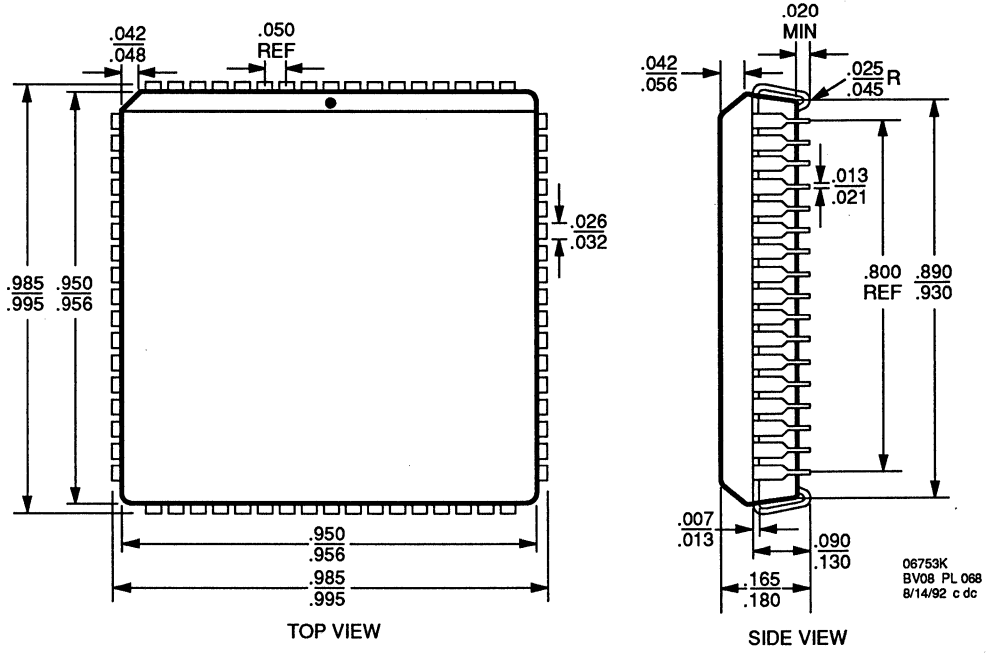
PL 020
20-Pin Plastic Leaded Chip Carrier (measured in inches)



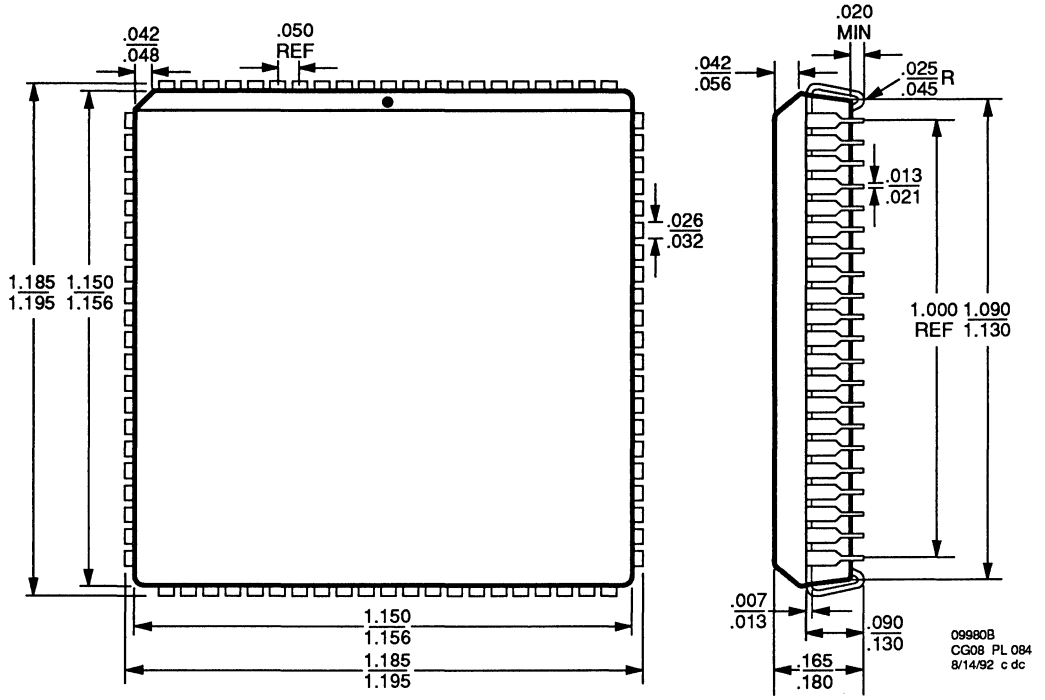
PL 028
28-Pin Plastic Leaded Chip Carrier (measured in inches)



