AMD

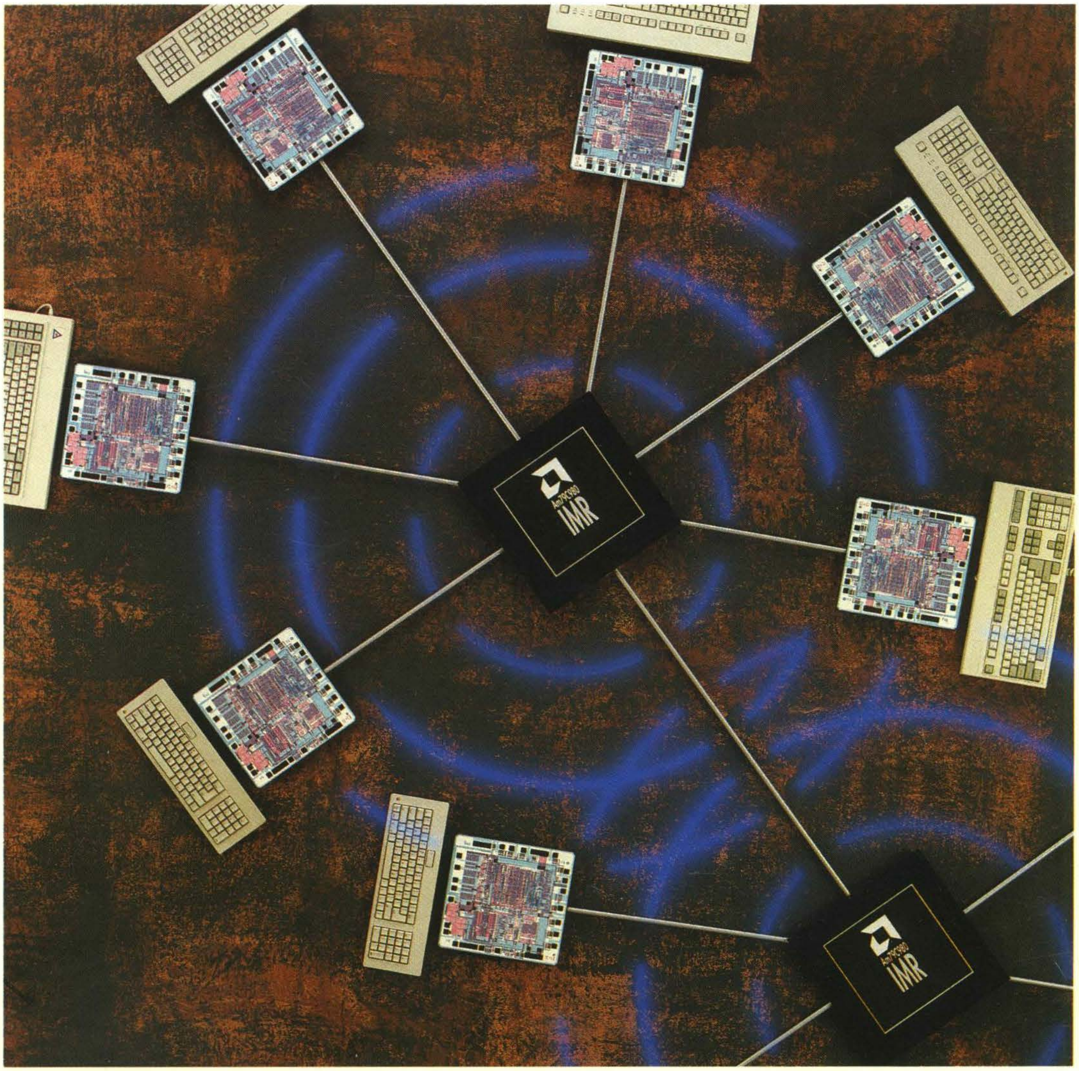
Ethernet/IEEE 802.3 Family

1992

Ethernet/IEEE 802.3 Family

1992 World Network Data Book/Handbok

Advanced
Micro
Devices



Ethernet/IEEE 802.3 Family

1992 World Network Data Book/Handbook

ADVANCED MICRO DEVICES

Advanced Micro Devices reserves the right to make changes in its products
without notice in order to improve design or performance characteristics.

This publication neither states nor implies any warranty of any kind, including but not limited to implied warrants of merchantability or fitness for a particular application. AMD assumes no responsibility for the use of any circuitry other than the circuitry in an AMD product.

The information in this publication is believed to be accurate in all respects at the time of publication, but is subject to change without notice. AMD assumes no responsibility for any errors or omissions, and disclaims responsibility for any consequences resulting from the use of the information included herein. Additionally, AMD assumes no responsibility for the functioning of undescribed features or parameters.

Trademarks

Product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

DAI, EADI, Integrated Local Area Communications Controller, ILACC are trademarks and PAL and SKINNYDIP are registered trademarks of Advanced Micro Devices, Inc.

Ethernet is a registered trademark of Xerox Corporation.

IBM PC/AT is a registered trademark of International Business Machines Corporation.

Macintosh II is a registered trademark of Apple Computer, Inc.

NetWare is a registered trademark of Novell, Inc.

NuBus is a trademark of Texas Instruments, Inc.

This publication provides the complete specifications for the semiconductors needed in designing 10BASE5, 10BASE2 or 10BASE–T IEEE 802.3/Ethernet network products.

It replaces the book we published less than twelve months ago. This quick cycle represents many exciting factors:

- AMD's continued introduction of new and next generation Ethernet products.
- Rapid adoption of compliant controller, physical layer and hub products.
- The accelerated pace of product introductions due to the diversity of the world market.

To those of you who have experience with our products, we are the advance guard in the increasing importance of sharing data. And, to you who are not familiar with our products, AMD's networking team welcomes the opportunity to share our newest specifications with you.

Remember our partnership helps you gain and keep the competitive edge. We are not your competition.

Robert M. Krueger
Vice President,
Networks Product Division

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 network marketplace. Our total portfolio includes products for 802.3/Ethernet, Fiber Distributed Data Interface (FDDI), ESCON, Fiber Channel 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 on 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.

Motherboards

AMD is the leading supplier of 802.3/Ethernet ICs to the motherboard marketplace. This market includes engineering workstations and personal computer platforms. AMD offers solutions for both 16-bit and 32-bit microprocessor busses. The industry's most widely designed in Ethernet controller, the Am7990 LANCE, defined the industry preferred architecture for efficient software interface in high-performance applications. AMD's second generation Ethernet controller, the Am79C900 ILACC, offers an easy migration path to 32-bit applications. The ILACC has a higher level of integration and performance, while taking advantage of the existing software development investment. Software drivers written for the Am7990 LANCE can easily be converted to serve the Am79C900 ILACC.

Complementing the controller offering are a Manchester encoder/decoder and several physical layer devices for either thick coax Ethernet/IEEE 802.3 (10BASE5), thin coax Cheapernet/IEEE 802.3 (10BASE2), or twisted pair Ethernet/IEEE 802.3 (10BASE-T). The Am79C98, Twisted Pair Ethernet Transceiver (TPEX), has a special power-down feature, sleep mode, optimal for PC laptop applications.

Medium Attachment Units

For medium attachment units (MAUs), also known as stand-alone transceivers, AMD offers two 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 second device, the Am79C98 Twisted Pair Ethernet Transceiver (TPEX), is AMD's offering for the 802.3 10BASE-T market. The Am79C98 is a highly integrated device that allows for a very cost-effective LAN system implementation using 10BASE-T medium attachment units. The Am79C98 is also supported by a stand-alone evaluation board, the Am79C98EVAL-HW, to speed progress along the learning curve.

Multiport Repeaters

Multiport repeaters, hubs and concentrators have been used in the industry for many years in coaxial 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 10BASE-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 large 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 Am79C980 Integrated Multiport Repeater (IMR). This device allows the system designer to easily develop reliable, maintainable 10BASE-T multiport repeaters of various complexity and functionality. The Am79C980 is supported by the IMR-VELCRO-HW kit to facilitate rapid design and production of IMR based 10BASE-T hubs.

PC Add-on cards

AMD's family of Ethernet controllers and transceivers are well suited for high performance add-on card applications.

In 1989, AMD developed the industry's first bus master PC add-on card for the IBM PC/AT™ and compatibles. The AT half card, the LANCE-AT-KT, is supported with Novell NetWare™ certified driver software and a development monitor/debug program. This board allows the system designer to easily evaluate the AMD devices. It also serves as a reference platform for product development of PC add-on card products.

Similarly, the 32-bit ILACC controller is supported by a 32-bit add-on card with a NU-bus™ interface, the ILACC-MAC-KT, for the Apple Macintosh II family of computers.

SECTION 1
Product Data Sheets

Ethernet Controllers

Physical Layer

Hub Product

Am7990

Local Area Network Controller for Ethernet (LANCE)

Advanced Micro Devices

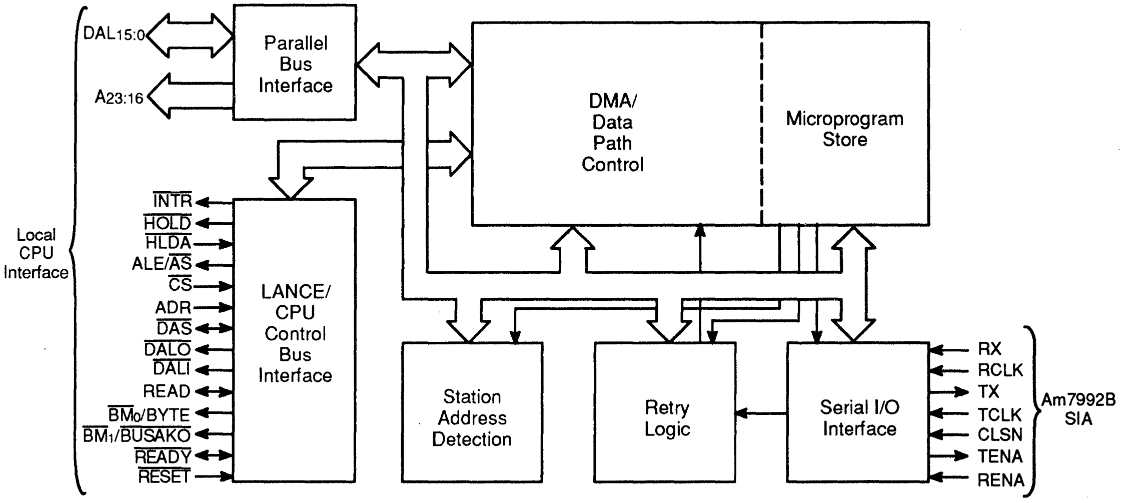
DISTINCTIVE CHARACTERISTICS

- **Compatible with Ethernet and IEEE 802.3 10BASE 5 Type A, and 10BASE 2 Type B, "Cheapernet"**
- **Easily interfaced with 80x86, 680x0, Am29000, Z8000™, LSI-II™ microprocessors**
- **On-board DMA and buffer management, 48 byte FIFO**
- **24-bit wide linear addressing (Bus Master Mode)**
- **Network and packet error reporting**
- **Back-to-back packet reception with as little as 4.1 µsec interpacket gap time**
- **Diagnostic Routines**
  - Internal/external loop back
  - CRC logic check
  - Time domain reflectometer

GENERAL DESCRIPTION

The Am7990 Local Area Network Controller for Ethernet (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 LANCE, in conjunction with the Am7992B Serial Interface Adapter (SIA), Am7996, Am7997 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 Am7990 is designed using a scaled NMOS 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



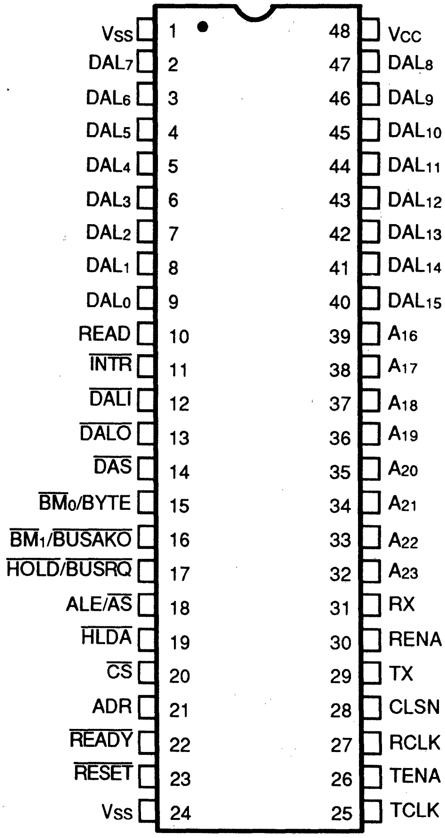
05698D-001A

Publication# 05698 Rev. D Amendment/0
Issue Date: January 1992

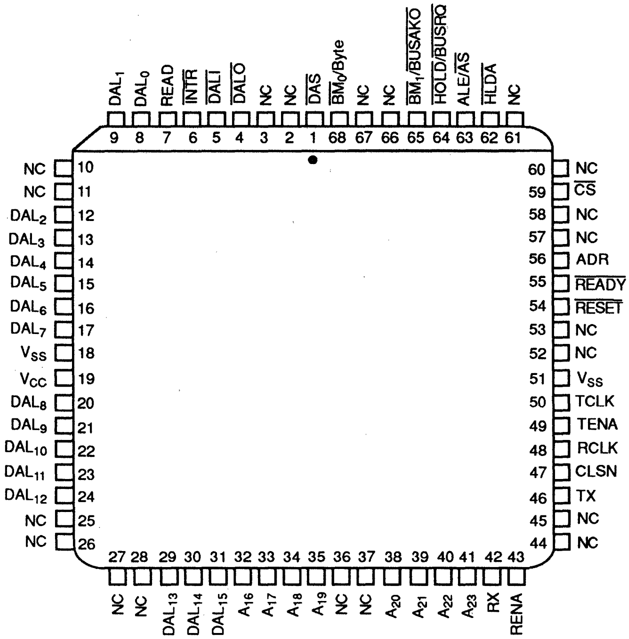
RELATED AMD PRODUCTS

Part No.	Description
Am7992B	Serial Interface Adaptor (SIA)
Am7996	IEEE 802.3/Ethernet/Cheapernet Transceiver
Am79C900	Integrated Local Area Communications Controller
Am79C98	Twisted Pair Ethernet Transceiver
Am79C980	Integrated Multiport Repeater

CONNECTION DIAGRAMS



05698D-002A

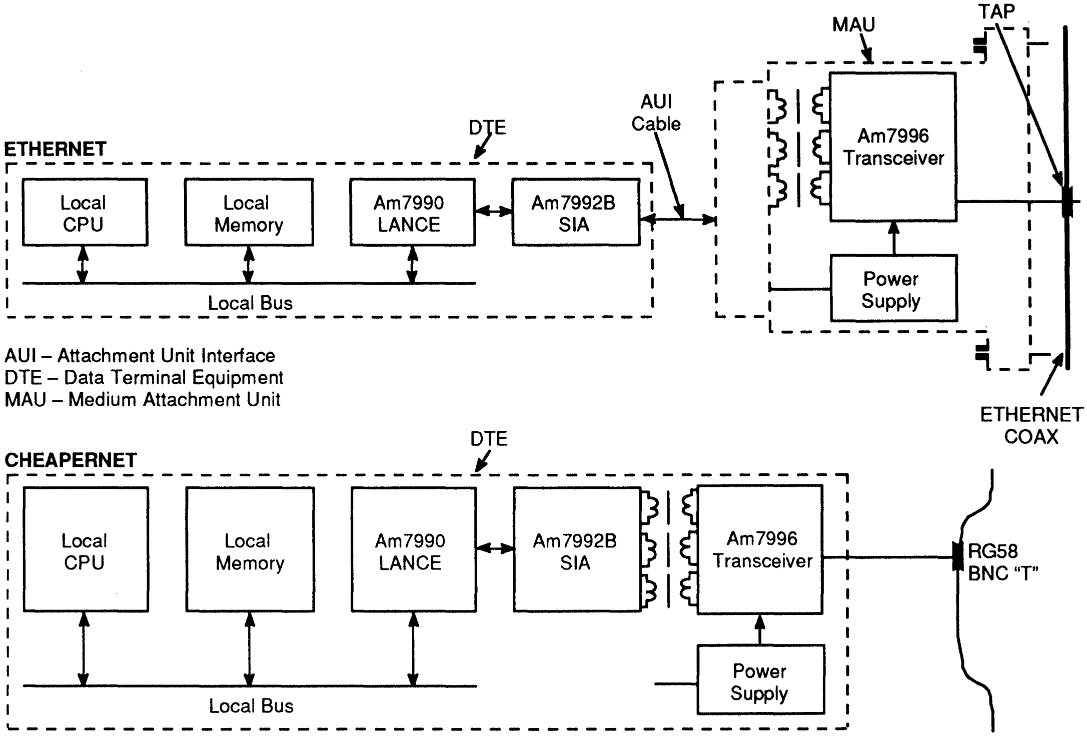


05698D-003A

Note:

Pin 1 is marked for orientation.

TYPICAL ETHERNET/CHEAPERNET NODE



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

05698D-004A

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:

AM7990 P C B /80

REVISION
Revision of Die
80 = Current Rev.

OPTIONAL PROCESSING
Blank = Standard Processing
B = Burn-in
TR = Tape and Reel Packaging

TEMPERATURE RANGE
C = Commercial (0 to +70°C)

PACKAGE TYPE
P = 48-Pin Plastic DIP (PD 048)
D = 48-Pin Sidebrazed Ceramic (SD 048)
J = 68-Plastic Leaded Chip Carrier
(PL 068)

SPEED OPTION
Not Applicable

DEVICE NUMBER/DESCRIPTION
Am7990
Local Area Network Controller for Ethernet

Valid Combinations		
AM7990	DC, DCB, PC, PCB, JC, JCTR	/80

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, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

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 LANCE is 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 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 CSR_3.

As ALE (CSR_3 (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 $\overline{AS}$ (CSR_3 (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 LANCE drives the ALE/$\overline{AS}$ line only as a Bus Master.

$\overline{BM_0}$/BYTE, $\overline{BM_1}$/$\overline{BUSAKO}$

(Output, Three-state)

The two pins are programmable through bit (00) of CSR_3

$\overline{BM_0}$, $\overline{BM_1}$ — If CSR_3 (00) BCON = 0
PIN 15 = $\overline{BM_0}$ (Output Three-state) (48-Pin DIPs)
PIN 16 = $\overline{BM_1}$ (Output Three-state) (48-Pin DIPs)

$\overline{BM_0}$, $\overline{BM_1}$ (Byte Mask). This indicates the byte(s) on the DAL are to be read or written during this bus transaction. The 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.

$\overline{BM_1}$	$\overline{BM_0}$	
LOW	LOW	Whole Word
LOW	HIGH	Upper Byte
HIGH	LOW	Lower Byte
HIGH	HIGH	None

BYTE, $\overline{BUSAKO}$ — If CSR_3 (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 DAL_{00} line, latched during the address portion of the bus cycle. The LANCE drives BYTE only as a Bus Master and ignores it when a Bus Slave selection is done (similar to $\overline{BM_0}$, $\overline{BM_1}$).

Byte selection is done as outlined in the following table.

BYTE	DAL_{00}	
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 LANCE is requesting the bus when it receives $\overline{HLDA}$, $\overline{BUSAKO}$ wlll remain HIGH.

Byte Swapping

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

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 FIFO.

Signal Line	Mode Bits	
	BSWP = 0 and BCON = 1	BSWP = 1 and BCON = 1
BYTE = L and DAL$_{00}$ = L	Word	Word
BYTE = L and DAL$_{00}$ = H	Illegal	Illegal
BYTE = H and DAL$_{00}$ = H	Upper Byte	Lower Byte
BYTE = H and DAL$_{00}$ = L	Lower Byte	Upper Byte

CLSN

Collision (Input)

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

CS

Chip Select (Input)

Indicates, when asserted, that the LANCE is the slave device of the data transfer. CS must be valid throughout the data portion of the bus cycle. CS must not be asserted when HLDA is LOW

DAL$_{00}$ – DAL$_{15}$

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

The time multiplexed Address/Data bus. During the address portion of a memory transfer, DAL$_{00}$ – DAL$_{15}$ contains the lower 16 bits of the memory address. The upper 8 bits of address are contained in A$_{16}$ – A$_{23}$.

During the data portion of a memory transfer, DAL$_{00}$ – DAL$_{15}$ contains the read or write data, depending on the type of transfer.

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

DALI

Data/Address Line In (Output, Three-State)

An external bus transceiver control line. DALI is asserted when the 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. DALI is driven only when LANCE is a Bus Master.

DALO

Data/Address Line Out (Output, Three-State)

An external bus transceiver control line. DALO is asserted when the LANCE drives the DAL lines. 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. DALO is driven only when LANCE is a Bus Master.

DAS

Data Strobe (Input/Output, Three-State)

Defines the data portion of the bus transaction. 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. DAS is driven only as a Bus Master.

HLDA

Bus Hold Acknowledge (Input)

A response to HOLD. When HLDA is LOW in response to the chip's assertion of HOLD, the chip is the Bus Master.

During bus master operation the LANCE waits for HLDA to be deasserted 'HIGH' before reasserting HOLD 'LOW'. This insures proper bus handshake under all situations.

HOLD/BUSRQ

Bus Hold Request (Output, Open Drain)

Asserted by the LANCE when it requires access to memory. HOLD is held LOW for the entire ensuing bus transaction. The function of this pin is programmed through bit (00) of CSR$_3$. Bit (00) of CSR$_3$ is cleared when RESET is asserted.

When CSR$_3$ (00) BCON = 0

PIN 17 = HOLD (Output Open Drain and input sense)
(48-Pin DIPs)

When CSR$_3$ (00) BCON = 1

PIN 17 = BUSRQ (I/O Sense, Open Drain) (48-Pin DIPs)

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

INTR

Interrupt (Output, Open Drain)

An attention signal that indicates, when active, that one or more of the following CSR$_0$ status flags is set: BABL, MERR, MISS, RINT, TINT or IDON. INTR is enabled by bit 06 of CSR0 (INEA = 1). 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 signals an output when the LANCE is a Bus Master.

High – Data is taken off the DAL by the LANCE.

Low – Data is placed on the DAL by the LANCE.

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

High – Data is placed on the DAL by the LANCE.

Low – Data is taken off the DAL by the LANCE.

$\overline{\text{READY}}$

(Input/Output, Open Drain)

When the 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 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 LANCE is a Bus Master and an output when the 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)

Bus Request Signal. Causes the 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 CSR_0 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_{CC}

Power supply pin +5 volt ±5%

It is recommended that a 0.1 µF and a 10 µF decoupling capacitors be used between V_{CC} 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 Local Area Network Controller for Ethernet (LANCE) has been designed to be "friendly" or easy to interface to a variety of popular 16-bit microprocessors. These microprocessors include the Z8000, Am29000, 80x86, 680x0 and LSI-11. The LANCE has a 24-bit wide linear address space when it is in the Bus Master Mode, allowing it to DMA directly into the entire address space of the above microprocessors. A programmable mode of operation allows byte addressing in one of two ways: a Byte/Word control signal compatible with the 8086 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 LANCE interfaces with both multiplexed and demultiplexed data busses and features control signals for address/data bus transceivers.