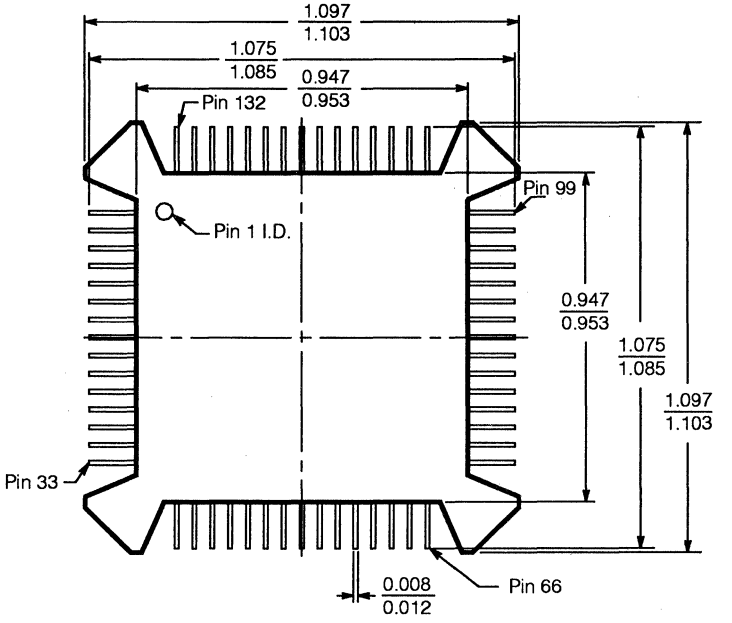
PL 068
68-Pin Plastic Leaded Chip Carrier (measured in inches)



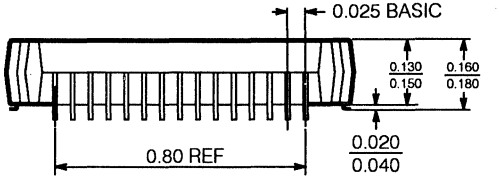
PL 084
84-Pin Plastic Leaded Chip Carrier (measured in inches)



PQB132
132-Pin Plastic Quad Flat Pack Trimmed and Formed (measured in millimeters)



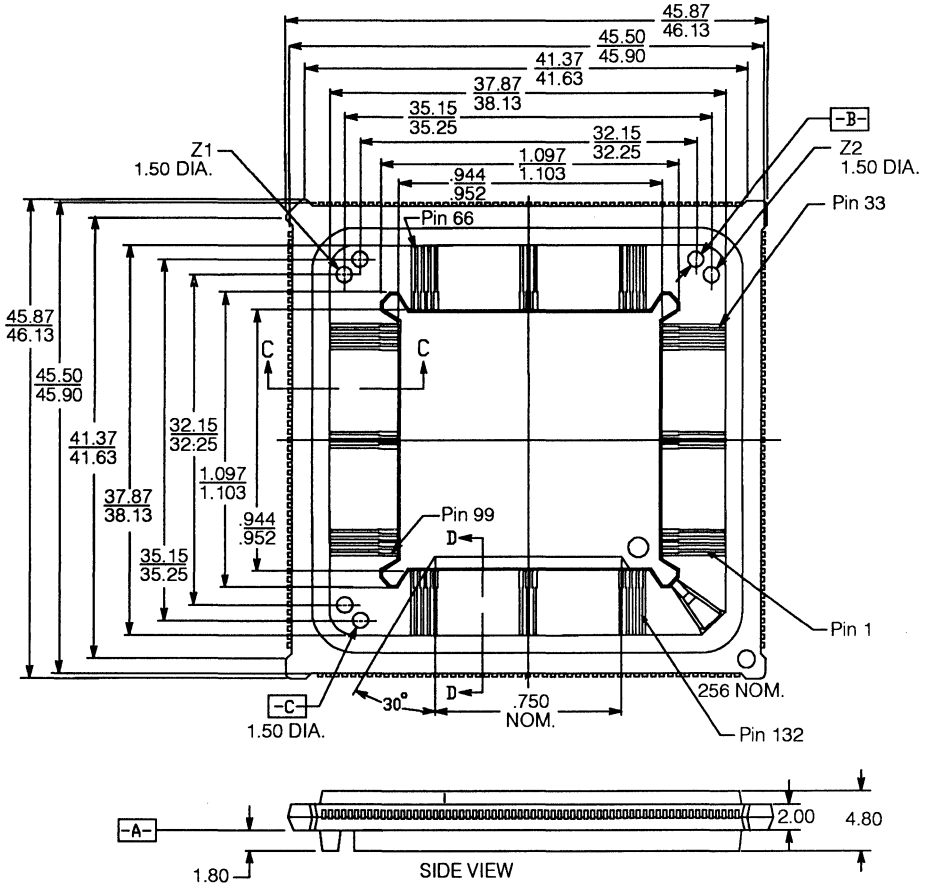
TOP VIEW



BOTTOM VIEW

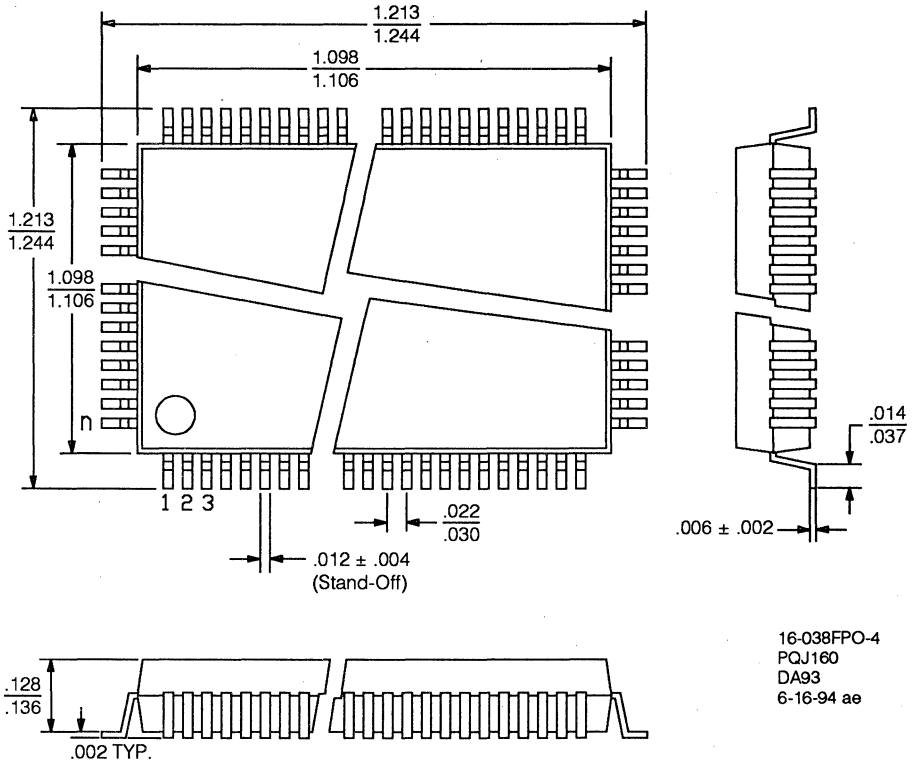
16-038-PQB
DA87
6-15-94 ae

PQB132
132-Pin Plastic Quad Flat Pack with Molded Carrier Ring
(measured in millimeters)