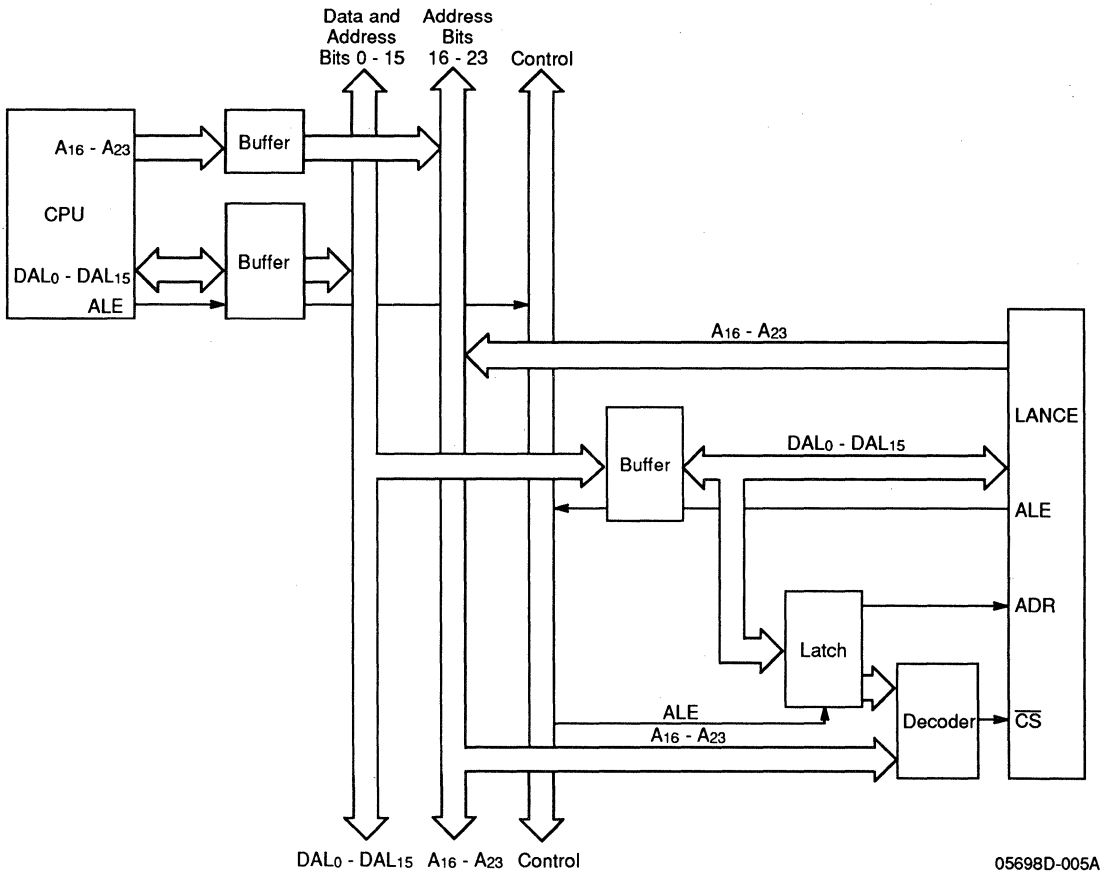


Figure 1-1. LANCE/CPU Interfacing — Multiplexed Bus

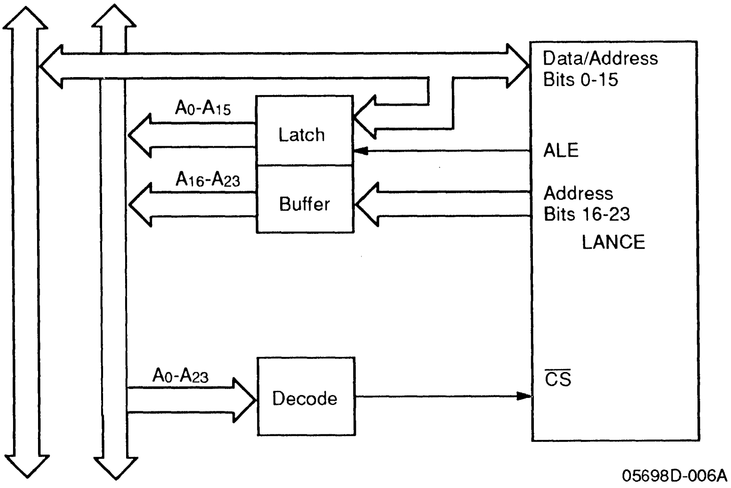


Figure 1-2. LANCE/CPU Interfacing—Demultiplexed Bus

During initialization, the CPU loads the starting address of the initialization block into two internal control registers. The LANCE has four internal control and status registers ($CSR_{0,1,2,3}$) which are used for various functions, such as the loading of the initialization block address, different programming modes and status conditions. The host processor communicates with the 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 LANCE upon: 1) completion of its initialization routine, 2) the reception of a packet, 3) the transmission of a packet, 4) transmitter timeout error, 5) a missed packet and 6) memory error.

The cause of the interrupt is ascertained by reading CSR_0. Bit (06) of CSR_0, (INEA), enables or disables interrupts to the microprocessor. In systems where polling is used in place of interrupts, bit (07) of CSR_0, (INTR), indicates an interrupt condition.

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

In the receive mode, packets are sent via the Am7992B SIA to the LANCE. The packets are loaded into the 48-byte 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 checksum doesn't agree with the packet CRC, an error bit is set.

Addressing

Packets can be received using 3 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 LANCE during an initialization cycle. There are two types of logical address. 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 the so-called "promiscuous mode" in which a node will accept all packets on the coax regardless of their destination address.

Collision Detection and Implementation

The Ethernet CSMA/CD network access algorithm is implemented completely within the LANCE. In addition to listening for a clear coax before transmitting, Ethernet handles collisions in a predetermined way. Should two transmitters attempt to seize the coax at the same time, they will collide and the data on the coax 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 don't try to repeatedly access the network at the same time. Up to 16 attempts to access the network are made by the LANCE before reporting back an error due to excessive collisions.

Error Reporting and Diagnostics

Extensive error reporting is provided by the LANCE. Error conditions reported relate either to the network as a whole or to 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 data bytes.
- **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 LANCE performs several diagnostic routines which enhance the reliability and integrity of the system. These include a CRC logic 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 LANCE to aid system designers locate faults in the Ethernet cable. Shorts and opens manifest themselves in reflections which are sensed by the TDR.

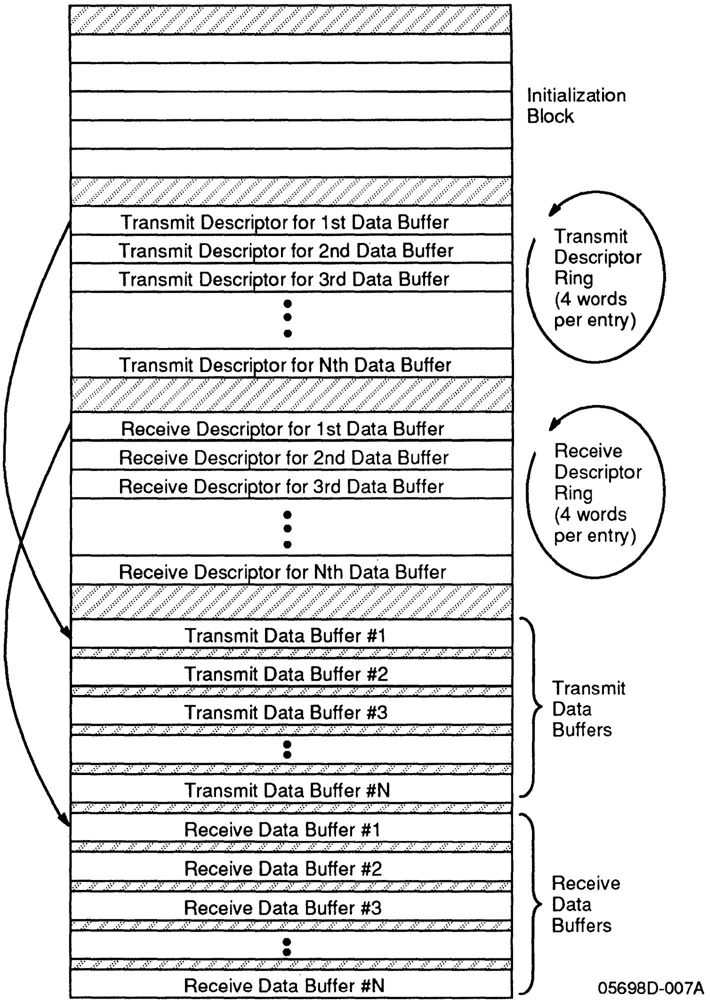


Figure 2-1. LANCE/Processor Memory Interface

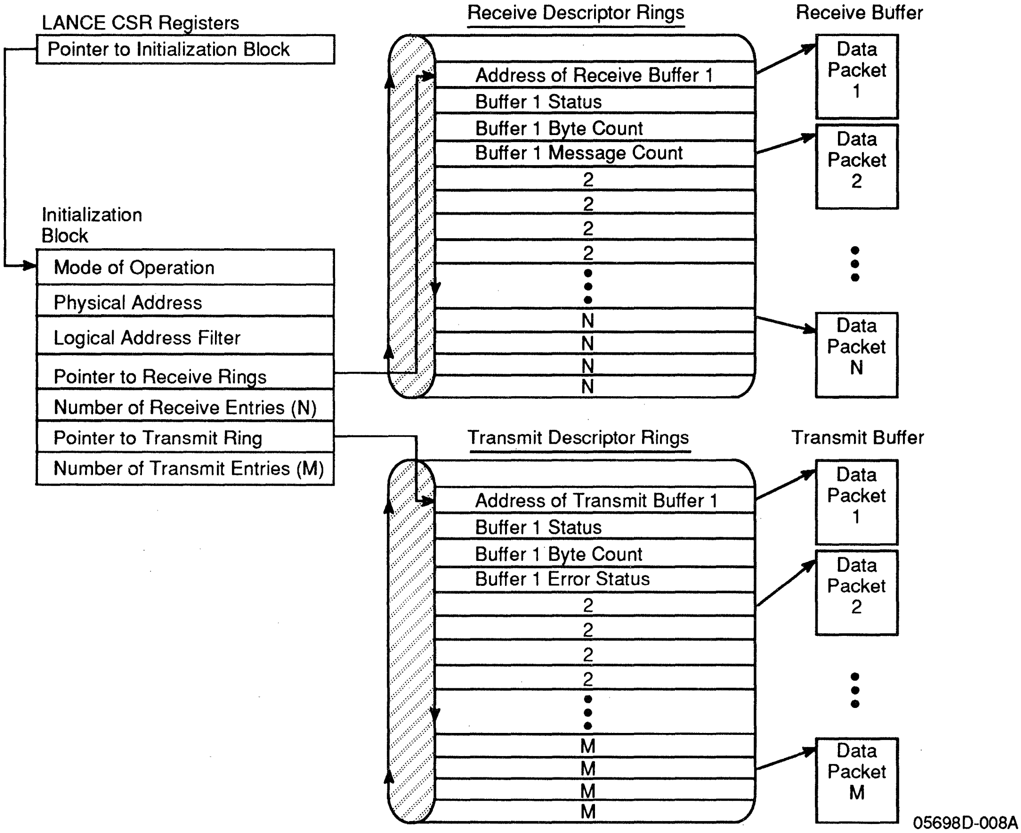


Figure 2-2. LANCE Memory Management

Buffer Management

A key feature of the LANCE and its on-board DMA channel is the flexibility and speed of communication between the 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 & 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 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 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.

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 ($\overline{BM_0}$ and $\overline{BM_1}$) indicate which byte is addressed. When the BYTE scheme is chosen, the $\overline{BM_1}$ 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.

LANCE in Bus Slave Mode

The 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 (CSR_0, CSR_1, CSR_2, and CSR_3) and the Register

Address Pointer (RAP). RAP and CSR_0 may be read or written to at anytime, but the LANCE must be stopped (by setting the stop bit in CSR_0) for CSR_1, CSR_2, and CSR_3 access.

Read Sequence (Slave Mode)

At the beginning of a read cycle, $\overline{CS}$, READ, and $\overline{DAS}$ are asserted. ADR also 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 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)

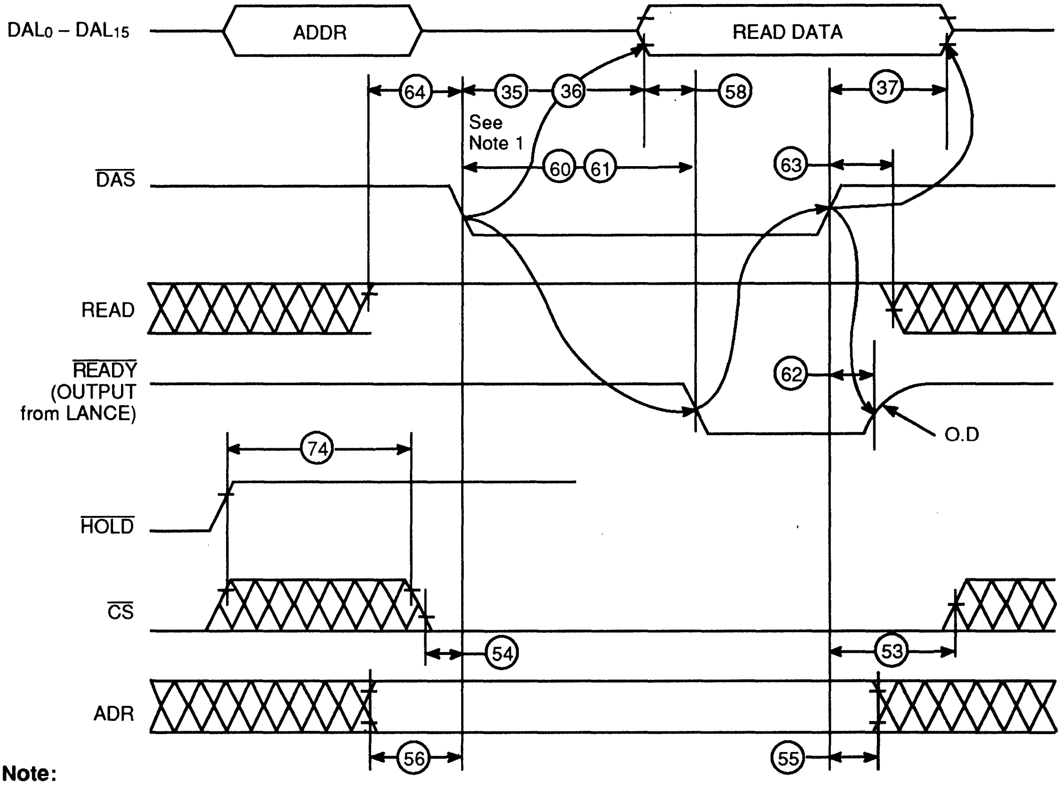
This cycle is similar to the read cycle, except that during this cycle, READ is not asserted (READ is LOW). The DAL buffers are tristated which configures these lines as inputs. The assertion of $\overline{READY}$ by LANCE indicates to the memory device that the data on the DAL lines have been stored by 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:

Timing parameter 62 does not apply in a slave write cycle that sets the STOP bit in CSR_0. Setting this bit generates a LANCE reset, which causes all bus control output signals (including $\overline{READY}$) to start floating about 100 nsec. after $\overline{READY}$ goes active. If $\overline{DAS}$ and $\overline{CS}$ are held active for more than 400 nsec. after $\overline{READY}$ start to float, the LANCE can start a second slave cycle. $\overline{DAS}$ and $\overline{CS}$ should be deasserted within 400 nsec. after $\overline{READY}$ starts to float to prevent this second slave cycle from happening.

LANCE in Bus Master Mode

All data transfers from the LANCE in the bus Master mode are timed by ALE, $\overline{DAS}$, and $\overline{READY}$. The automatic adjustment of the LANCE cycle by the $\overline{READY}$ signal allows synchronization with variable cycle time memory due either to memory refresh or to dual port access. Bus cycles are a minimum of 600 ns in length and 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 CSR_0, CSR_3 and RAP.
   Type 2 refers to access of CSR_1 and CSR_2 which are longer than Type 1 delay.

05698D-009A

Figure 3. Bus Slave Read Timing

Read Sequence (Master Mode)

The read cycle is begun by valid addresses being placed on DAL_{00} – DAL_{15} and A_{16} – A_{23}. 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 $\overline{\text{AS}}$ are pulsed, and the trailing edge of either can be used to latch addresses. DAL_{00} – DAL_{15} go into a 3-state mode, and $\overline{\text{DAS}}$ falls LOW to signal the beginning of the memory access. The memory responds by placing $\overline{\text{READY}}$ LOW to indicate that the DAL lines have valid data. The LANCE then latches memory data on the rising edge of $\overline{\text{DAS}}$, which in turn ends the memory cycle and $\overline{\text{READY}}$ returns HIGH. Refer to Figure 5-1.

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

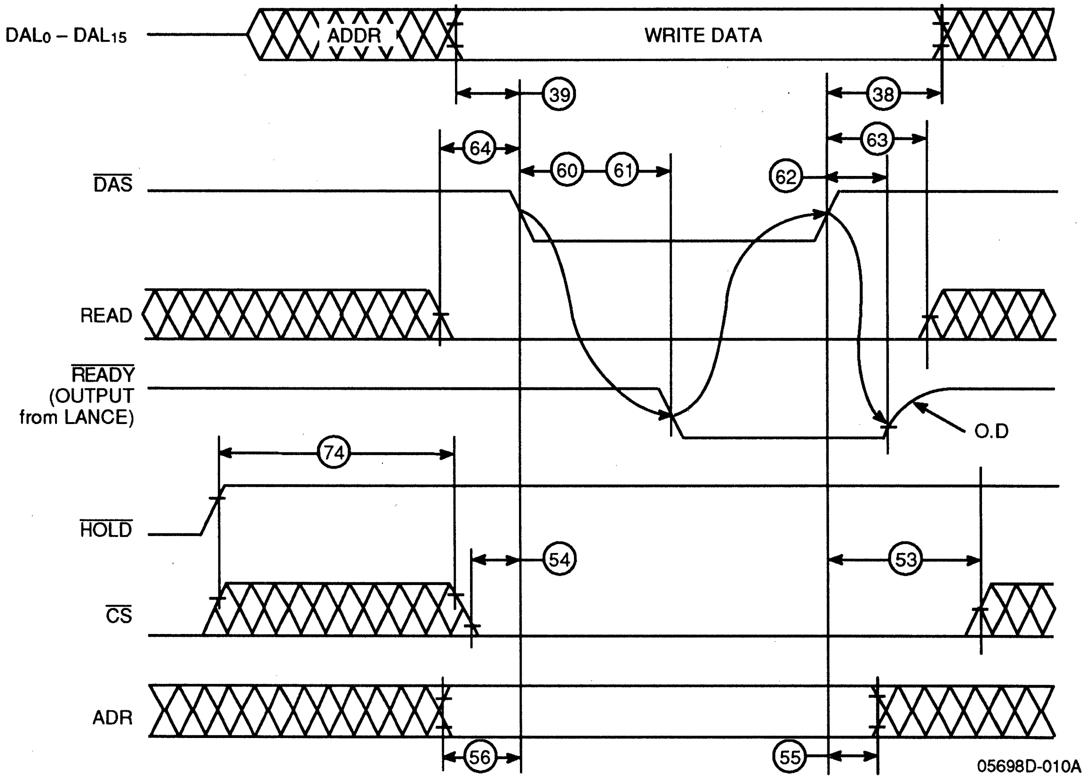
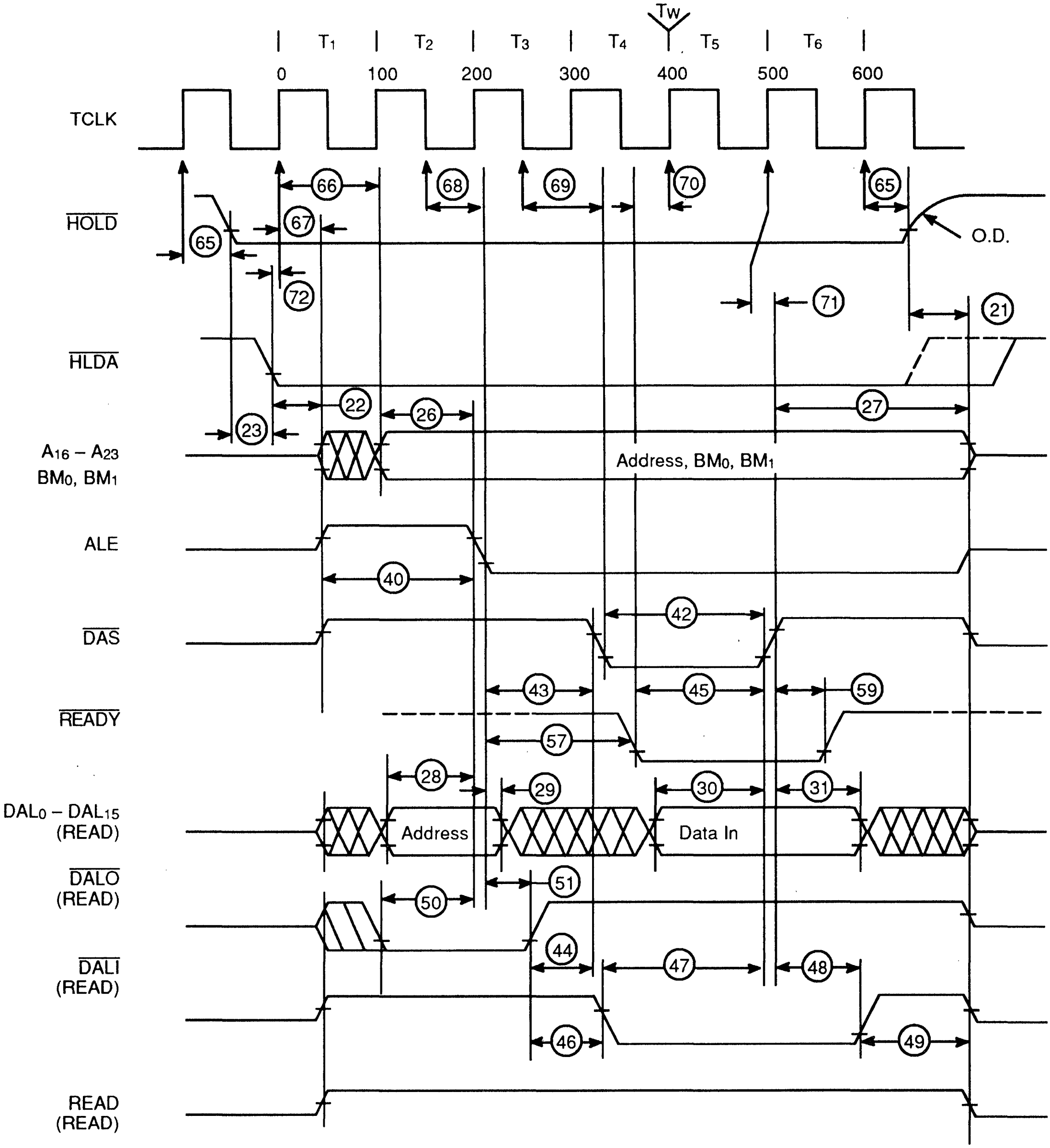


Figure 4. Bus Slave Write Timing

Write Sequence (Master Mode)

The write cycle is similar to the read cycle except that the DAL_{00} – DAL_{15} lines change from containing addresses to data after either ALE or $\overline{\text{AS}}$ goes inactive. After data is valid on the bus, $\overline{\text{DAS}}$ goes active. Data to memory is held valid after $\overline{\text{DAS}}$ goes inactive. Refer to Figure 5-2.



05698D-011A

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

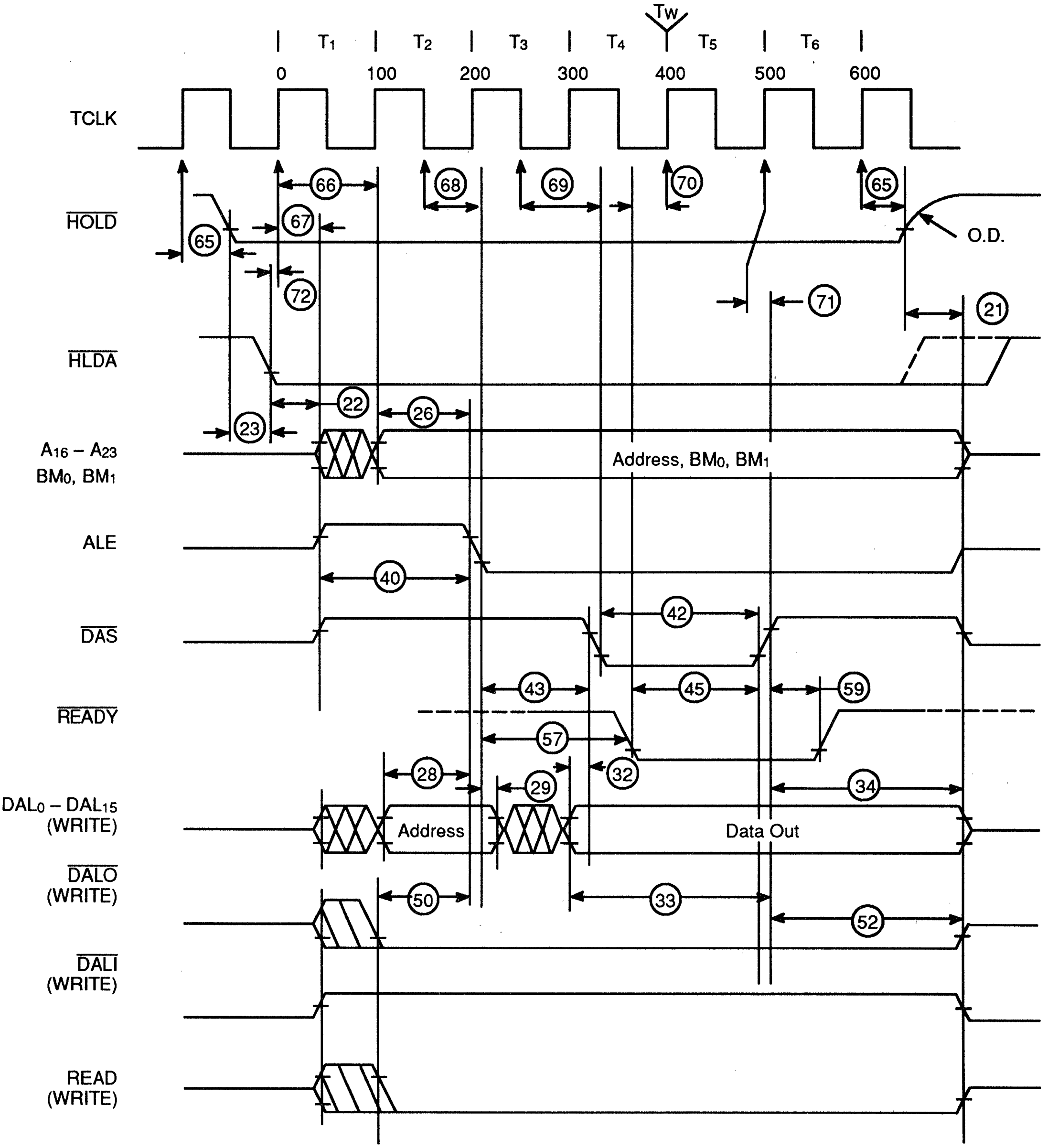
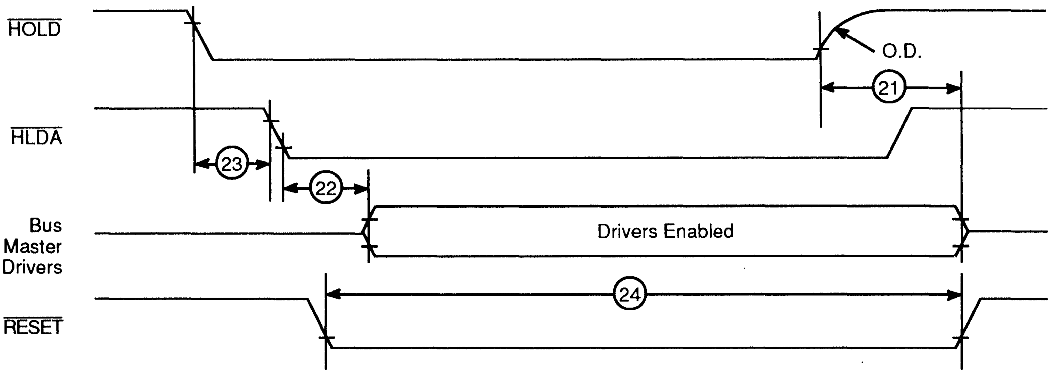


Figure 5-2. Bus Master Write Timing (Single DMA Cycle)



05698D-013A

Note:

1. RESET is an asynchronous input to the LANCE and is not part of the Bus Acquisition timing. When RESET is asserted, the LANCE becomes a Bus Slave.

Figure 6. Bus Acquisition Timing

Differences Between Ethernet Versions 1 and 2

a. Version 2 specifies that the collision detect of the transceiver must be activated during the interpacket 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 signifcant 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 Am7990 (LANCE).

Programming the Am7990 (LANCE)

The Am7990 (LANCE) is designed to operate in an environment that includes close coupling with local memory and microprocessor (HOST). The Am7990 LANCE is programmed by a combination of registers and data structures resident within the LANCE and memory registers. There are four Control and Status Registers (CSRs) within the LANCE which are programmed by the HOST device. Once enabled, the LANCE has the ability to access memory locations to acquire additional operating parameters.

The Am7990 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:

1. 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
2. Receive and Transmit Descriptor Rings – Two ring structures, one each 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
3. 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 LANCE may be summarized as:

1. Program the 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.
2. The LANCE loads itself with the information contained within the initialization block.
3. The 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

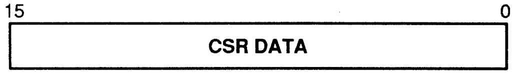
The CSRs are read (or written) in a two step operation. The address of the CSR to be accessed is written into the RAP during a bus slave transaction. During a subsequent bus slave transaction, the data being read from (or written into) the RDP is read from (or written into) the CSR selected in the RAP.

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

To distinguish the data port from the address port, a discrete I/O pin is provided.

ADR I/O Pin	Port
L	Register Data Port (RDP)
H	Register Address Port (RAP)

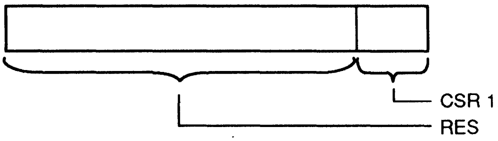
Register Data Port (RDP)



05698D-014A

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. CSR_1, CSR_2 and CSR_3 are accessible only when the STOP bit of CSR_0 is set. If the STOP bit is not set while attempting to access CSR_1, CSR_2 or CSR_3, the LANCE will return READY, but a READ operation will return undefined data. WRITE operation is ignored.

Register Address Port (RAP)



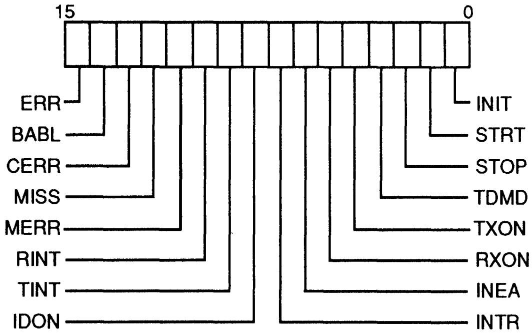
05698D-015A

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 $\overline{\text{RESET}}$.

CSR(1 :0)	CSR
00	CSR_0
01	CSR_1
1 0	CSR_2
1 1	CSR_3

Control and Status Register Definition

Control and Status Register 0 (CSR_0)



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

05698D-016A

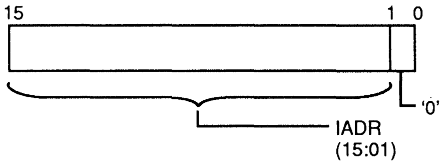
Bit	Name	Description
15	ERR	ERROR summary is set by the "ORing" of BABL, CERR, MISS and MERR. ERR remains set as long as any of the error flags are true. ERR is read only; writing it has no effect. It is cleared by Bus $\overline{\text{RESET}}$, setting the STOP bit, or clearing the individual error 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 data bytes have been transmitted; the 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 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.
13	CERR	COLLISION ERROR indicates that the collision input to the LANCE failed to activate within 2 µs after a LANCE-initiated transmission was 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 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. 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 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.

Bit	Name	Description
11	MERR	MEMORY ERROR is set when the LANCE is the Bus Master and has not received $\overline{READY}$ 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 (CSR_0, TXON = 0, RXON = 0) and an interrupt is generated if INEA = 1. MERR is READ/CLEAR ONLY, and is set by the LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{RESET}$ or by setting the STOP bit.
10	RINT	RECEIVER INTERRUPT is set when the 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 LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{RESET}$ or by setting the STOP bit.
09	TINT	TRANSMITTER INTERRUPT is set when the 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 LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{RESET}$ or by setting the STOP bit.
08	IDON	INITIALIZATION DONE indicates that the LANCE has completed the initialization procedure started by setting the INIT bit. When IDON is set, the 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 LANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{RESET}$ 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 $\overline{INTR}$ 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 $\overline{INTR}$ pin to be driven LOW when the Interrupt Flag is set. If INEA = 1 and INTR = 1, the $\overline{INTR}$ pin will be Low. If INEA = 0, the $\overline{INTR}$ 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 cannot be set while STOP bit is set. INEA can be set in parallel or after INIT and/or STRT bit are set.
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 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 $\overline{RESET}$ 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.

Bit	Name	Description
		TXON is READ ONLY; writing this bit has no effect. TXON is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
03	TDMD	TRANSMIT DEMAND, when set, causes the 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 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 $\overline{\text{RESET}}$ or by setting the STOP bit. Writing a "0" in this bit has no effect.
02	STOP	STOP disables the LANCE from all external activity when set and clears the internal logic. Setting STOP is the equivalent of asserting $\overline{\text{RESET}}$. The 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. CSR_1, CSR_2, and CSR_3 must be reloaded when the STOP bit is set.
01	STRT	START enables the 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 $\overline{\text{RESET}}$ or by setting the STOP bit.
00	INIT	INITIALIZE, when set, causes the 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 $\overline{\text{RESET}}$ or by setting the STOP bit. Since the setting of status bits in CSR_0 is independent of the timing of the slave read cycle, it is possible for external events to cause some of the bits to change in the middle of a read cycle. In particular the ERR, BABL, CERR, MISS, IDON, and INTR bits can change during a read cycle, while MERR, RINT and TINT can not. This is not a problem if CSR_0 is read only within the first few instructions of an interrupt service routine since the events that cause these bits to change are widely spaced in time relative to the time required to execute processor instructions.

Control and Status Register 1 (CSR_1)

READ/WRITE: Accessible only when the STOP bit of CSR_0 is a ONE and RAP = 01. Content of CSR_1 is not preserved after CSR_0 s STOP bit is set to one.

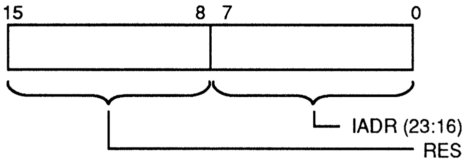


05698D-017A

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 (CSR_2)

READ/WRITE: Accessible only when the STOP bit of CSR_0 is a ONE and RAP = 10. Content of CSR_2 is not preserved after CSR's STOP bit is set to one.



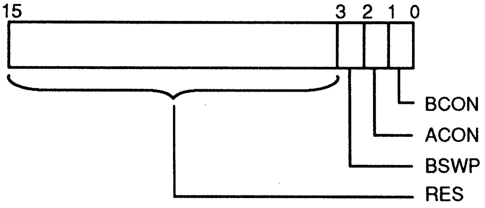
05698D-018A

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 (CSR_3)

CSR_3 allows redefinition of the Bus Master interface.

READ/WRITE: Accessible only when the STOP bit of CSR_0 is ONE and RAP = 11. CSR_3 is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit in CSR_0.



05698D-019A

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 LANCE will swap the high and low bytes on DMA data transfers between the FIFO and bus memory. Only data from 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 CSR_0.
01	ACON	ALE CONTROL defines the assertive state of ALE when the LANCE is a Bus Master. ACON is READ/WRITE and cleared by $\overline{\text{RESET}}$ and by setting the STOP bit in CSR_0.

ACON	ALE
0	Asserted HIGH
1	Asserted LOW

When ALE is programmed to be asserted LOW, a negative going pulse of less than 10 ns. duration can occur at the end of a bus master cycle just after $\overline{\text{HOLD}}$ is deasserted.

Bit	Name	Description
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 CSR_0.

BCON	Pin 16	Pin 15	Pin 17
0	$\overline{BM_1}$	$\overline{BM_0}$	$\overline{\text{HOLD}}$
1	$\overline{\text{BUSAKO}}$	BYTE	$\overline{\text{BUSRQ}}$

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

Initialization

Initialization Block

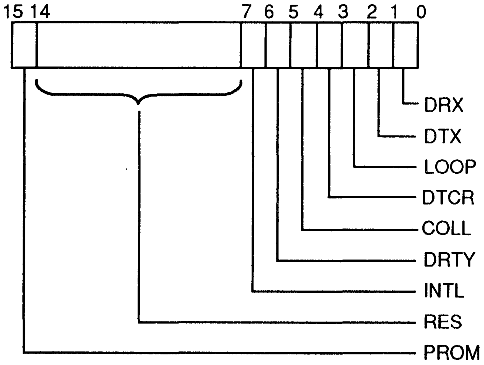
Chip initialization includes the reading of the initialization block in memory to obtain the operating parameters. The following is a definition of the Initialization Block.

The Initialization Block is read by the LANCE when the INIT bit in CSR_0 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 LANCE has read the Initialization Block, IDON is set in CSR_0 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 LANCE's operating parameters. Normal operation is with the Mode Register clear.



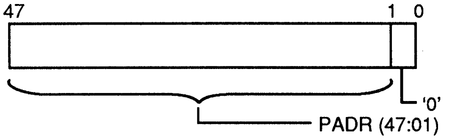
05698D-020A

Bit	Name	Description
15	PROM	PROMISCUOUS mode. When PROM = 1, all incoming packets are accepted.
14:07	RES	RESERVED. Read as zeroes. Write as zeroes.
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 LANCE is operational when the packets are addressed to the node itself. The Lance will not receive any packets externally when it is in internal loopback mode. EXTERNAL LOOPBACK allows the LANCE to transmit a packet through the SIA transceiver cable out to the Ethernet coax. It is used to determine the operability of all circuitry and connections between the LANCE and the coaxial cable. Multicast addressing in external loopback is valid only when DTCR = 1 (user needs to append the 4 bytes CRC). In external loopback, the LANCE also receives packets from other nodes. The FIFO READ/WRITE pointers may misalign in the 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.

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

05	DRTY	DISABLE RETRY. When DRTY = 1, the 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 (TMD_3).

Bit	Name	Description
04	COLL	FORCE COLLISION. This bit allows the collision logic to be tested. The 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 TMD$_3$.
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. 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 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 FIFO. The LANCE waits until the entire message is in the FIFO before serial transmission begins. The incoming data stream fills the FIFO from behind as it is being emptied. Moving the received message out of the FIFO to memory does not begin until reception has ceased. 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 LANCE to not access the Transmitter Descriptor Ring, and therefore, no transmissions are attempted. DTX = 1 will clear the TXON bit in CSR$_0$ when initialization is complete.
00	DRX	DISABLE THE RECEIVER causes the LANCE to reject all incoming packets and not access the Receive Descriptor Ring. DRX = 1 will clear the RXON bit in the CSR$_0$ when initialization is complete.



05698D-021A

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

Logical Address Filter

05698D-022A

Bit	Name	Description
63:00	LADRF	The 64-bit mask used by the 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 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.

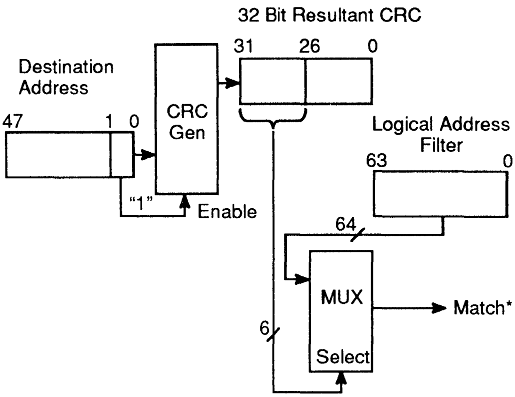
The logical address filter is a 64-bit mask composed of four sixteen-bit registers, LADRF (63:00) in the initialization block, that is used to accept incoming Logical Addresses. The incoming address is sent through the CRC circuit. After all 48 bits of the address have gone through the CRC circuit, the 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 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.

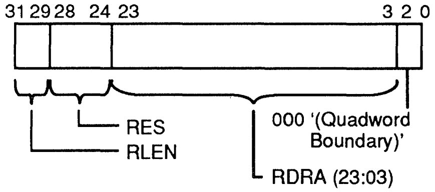


*Match - 1, the packet is accepted
Match - 0, the packet is rejected

05698D-023A

Figure 7. Logical Address Filter Operation

Receive Descriptor Ring Pointer

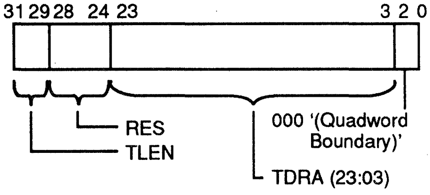


05698D-024A

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.

Bit	Name	Description
02:00		MUST BE ZEROES. These bits are RDRA (02:00) and must be zeroes because the Receive Rings are aligned on quadword boundaries.

Transmit Descriptor Ring Pointer



05698D-025A

Bit	Name	Description
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

Bit	Name	Description
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 Rings are aligned on quadword boundaries.

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. LANCE polling is limited to looking one ahead of the descriptor entry the 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 LANCE. Writing a "ONE" into the STRT bit of CSR$_0$ will cause the LANCE to start accessing the descriptor rings and enable it to send and receive packets.

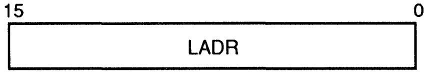
The LANCE communicates with a HOST device through the ring structures in memory. Each entry in the ring is either owned by the LANCE or the HOST. There is an ownership bit (OWN) in the message descriptor entry. Mutual exclusion is accomplished by a protocol which states that each device can only relinquish 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

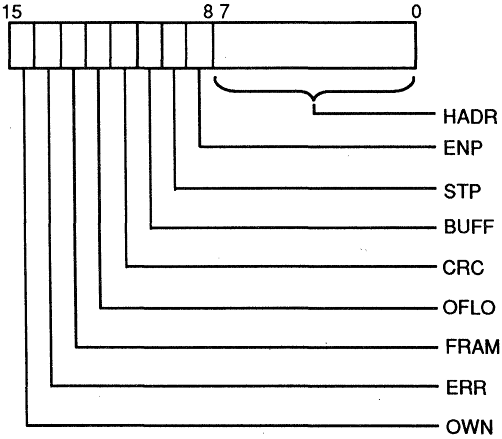
Receive Message Descriptor 0 (RMD$_0$)



05698D-026A

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 LANCE.

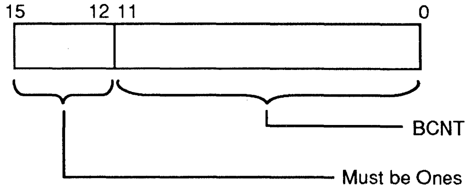
Receive Message Descriptor 1 (RMD$_1$)



05698D-027A

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = 0) or by the LANCE (OWN = 1). The 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 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.
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 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 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) FIFO overflow occurred before the LANCE received the next STATUS. 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.
09	STP	START OF PACKET indicates that this is the first buffer used by the 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 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 LANCE.

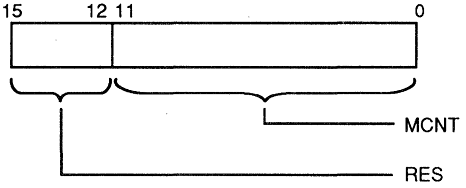
Receive Message Descriptor 2 (RMD$_2$)



05698D-028A

Bit	Name	Description
15:12		MUST BE ONES. This field is written by the host and is not changed by the 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 LANCE. Minimum buffer size is 64 bytes for the first buffer of packet.

Receive Message Descriptor 3 (RMD_3)

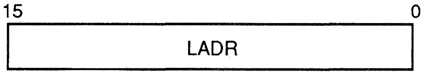


05698D-029A

Bit	Name	Description
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

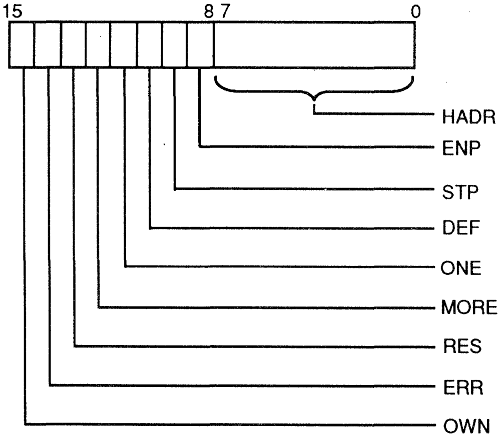
Transmit Message Descriptor 0 (TMD_0)



05698D-030A

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 LANCE.

Transmit Message Descriptor 1 (TMD_1)

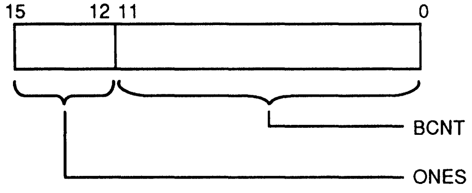


05698D-031A

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = O) or by the LANCE (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by this descriptor. The LANCE clears the OWN bit after transmitting the contents of the buffer. Neither the host nor the 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	RES	RESERVED bit. The LANCE will write this bit with a "0".
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 LANCE had to defer while trying to transmit a packet. This condition occurs if the channel is busy when the LANCE is ready to transmit.
09	STP	START OF PACKET indicates that this is the first buffer to be used by the LANCE for this packet. It is used for data chaining buffers. STP is set by the host and is not changed by the LANCE. The STP bit must be set in the first buffer of the packet, or the 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 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 LANCE.

Bit	Name	Description
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 LANCE.

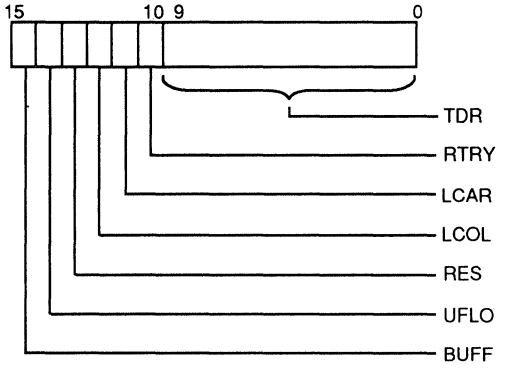
Transmit Message Descriptor 2 (TMD$_2$)



05698D-032A

Bit	Name	Description
15:12	ONES	Must be ones. This field is set by the host and is not changed by the 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 LANCE. This field is written by the host and is not changed by the 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 (TMD$_3$)



05698D-033A

Bit	Name	Description
15	BUFF	BUFFER ERROR is set by the LANCE during transmission when the 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 FIFO underflow occurred before the LANCE received the next STATUS signal. BUFF is set by the LANCE and cleared by the host. BUFF error will turn off the transmitter (CSR$_0$, 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 FIFO has emptied before the end of the packet was reached. Upon UFLO error, transmitter is turned off (CSR$_0$, TXON = 0).
13	RES	RESERVED bit. The 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 LANCE does not retry on late collisions.
11	LCAR	LOSS OF CARRIER is set when the carrier input (RENA) to the LANCE goes false during a LANCE-initiated transmission. The 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.

Bit	Name	Description
09:00	TDR	TIME DOMAIN REFLECTOMETRY reflects the state of an internal 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 LANCE and is valid only if RTRY is set.

Ring Access Mechanism in the LANCE

Once the LANCE is initialized through the initialization block and started, the CPU and the 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 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 LANCE obtained from the lookahead operation.

There are three types of ring access in the LANCE. The first type is when the LANCE polls the rings to own a buffer. The second type is when the buffers are data chained. The LANCE does a lookahead operation between the time that it is transferring data to/from the FIFO; this lookahead is done only once. The third type is when the 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 LANCE will automatically poll the transmit ring in the memory once it has started (CSR_0, STRT = 1). This polling occurs every 1.6 ms, (CSR_0 TDMD bit = 0) and consists of reading the status word of the transmit Ring, TMD_1, until the LANCE owns the descriptor. The LANCE will read TMD_0 and TMD_2 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 LANCE will look ahead to the next descriptor in the ring while transferring the current buffer into the FIFO (see Figure 8-1). The LANCE does this lookahead only once. If it does not own the next transmit Descriptor Table Entry (DTE) (2nd T_x 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 LANCE owns the 2nd DTE, it will also read the buffer address and the buffer byte count of this entry. Once the LANCE has finished emptying the current buffer, it clears the OWN bit for this buffer, and immediately starts loading the FIFO from the next (2nd) buffer. Between DMA bursts, starting from the 2nd buffer, the 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 (TMD_1, ENP = 1). Once the last part of the packet has been transmitted out from the FIFO to the cable, the LANCE will update the status in TMD_1, TMD_3 (TMD_3 is updated only when there is an error) and will relinquish the last buffer to the CPU. The 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 LANCE does not own the next buffer, it then polls the T_x ring every 1.6 ms.

When an error occurs before all of the buffers get transmitted, the status, TMD_3, is updated in the current DTE, own bit is cleared in TMD_1, and TINT bit is set in CSR_0 which causes an interrupt if INEA = 1. The LANCE will then skip over the rest of the descriptors for this packet (clears the OWN bit and sets the TINT bit in CSR_0) 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 LANCE will not perform any lookahead operation. It will transmit the current buffer, update the TMD_3 if any error, and then update the status and clear the OWN bit in TMD_1. The LANCE will then immediately check the next descriptor in the ring to see if it owns it. If it does, the LANCE will also read the rest of the entries from the descriptor table. If the 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 CSR_0 when it has relinquished a buffer to the LANCE. This will force the LANCE to check the OWN bit at this buffer without waiting for the polling time to elapse.

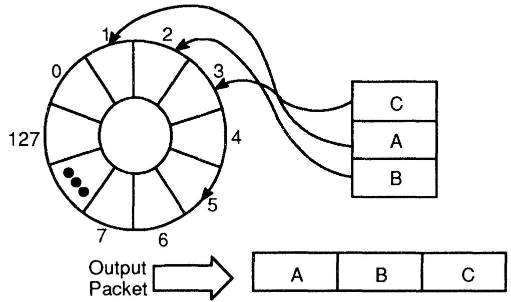
Receive Ring Buffer Management

Receive Ring access is similar to the transmit ring access. Once the receiver is enabled, the LANCE will always try to have a receive buffer available, should there be a packet addressed to this node for reception. Therefore, when the LANCE is idle, it will poll the receive ring entry, once every 1.6 ms, until it owns the current receive DTE. Once the LANCE owns the buffer, it will read RMD_0 and RMD_2 to get the rest of buffer address and buffer byte count. When a packet arrives from the cable, after the Address Recognition Logic accepts the packet, the 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 CSR_0 and will not poll the receive ring until the packet ends.

Assuming the 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 FIFO to the first receive buffer in case the current buffer requires data chaining. When the LANCE owns the buffer, the lookahead operation consists of three separate single word DMA reads: RMD_1, RMD_0, and RMD_2. When the LANCE does not own the next buffer, the lookahead operation consists of only one single DMA read, RMD_1. Either lookahead operation is done only once. Following the lookahead operation, whether LANCE owns the next buffer or not, the LANCE will transfer the data from FIFO to the first receive buffer for this packet in burst mode (8 word transfer per one DMA cycle arbitration).

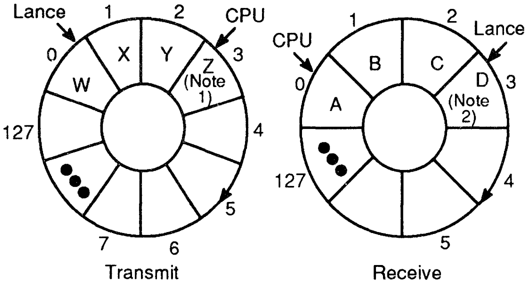
If the packet being received requires data chaining, and the LANCE does not own the 2nd DTE, the LANCE will update the current buffer status, RMD_1, with the BUFF and/or OFLO error bits set. If the LANCE does own the next buffer (2nd DTE) from previous lookahead, the LANCE will relinquish the current buffer and start filling up the 2nd buffer for this packet. Between the time that the LANCE is transferring data from the FIFO to 2nd buffer, it does a lookahead operation again to see if it owns the next (3rd) buffer. If the LANCE does own the third DTE, it will also read RMD_0, and RMD_2 to get the rest of buffer pointer address and buffer byte count.