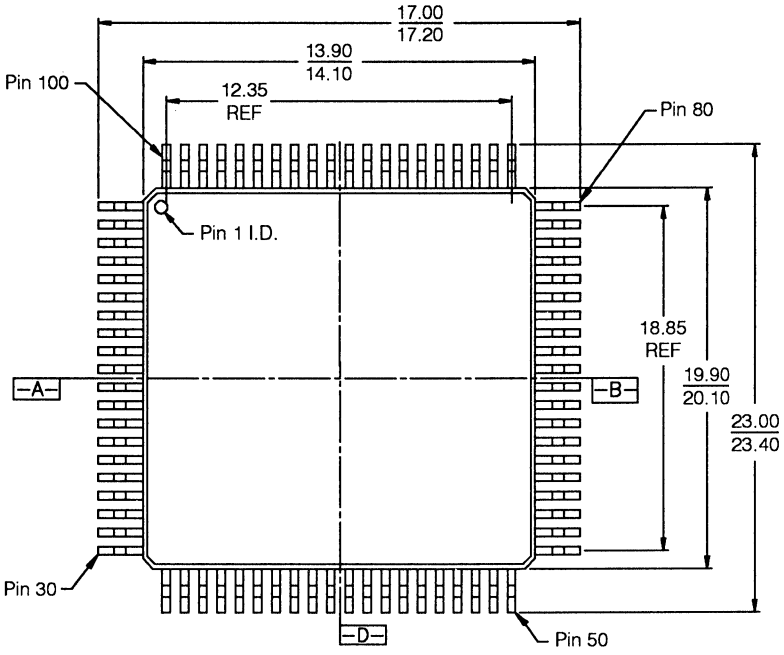
16-0000038-PQB-1
DA84
6-14-94 ae

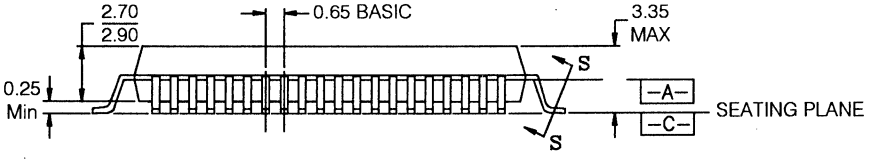
PQJ160
160-Pin EIAJ Quad Flat Pack Package (measured in inches)



16-038FPO-4
PQJ160
DA93
6-16-94 ae

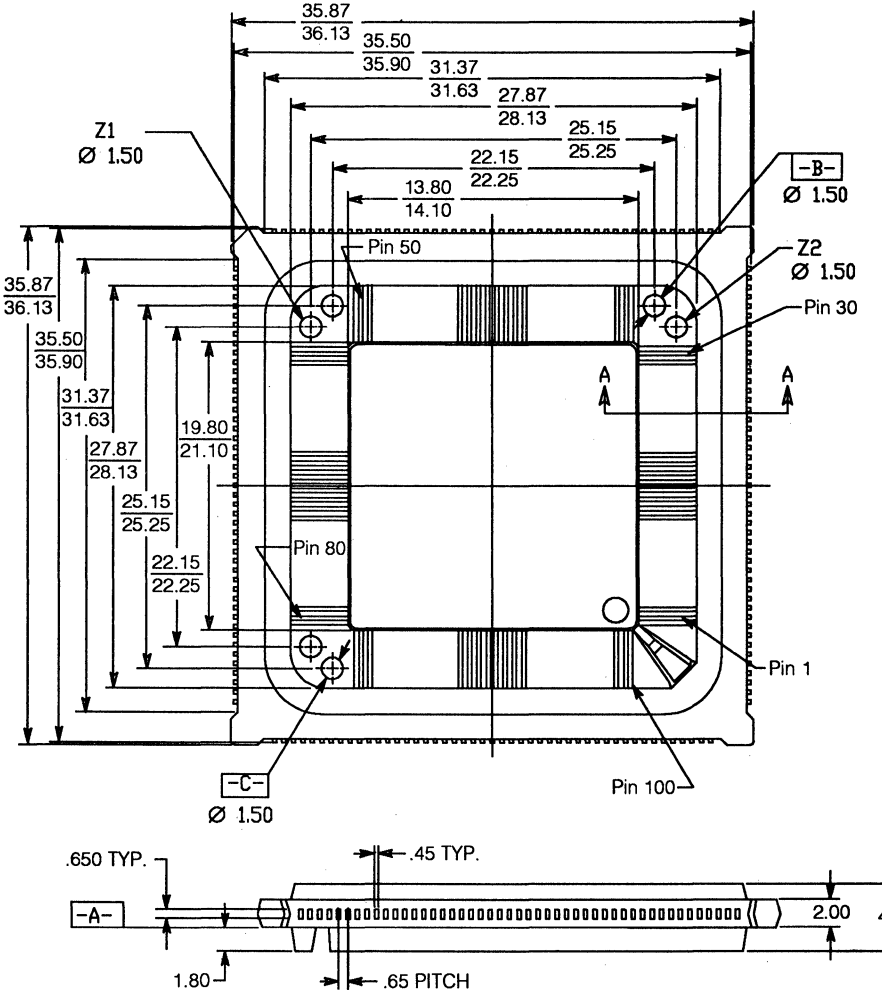
PQR100
100-Pin Plastic Quad Flat Pack Package Trimmed and Formed (measured in millimeters)