This activity continues on until the LANCE recognizes the end of the packet (cable is idle); it then updates the current buffer status with the end of packet bit (ENP) set. The LANCE will also update the message byte count (RMD_2) with the total number of bytes received for this packet in the current buffer (the last buffer for this packet).



05698D-034A

Figure 8-1. Data Chaining (Transmit)



05698D-035A

Notes:

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

Figure 8-2. Buffer Management Descriptor Rings

LANCE DMA Transfer (Bus Master Mode)

There are two types of DMA Transfers with the 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 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 FIFO, in transmit mode, or at least 16 bytes of data, in the FIFO in receive mode, when the LANCE releases the bus ($\overline{HOLD}$ deasserted), the LANCE will request the bus again within 700 ns. ($\overline{HOLD}$ dwell time). Burst DMAs are always 8 cycle transfers unless there are fewer than 8 words left to be transferred in to/from the SILO.

Single Word DMA Transfer

The LANCE initiates single word DMA transfers to access the transmit, receive rings or initialization block. The LANCE will not initiate any burst DMA transfer between the time that it gets to own the descriptor, and accessing the descriptor entries in the ring (an average of 3–4 separate DMA cycles for a multibuffer packet) or reading the initialization block.

Bus Latency Requirements

If the time between $\overline{HOLD}$ and $\overline{HLDA}$ is such that three consecutive single word DMA transfers can take more than 33.5 µsec., under certain rather unusual conditions the receiver can lock up and stop receiving packets. This problem occurs if during the time that the LANCE is polling a descriptor ring, a packet addressed to this node arrives and causes the FIFO to overflow before the polling is complete.

If the system design can not guarantee a short enough bus latency, the problem can be solved by either external hardware or software. For the hardware solution, an external circuit could interrupt the processor if fewer than 3 DMA transfers occur within 47 µsec. after the RENA signal goes active. This interrupt would signal the software to reset the LANCE by setting the STOP bit in CSR_0.

For a software solution a timer interrupt can cause the software to reset the LANCE after no packets have been received for a certain period of time. The length of this time period can vary with the amount of traffic on the network. When traffic is heavy, the timeout delay should be short. When traffic is light, the timeout delay can be made longer.

FIFO Operation

The FIFO provides temporary buffer storage for data being transferred between the parallel bus I/O pins and serial bus I/O pins. The capacity of the FIFO is 48 bytes.

Transmit

Data is loaded into the FIFO under internal microprogram control. FIFO has to have more than 16 bytes empty before the LANCE requests the bus ($\overline{\text{HOLD}}$ is asserted). The LANCE will start sending the preamble (if the line is idle) as soon as the first byte is loaded to the FIFO from memory. Should transmitter be required to back off, there could be up to 32 bytes of data in the FIFO ready for transmission. Reception has priority over transmission during the time that the transmitter is backing off.

Receive

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

FIFO – Memory Byte Alignment

Memory buffers may begin and end on arbitrary byte boundaries. Parallel data is byte aligned between the FIFO and DAL lines (DAL$_0$–DAL$_{15}$). Byte alignment can be reversed by setting the Byte Swap (BSWP) bit in CSR$_3$.

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

TRANSMISSION – BYTE READ FROM ODD MEMORY ADDRESS

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

RECEPTION – WORD WRITE TO EVEN MEMORY ADDRESS

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

RECEPTION – BYTE WRITE TO EVEN MEMORY ADDRESS

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

RECEPTION – BYTE WRITE TO ODD MEMORY ADDRESS

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

The LANCE Recovery and Reinitialization

The transmitter and receiver section of the 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 CSR$_0$ register (RXON, TXON bits). The 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. Another reason why it may be desirable to reinitialize the LANCE, to turn the transmitter and/or receiver back on again, is when either section shuts off because of an error (MERR, UFLO, TX BUFF error). Care must be taken when the LANCE is reinitialized. 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 LANCE, the user must stop the LANCE by setting the stop bit in CSR$_0$ prior to reinitialization (setting INIT bit in CSR$_0$). The user needs to reprogram CSR$_3$ because its content gets cleared when the stop bit gets set (soft reset). CSR$_3$ reprogramming is not needed when default values BCON, ACON, and BSWP are used. CSR$_1$ and CSR$_2$ must be reloaded after the STOP bit is set.

It is recommended that the LANCE not be re-started, once it has been stopped (STOP = 1 in CSR$_0$), by simply setting the STRT bit in CSR$_0$. Re-starting the LANCE by setting the STRT bit puts the LANCE in operation in accordance with the parameters set up in the mode register. However, contents of the descriptor pointers in the LANCE are not guaranteed upon re-start.

Frame Formatting

The LANCE performs the encapsulation/decapsulation function of the data link layer (2nd 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 LANCE will append the preamble, synch, and CRC (FCS) to the frame as is shown in Figures 9-1 and 9-2.

Receive

In receive mode, the LANCE strips off the preamble and synch bits and transfers the rest of the frame, including the CRC bytes (4 bytes), to the memory. The 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. A runt packet is normally the result of a collision.

Preamble 1010 ... 1010	Synch 1 1	Dest. ADR	Source ADR	Type	Data	FCS
62 Bits	2 Bits	6 Bytes	6 Bytes	2 Bytes	46–1500 Bytes	4 Bytes

05698D-036A

Figure 9-1. Ethernet Frame Format

Preamble 1010 ... 1010	SFD 10101011	Dest. ADR	Source ADR	Length	LLC Data	PAD	FCS
56 Bits	8 Bits	6 Bytes	6 Bytes	2 Bytes	46–1500 Bytes		4 Bytes

05698D-037A

Figure 9-2. IEEE 802.3 MAC Frame Format

Framing Error (Dribbling Bits)

The LANCE can handle up to 7 dribbling bits when a received packet terminates; the input to the 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 cable, and it gets stored internally on byte boundary. The framing error is reported to the user as follows:

- If the number of the dribbling bits are 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 are 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.

Interpacket Gap Time (IPG)

The interpacket gap time for back-to-back transmission is 9.6 to 10.6 microseconds, including synchronization. The interpacket delay interval begins immediately after the negation of the RENA signal. During the first 4.1 µs of the IPG, RENA activity is masked off internally in the LANCE. If RENA is asserted and remains asserted during the first 4.1 µs of IPG following a receive, the LANCE will defer to the packet (it will not receive it). If this condition occurs following a transmit, the LANCE will start to look for the synch bits (011) and about 800 ns (8 bit time) after the 4.1 µs window has elapsed. Therefore, the packet may be received correctly if at least 8 bits of the preamble are left following the 4.1 µs window, or the received packet may contain CRC error (not enough preamble bits left, LANCE may be locking to the synch bits in the middle of data), or the received packet may be discarded because of the runt packet filter (some data is lost during the 4.1 µs window).

If RENA is asserted after the 4.1 µs window, the LANCE will treat this as the start of a new packet. It will start to look for the synch bits (011) 8-bit time after RENA becomes active. Whenever the LANCE is about to transmit and is waiting for the interpacket delay to elapse, it will begin transmission immediately after the interpacket delay interval, independent of the state of RENA. However, RENA must be asserted during the time that TENA is high. The LCAR (loss of carrier) error bit is otherwise set in TMD3, after the packet has been transmitted.

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 LANCE continues to send the preamble, and sends the JAM pattern following the preamble. If collision occurs after the preamble, the 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, the following will occur. A collision that occurs within 6 byte times (4.8 µs) will result in the packet being rejected because of an address mismatch with the FIFO write pointer being 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 recognized 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 LANCE will attempt to retry it 15 more times. The LANCE does not try to reread the descriptor entries from the Tx ring upon each collision. The descriptor entries for the current buffer are internally saved. 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 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).

If all 16 attempts fail, the LANCE sets the RTRY bit in the current Transmit Message Descriptor 3, TMD_3, 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 LANCE will not transmit again; it will terminate the transmission, note the LCOL error in TMD_3, 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 FIFO in anticipation of retransmission. It is important that LANCE be ready to transmit when the backoff interval elapses to utilize the channel properly.

Time Domain Reflectometry

The 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, that 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 2 µs following the completion of transmission, then the LANCE will set the CERR bit in CSR_0. CERR error will not cause an interrupt to occur (INTR = 0).

Cyclic Redundancy Check (CRC)

The LANCE utilizes the 32-bit CRC function used in the Autodin-II network. Refer to the Ethernet specification (section 6.2.4 Frame Check Sequence Field and Appendix C; CRC Implementation) for more detail. The LANCE requirements for the CRC logic are the following:

1. TRANSMISSION – MODE <02> LOOP = 0, MODE <03> DTCR = 0. The LANCE calculates the CRC from the first bit following the Start bit to the last bit of the data field. The CRC value inverted is appended onto the transmission in one unbroken bit stream.
2. RECEPTION – MODE <02> LOOP = 0. The LANCE performs a check on the input bit stream from the first bit following the Start bit to the last bit in the frame. The 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.
3. LOOPBACK – MODE <02> LOOP =1, MODE <03> DTRC = 0. The 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.
4. LOOPBACK – MODE <02> LOOP = 1 MODE <03> DTRC = 1. 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 LANCE is as a half-duplex device. However, to provide an on-line operational test of the 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:

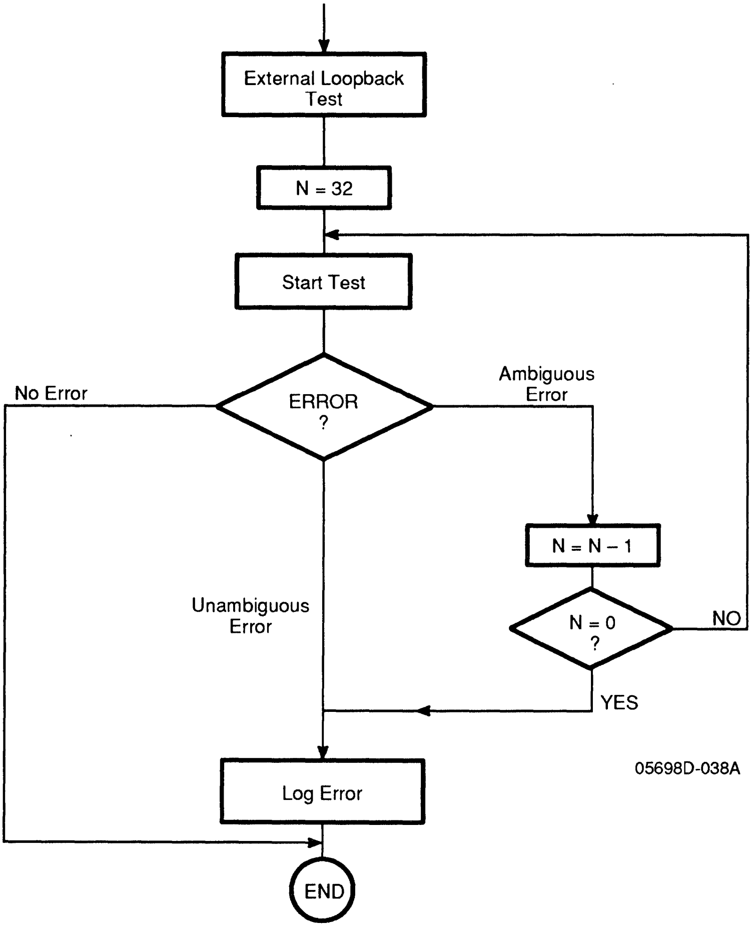
1. The packet length must be no longer than 32 bytes, and no shorter than eight bytes, exclusive of the CRC.
2. Serial transmission does not begin until the FIFO contains the entire output packet.
3. Moving the input packet from the FIFO to the memory does not begin until the serial input bit stream terminates.
4. CRC may be generated and appended to the output serial bit stream or may be checked on the input serial bit stream, but not both in the same transaction.
5. In internal loopback, the packets should be addressed to the node itself.
6. 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.

External Loopback Test Procedure

If the LANCE performs an external loopback test in a live network in which packets can be sent to the LANCE during the loopback test, a failure can be reported when in fact there is no problem. This problem occurs when a packet addressed to this node (or a broadcast packet) arrives after the LANCE has started to load the FIFO for loopback transmission but before the LANCE has started to transmit. When this happens, the data in the FIFO can become corrupted such that the data transmitted are not the same as the data in the transmit buffer. As a result CRC, OFLO, UFLO, RTRY, or LCAR errors can be reported when there is nothing wrong with the system.

Therefore to eliminate false errors, the external loopback routine should run the test a predetermined number of times (20 to 30 times are more than enough) or until the test passes or until an unambiguous error (BABL, CERR, MISS, MERR, FRAM, BUFF, or LCOL) occurs.



N = Max. number of times to repeat the test.

Figure 9. External Loopback Test Flow Chart

Serial Transmission

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

1. Preamble/Start bit: 62 alternating ONES and ZEROES terminating with the synch in two ONEs. The last ONE is the Start bit.
2. Data: The serialized byte stream from the FIFO Shifted out with LSB first.
3. CRC: The inverted 32-bit polynomial calculated from the Data, address, and type field. CRC is not transmitted if:
   i. Transmission of the Data field is truncated for any reason.
   ii. CLSN becomes asserted any time during transmission.
   iii. MODE <03> DTCR = 1 in a normal or loopback transmission mode.

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 LANCE starts transmitting the preamble when the following are satisfied:

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

Serial Reception

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

1. Preamble/Start bit: Two ONES occurring a minimum of 8 bit times after the assertion of RENA. The last ONE is the Start bit.
2. Destination Address: The 48 bits (6 bytes) following the Start bit.
3. Data: The serialized byte 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 FIFO. Source Address and Type field are part of the data which are transparent to the 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 LANCE does not not sample the received data until about 800 ns AFTER RENA goes high.

APPENDIX A

8086 computer program example to generate the hash filter, for multicast addressing in the LANCE.

```
                  ;         SUBROUTINE TO SET A BIT IN THE HASH FILTER FROM A
                  ;         GIVEN ETHERNET LOGICAL ADDRESS
                  ;         ON ENTRY SI POINTS TO THE LOGICAL ADDRESS WITH LSB FIRST
                  ;                  DI POINTS TO THE HASH FILTER WITH LSB FIRST
                  ;         ON RETURN SI POINTS TO THE BYTE AFTER THE LOGICAL ADDRESS
                  ;             ALL OTHER REGISTERS ARE UNMODIFIED
                  ;
                            PUBLIC SETHASH
                            ASSUME CS:CSE61
                  ;
= 1DB6            POLYL     EOU      1DB6H      ;CRC POLYNOMINAL TERMS
= 04C1            POLYH     EQU      04C1H
                  ;
0000              CSE61     SEGMENT PUBLIC ‘CODE’
                  ;
0000              SETHASH   PROC     NEAR
0000 50                     PUSH     AX         ;SAVE ALL REGISTERS
0001 53                     PUSH     BX
0002 51                     PUSH     CX
0003 52                     PUSH     DX
0004 55                     PUSH     BP
                  ;
0005 B8 FFFF                MOV      AX,0FFFFH  ;AX,DX =CRC ACCUMULATOR
0008 BA FFFF                MOV      DX,0FFFFH  ;PRESET CRC ACCUMULATOR TO ALL 1'S
000B B5 03                  MOV      CH,3       ;CH =WORD COUNTER
                  ;
000D 8B 2C        SETH10:   MOV      BP,[S1]    ;GET A WORD OF ADDRESS
000F 83 C6 02               ADD      S1,2       ;POINT TO NEXT ADDRESS
0012 B1 10                  MOV      CL,16      ;CL=BIT COUNTER
                  ;
0014 8B DA        SETH20:   MOV      BX,DX      ;GET HIGH WORD OF CRC
0016 D1 C3                  ROL      BX,1       ;PUT CRC31 TO LSB
0018 33 DD                  XOR      BX,BP      ;COMBINE CRC31 WITH INCOMING BIT
001A D1 E0                  SAL      AX,1       ;LEFT SHIFT CRC ACCUMULATOR
001C D1 D2                  RCL      DX,1
001E 81 E3 0001             AND      BX,0001H   ;BX=CONTROL BIT
0022 74 07                  JZ       SETH30     ;DO NOT XOR IF CONTROL BIT = 0
                  ;
                  ;         PERFORM XOR OPERATION WHEN CONTROL BIT= 1
                  ;
0024 35 1D 86               XOR      AX,POLYL
0027 81 F2 04C1             XOR      DX,POLYH
                  ;
002B 0B C3        SETH30:   OR       AX,BX      ;PUT CONTROL BIT IN CRC0
002D D1 CD                  ROR      BP,1       ;ROTATE ADDRESS WORD
```

APPENDIX A (Continued)

```
002F FE C9                    DEC     CL            ;DECREMENT BIT COUNTER
0031 75 E1                    JNZ     SETH20
0033 FE CD                    DEC     CH            ;DECREMENT WORD COUNTER
0035 75 D6                    JNZ     SETH10
                  ;           FORMATION OF CRC COMPLETE, AL CONTAINS THE REVERSED HASH
                  ;           CODE
0037 B9 000A                  MOV     CX,10
003A D0 E0        SETH40:     SAL     AL,1          ;REVERSE THE ORDER OF BITS IN AL
003C D0 DC                    RCR     AH,1          ;AND PUT IT IN AH
003E E2 FA                    LOOP    SETH40

                  ;           AH NOW CONTAINS THE HASH CODE
                  ;
0040 8A DC                    MOV     BL,AH         ;BL = HASH CODE, BH IS ALREADY ZERO
0042 B1 03                    MOV     CL,3          ;DIVIDE HASH CODE BY 8
0044 D2 EB                    SHR     BL,CL         ;TO GET TO THE CORRECT BYTE
0046 B0 01                    MOV     AL,01H        ;PRESET FILTER BIT
0048 80 E45 07                AND     AH,7H         ;EXTRACT BIT COUNT
004B 8A CC                    MOV     CL,AH
004D D2 E0                    SHL     AL,CL         ;SHIFT BIT TO CORRECT POSITION
004F 08 01                    OR      [DI + BX],AL  ;SET IN HASH FILTER
0051 5D                       POP     BP
0052 5A                       POP     DX
0053 59                       POP     CX
0054 5B                       POP     BX
0055 58                       POP     AX
0056 C3                       RET
                  ;
0057              SETHASH     ENDP
                  ;
0057              CSEG1       ENDS
                  ;
                              END
```

Program example in BASIC to generate the hash filter, for multicast addressing, in the 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 ""
```

APPENDIX A (Continued)

```
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
```

APPENDIX A (Continued)

```
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 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 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];
}
```

The table "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.

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 OO 00 00 00 00
3	7	31	E9 00 00 00 00 00	7	7	63	4D 00 00 00 00 00

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	−25 to +125°C
Supply Voltages to Ground Potential Continuous	−0.3 V to +7 V
Commercial Power Dissipation	1.5 W

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 to +70°C
Supply Voltage (V_{CC})	+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 (Except RX, TCLK)				0.8	V
V_{IH}	Input HIGH Voltage (Except RX, TCLK)		2			V
V_{LL}	Input LOW Voltage (RX, TCLK)				0.8	V
V_{CH}	Input HIGH Voltage (RX, TCLK)		2			V
V_{OL}	Output LOW Voltage	COM'L I_{OL} = 3.2 mA MIL I_{OL} = 1.6 mA			0.5	V
V_{OH}	Output HIGH Voltage	COM'L I_{OH} = −0.4 mA MIL I_{OH} = −0.2 mA	2.4			V
I_{IL}	Input Leakage	V_{IN} = 0.4 V to V_{CC}			±10	µA
I_{CC}**	Power Supply Current			200	270	mA

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

CAPACITANCE* (T_A = 25°C; V_{CC} = 0)

Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Unit
C_{IN}	Input Capacitance	F = 1 MHz			10	pF
C_{OUT}	Output Capacitance	F = 1 MHz			15	pF
C_{IO}	Capacitance	F = 1 MHz			20	pF

*Parameters are not tested.

SWITCHING CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified

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			ns
3	t_{TCH}	TCLK HIGH Time		45			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	C_L = 50 pF			70	ns
7	t_{TEH}	TENA Hold Time After the Rising Edge of TCLK	C_L = 50 pF	5			ns
8	t_{TDP}	TX Data Propagation Delay After the Rising Edge of TCLK	C_L = 50 pF			70	ns
9	t_{TDH}	TX Data Hold Time After the Rising Edge of TCLK	C_L = 50 pF	5			ns
10	t_{RCT}	RCLK Period	(Note 3)	85		118	ns
11	t_{RCH}	RCLK HIGH Time		38			ns
12	t_{RCL}	RCLK LOW Time		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)		5			ns
18	t_{RDS}	RX Data Setup Time (RX Data Stable to the Rising Edge of RCLK)		40			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 $\overline{\text{HOLD}}$				50	ns
22	t_{DON}	Bus Master Driver Disable After Falling Edge of $\overline{\text{HLDA}}$				2t_{TCT} + 50	ns
23	t_{HHA}	Delay to Falling Edge of $\overline{\text{HLDA}}$ from Falling Edge of $\overline{\text{HOLD}}$ (Bus Master)		0			ns
24	t_{RW}	$\overline{\text{RESET}}$ Pulse Width LOW		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 $\overline{\text{DAS}}$		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 $\overline{\text{DAS}}$ (Bus Master Read)		50			ns

SWITCHING CHARACTERISTICS (Continued)

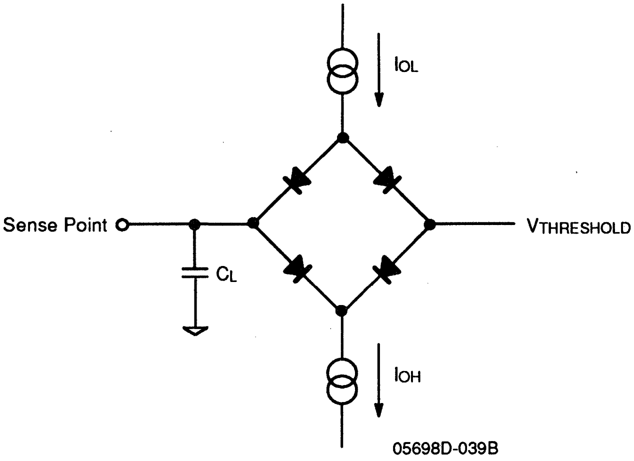
No.	Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Unit
31	t_{RDAH}	Data Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Master Read)		0			ns
32	t_{DDAS}	Data Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (Bus Master Write)		10			ns
33	t_{WDS}	Data Setup Time to the Rising Edge of $\overline{\text{DAS}}$ (Bus Master Write)		200			ns
34	t_{WDH}	Data Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Master Write)		35			ns
35	t_{SD01}	Data Driver Delay After the Falling Edge of $\overline{\text{DAS}}$ (Bus Slave Read)	(CRS 0, 3, RAP)		4t_{TCT}		ns
36	t_{SD02}	Data Driver Delay After the Falling Edge of $\overline{\text{DAS}}$ (Bus Slave Read)	(CSR 1, 2)		12t_{TCT}		ns
37	t_{SRDH}	Data Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Slave Read)		0		55	ns
38	t_{SWDH}	Data Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Slave Write)		0			ns
39	t_{SWDS}	Data Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (Bus Slave Write)		0			ns
40	t_{ALEW}	ALE Width HIGH		120			ns
41	t_{DALE}	Delay from Rising Edge of $\overline{\text{DAS}}$ to the Rising Edge of ALE		70			ns
42	t_{DSW}	$\overline{\text{DAS}}$ Width LOW		200			ns
43	t_{ADAS}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{\text{DAS}}$		80			ns
44	t_{RIDF}	Delay from the Rising of $\overline{\text{DALO}}$ to the Falling Edge of $\overline{\text{DAS}}$ (Bus Master Read)		15			ns
45	t_{RDYS}	Delay from the Falling Edge of $\overline{\text{READY}}$ to the Rising Edge of $\overline{\text{DAS}}$		75		250	ns
46	t_{ROIF}	Delay from the Rising Edge of $\overline{\text{DALO}}$ to the Falling Edge of $\overline{\text{DALI}}$ (Bus Master Read)		15			ns
47	t_{RIS}	$\overline{\text{DALI}}$ Setup Time to the Rising Edge of $\overline{\text{DAS}}$ (Bus Master)		135			ns
48	t_{RIH}	$\overline{\text{DALI}}$ Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Master Read)		0			ns
49	t_{RIOF}	Delay from the Rising Edge of $\overline{\text{DALI}}$ to the Falling Edge of $\overline{\text{DALO}}$ (Bus Master Read)		55			ns
50	t_{OS}	$\overline{\text{DALO}}$ and READ Setup Time to the Falling Edge of ALE (Bus Master Write and Read)		110			ns
51	t_{ROH}	$\overline{\text{DALO}}$ Hold Time After the Falling Edge of ALE (Bus Master Read)		35			ns
52	t_{WDSI}	Delay from the Rising Edge of $\overline{\text{DAS}}$ to the Rising Edge of $\overline{\text{DALO}}$ (Bus Master Write)		35			ns
53	t_{CSH}	$\overline{\text{CS}}$ Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Slave)		0			ns
54	t_{CSS}	$\overline{\text{CS}}$ Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (Bus Slave)		0			ns

SWITCHING CHARACTERISTICS (Continued)

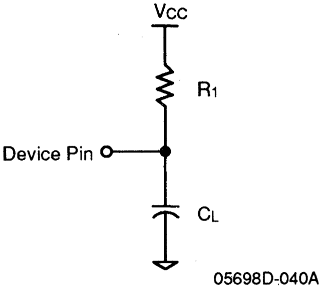
No.	Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Units
55	t_{SAH}	ADR Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Slave)		0			ns
56	t_{SAS}	ADR Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (Bus Slave)		0			ns
57	t_{ARYD}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{\text{READY}}$ to insure a Minimum Bus Cycle Time (600 ns)	(Note 5)			80	ns
58	t_{SRDS}	Data Setup Time to the Falling Edge of $\overline{\text{READY}}$ (Bus Slave Read)		75			ns
59	t_{RDYH}	$\overline{\text{READY}}$ Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Master)		0			ns
60	t_{SR01}	$\overline{\text{READY}}$ Driver Turn On After the Falling Edge of $\overline{\text{DAS}}$ (Bus Slave)	(CSR 0, 3, RAP) (Notes 4, 6)		6t_{TCT}		ns
61	t_{SR02}	$\overline{\text{READY}}$ Driver Turn On After the Falling Edge of $\overline{\text{DAS}}$ (Bus Slave)	(CSR 1, 2) (Note 6)		14t_{TCT}		ns
62	t_{SRYH}	$\overline{\text{READY}}$ Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Slave)		0		35	ns
63	t_{SRH}	READ Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Slave)		0			ns
64	t_{SRS}	READ Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (Bus Slave)		0			ns
65	t_{CHL}	TCLK Rising Edge to 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 $\overline{\text{DAS}}$ Falling Edge				90	ns
70	t_{RCS}	Ready Setup Time to TCLK	(Note 5)	0			ns
71	t_{CDH}	TCLK Rising Edge to $\overline{\text{DAS}}$ HIGH				90	ns
72	t_{HCS}	HLDA Setup to TCLK		0			ns
73	t_{RENH}	RENA Hold Time After the Rising Edge of RCLK		0			ns
74	t_{CSR}	$\overline{\text{CS}}$ recovery time between deassertion of $\overline{\text{CS}}$ or $\overline{\text{HOLD}}$ and assertion of $\overline{\text{CS}}$		t_{TCT}+60			ns

Notes:

1. Not shown in the timing diagrams, specifies the minimum bus cycle for a single DMA 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); RCLK and TCLK LOW/HIGH times tested at Min./Max. and Max./Min. specifications.
3. Not tested.
4. CRS0 write access time (t_{SR01}) when STOP bit is set can be as long as 12t_{TCT}.
5. The $\overline{\text{READY}}$ Setup time before negation of $\overline{\text{DAS}}$ is a function of the synchronization time of $\overline{\text{READY}}$. The synchronization must occur within 100 ns. Therefore, the setup time is 100 ns plus any accumulated propagation delays. Ready slips occur on 100 ns increments. It is guaranteed that no wait states will be added by the LANCE if either parameter #57 or #70 is met. Parameter #70 is intended for systems in which TCLK is synchronized with the processor bus interface. Parameter #57 is intended for asynchronous systems.
6. Parameter is for design reference only. Functional testing uses typical value ± 1 T_{TCT}.



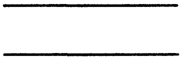
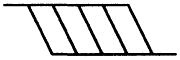
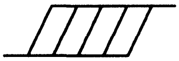
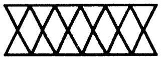
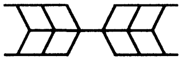
A. Normal & Three-State Outputs



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

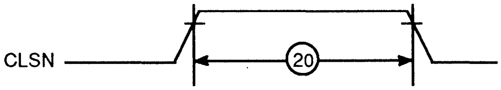
TEST OUTPUT LOADS			
Pin Name	**Test Circuit**	**R_1 (kΩ)**	**C_L (pF)**
All Outputs and I/O Pins except $\overline{\text{INTR}}$, $\overline{\text{HOLD}}/\overline{\text{BUSRQ}}$, $\overline{\text{READY}}$	A	–	100
$\overline{\text{INTR}}$, $\overline{\text{HOLD}}/\overline{\text{BUSRQ}}$, $\overline{\text{READY}}$	B	1.5	50

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

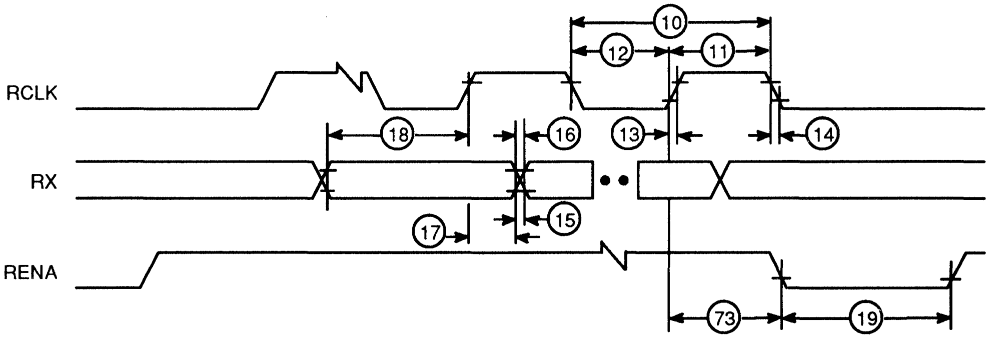
KS000010

SWITCHING WAVEFORMS (Note 1)



Serial Link Timing (Collision)

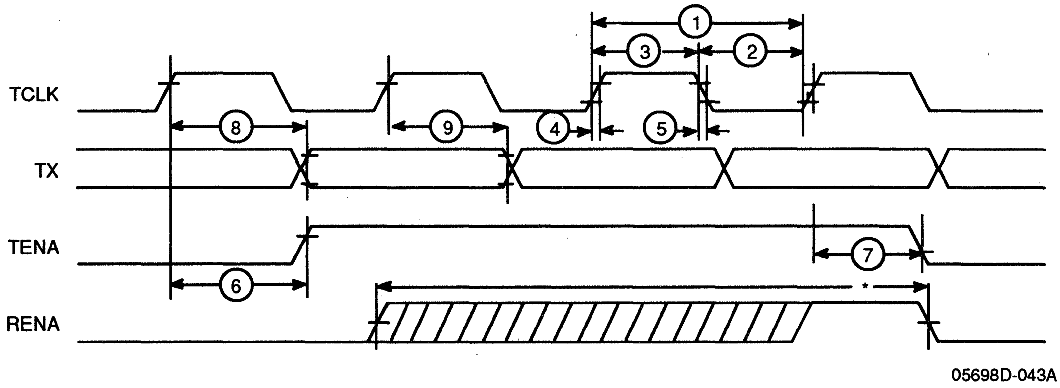
05698D-041A



05698D-042A

Serial Link Timing (Receive)

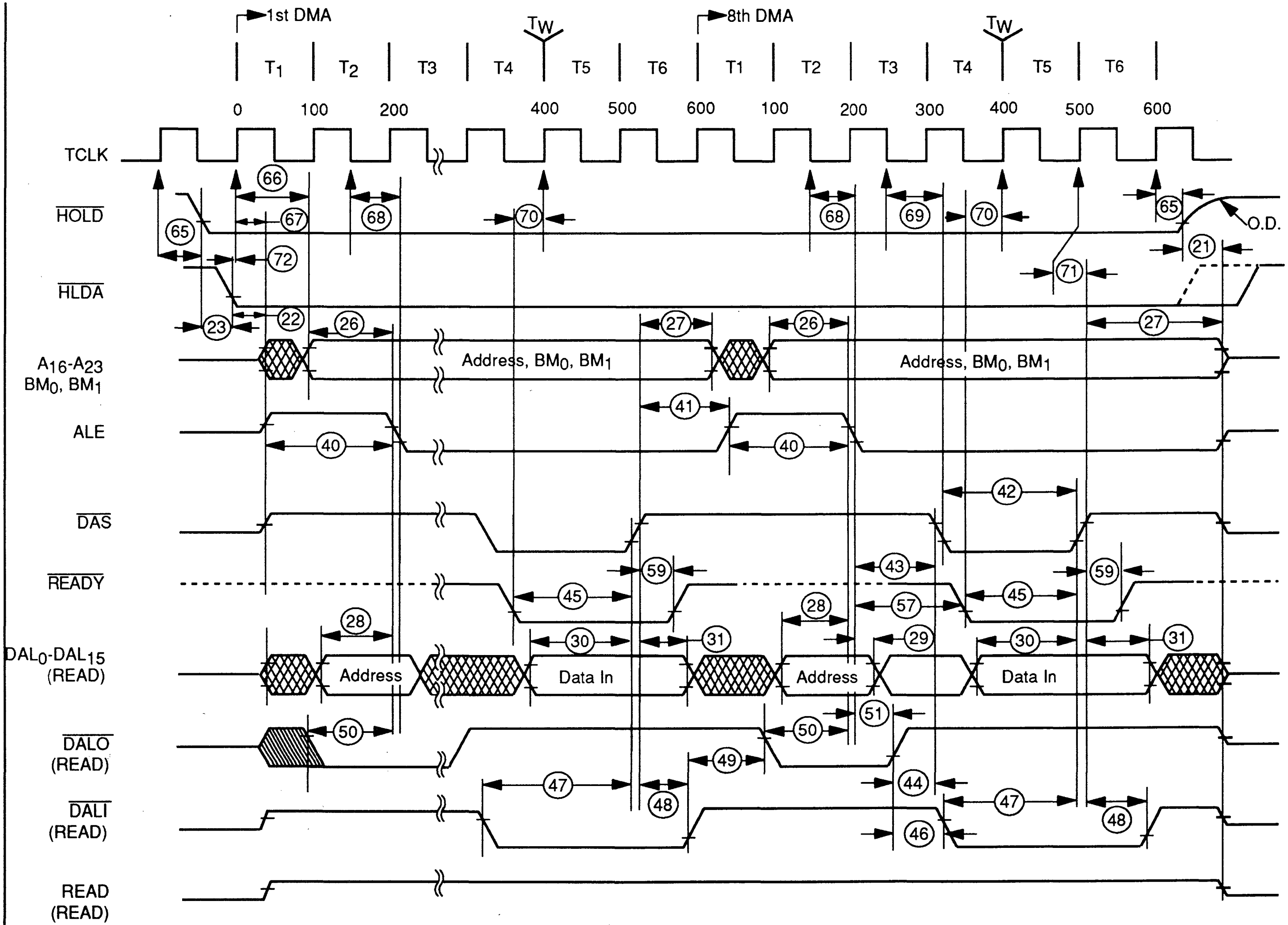
SWITCHING WAVEFORMS



*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 LANCE.

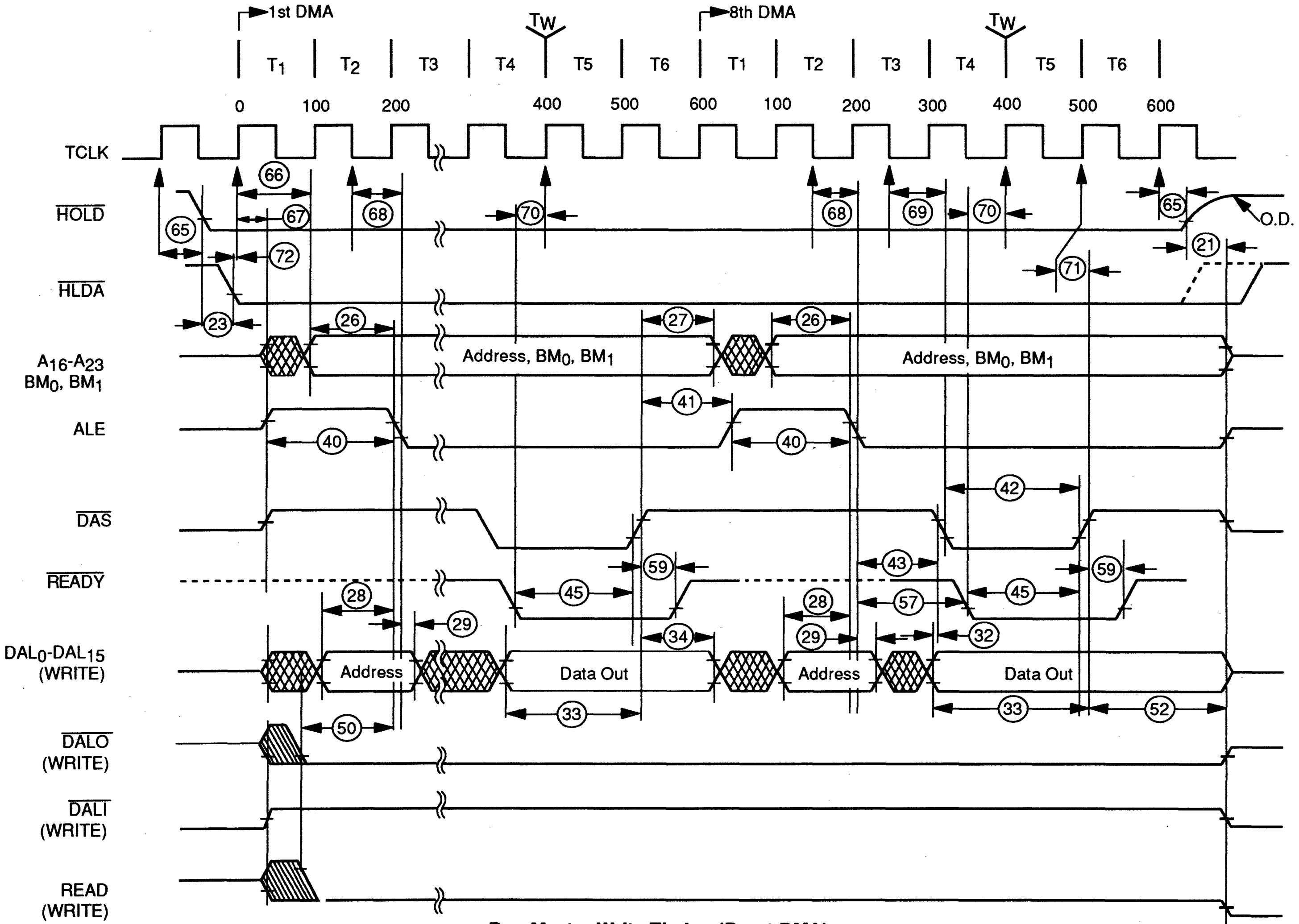
Note:
Please refer to Figures 3 to 6 for additional waveform diagrams.

Serial Link Timing (Transmit)



Bus Master Read Timing (Burst DMA)

05698D-044A



Bus Master Write Timing (Burst DMA)

05698D-045A

PRELIMINARY

Am79C900

Integrated Local Area Communications Controller™ (ILACC™)

Advanced Micro Devices

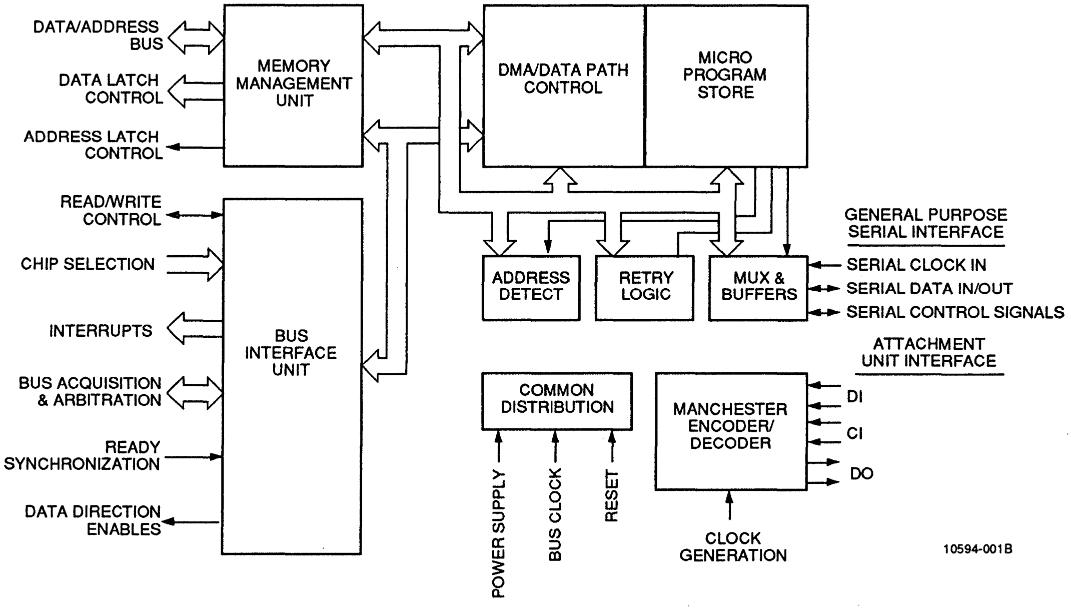
DISTINCTIVE CHARACTERISTICS

- **Integrated Ethernet controller and Serial Interface Adapter.**
- **32-bit bus interface with programmable capability for easy interface to popular bus architectures such as: 29000, 80X86, 680X0.**
- **Compatible with Ethernet and ISD 8802-3 ANSI/IEEE Std. 802.3 10BASE5, 10BASE2, 10BASE-T, and 10BASE-F.**
- **On board 48-byte FIFO, DMA controller, and advanced buffer management scheme.**
- **Extensive network diagnostics capabilities including: CRC, loop back, collision retry/runt packet counters, and TDR.**
- **State of the art CMOS technology and surface mount packaging.**

GENERAL DESCRIPTION

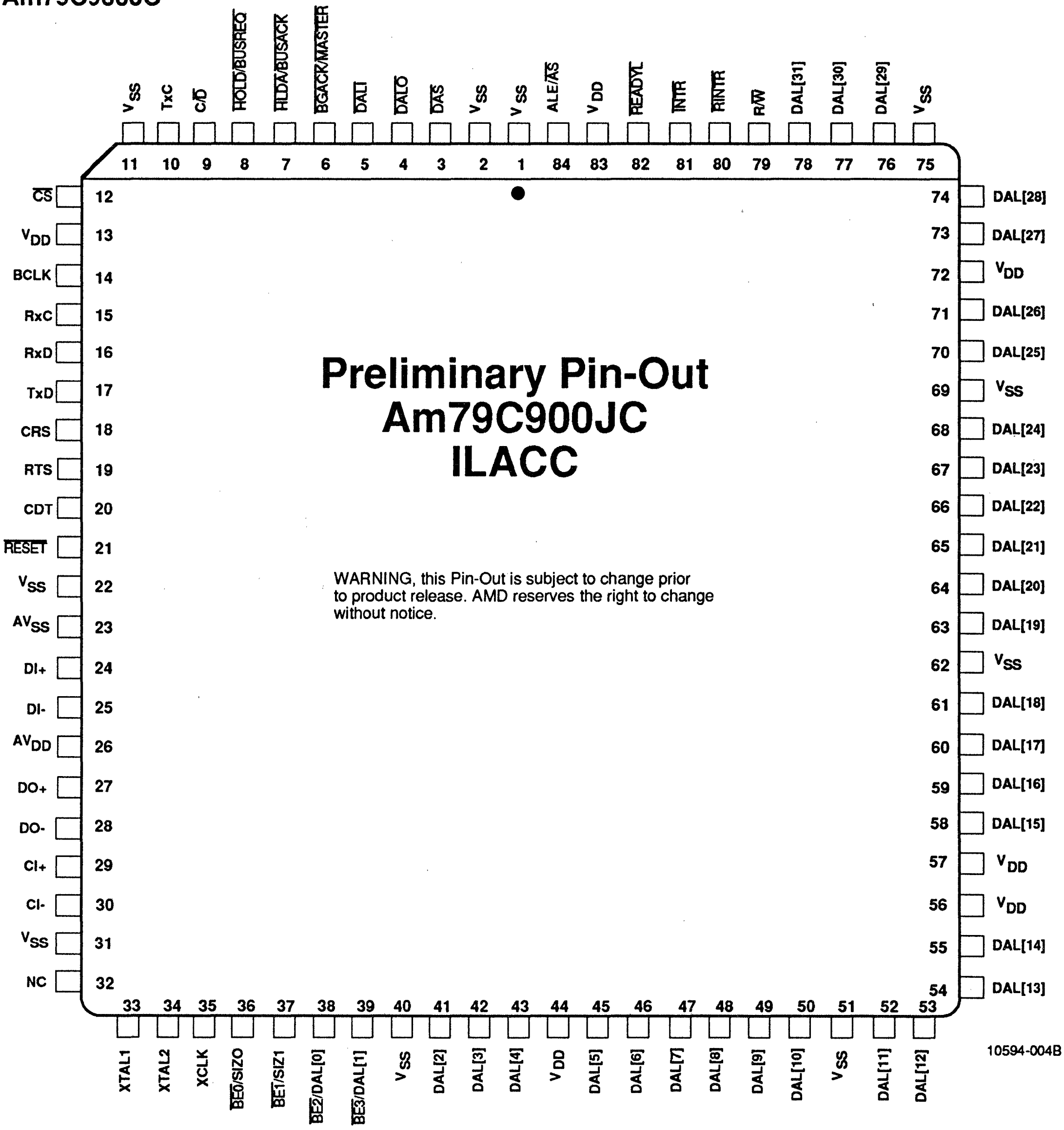
The Am79C900 Integrated Local Area Communications Controller (ILACC) is a second generation Ethernet/802.3 integrated controller and serial interface encoder/decoder. The ILACC has been designed to easily interface to popular microprocessor bus architectures through its programmable bus interface. The ILACC's on board DMA controller and its sophisticated buffer management scheme allows the system designer to achieve maximum performance in tightly coupled systems such as PC mother board applications and node processor based adapter cards. In open bus architectures such as personal computer add-on LAN cards the ILACC gives the system designer the flexibility to chose the optimal cost-performance ratio by allowing both inexpensive bus master and shared memory applications. The ILACC will support thick coax, thin coax, twisted pair and fiber optic cable networking schemes, such as Ethernet and ISO 8802-3 ANSI/IEEE Std. 802.3 10BASE5, 10BASE2, 10BASE-T, and 10BASE-F through its AUI interface in conjunction with an external transceiver chip.

BLOCK DIAGRAM



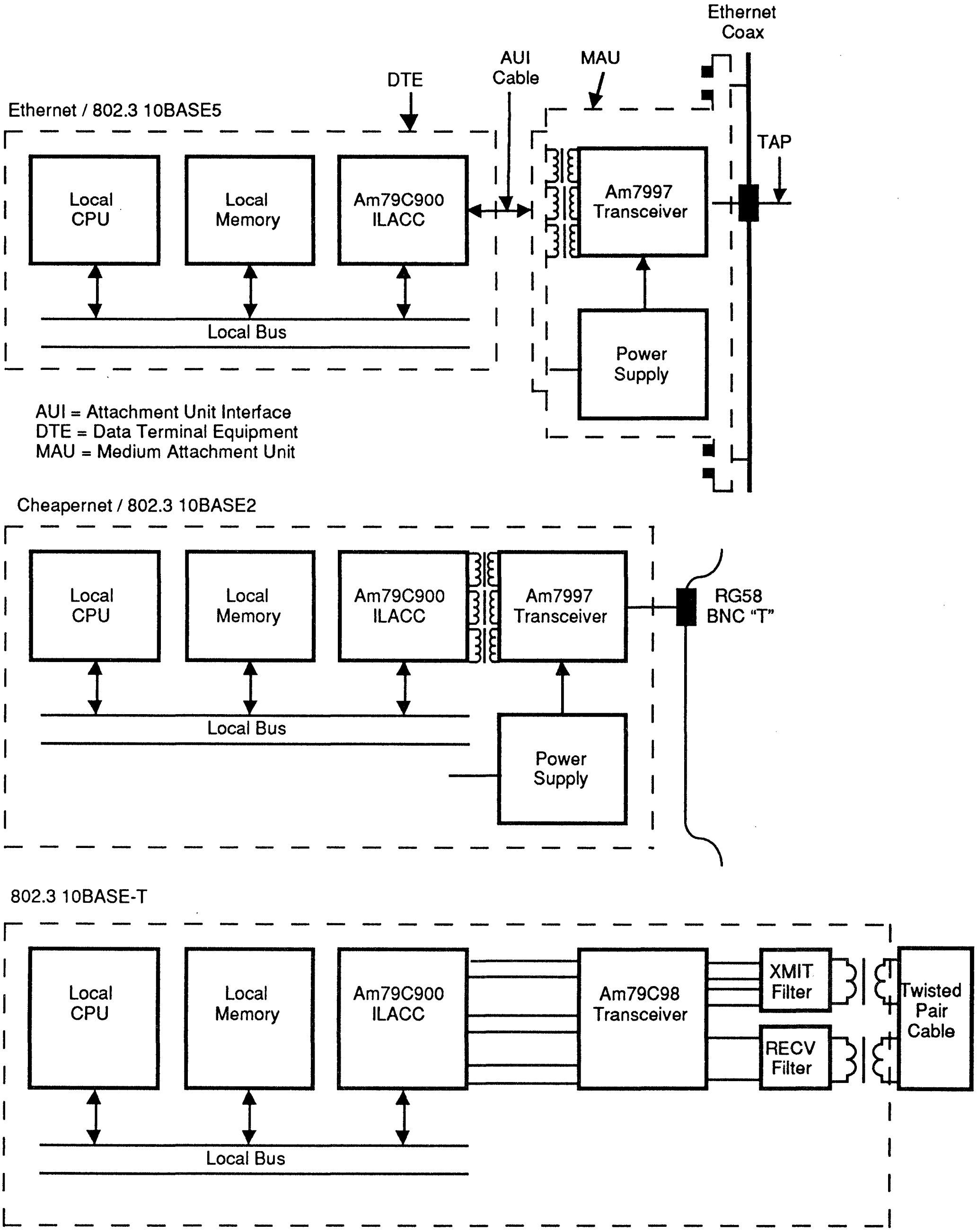
Publication #10594 Rev. C Amendment /0
Issue Date: January 1992

CONNECTION DIAGRAMS
Am79C900JC



10594-004B

TYPICAL ETHERNET NODE

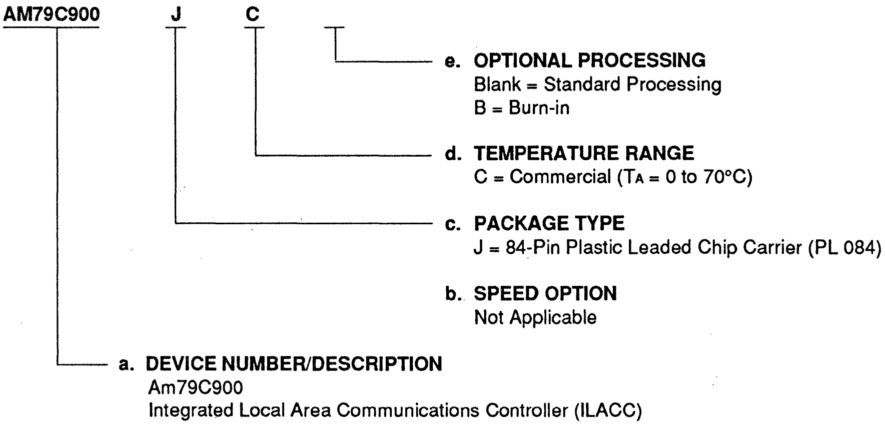


AUI = Attachment Unit Interface
DTE = Data Terminal Equipment
MAU = Medium Attachment Unit

10594-005B

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:



AM79C900 J C

e. OPTIONAL PROCESSING
Blank = Standard Processing
B = Burn-in

d. TEMPERATURE RANGE
C = Commercial (T_A = 0 to 70°C)

c. PACKAGE TYPE
J = 84-Pin Plastic Leaded Chip Carrier (PL 084)

b. SPEED OPTION
Not Applicable

a. DEVICE NUMBER/DESCRIPTION
Am79C900
Integrated Local Area Communications Controller (ILACC)

Valid Combinations	
AM79C900	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, to check on newly released combinations.

PIN DESCRIPTION

ALE/$\overline{\text{AS}}$

Address Latch Enable/Address Strobe (Input/Output, High Impedance)

Used to demultiplex the DAL bus and define the address portion of the memory cycle. $\overline{\text{AS}}$ is the logical inversion of ALE. The polarity of the pin is programmable through ACON (CSR3 bit 1) as follows:

ACON = 0: ALE (falling edge latches address).
ACON = 1: $\overline{\text{AS}}$ (rising edge latches address).

Used as input during bus arbitration to detect completion of previous bus master transactions.

BCLK

Bus Clock (Input)

BCLK determines the operating clock rate for the microprocessor interface.

$\overline{\text{BE0}}$–$\overline{\text{BE1}}$/SIZ0–SIZ1

Byte Enable (Output, High Impedance)

With BACON = 00 , these lines become $\overline{\text{BE0}}$ and $\overline{\text{BE1}}$ (DAL[0] and DAL[1] become BYTE ENABLE 2 and 3 respectively). These signals are used for the 80X86 interface.

Size (Output, High Impedance)

With BACON = 01, SIZ0 and SIZ1 are produced for 680X0 or Am29000 environments.

$\overline{\text{BE2}}$-$\overline{\text{BE3}}$/DAL[0]–DAL[1],

Byte enable/Data/Address lines (Input/Output, High Impedance)

For BACON = 00. In master cycles, during the address portion of a memory transfer, the pins function as $\overline{\text{BE2}}$–$\overline{\text{BE3}}$, the memory bank selected signals for an 80X86-type environment. During the data portion of the memory transfer, DAL[0]–DAL[1] contain the read or write data depending on the type of transfer. In slave cycles, these lines operate as data lines.

Data/Address lines (Input/Output, High Impedance)

For BACON = 01. In master cycles, during the address portion of a memory transfer, the pins function as A[1] and A[0] the byte offset signals for a 680X0-type environment. During the data portion of the memory transfer, DAL[1:0] contain the read or write data depending on the type of transfer. In slave cycles, these lines operate as data lines.

$\overline{\text{BGACK}}$/ $\overline{\text{MASTER}}$

Bus Grant Acknowledge (Input/Output, Open Drain)

Bus Grant Acknowledge indicates the current bus master (BACON = 01). If the ILACC has requested the bus (asserted $\overline{\text{BUSREQ}}$), it must wait until $\overline{\text{BUSACK}}$ becomes asserted (indicating the current master will relinquish the bus on completion of its transaction) at which time the ILACC will sample $\overline{\text{BGACK}}$, $\overline{\text{DAS}}$ and ALE/$\overline{\text{AS}}$. If they are in their inactive state (indicating the current master has completed its last cycle and no other device is claiming bus mastership), the ILACC will assert $\overline{\text{BGACK}}$. Bus Grant Acknowledge will remain asserted as long as the ILACC remains bus master. Used with 680X0 family of processors.

Master (Output, Open Drain)

Asserted when ILACC is the bus master (BACON=00) to enable data/address bus drivers.

C/$\overline{\text{D}}$

Control/Data Select (Input)

Used during slave cycles to determine if the current transaction is transferring control or data information.

C/$\overline{\text{D}}$ = 0: Data Port select.
C/$\overline{\text{D}}$ = 1: Register Address Port select.

The input is ignored when the ILACC is a bus master.

CDT

Collision Detect (Input)

When asserted, indicates that there is more than one node transmitting on the medium concurrently. CDT is only required when using the general purpose serial interface. CDT should be tied low when using the internal SIA (PORTSEL = 0, CSR15 bit 8).

CI+
CI-

Control In (Input)

A differential line input signaling that there is a collision when operating the integrated SIA. Used in Ethernet/ 802.3 applications. Operates at pseudo-ECL levels. When using the general purpose serial interface (PORTSEL = 1, CSR15 bit 8), CI+/- should be tied to ground.

CRS

Carrier Sense (Input)

CRS must be asserted when valid data is being received by an external transceiver connected via the general purpose interface port. CRS is only required when using the general purpose serial interface. CRS should be tied low when using the internal SIA (PORTSEL = 0, CSR15 bit 8).

$\overline{\text{CS}}$

ChipSelect(Input)

Used to access the ILACC internal registers in conjunction with C/$\overline{\text{D}}$. Ignored during bus mastership cycles.

DAL[31:2]

Data/Address Lines (Input Output, High Impedance)

During the address portion of a memory transfer DAL[31:2] contain memory address information. During the data portion of the memory transfer DAL[31:2] contain the read or the write data depending on the type of transfer.

DALI

Data/Address Line In (Output)

An external bus transceiver control line used to enable the data path into the ILACC. Active in both master and slave cycles.

DALO

Data/Address Line Out (Output)

An external bus transceiver control line used to enable the data path away from the ILACC. Active in both master and slave cycles.

DAS

Data Strobe (Input/Output, High Impedance)

Defines the data portion of the bus transfer. Input during bus slave. Output during bus master cycles.

DI+
DI-

Data In (Input)

A differential line input to the integrated SIA for receiving Manchester encoded data from the network. Operates at pseudo-ECL levels. When using the general purpose serial interface (PORTSEL = 1, CSR15 bit 8), DI ± should be tied to ground.

DO+
DO-

Data Out (Output)

A differential line output for transmitting Manchester encoded data from the integrated SIA. Operates at pseudo-ECL levels. When using the general purpose serial interface port (PORTSEL = 1, CSR15 bit 8), DO± should be left unconnected.

HLDA/BUSACK

Hold Acknowledge (Input)

Response from other potential bus masters to indicate they have relinquished bus mastership in an 80X86-type processor environment (BACON = 00). Any host which allows preemptive DMA may deassert Hold Acknowledge at any time requiring the ILACC to deassert HOLD .

Bus Acknowledge (Input)

This signal is asserted by the host in response to a Bus Request. When Bus Acknowledge is received in response to the chip's assertion of Bus Request, the ILACC becomes the bus master after ALE/AS, DAS and BGACK are sampled inactive. Intended for use in 680X0-type processor environments (BACON = 01).

HOLD/BUSREQ

Hold (Output, Open Drain)

Asserted by the ILACC to request bus mastership in 80X86 processor configurations. The output can be wire-ORed with other potential bus masters. HOLD will be deasserted by the ILACC within a maximum of five bus cycles if another master preempts the ILACC (removes HLDA).

Bus Request (Output Open Drain)

Bus Request is asserted when the chip requires the bus for direct memory transfer in 680X0-type processor configurations. The output may be wire-ORed with other potential bus masters.

INTR

Interrupt (Output, Open Drain)

An attention signal that indicates that one or more of the following status flags are set: BABL, MERR, MISS, TINT, IDON (all in CSR0), TXSTRT or LBE (in CSR4). INTR is enabled by IENA = 1 (CSR0 bit 6).

NC

No Connection

Do not connect.

R/W

Read/Write (Input/Output, High Impedance)

Indicates the direction of data flow to or from the ILACC. An output during bus master cycles. An input during slave cycles.

READYL

Ready Low (Input/Output Open Drain)

When the ILACC is a bus slave, READYL is the output used to request wait states to be inserted in host read/write operations. When the ILACC is a bus master, READYL is the input acknowledge from target memory to indicate it will accept data in a write cycle or that valid data is available on the DAL bus in a read cycle.

RESET

System Reset (Input)

Reset clears the internal logic. All outputs go to their high impedance state or are driven inactive. All bus-related outputs are high impedance until the Initialize command is given by the host.

RINTR

Receive Interrupt (Output, Open Drain)

When active, indicates that RINT in CSR0 is set (bit 10). RINTR is enabled by INEA (CSR0 bit 6). Receive interrupts can be masked by setting the mask bit RINTM in CSR3 (bit 10). RINTR will remain asserted until RINT is cleared, RINTM is set, or INEA is cleared. RINT set in CSR0 does not cause the external INTR to become asserted although the INTR summary bit in CSR0 will be set providing RINTM in CSR3 is clear.

RTS

Request To Send (Output)

RTS is asserted when the chip wishes access to the channel. RTS remains asserted during the transmission cycle. RTS will only be activated by the ILACC if the general purpose serial interface has been selected. RTS should be left unconnected if the integral SIA has been selected (PORTSEL = 0, CSR15 bit 8).

RxC
Receive Clock (Input)
The receive data clock operates at the network data rate 10 MHz. Only required if the general purpose serial interface has been selected. RxC should be tied low if the internal SIA has been selected (PORTSEL = 0, CSR15 bit 8).

RxD
Receive Data (Input)
The receive serial data path to the general purpose serial interface. Serial data presented on this input will be clocked into the ILACC by the positive edge of the RxC. RxC should be tied low when using the internal SIA (PORTSEL=0, CSR15 bit 8).

TxC
Transmit Clock (Input)
The transmit data clock operates at the network data rate (10 MHz). Only required if the general purpose serial interface has been selected. TxC should be tied low if the internal SIA has been selected (PORTSEL = 0, CSR15 bit 8).

TxD
Transmit Data (Output)
The transmit serial data path is only activated by the ILACC if the general purpose serial interface has been selected. The ILACC will clock out serial data onto TxD on the positive edge of TxC. TxD should be left unconnected if the internal SIA has been selected (PORTSEL = 0, CSR15 bit 8).

V_{DD}
Power supply (6 pins)
+5 V supply for internal interface logic and I/O pin driver functions.

V_{SS}
Ground (10 pins).
0 V reference for internal interface logic and I/O pin driver functions.

AV_{DD}
Analog Power supply
+5 V supply for the analog functions of the internal SIA. This supply should be separated from the digital V_{DD} supplies as far back to the system power supply as practical.

AV_{SS}
Analog Ground Reference
0 V reference for the analog functions of the internal SIA. This ground reference should be separated from the digital V_{SS} supplies as far back to the system power supply as practical.

XCLK
Clock (Output)
XCLK is derived from the crystal oscillator.

XTAL1
XTAL2
20 MHz Crystal Oscillator (Input)