16-038-PQR-2
DA92
6-14-94 ae

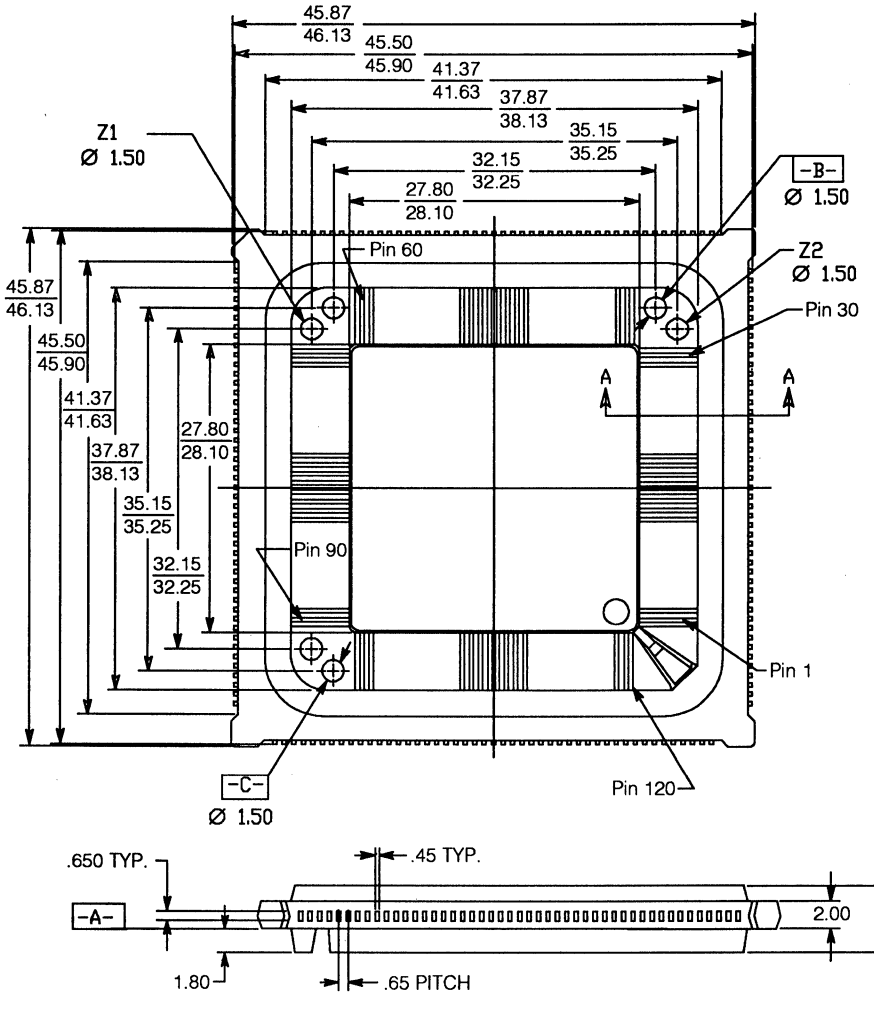
PQR100
100-Pin Plastic Quad Flat Pack Package with Molded Carrier Ring (measured in millimeters)