The crystal frequency determines the network data rate. When using an external crystal, two 100-pF capacitors are required between XTAL1 and ground and XTAL2 and ground. XTAL1 may be driven from an external source in which case XTAL2 must be left floating. When using the internal SIA, the network data rate (10 MHz) will be one half of the external crystal frequency.

FUNCTIONAL DESCRIPTION

General

The Am79C900 (ILACC) is designed to operate in an environment that allows close coupling with a local memory and/or microprocessor (host), or alternately it can reside on a system bus and act as an intelligent bus master device.

The ILACC is programmed by a combination of registers resident within the chip and data structures located in user memory. There are 60 user-accessible Control and Status Registers (CSRs) within the chip. The host is responsible for initial programming of a small subset. Once enabled, the ILACC accesses memory directly to acquire additional operating parameters.

The Am79C900 has the ability to perform independent buffer management as well as transfer data packets to and from the network. There are three memory structures accessed by the chip:

1. **Initialization Block** – Seven 32-bit entries in memory starting on a long word boundary. It contains the parameters necessary for device operation. The Initialization Block is comprised of:
   - Mode of Operation
   - Physical Address
   - Logical Address Filter
   - Pointers to Receive and Transmit Descriptor Rings
   - Number of Entries in Receive and Transmit Descriptor Rings
2. **Receive and Transmit Descriptor Rings** – Two contiguous ring structures in memory for control of Receive and Transmit packets. The descriptor rings are comprised of:
   - The address of a data buffer
   - Status and error information associated with the buffer
   - The length of the data buffer
3. **Data buffers** – Area(s) of memory reserved for packet buffering. Data buffers may begin on arbitrary byte boundaries. Each buffer must be contiguous in memory, although multiple buffers can be located anywhere in addressable memory.

   In general the programming sequence of the ILACC may be summarized as:

1. Program the ILACC's CSRs to locate the Initialization Block in memory.
2. Define the byte control, byte addressing, address latch, and bus arbitration.
3. Fetch the Initialization Block via DMA.
4. Access the descriptor rings and data buffers for packet handling.

The parallel interface of the ILACC has been designed to be easily interfaced to a variety of popular 32-bit microprocessor buses; examples include the 80X86, 680X0 and AMD 29000 series. The ILACC is user-configurable so that it directly interfaces to the bus arbitration schemes of the above architectures.

The ILACC has a 32-bit wide linear address space when acting as Bus Master allowing it to DMA directly into the entire address space of the above microprocessors and system buses.

Interrupts to the processor are generated by the ILACC upon:

1. Completion of ILACC's Initialization routine
2. The reception of a packet
3. Start of transmit packet
4. Completion of transmit activity
5. A transmitter time-out error
6. A missed packet
7. A memory error

The cause of interrupt is determined by reading CSR0 and/or CSR4. Bit 6 of CSR0 (INEA) enables or disables interrupts to the host. In systems where polling is used in place of interrupts, bit 7 of CSR0 (INTR) indicates an interrupt condition.

The basic operation of the ILACC consists of two distinct modes: transmit and receive. In the transmit mode, the ILACC directly addresses data in a transmit buffer in memory. It prefaces the data with a preamble and synchronization pattern and calculates and appends a 32-bit CRC. This packet is then Manchester encoded by the internal SIA or sent out in NRZ format with clock depending on which transceiver port is selected.

In the receive mode, packets are received via the external transceiver and passed to either the SIA port or the general purpose serial interface of the ILACC. If the internal SIA is used, clock and data separation occur and the packet is loaded into buffer memory. If the general purpose interface is used, clock and data separation must occur externally. A CRC is calculated for the received packet 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.

ILACC Bus Configurations

The ILACC supports a 32-bit data and address bus. Memory byte selection during ILACC bus mastership can be software-programmed according to the target microprocessor using the BACON bits. Arbitration schemes for 80X86 and the 680X0 are supported.

Bus Cycles

Depending on the operation, the ILACC can function as a bus slave (memory or I/O mapped) or as a bus master (DMA) device.

BUS SLAVE CYCLES

Slave cycles are executed by the host system on the ILACC to program the initial conditions of the device or to examine its state during operation.

The host can gain read or write access to the ILACC's internal Control and Status Registers (CSRs) by asserting the CS and DAS Lines (CS = DAS = LOW), causing the ILACC to enter the bus slave mode. The CSRs are accessed in a two-stage process. The host must first write the address of the register to be accessed into the Register Address Pointer (RAP). The host can subsequently perform read or write operations on the register addressed by the contents of RAP by accessing the Data Port.

RAP or Data Port selection is performed using the C/D input pin (C/D = 1 for RAP access). For more details, see the heading "User Programmable Registers."

All slave accesses to/from the ILACC's internal CSRs take place over DAL[15:0]. The high order address and data bus lines (DAL[31:16]) are driven but contain undefined data during slave read operations and are ignored during slave write operations.

Read Sequence

At the beginning of the read cycle, CS, C/D and R/W are asserted by the host. The host will assert DAS which will latch both the read request and the state of C/D within the ILACC. R/W and C/D need not be active for the remainder of the cycle. The ILACC will subsequently assert DALO to enable the external output bus transceiver(s). If C/D was latched as a "1", the contents of RAP will be placed on the DAL bus. If C/D was a "0", the contents of the CSR addressed by RAP will be placed on the DAL bus. After the data on DAL[31:0] becomes valid, the ILACC asserts READYL signalling the host to strobe in the data using the rising edge of DAS and relinquish CS. The ILACC subsequently releases DALO, READYL, and the DAL bus in response to DAS going inactive. CS and DAS must be valid during the entire slave read cycle.

Write Sequence

At the beginning of the write cycle, CS, C/D and R/W are asserted by the host. The host will assert DAS which will latch both the write request and the state of C/D within the ILACC. R/W and C/D need not be active for the remainder of the cycle. The ILACC will subsequently assert DALI to enable the external input bus transceiver(s). The host will output the write data on DAL[15:0]. If C/D was latched as a "1", the contents of the DAL bus will be written to RAP. If C/D was a "0", the contents of the DAL bus will be written to the CSR addressed by the RAP. When the ILACC asserts READYL, the host strobes the data into the ILACC using the rising edge of DAS and subsequently releases the CS line and the DAL bus. The ILACC will deassert DALI and READYL in response to DAS going inactive. CS and DAS must be valid during the entire slave write cycle.

BUS ACQUISITION

The ILACC bus acquisition mechanism can be optimized to suit common two- or three-wire bus arbitration schemes, using the Bus Acquisition Control (BACON) bits in CSR4, as defined below:

BACON	Bus Configuration
00	80X86
01	680X0
10	RESERVED
11	RESERVED

For 80X86-type processors, bus acquisition is controlled with a two-wire handshake of HOLD (HOLD REQUEST) and HLDA (HOLD ACKNOWLEDGE). If BACON = 00 (80X86 operation), the burst transfer may be preempted by the host or system arbiter deasserting the HLDA line. The ILACC will complete its current bus transaction before relinquishing the HOLD request.

For 680X0-type processors, bus acquisition is controlled with a three-wire handshake of BUSREQ (BUS REQUEST), BUSACK (BUS ACKNOWLEDGE) and BGACK (BUS GRANT ACKNOWLEDGE). Preemption is not supported in this configuration.

The ILACC will request the bus to enable the movement of a received packet into the receive buffer area or to check for the presence of a transmit message and to move it from the transmit buffer area if required.

If there are 16 bytes or more empty in the FIFO in transmit mode, or at least 16 bytes of data in the FIFO in receive mode when the ILACC releases the bus (HOLD or BGACK deasserted), it will request the bus again within 4 bus clock periods for receive or 10 bus clock periods for transmit.

BUS MASTER CYCLES

All data transfers from the ILACC during bus mastership are timed by ALE/AS or DAS and READYL. The automatic adjustment of the ILACC cycle by the READYL signal allows synchronization with variable cycle time memory. Bus cycles are a minimum of 4 BCLK cycles and can be increased in increments of 1 BCLK cycle.

DALI and DALO are used to control external bus transceivers. DALI is used to enable the data path toward the ILACC. DALO to enable the data path away from the ILACC.

Read Sequence

The read cycle is commenced by placing valid addresses on DAL[31:0] for the 680X0 mode or DAL[31:2] for the 80X86 mode. The appropriate byte transfer signals (A_{0-1} and SIZ0 - SIZ1 for the 680X0 mode; BE0 - BE3 for the 80X86 mode) are asserted to indicate the active data bus width. The R/W signal is driven HIGH to indicate a read cycle. The ALE/AS pulse allows the external latch to load and store the long-word address. The DAL lines go into a high impedance state and DAS falls low to signal the start of the memory access. DALI is activated by the ILACC to enable the external input bus transceiver(s). The memory responds by asserting the READYL input to the ILACC to indicate that DAL[31:0] have valid data. The ILACC latches the memory data on the rising edge of DAS which in turn ends the memory cycle.

Write Sequence

The write cycle is very similar except the write transfer is indicated by R/W being driven LOW. The DAL lines change from containing addresses to data after ALE/AS goes inactive. DALO is used to enable the ILACC data onto the data bus. Data to memory is held valid after DAS goes inactive.

Variable Length ILACC DMA Transfers

The ILACC will initiate DMA transfers according to the type of operation being performed. All DMA transfers will fall into one of the following catagories:

- Single-cycle DMA
- Dual-cycle DMA
- Burst-cycle DMA

Single-cycle DMA Transfers

Once the ILACC has been granted bus mastership, it will issue a single long-word memory address and perform a single long word (32-bit) or partial long word access to that location. Following the completion of that single location access, the ILACC will relinquish bus mastership. An example of this type of access can be found at the end of ILACC initialization. Another example would be at the end of a transmit packet when only one access is necessary to finish reading the packet into the FIFO.

Dual-cycle DMA Transfers

Once the ILACC has been granted bus mastership, it will perform two data transfer cycles before relinquishing the bus. The two transfers within the mastership period will always be of the same type (either both read or both write) but may be to non-contiguous addresses. Dual cycle DMA transfers cannot be preempted. Dual-cycle DMA transfers are typically performed during descriptor accesses.

Burst-Cycle DMA Transfers

Once the ILACC 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 long-word 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 long-word 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 four transfers will be performed. If DMAPLUS = 1, the burst will continue until the FIFO is filled to its high threshold (34 bytes in transmit operation) or emptied to its low threshold (14 bytes in receive operation). Due to pipelining delays internal to the ILACC, the actual number of bytes in the FIFO at the end of the DMA burst may exceed the above thresholds (i.e. filled with ≥34 bytes or emptied to ≤ 14 bytes).The exact number of transfer cycles in this case will be dependent on the latency of the system bus to the ILACC's mastership request (HOLD/BUSREQ to HLDA/BGACK delay) and the speed of bus operation. The burst cycle may be preempted when using the 80X86 mode of operation by removing the mastership privilege (HLDA = HIGH). The ILACC will complete the current read or write cycle before returning the mastership request inactive (HOLD = HIGH).

BUS MASTER TRANSACTIONS

Transactions during ILACC bus mastership consist of use of the allowable DMA cycle types as previously defined and the type of memory transaction being performed. Bus transactions fall into one of the following three catagories:

- Initialization block read access
- Descriptor read/write access
- Data buffer read/write access

Initialization Block Access

This transaction reads all 7 long-words (28 bytes) of the Initialization Block.

Data is read from the Initialization Block as a sequence of four separate arbitration/relinquish cycles. The first three exchanges will be performed as dual-cycle DMA transfers, performing two contiguous long-word reads, commencing at the base address programmed in CSR1 and CSR2. This sequence continues until the fourth cycle which will perform a single-cycle DMA transfer to read the last long-word entry in the Initialization Block. The time between each mastership cycle may vary if another device is also contending for bus mastership during the initialization sequence. The ILACC will always complete the 2 long-word read operations within the bus mastership period, even if preempted.

Initialization Block entries are not byte-swapped regardless of target configuration. The internal ILACC registers read in the memory-based initialization parameters on the basis that bit 31 of the memory location is the high order bit of the word.

The base location of the Initialization Block is constrained to be on an even word boundary. Bits 0 and 1 of CSR1 must be zero.

Descriptor Access

These transactions read and write the appropriate entries of the transmit and receive descriptor rings to manage the transfer of transmit buffers to the network and messages from the network to the receive buffers.

Accesses to descriptor entries are performed by only two methods:

1. For chained buffers (multiple descriptors/buffers containing a single message), a single-cycle write will be used to update the transmit or receive descriptor status (TMD1 or RMD1) of all but the last descriptor in the chain.
2. A dual-cycle DMA transfer will be performed on all descriptor read operations and on write operations to the last descriptor in a chain including the case where only one descriptor was used for the packet.

In the case of a transmit error in the middle of a chain, a dual-cycle transfer will be written to the current descriptor, = regardless of its position in the packet. The dual access is needed at this point in order to report the error condition to TMD2.

Note that during all descriptor read operations, the entire data bus will be activated, = as if a full long-word data fetch were being performed. The ILACC will internally route and use only the required data and discard any superfluous information fields. For write cycles, only the appropriate data bus bytes are activated to ensure adjacent memory locations are not corrupted.

Descriptor entries are not byte swapped regardless of target configuration. The internal ILACC registers read in the memory-based initialization parameters on the basis that bit 31 of the memory location is the high order bit of the word.

Descriptors must be constrained to be on 16-byte boundary as defined by the TRANSMIT/RECEIVE DESCRIPTOR RING ADDRESS fields (TDRA and RDRA), within the Initialization Block (TDRA[3-0] = 0, RDRA[3-0] = 0). Hence all descriptor entries will appear on long-word boundaries to the ILACC.

Data Buffer Access

Burst-cycle DMA is used to read transmit buffer information and transfer it to the FIFO or to write receive message information from the FIFO to the receive buffer area.

Accesses to buffer entries are performed by only two burst mechanisms:

1. If DMAPLUS = 0 (CSR4, bit 14), the burst transfer will consist of up to four read/write cycles providing the ILACC is not preempted (applicable to 80X86 mode only). See timing diagrams for preemption timing.
2. If DMAPLUS = 1, burst transfers continue until the FIFO is filled to at least its high threshold (34 bytes in transmit operation) or emptied to its low threshold (≤ 14 bytes in receive operation) within a single DMA burst cycle (unless preempted). See timing diagrams for preemption timing.

To maximize system bus bandwidth, the ILACC will always use its first DMA transfer to or from a buffer to long-word align its remaining transfers. For example, if the buffer is located on an odd-word boundary (A1 = 1, A0 = 0), the first DMA transfer will read or write 1 word (2 bytes) of data. Subsequent cycles will DMA long-word data (4 bytes) since the addresses will now be long-word aligned (A1 = 0, A0 = 0).

Note that during all transmit buffer read operations, the entire 32-bit data bus will be activated and read as a single long word. In those instances where fewer than 4 bytes are to be read from a long word location in a buffer, the ILACC will internally route the byte(s) being transferred and will discard the remaining bytes from the 32-bit read operation. These conditions might exist only at the start of a transmit buffer, the end of the used portion of a transmit buffer, or the end of the transmit buffer's memory allocation. For receive buffer write operations, the ILACC will always perform 32-bit writes with the following exceptions:

1. When writing to the beginning of a receive buffer where the first byte of that buffer does not align with a long word boundary.
2. When writing to the long word location that contains the last (<4) byte(s) of a receive buffer's memory allocation.

For these two instances, the ILACC will employ the bus control signals ($\overline{BE0}$-$\overline{BE3}$, SIZ0-SIZ1) to ensure that only the appropriate bytes in the 32-bit word are written to memory. Note that when fewer than 4 bytes are being written to the end of a receive buffer where there are more than sufficient bytes in the 32-bit location that are part of the current buffer, then undefined data will be written to the unused buffer byte(s) in that 32-bit location.

Buffer data will be byte swapped according to the target memory architecture due to the byte orientation of the 802.3 protocol ("little-endien"). Data is transferred across the network in byte-ascending order (i.e., starting with byte 0, then byte 1, 2, 3, etc.).

There are a number of additional restrictions which apply to transmit and receive buffers:

1. The BUFFER BYTE COUNT for receive buffers (BCNT in RMD1) must be a minimum of 64 bytes.
2. The MESSAGE BYTE COUNT for receive buffers (MCNT in RMD2) will contain the exact number of bytes received in the packet and is written by the ILACC in the last receive descriptor table entry (DTE) for the message (valid where the ENP bit is set in RMD1, and assuming ERR is clear).
3. The BUFFER BYTE COUNT (BCNT in TMD1) for the first buffer in a chained transmit packet must be a minimum of 116 bytes if DMAPLUS (CSR4[14]) is set or 100 bytes minimum if DMAPLUS is reset.

Note that 80X86 type processors have bus transfer restrictions, namely:

(i) that a single 3-byte transfer will not be observed without an accompanying byte cycle either before or after.

(ii) misaligned transfers will move the data at the high addressed long-word location first then decrement to the previous long-word location to complete the transfer.

These restrictions do not apply to the ILACC even when configured for the 80X86 interface. 3-byte cycles can be observed (i.e., at the end of a buffer), and the ILACC will transfer data logically incrementing to each long word location and performing the appropriate transfer.

SUMMARY OF 32-BIT MEMORY TRANSFERS

From the cases outlined previously, the following set of 32-bit bus transfer conditions are required.

Initialization Block:

LONG-WORD TRANSFER FROM EVEN-WORD ADDRESS

Example: Normal 32-bit Initialization Block entry read.

680X0: SIZ1=0, SIZ0=0, A1=0, A0=0

MS	byte	from	DAL [31:24]
	byte	from	DAL [23:16]
	byte	from	DAL [15:08]
LS	byte	from	DAL [07:00]

80X86: $\overline{\text{BE3}}$=0, $\overline{\text{BE2}}$=0, $\overline{\text{BE1}}$=0, $\overline{\text{BE0}}$=0

MS	byte	from	DAL [31:24]
	byte	from	DAL [23:16]
	byte	from	DAL [15:08]
LS	byte	from	DAL [07:00]

Transmit/Receive Descriptors:

LONG-WORD TRANSFER TO/FROM EVEN-WORD ADDRESS

Example: Normal 32-bit descriptor access.

680X0: SIZ1=0, SIZ0=0, A1=0, A0=0

MS	byte	to/from	DAL [31:24]
	byte	to/from	DAL [23:16]
	byte	to/from	DAL [15:08]
LS	byte	to/from	DAL [07:00]

80X86: $\overline{\text{BE3}}$=0, $\overline{\text{BE2}}$=0, $\overline{\text{BE1}}$=0, $\overline{\text{BE0}}$=0

MS	byte	to/from	DAL [31:24]
	byte	to/from	DAL [23:16]
	byte	to/from	DAL [15:08]
LS	byte	to/from	DAL [07:00]

BYTE TRANSFER FROM ODD-BYTE (3 MOD 4) ADDRESS

Example: Read status from RMD1/TMD1.

680X0: SIZ1=0, SIZ0=0, A1=1, A0=1

MS	byte	from	DAL [31:24]

80X86: $\overline{\text{BE3}}$=0, $\overline{\text{BE2}}$=0, $\overline{\text{BE1}}$=0, $\overline{\text{BE0}}$=0

MS	byte	from	DAL [31:24]

Note: Although all bytes are active, only required bytes are used internally.

BYTE TRANSFER TO ODD-BYTE (3 MOD 4) ADDRESS

Example: Write status to RMD1/TMD1.

680X0: SIZ1=0, SIZ0=1, A1=1, A0=1

MS	byte	to	DAL [31:24]

80X86: $\overline{\text{BE3}}$=0, $\overline{\text{BE2}}$=1, $\overline{\text{BE1}}$=1, $\overline{\text{BE0}}$=1

MS	byte	to	DAL [31:24]

Transmit/Receive Buffers:

The examples shown below of transfers of fewer than 4 bytes to receive buffers assume that there are exactly enough bytes left in the receive buffer's memory allocation to accomodate the particular example. These examples are shown to illustrate the states of the bus control signals under those circumstances. When the number of bytes to be written is determined by the number of remaining received packet bytes and not by the available space left in the buffer, then the ILACC will write undefined data to the remaining unused bytes in that 32-bit location (as long as those unused bytes are part of the current buffer's memory allocation). It is possible that as much as a full long-word transfer could be used by the ILACC to transfer as little as a single byte.