SIDE VIEW

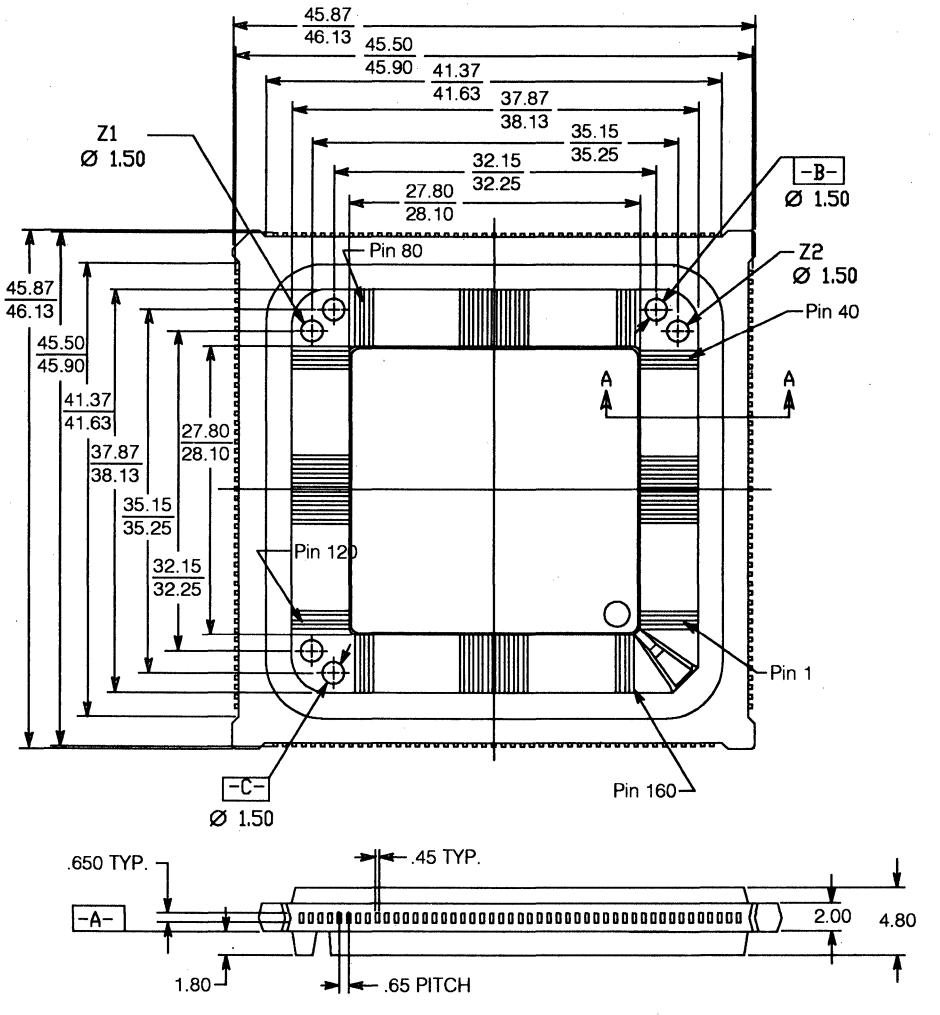
16-038-MOL-2
PQR100
DB89
6-16-94 ae

PQR120
120-Pin Plastic Quad Flat Pack Package with Molded Carrier Ring
(measured in millimeters)



16-038-MOL-2
PQR120
DB89
6-16-94 ae

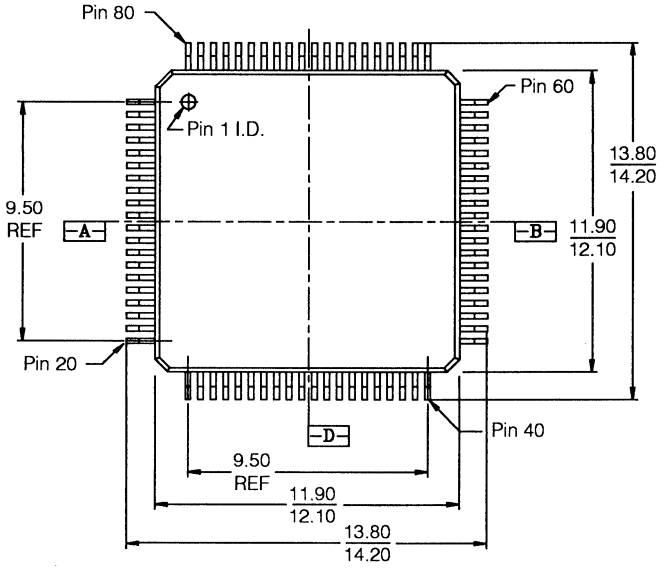
PQR160
160-Pin Plastic Quad Flat Pack Package with Molded Carrier Ring (measured in millimeters)

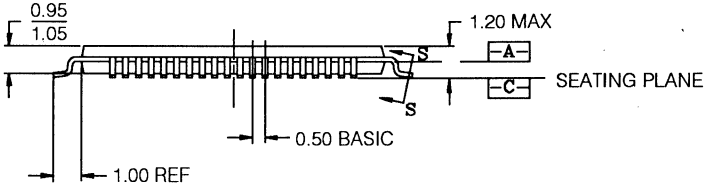


SIDE VIEW

16-038-MOL-2
PQR160
DB89
6-16-94 ae

PQT080
80-Pin Thin Quad Flat Pack Package (measured in millimeters)





Sales Offices

North American

ALABAMA ..... (205) 882-9122
ARIZONA ..... (602) 242-4400
CALIFORNIA,
- Culver City ..... (310) 645-1524
- Newport Beach ..... (714) 752-6262
- Sacramento(Roseville) ..... (916) 786-6700
- San Diego ..... (619) 560-7030
- San Jose ..... (408) 922-0500
- Woodland Hills ..... (818) 878-9988

CANADA, Ontario,
- Kanata ..... (613) 592-0060
- Willowdale ..... (416) 222-7800

COLORADO ..... (303) 741-2900
CONNECTICUT ..... (203) 264-7800
FLORIDA,
- Clearwater ..... (813) 530-9971
- Boca Raton ..... (407) 361-0050
- Orlando (Longwood) ..... (407) 862-9292

GEORGIA ..... (404) 449-7920
IDAHO ..... (208) 377-0393
ILLINOIS,
- Chicago (Itasca) ..... (708) 773-4422
- Naperville ..... (708) 505-9517

MARYLAND ..... (301) 381-3790
MASSACHUSETTS ..... (617) 273-3970
MINNESOTA ..... (612) 938-0001
NEW JERSEY,
- Cherry Hill ..... (609) 662-2900
- Parsippany ..... (201) 299-0002

NEW YORK,
- Brewster ..... (914) 279-8323
- Rochester ..... (716) 425-8050

NORTH CAROLINA
- Charlotte ..... (704) 875-3091
- Raleigh ..... (919) 878-8111

OHIO,
- Columbus (Westerville) ..... (614) 891-6455
- Dayton ..... (513) 439-0268

OREGON ..... (503) 245-0080
PENNSYLVANIA ..... (215) 398-8006
TEXAS,
- Austin ..... (512) 346-7830
- Dallas ..... (214) 934-9099
- Houston ..... (713) 376-8084

International

BELGIUM, Antwerpen	TEL	(03) 248 43 00
	FAX	(03) 248 46 42
FRANCE, Paris	TEL	(1) 49-75-10-10
	FAX	(1) 49-75-10-13
GERMANY,		
Bad Homburg	TEL	(06172)-24061
	FAX	(06172)-23195
München	TEL	(089) 45053-0
	FAX	(089) 406490
HONG KONG,	TEL	(852) 865-4525
Wanchai	FAX	(852) 865-4335
ITALY, Milano	TEL	(02) 3390541
	FAX	(02) 38103458
JAPAN,		
Tokyo	TEL	(03) 3346-7550
	FAX	(03) 3342-5196
Osaka	TEL	(06) 243-3250
	FAX	(06) 243-3253
KOREA, Seoul	TEL	(82) 2-784-0030
	FAX	(82) 2-784-8014