LONG-WORD TRANSFER TO/FROM EVEN-WORD ADDRESS

Example: 4 bytes in buffer on even-word address.

680X0: SIZ1=0, SIZ0=0, A1=0, A0=0

FIFO	byte n	to/from	DAL [31:24]
FIFO	byte n + 1	to/from	DAL [23:16]
FIFO	byte n + 2	to/from	DAL [15:08]
FIFO	byte n + 3	to/from	DAL [07:00]

80X86: $\overline{BE3}$=0, $\overline{BE2}$=0, $\overline{BE1}$=0, $\overline{BE0}$=0

FIFO	byte n	to/from	DAL [07:00]
FIFO	byte n + 1	to/from	DAL [15:08]
FIFO	byte n + 2	to/from	DAL [23:16]
FIFO	byte n + 3	to/from	DAL [31:24]

3-BYTE TRANSFER FROM EVEN-WORD ADDRESS

Example: 3 bytes in transmit buffer.

680X0: SIZ1=0, SIZ0=0, A1=0, A0=0

FIFO	byte n	from	DAL [31:24]
FIFO	byte n + 1	from	DAL [23:16]
FIFO	byte n + 2	from	DAL [15:08]

80X86: $\overline{BE3}$=0, $\overline{BE2}$=0, $\overline{BE1}$=0, $\overline{BE0}$=0

FIFO	byte n	from	DAL [07:00]
FIFO	byte n + 1	from	DAL [15:08]
FIFO	byte n + 2	from	DAL [23:16]

Note: Although all bytes are active, only required bytes are used internally.

3-BYTE TRANSFER TO EVEN-WORD ADDRESS

Example: Last 3 bytes in receive buffer.

680X0: SIZ1=1, SIZ0=1, A1=0, A0=0

FIFO	byte n	to	DAL [31:24]
FIFO	byte n + 1	to	DAL [23:16]
FIFO	byte n + 2	to	DAL [15:08]

80X86: $\overline{BE3}$=1, $\overline{BE2}$=0, $\overline{BE1}$=0, $\overline{BE0}$=0

FIFO	byte n	to	DAL [07:00]
FIFO	byte n + 1	to	DAL [15:08]
FIFO	byte n + 2	to	DAL [23:16]

3-BYTE TRANSFER FROM ODD BYTE (1 MOD 4) ADDRESS

Example: Last 3 bytes in transmit buffer.

680X0: SIZ1=0, SIZ0=0, A1=0, A0=0

FIFO	byte n	from	DAL[23:16]
FIFO	byte n+1	from	DAL[15:8]
FIFO	byte n+2	from	DAL[7:0]

80X86: $\overline{BE3}$=0, $\overline{BE2}$=0, $\overline{BE1}$=0, $\overline{BE0}$=0

FIFO	byte n	from	DAL[15:8]
FIFO	byte n+1	from	DAL[23:16]
FIFO	byte n+2	from	DAL[31:24]

Note: Although all bytes are active, only required bytes are used internally.

3-BYTE TRANSFER TO ODD BYTE (1 MOD 4) ADDRESS

Example: Last 3 bytes in receive buffer's allocated memory.

680X0: SIZ1=1, SIZ0=1, A1=0, A0=1

FIFO	byte n	from	DAL[23:16]
FIFO	byte n+1	from	DAL[15:8]
FIFO	byte n+2	from	DAL[7:0]

80X86: $\overline{BE3}$=0, $\overline{BE2}$=0, $\overline{BE1}$=0, $\overline{BE0}$=1

FIFO	byte n	from	DAL[15:8]
FIFO	byte n+1	from	DAL[23:16]
FIFO	byte n+2	from	DAL[31:24]

WORD TRANSFER FROM EVEN-WORD ADDRESS

Example: Last 2 bytes in transmit buffer on even-word address.

680X0: SIZ1=0, SIZ0=0, A1=0, A0=0

FIFO	byte n	from	DAL [31:24]
FIFO	byte n + 1	from	DAL [23:16]

80X86: $\overline{BE3}$=0, $\overline{BE2}$=0, $\overline{BE1}$=0, $\overline{BE0}$=0

FIFO	byte n	from	DAL [07:00]
FIFO	byte n + 1	from	DAL [15:08]

Note: Although all bytes are active, only required bytes are used internally.

WORD TRANSFER TO EVEN-WORD ADDRESS

Example: Last 2 bytes in receive buffer on even-word address.

680X0: SIZ1=1, SIZ0=0, A1=0, A0=0

FIFO	byte n	to	DAL [31:24]
FIFO	byte n + 1	to	DAL [23:16]

80X86: $\overline{BE3}$=1, $\overline{BE2}$=1, $\overline{BE1}$=0, $\overline{BE0}$=0

FIFO	byte n	to	DAL [07:00]
FIFO	byte n + 1	to	DAL [15:08]

WORD TRANSFER FROM ODD-WORD ADDRESS

Example: First 2 bytes in transmit buffer on odd-word address.

680X0: SIZ1=0, SIZ0=0, A1=0, A0=0

FIFO	byte n	from	DAL [15:08]
FIFO	byte n + 1	from	DAL [07:00]

80X86: $\overline{BE3}$=0, $\overline{BE2}$=0, $\overline{BE1}$=0, $\overline{BE0}$=0

FIFO	byte n	from	DAL [23:16]
FIFO	byte n + 1	from	DAL [31:24]

Note: Although all bytes are active, only required bytes are used internally.

WORD TRANSFER TO ODD-WORD ADDRESS

Example: First 2 bytes in receive buffer on odd-word address.

680X0: SIZ1=1, SIZ0=0, A1=1, A0=0

FIFO	byte n	to	DAL [15:08]
FIFO	byte n + 1	to	DAL [07:00]

80X86: $\overline{BE3}$=0, $\overline{BE2}$=0, $\overline{BE1}$=1, $\overline{BE0}$=1

FIFO	byte n	to	DAL [23:16]
FIFO	byte n + 1	to	DAL [31:24]

BYTE TRANSFER FROM EVEN-WORD ADDRESS
Example: Last byte in transmit buffer on even-word address.
680X0: SIZ1=0, SIZ0=0, A1=0, A0=0
FIFO byte n from DAL [31:24]
80X86: $\overline{\text{BE3}}$=0, $\overline{\text{BE2}}$=0, $\overline{\text{BE1}}$=0, $\overline{\text{BE0}}$=0
FIFO byte n from DAL [07:00]
Note: Although all bytes are active, only required bytes are used internally.

BYTE TRANSFER TO EVEN-WORD ADDRESS
Example: Last byte in receive buffer on evenword address.
680X0: SIZ1=0, SIZ0=1, A1=0, A0=0
FIFO byte n to DAL [31:24]
80X86: $\overline{\text{BE3}}$=1, $\overline{\text{BE2}}$=1, $\overline{\text{BE1}}$=1, $\overline{\text{BE0}}$=0
FIFO byte n to DAL [07:00]

BYTE TRANSFER FROM ODD-BYTE (3 MOD 4) ADDRESS
Example: First byte in transmit buffer on odd-byte (3 MOD 4) address.
680X0: SIZ1=0, SIZ0=0, A1=0, A0=0
FIFO byte n from DAL [15:08]
80X86: $\overline{\text{BE3}}$=0, $\overline{\text{BE2}}$=0, $\overline{\text{BE1}}$=0, $\overline{\text{BE0}}$=0
FIFO byte n from DAL [23:16]
Note: Although all bytes are active, only required bytes are used internally.

BYTE TRANSFER TO ODD-BYTE (3 MOD 4) ADDRESS
Example: First byte in receive buffer on odd-byte (3 MOD 4) address.
680X0: SIZ1=0, SIZ0=1, A1=1, A0=1
FIFO byte n to DAL [15:08]
80X86: $\overline{\text{BE3}}$=0, $\overline{\text{BE2}}$=1, $\overline{\text{BE1}}$=1, $\overline{\text{BE0}}$=1
FIFO byte n to DAL [23:16]

FIFO OPERATIONS

The FIFO provides temporary buffer storage for data being transferred between the parallel bus I/O pins and serial bus I/O pins. The capacity of the FIFO is 48 bytes.

TRANSMIT

Data is loaded into the FIFO by the ILACC under microprogram control

During transmission, the ILACC transfers data from the FIFO onto the network via an external transceiver (and optional external SIA). Once the FIFO has been emptied to the point where fewer than 33 bytes remain in the FIFO (or transmit data has yet to be loaded into the FIFO prior to transmission), the ILACC will request the bus by asserting $\overline{\text{HOLD/BUSREQ}}$. The ILACC will start sending the preamble (provided the network is quiet) as soon as the first transmit data byte is loaded into the FIFO. Should the transmitter be required to back off, there will be up to 48 bytes of data in the FIFO ready for transmission once the network again becomes quiet. Reception has priority over transmission during the time that the transmitter is backing off.

RECEIVE

Data is received from the network via an external transceiver (and optional external SIA) and is transferred into the FIFO under microprogram control. Once the FIFO has filled to the point where more than 16 bytes are stored, the ILACC will request the bus by asserting $\overline{\text{HOLD/BUSREQ}}$. Preamble (including the synchronization bits) is not loaded into the FIFO.

Serial Interface

The ILACC has two serial interfaces:

- General Purpose Serial Interface
- IEEE 802.3 Attachment Unit Interface (AUI)

GENERAL PURPOSE SERIAL INTERFACE

This is provided to allow alternate clock/data encoder transceivers. When the GPSI port is not in use (PORTSEL = 0, CSR15 bit 8), all inputs should be tied low and all outputs should be left floating.

SERIAL INTERFACE ADAPTER

This 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 in the collision path. In addition, the integral SIA provides the interface between the CMOS logic environment of the controller and the differential signaling environment of the transceiver.

SIA-Controller Interface

Since the ILACC incorporates the facilities of both the LANCE (Am7990) and SIA (Am7992A), the interface signals which previously appeared as hardwired pins are now internal.

To more easily understand the operation of the ILACC, this internal interface is described as a set of signals, defined as follows:

Internal Receive Enable (IRENA) – An output from the SIA to the controller section to indicate carrier presence. IRENA goes active when there is a negative transition on DI± that is more negative than the amplitude "Squelch Limit" and meets the pulse width requirements of the input filtering. IRENA goes inactive within 2 bit times of the last positive transition at DI±.

Internal Receive Clock (IRCLK) – The recovered clock from the differential input at DI±. An output from the SIA block to the controller section to clock in the serial bit stream. IRCLK is activated 1/4 bit time after the second negative Manchester preamble clock transition at DI± and remains active until the end of message.

Internal Receive Data (IRXD) – The recovered serial data stream from the SIA block to the controller section of the ILACC. When IRENA is active, signals at DI± meeting threshold and pulse width requirements will be clocked in by IRCLK and passed to the controller section of the ILACC.

Internal Transmit Enable (ITENA) – Identical in function to the RTS output. It is asserted (high) by the controller section to indicate that serial data is available for encoding and driving DO±.

Internal Transmit Clock (ITCLK) – An output from the SIA block to the controller section to clock out the serial bit stream and permit output data to be encoded.

Internal Transmit Data (ITXD) – The serial bit stream output from the controller section. When ITENA is active, signals at ITXD will be clocked out by ITCLK and appear as Manchester encoded data at the DO± outputs.

Internal Collision Detect (ICLSN) – ICLSN is an output from the SIA block to the controller section. When signals at the CI± differential inputs are driven by an external transceiver to indicate a collision, ICLSN will go high.

Transmit Path

The transmit section encodes separate clock and NRZ data input signals into a standard Manchester 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 (XCLK) for the SIA portion of the ILACC. It is divided by two to create the internal transmit clock reference (ITCLK). Both XCLK and ITCLK are fed into the SIA's Manchester Encoder to generate the transitions in the encoded data stream. ITCLK is used by the SIA to internally synchronize the Internal Transmit Data (ITXD) from the controller and Internal Transmit Enable (ITENA). ITCLK 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 CMOS-level input as a reference. Transmit accuracy of 0.01% is achieved (no external adjustments are required).

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+ and 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):

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

SIA Oscillator

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±.

	Min	Nom	Max	Units
1. Parallel Resonant Frequency		20		MHz
2. Resonant Frequency Error (CL = 50 pF)	–50	0	+50	PPM
3. Change in Resonant Frequency With Respect To Temperature (CL = 50 pF)	–40		+40	PPM
4. Motional Crystal Capacitance (C1)		0.022		pF
5. Series Resistance			35	Ω
6. Shunt Capacitance			7	pF
7. Drive Level			2	mW

Crystal Manufacturers include: Reeves-Hoffman P/N 04-20423-312
Epson P/N MA-506-20.0M-50PF (surface mount crystal)

External Clock Drive Characteristics

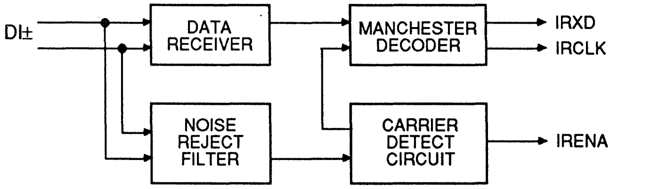
When driving the oscillator from an external clock source, XTAL2 must be left floating (unconnected). An external clock having the following characteristics must be used to ensure less than ±0.5 ns jitter at DO±.

Clock Frequency:	20 MHz +0.01%
Rise/Fall Time (tR/tF):	< 2 ns from 0.8 V to 2.0 V
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

Receiver Path

The principle functions of the Receiver are to signal the ILACC 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 à 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.



10594-006B

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 positive than minus 100 mV are also suppressed.

The Carrier Detection circuitry controls the stop and start of the phase-locked loop during clock acquisition. Clock acquisition requires a valid Manchester bit pattern of 1010 to lock onto the incoming message (see Receive Timing – Start of Reception Clock Acquisition waveform diagram).

When input amplitude and pulse width conditions are met at DI±, the internal enable signal from the SIA to controller (IRENA) 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 ITCLK. The first negative clock transition after IRENA 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.

IRCLK and IRXD are enabled 1/4 bit time after clock acquisition in bit cell 5. IRXD is at a HIGH state when the receiver is idle (no IRCLK). IRXD, however, is undefined when clock is acquired and may remain HIGH or change to LOW state whenever IRCLK is enabled. At 1/4 bit time through bit cell 5, the controller portion of the ILACC sees the first IRCLK transition. This also strobes in the incoming fifth bit to the SIA as Manchester "1". IRXD may make a transition after the IRCLK rising edge in bit cell 5, but its state is still undefined. The Manchester "1" at bit 5 is clocked to IRXD 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 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 RCLK by 10 to 1.

Carrier Tracking and End of Message

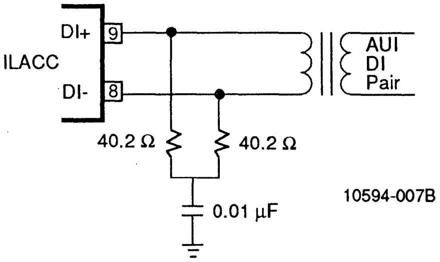
The carrier detection circuit monitors the DI± inputs after IRENA is asserted for an end of message. IRENA deasserts 1 to 2 bit times after the last positive transition on the incoming message. This initiates the end of reception cycle. The time delay from the last rising edge of the message to IRENA deassert allows the last bit to be strobed by IRCLK and transferred to the controller section but prevents any extra bit(s) at the end of message. When IRENA deasserts (see Receive Timing-End of Reception (Last Bit = 0) and Receive Timing-End of Reception (Last Bit = 1) waveform diagrams) an IRENA hold off timer inhibits IRENA assertion for 1 to 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. IRCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit and clocks the data out on IRXD on the following IRCLK. The data receiver also generates the signal used for phase detector comparison to the internal SIA VCO.

Differential Input Termination

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 if direct coupling is used (as shown in theDifferential Input Termination diagram below). Thedifferential input impedance, Z_{IDF}, and the common-mode input impendance, Z_{ICM}, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. 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 SIA 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 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 will properly decode data

Data Flow Overview

DESCRIPTOR RING ACCESS MECHANISM – DETAILED DESCRIPTION

At initialization the ILACC will have read the base address of both the transmit and receive descriptor rings. These will be stored in CSRs for use by the ILACC during subsequent operation (CSR24, 25 = Base Address of Rx Ring, CSR30, 31 = Base Address of Tx Ring).

With the ILACC started and the transmit and receive functions enabled, the base address of each ring will be loaded into the current descriptor address registers (CSR28, 29 = Current Address of Rx Ring, CSR34, 35 = Current Address of Tx Ring).

The address of the next descriptor in the transmit and receive rings will be computed and loaded into CSR26, 27 (Next Address of Rx Ring) and CSR32, 33 (Next Address of Tx Ring).

Polling:

When there is no channel activity and there is no pre or post receive or transmit activity being performed by the ILACC, the ILACC will periodically poll the current receive and transmit descriptor entries in order to ascertain their ownership.

A typical polling operation consists of the following: the ILACC 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 RMD0 and RMD1 of the current TDTE at periodic polling intervals. The accesses will be performed as dual-cycle DMA transfers which are described in the BUS MASTER CYCLES section. In one dual-cycle DMA transfer, RMD0 will be read followed by RMD1. Following that operation, another dual-cycle DMA transfer will be executed with TMD0 and TMD1 being read in that order. All information collected during polling activity will be stored internally in the appropriate CSRs (CSR18, 19, 40, 20, 21, 42, 50, and 52).

A typical polling operation is the product of the following conditions: The RDTE access is only performed if the ILACC does not possess ownership of the current RDTE. The TDTE access is only performed if the ILACC does not possess ownership of the current TDTE. Additionally, if RXON=0, the ILACC will never poll RDTE locations. Similarly, if TXON=0, the ILACC will never poll TDTE locations. It should be noted that the typical system should always have at least one RDTE available for the possibility of an unpredictable receive event. Given that this condition is satisfied, the RDTE poll would rarely be seen and hence, the typical poll operation simply consists of a check of the current TDTE.

The poll time interval is nominally defined as 32 768 BCLK periods, 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 ILACC, 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, an obligatory poll operation will always be performed). The poll time count register is never reset.

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 ILACC finds that the OWN bit of that TDTE is not set, then the ILACC resumes the poll time count and reexamines the same TDTE at the next expiration of the poll time count. Note that the information collected during the previous poll remains in the device unused.

If the OWN bit of the TDTE is set but STP=0, the ILACC 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 ILACC 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 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 ILACC 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 operation is performed as a dual-cycle DMA with TMD0 being read first and TMD1 last. The contents of TMD0 and TMD1 will be stored in the appropriate CSR locations, i.e. next RX/TX Buffer Address (CSR22, 23), next RX/TX Byte Count (CSR44), and next RX/TX Status (CSR54), regardless of the state of the OWN bit. This transmit descriptor lookahead operation is performed only once. There will be no second chance.

If the ILACC does not own the next TDTE (i.e. the second TDTE for this packet), it will complete transmission of the current buffer and 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 ILACC 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 ILACC 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 ILACC does own the second TDTE in a chain, it will gradually empty the contents of the first buffer (as they 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 may perform one data DMA access on the next buffer in the chain before executing another lookahead operation. (i.e. a lookahead to the third descriptor , if the ENP has not yet been found.)

In the case where the ENP bit has been found in a descriptor, no more lookahead accesses will be performed. This condition applies regardless of whether the transmit packet is contained in a single buffer or is chained.

The ILACC will commence to DMA transfer the remaining transmit buffer data into the FIFO, and once the packet has been successfully transmitted will write to TMD2 and TMD1 (in that order), updating the status and OWN bits.

If an error occurs in the transmission before all of the bytes of the current buffer have been transferred, a dual-cycle DMA transfer will be executed in order to write to TMD2 and TMD1 (in that order) of the current buffer. In that case, data transfers from the next buffer will not commence. Instead, following the TMD2/TMD1 update, the ILACC will return to the polling microcode where it will immediately access the next descriptor and find the condition OWN=1 and STP=0 as described earlier. As described for that case, the ILACC 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 (i.e. either successful or with errors) and the completion of the descriptor updates, the ILACC will always perform another poll operation. As described earlier, this poll operation will begin with a check of the current RDTE unless the ILACC already owns that descriptor.The ILACC will proceed to polling the next TDTE. If the transmit descriptor OWN bit has a zero value the ILACC will resume poll time count incrementing. If the transmit descriptor OWN bit has a value of ONE, the ILACC 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 ILACC completes a transmit packet (either with or without error) and writes the status information to the current descriptor, the TINT bit of CSR0 is set to indicate the completion of a transmission. This causes an interrupt signal if the INEA bit of CSR0 has been set and the TINTM bit of CSR3 is reset.

When transmit packets are not chained, ENP=1 is found in the first descriptor for the packet and no transmit descriptor lookahead will be performed.

Receive Descriptor Table Entry (RDTE)

If a poll operation has revealed that the current Receive Descriptor Table Entry is not owned by the ILACC, the ILACC will continue to poll the current RDTE according to the polling sequence described above. The RMD0 and RMD1 information that was gathered during the poll operation will remain on the device unused.

If a poll operation has revealed that the current RDTE belongs to the ILACC, additional poll accesses to the current RDTE are not necessary. Future poll operations will not include RDTE accesses as long as the ILACC retains ownership of the current RDTE.

When receive activity is present on the channel, the ILACC 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 ILACC checks the ownership of the current buffer, as stored internally.

If ownership is lacking, then the ILACC will immediately perform a poll of the current RDTE. This poll of the current RDTE will be a dual-cycle transfer, reading RMD1 and RMD0 (in that order). If ownership is still denied, the ILACC has no buffer in which to store the incoming message. The MISS bit will be set in CSR0; An interrupt will be generated if INEA=1 (CSR0) and MISSM=0 (CSR3). Another poll of the current RDTE will not occur until the packet has finished.

If the ILACC 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, the ILACC will perform a normal lookahead operation of the next RDTE, reading RMD1 and RMD0 (in that order). Following this lookahead poll, transfers of the receive FIFO data may begin.

Regardless of ownership of the second receive descriptor, the ILACC 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 ILACC 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 ILACC does own the second (next) buffer, and when the first buffer is full, ownership will be passed back to the system by writing a zero to the OWN bit of RMD1. Receive data transfers to the second buffer may occur before the ILACC 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 for each descriptor. There will be no second chance.