International (Continued)

LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
SINGAPORE	TEL	(65) 3481188
	FAX	(65) 3480161
SWEDEN,		
Stockholm area	TEL	(08) 98 61 80
(Bromma)	FAX	(08) 98 09 06
TAIWAN, Taipei	TEL	(886) 2-7153536
	FAX	(886) 2-7122183
UNITED KINGDOM,		
Manchester area	TEL	(0925) 830380
(Warrington)	FAX	(0925) 830204
London area	TEL	(0483) 740440
(Woking)	FAX	(0483) 756196

North American Representatives

CANADA
- Burnaby, B.C. – DAVETEK MARKETING ..... (604) 430-3680
- Kanata, Ontario – VITEL ELECTRONICS ..... (613) 592-0060
- Mississauga, Ontario – VITEL ELECTRONICS ..... (905) 564-9720

Lachine, Quebec – VITEL ELECTRONICS ..... (514) 636-5951
ILLINOIS
- Skokie – INDUSTRIAL REPRESENTATIVES,INC ..... (708) 967-8430

IOWA
- LORENZ SALES ..... (319) 377-4666

KANSAS
- Merriam – LORENZ SALES ..... (913) 469-1312
- Wichita – LORENZ SALES ..... (316) 721-0500

MEXICO
- Chula Vista – SONIKA ELECTRONICA ..... (619) 498-8340
- Guadalajara – SONIKA ELECTRONICA ..... (523) 647-4250
- Mexico City – SONIKA ELECTRONICA ..... (523) 754-6480
- Monterrey – SONIKA ELECTRONICA ..... (523) 358-9280

MICHIGAN
- Holland – COM-TEK SALES, INC ..... (616) 335-8418
- Brighton – COM-TEK SALES, INC ..... (313) 227-0007

MINNESOTA
- Mel Foster Tech. Sales, Inc. ..... (612) 941-9790

MISSOURI
- LORENZ SALES ..... (314) 997-4558

NEBRASKA
- LORENZ SALES ..... (402) 475-4660

NEW MEXICO
- THORSON DESERT STATES ..... (505) 883-4343

NEW YORK
- East Syracuse – NYCOM, INC ..... (315) 437-8343
- Hauppauge – COMPONENT CONSULTANTS, INC ..... (516) 273-5050

OHIO
- Centerville – DOLFUSS ROOT & CO ..... (513) 433-6776
- Westlake – DOLFUSS ROOT & CO ..... (216) 899-9370

PENNSYLVANIA
- RUSSELL F. CLARK CO.,INC. ..... (412) 242-9500

PUERTO RICO
- COMP REP ASSOC, INC ..... (809) 746-6550

UTAH
- FRONT RANGE MARKETING ..... (801) 288-2500

WASHINGTON
- ELECTRA TECHNICAL SALES ..... (206) 821-7442

WISCONSIN
- Brookfield – INDUSTRIAL REPRESENTATIVES,INC ..... (414) 574-9393

Advanced Micro Devices, Inc. 901 Thompson Place, P.O. Box 3453, Sunnyvale, CA 94088, USA
Tel: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450
APPLICATIONS HOTLINE & LITERATURE ORDERING • TOLL FREE: (800) 222-9323 • (408) 749-5703

BAN-29.8M-7/94-0 4/7/94
14287C Printed in USA

ADVANCED
MICRO
DEVICES, INC.
901 Thompson Place
P.O. Box 3453
Sunnyvale,
California 94088-3453
(408) 732-2400
TWX: 910-339-9280
TELEX: 34-6306

APPLICATIONS HOTLINE &
LITERATURE ORDERING
USA (800) 222-9323
USA (408) 749-5703
JAPAN 3346-7550
UK & EUROPE 44-(0)256-811101
TOLL FREE
USA (800) 538-8450
FRANCE 0590-8621
GERMANY 0130-813875
ITALY 1678-77224

RECYCLED &
RECYCLABLE

Printed in USA
BAN-29.8M-6/94-0
14287C