This activity continues until the ILACC recognizes the completion of the packet (the last byte of this receive message has bee removed from the FIFO). The ILACC will subsequently update the current RDTE by writing to RMD2 and RMD1 (in that order). This operation updates the ENP bit, message byte count (MCNT), runt packet count (RPC), and the receive collision count (RCC). The ILACC will then overwrite the 'current' entries in the CSRs with the 'next' entries.

If after the last byte of data has been read out of the FIFO, the packet is detected to be a runt, the "current" buffer address and status in the RDTE are not updated, and the "current" CSRs are reloaded with the "backup" CSR values. The runt packet data buffer will be overwritten by the next received message data.

SERIAL TRANSMISSION

Serial transmission consists of sending an unbroken bit stream from either the TxD output or DO± pair consisting of:

1. Preamble/Start Frame Delimeter (SFD): 56 bits (7 bytes) of alternating ONES and ZEROES terminating with the 8-bit (1-byte) SFD sequenceof 10101011.
2. Data: The serialized byte stream from the FIFO shifted out with the LSB first.
3. CRC: The inverted 32-bit polynomial calculated from the data, address, and type fields, shifted out with the MSB first. The CRC is not transmitted if:

   a. Transmission of the data field is truncated for any reason.

   b. CI± becomes active at any time during transmission.

   c. DTCR = 1 (CSR15, bit 03) in a normal or loopback transmission mode.

The transmission is indicated at the general purpose serial interface by the assertion of RTS with the first bit of the preamble and the negation of RTS after the last transmitted bit.

The ILACC starts transmitting the preamble when the following are satisfied:

1. The buffer managment unit of the ILACC has determined that there is a pending transmission.
2. The interpacket gap time (IPG) has elapsed.
3. The backoff interval has elapsed, if a retransmission is required.

SERIAL RECEPTION

Serial reception consists of receiving an unbroken bit stream on the RxD input or DI± pair consisting of:

1. Preamble/SFD: The two consecutive ONES of the SFD occurring a minimum of 8 bit times after the carrier is detected (assertion of internal or external Carrier Sense) commence the serial to parallel conversion process and movement of the receive bit stream into the FIFO.
2. Data: The serialized byte stream following the Destination Address. The last four complete bytes of data are the CRC. The Destination Address and the Data are framed into bytes and enter the FIFO. Source Address and Length/Type field are part of the data which are transparent to the ILACC.

PREAMBLE 1010...1010	SYNCH 1 1	DEST. ADR	SOURCE ADR	TYPE	DATA	FCS
62 BITS	2 BITS	6 BYTES	6 BYTES	2 BYTES	46–1500 BYTES	4 BYTES

Ethernet Frame Format

PREAMBLE 1010...1010	SYNCH 10101011	DEST. ADDR	SOURCE ADDR	LENGTH	LLC DATA	PAD	FCS
56 BITS	8 BITS	6 BYTES	6 BYTES	2 BYTES	46–1500 BYTES		4 BYTES

10594-009A

IEEE 802.3 MAC Frame Format

Reception is indicated at the general purpose serial interface by the assertion of CRS and the presence of clock on RxC while RTS is inactive. The ILACC does not sample the received data until about 8 bit times (800 ns for 10-MHz operation) after CRS goes high. Note that the receive process will be aborted if two consecutive ZEROES occur during the preamble/SFD sequence prior to the two ONES pattern within the SFD.

FRAME FORMATTING

The ILACC performs the encapsulation/decapsulation function of the data link layer (2nd layer of ISO model) as follows:

Transmit

In transmit mode, the user must supply the Destination Address, Source Address, and Length/Type fields as part of the data field in the transmit buffer memory. The ILACC will append the preamble, SFD (or synchronization bits), and CRC (Frame Check Sequence) to the frame as is shown in the figures below.

Receive

In receive mode, the ILACC strips off the preamble and SFD (or synch bits) and transfers the rest of the frame, including the four CRC bytes, to memory. The ILACC 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 data is discarded. A runt packet is normally the result of a collision.

Error Reporting and Diagnostics

Extensive status reporting and diagnostics are provided by the ILACC.

Error Reporting

Error conditions reported relate either to the network as a whole or to the individual node. Network error and status information is reported in the ILACC internal CSRs, and also within the message buffer descriptors passed between the ILACC and the host or user.

Node Errors Include:

Babbling Transmitter – Transmitter attempting to transmit more than 1518 bytes.

Collision – Collision detection circuitry nonfunctional.

Missed packet – Insufficient buffer space.

Memory time-out – Memory response failure.

Overflow – The receiver has lost all or part of the incoming packet due to overflow of internal FIFO.

Buffer error – The receiver or transmitter does not own the next buffer when data chaining.

Underflow – When transmitting, the FIFO has emptied before the end of the packet was loaded into the ILACC.

Network Related Errors:

Framing – Packet did not end on a byte boundary.

CRC – A CRC error was detected on the incoming packet.

Receive Collision Count – Counts collisions on the network between any two receive packets.

Runt Packet Count – Counts undersize packets on the network between any two received packets.

Transmit Collision Count – Counts the number of retrys to send an individual packet.

The ILACC performs several diagnostic routines which enhance the reliability and integrity of the system. These include a CRC logic check and two loopback modes (internal/external). Errors may be introduced into the system to check error detection logic. A Time Domain Reflectometer is incorporated into the ILACC to aid the location of cable faults. Short or open circuit conditions manifest themselves in reflections which are sensed by the TDR.

FRAMING ERROR (DRIBBLING BITS)

The ILACC 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, 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, there is also a Framing error (FRAM = 1).
3. If the number of the dribbling bits = 0, there is no Framing error. There may or may not be a CRC error.

LOSS OF CARRIER

After the ILACC initiates a request for transmission (either internally to the embedded SIA, or externally via the RTS output), it will expect to see a "carrier sense" returned from the internal SIA or the external transceiver (CRS should become active when the general purpose serial interface is used). The "carrier sense" signal must be asserted during the time that the request to send is active. If "carrier sense" does not become active in response to the request or becomes inactive before the end of transmission, the LCAR (loss of carrier) error bit will be set in TMD2 after the packet has been transmitted.

DIAGNOSTICS

Loopback

The normal operation of the ILACC functions as a half-duplex device. However, a pseudo-full duplex mode is provided for on-line operational test of the ILACC. In this configuration, simultaneous transmission and reception of a loopback packet is enabled with the following constraints:

1. The packet length must be no longer than 41 bytes with DTCR=0 and 45 bytes with DTRC=1. If the transmit packet size exceeds the recommended length, LBE is set to indicate an overwrite in the FIFO.
2. Serial transmission does not begin until the FIFO contains the entire output packet.
3. Moving the input packet from the FIFO to the memory does not begin until the serial input bit stream terminates.
4. The CRC may be generated and appended to the output serial bit stream or may be checked on the input serial bit steam but not both in the same transaction.
5. The packets should be addressed to the node itself.
6. During normal loopback, all address schemes remain valid.
7. Multicast addressing can be used only when DTCR = 1 (CSR15 bit 3). In this case, the user needs to append the CRC bytes.
8. Ordinary receive activity will be ignored.

Loopback is controlled by INTL, DTCR, and LOOP (CSR15 bits 6, 3, 2).

Loop	INTL	Function
0	X	No loopback, normal operation
1	0	External loopback
1	1	Internal

Interpacket Gap Time (IPG)

General Purpose Serial Interface

The IPG time is 96 network data bit times (9.6 microseconds). The interpacket gap time for back-to-back transmission is 96 to 106 network data bit times, including synchronization. The interpacket delay interval begins immediately after the negation of the CRS signal.

In the following paragraphs, the network data rate is assumed to be 10MHz.

Following receive activity, the IPG count will begin when CRS falls. If at any time during the first part of the IPG (the first 6.0 microseconds) CRS is reasserted, then the IPG count will be reset to zero. The counter is restarted when CRS falls again. If CRS is asserted during the second part of the IPG (after the first 6.0 microseconds), then the IPG count will NOT be reset but will instead proceed as scheduled to 9.6 microseconds. More succinctly stated, CRS activity in the first part of the IPG will cause the counter to be reset. CRS activity in the second part of the IPG will not cause the counter to be reset. This is the two part deferral process outlined in the ANSI/IEEE Std 802.3 1990-09-21 specification section 4.2.8. (The newly arriving data will be examined for preamble, sync and address, and will be accepted if the packet address matches the station's.)

Following transmit activity, the IPG count will begin when CRS falls. For the first 4.0 microseconds of the IPG CRS activity will be ignored to allow the passing of the SQE test signal. After the expiration of this SQE window (ANSI/IEEE Std 802.3 section 7.2.4.6), the ILACC will again accept receive activity examining incoming bit streams for preamble, sync and addressing information. However, the IPG counter will proceed to the 9.6 microsecond value regardless of the new CRS activity (deference process, section 4.2.8), and, if another transmission is pending during this time, the transmission will commence at the expiration of the current IPG count regardless of the state of CRS. In other words, following a transmission there is no distinction made between the first and second parts of the IPG with respect to CRS activity. The IPG counter always counts directly to 9.6 microseconds with no interruptions beginning with the fall of CRS(section 4.2.8).

Asserting the optional Transmit Two Part Defer bit in CSR3 (TX2PTDFR) will cause the counting of the IPG following transmit operations to be similar to that of the receive case except that the SQE window will still be in place following transmit activity. In other words, asserting TX2PTDFR will allow CRS activity to reset the IPG counter during the first part of the IPG count when it follows a transmission (with the caveat that CRS is masked internally for the first 4.0 microseconds of the IPG count due to the SQE window). This has the affect of allowing IPG reset after transmit only between 4.0 and 6.0 microseconds following transmission when TX2PTDFR is set. The TX2PTDFR bit is provided as an option. Implementation of TX2PTDFR is not currently included in the ANSI/IEEE 802.3 standard.

Following either the de-assertion of the previous reception's CRS or the expiration of the SQE window following a transmission, the ILACC can begin examining a new incoming packet. After the assertion of CRS and the start of RXC, about 8 bit times will be required before the ILACC starts to look for the sync bits (011). If CRS is asserted during the 4.0 microsecond SQE window following a transmission and remains asserted at the expiration of the SQE window, any of the following scenarios are possible: If 10 bits of preamble plus the sync bit occur after the SQE window expires, the address portion of the message will be examined for a match. If there is not enough preamble left in the message at the end of the SQE window to satisfy the 10 bit requirement, the ILACC will respond in one of two ways.

Either a 011 sequence of bits will appear in the message as the receive function is enabled and the receiver will believe that it has just found the sync bit, or a 00 sequence of bits will be found and the ILACC will completely ignore the rest of the incoming message just as though there occurred an address mismatch. In the case where 011 is found, if the next six bytes of data manage to be interpreted as a correct address match (through perhaps logical addressing or promiscuous addressing), the remaining data will be received and (if not a runt) will be stored in a buffer in memory. However, it is very likely that CRC error and possibly the FRAM error will be indicated for this packet.

If receive activity begins in the second part of the IPG as detected by the ILACC and there exists a pending transmission, the ILACC will ignore the receive activity since at the expiration of IPG the pending transmission will be broadcast onto the network as specified in the ANSI/IEEE spec (see section 4.2.8). (There is no need to accept the receive activity in this case since the ILACC can predict that it will begin transmission and cause collision with the receive packet.)

Internal SIA Interface

The timing of the integrated SIA is identical to that described above. However, no external signals are available to indicate the start of the IPG time out.

COLLISION DETECTION AND COLLISION JAM

General Purpose Serial Interface

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

Internal SIA Interface

When using the SIA interface, collisions are detected by monitoring the differential CI± signals returned from the bus transceiver. JAM is issued the same way as General Purpose Serial Interface.

Receive-Based Collision

If CDT or CI± (serial interface dependent) is asserted during the reception of a packet, the reception is immediately terminated. If a collision occurs within 6 byte times (4.8 ms at 10M bit/s network data rate), the packet will be rejected because of an address mismatch and the FIFO write pointer will be reset. If a collision occurs within 64 byte times (51.2 ms at 10M bit/s), the packet will be rejected since it is a runt packet. If a collision occurs after 64 byte times (late collision), this will result in a truncated packet being written to the memory buffer. The CRC and FRAM bits may be set, and the MCNT will not match the length given in the length field.

Transmit-Based Collision

When a transmission attempt has been terminated due to the assertion of CDT or CI± (a collision that occurs within 64 byte times), the ILACC will retry the transmission up to a maximum of 15 times. The ILACC does not try to reread the descriptor entries from the transmit TDTE upon each collision. The descriptor entries for the current buffer are internally saved in the CSRs. The scheduling of each re-transmission is determined by a controlled randomized process called the "truncated binary exponential backoff". Upon completion of the COLLISION JAM interval, the ILACC calculates a delay before re-transmitting. The delay is an integral multiple of the SLOT TIME. The SLOT TIME is defined as 512 bit times (64 bytes). If a collision is detected during transmission, the SLOT TIME counter is started at the end of the COLLISION JAM sequence.

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 of:

$0 < r < 2^k$, where $k = \min(n, 10)$
For example, if this is the third retry:
$n = 3$
$k = \min(3,10) = 3$
$2^3 = 8$

If the number selected is zero (r = 0), the ILACC will begin re-transmission at the end of the 96 clock IPG time.

If all 15 retry attempts fail, the ILACC 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.

When there are excessive collisions of any buffer in a multi-buffer packet, the status will be written in the current descriptor. The OWN bit in the subsequent descriptor will be reset until the STP is found.

If there is a late collision (collision occurring after 64 byte times), the ILACC will not retransmit. It will terminate the transmission, note the LCOL error (TMD2), 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 FIFO in anticipation of retransmission. It is important that the ILACC be ready to transmit when the backoff interval elapses to utilize the channel properly.

Time Domain Reflectometry

The ILACC contains a time domain reflectometry counter. The TDR counter is ten bits wide. It counts at the internal ILACC network crystal frequency (XCLK). It is cleared by the microprogram and commences counting once the carrier is detected during transmission. Counting ceases if a collision is detected (CDT or CI+) or the request for transmission (RTS for general interface, internal request for integrated SIA) is dropped. The counter does not wrap around but will freeze at the maximum count (all ones) until cleared. The value in the TDR is written to the TDTE TMD2 following the transmission of the packet. The TDR is used to determine the location of suspected cable faults.

Heartbeat

General Purpose Serial Interface

During the interpacket gap delay following the negation of RTS, the CDT input may be asserted by some transceivers as a self-test (SQE TEST in ANSI/IEEE 802.3). If the CDT input is not asserted within the 40 network bit times (4 microseconds) period following the completion of transmission (after CRS goes inactive), then the ILACC will set the CERR bit in CSR0. CERR error will not cause an interrupt to occur. (This window of 40 network bit times following a transmission is defined in the ANSI/ IEEE 802.3 specification as the carrier-inhibit-time.)

Internal SIA

When PORTSEL=0 (MODE register bit 8), the integrated SIA will expect the SQE TEST signal to appear as a 10MHz waveform on the CI± pair within the 40 network bit times described above. Again, should this signal not appear, the CERR bit of CSR0 will be asserted.

Cyclic Redundancy Check (CRC)

The ILACC utilizes the 32-bit CRC function used in the Autodin-II network. Refer to the Ethernet specification (Frame Check Sequence Field and Appendix C: CRC Implementation) or ISO 8802-3 ANSI/IEEE Std. 802-3 can be verified using the LOOP (CSR15 bit 2) and DTCR (CSR15 bit 3) bits. See Frame Check Sequence Field section on page 21 for more detail. The ILACC CRC logic is as follows:

1. **TRANSMISSION – LOOP = 0 (CSR15 bit 2) and DTCR = 0 (CSR15 bit 3).** The ILACC calculates the CRC from the first bit following the Start bit to the last bit of the data field. The CRC value inverted is appended onto the transmission in one unbroken bit strea, with the most significant bit transmitted first.
2. **RECEPTION – LOOP = 0 (CSR15 bit 2).** The ILACC performs a check on the input bit stream from the first bit following the start bit to the last bit in the frame. The ILACC continually samples the state of the CRC checked 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.
3. **LOOPBACK – LOOP = 1 (CSR15 bit 2) and DTCR = 0 (CSR15 bit 3).** The ILACC 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.
4. **LOOPBACK – LOOP = 1 (CSR15 bit 2) and DTCR = 1 (CSR15 bit 3).** The ILACC 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 steam during transmission.

PROGRAMMABLE RESOURCES

This section defines the control and status registers and the memory data structures required to program the ILACC.

User Programmable Registers

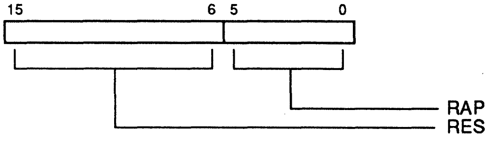
Internal programmable registers are accessed in a two step operation. First, the address of the programmable register is written into the Register Address Port (RAP). Subsequent read or write operations will access the register pointed to by the contents of the RAP. The data will be read from (or written into) the selected Register through the Data Port.

The C/$\overline{\text{D}}$ pin permits external selection of either the Data or the Register Address Port as follows:

C/$\overline{\text{D}}$	Port
0	Data Port
1	Register Address Port

REGISTER ADDRESS PORT (C/$\overline{\text{D}}$ = H)

The high-order 16 bit (i.e., DAL 16-31) are undefined. RAP is defined as follows:



10594-010A

Bit	Name	Description
15-06	RES	RESERVED, written and read as zero.
05-00	RAP	Register Address Port select. Selects the Control and Status Register location to be accessed. RAP is cleared by $\overline{\text{RESET}}$.

CONTROL AND STATUS REGISTERS

The Control and Status Registers (CSRs) are internal to the ILACC, and accessed on an individual basis by first writing the appropriate CSR address into the RAP.

Regardless the state of the STOP bit, READ/WRITE access is permitted to CSR0 and CSR3–4. To access CSR1–2, the STOP bit in CSR0 must be set.

All CSR data transfers will take place over the lower 2 bytes (DAL$_{15-0}$) for all slave operations. DAL$_{31-16}$ are undefined.

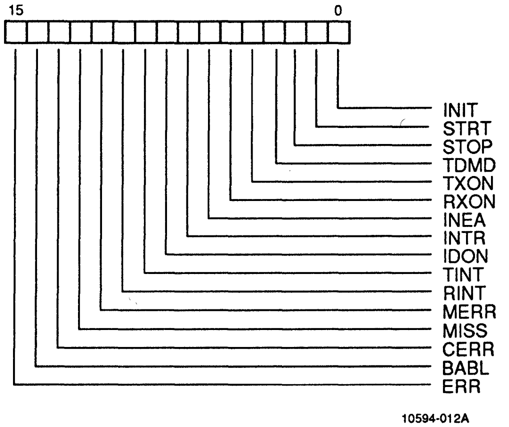
Addresses 5-58 are available as READ ONLY, providing the STOP bit is set. These allow the host to verify data loaded during initialization and/or monitor ILACC functions.

The internal programmable registers are mapped as follows:

RAP	CSR	Contents
00	CSR0	CONTROL AND STATUS REGISTER 0
01	CSR1	CONTROL AND STATUS REGISTER 1
02	CSR2	CONTROL AND STATUS REGISTER 2
03	CSR3	CONTROL AND STATUS REGISTER 3
04	CSR4	CONTROL AND STATUS REGISTER 4
05	CSR5	TRANSMIT RETRY COUNT
06	CSR6	RX/TX DESCRIPTOR TABLE LENGTH
07	CSR7	RECEIVE COLL./RUNT PACKET CNT.
08	CSR8	LOGICAL ADDRESS FILTER 15-0
09	CSR9	LOGICAL ADDRESS FILTER 31-16
10	CSR10	LOGICAL ADDRESS FILTER 47-32
11	CSR11	LOGICAL ADDRESS FILTER 63-48
12	CSR12	PHYSICAL ADDRESS 15-0
13	CSR13	PHYSICAL ADDRESS 31-16
14	CSR14	PHYSICAL ADDRESS 47-32
15	CSR15	MODE REGISTER
16	CSR16	INITIALIZATION BLOCK ADDRESS 15-0
17	CSR17	INITIALIZATION BLOCK ADDRESS 31-16
18	CSR18	CURRENT RX BUFFER ADDRESS 15-0
19	CSR19	CURRENT RX BUFFER ADDRESS 31-16
20	CSR20	CURRENT TX BUFFER ADDRESS 15-0
21	CSR21	CURRENT TX BUFFER ADDRESS 31-16
22	CSR22	NEXT RX/TX BUFFER ADDRESS 15-0
23	CSR23	NEXT RX/TX BUFFER ADDRESS 31-16
24	CSR24	BASE ADDRESS OF RX RING 15-0
25	CSR25	BASE ADDRESS OF RX RING 31-16
26	CSR26	NEXT ADDRESS OF RX RING 15-0
27	CSR27	NEXT ADDRESS OF RX RING 31-16
28	CSR28	CURRENT ADDRESS OF RX RING 15-0
29	CSR29	CURRENT ADDRESS OF RX RING 31-16
30	CSR30	BASE ADDRESS OF TX RING 15-0
31	CSR31	BASE ADDRESS OF TX RING 31-16
32	CSR32	NEXT ADDRESS OF TX RING 15-0
33	CSR33	NEXT ADDRESS OF TX RING 31-16
34	CSR34	CURRENT ADDRESS OF TX RING 15-0
35	CSR35	CURRENT ADDRESS OF TX RING 31-16
36	CSR36	RUNT BACKUP BUFFER ADDRESS 15-0
37	CSR37	RUNT BACKUP BUFFER ADDRESS 31-16
38	CSR38	RETRY BACKUP BUFFER ADDRESS 15-0
39	CSR39	RETRY BACKUP BUFFER ADDRESS 31-16
40	CSR40	CURRENT RX BYTE COUNT
42	CSR42	CURRENT TX BYTE COUNT
44	CSR44	NEXT RX/TX BYTE COUNT
46	CSR46	POLL TIME COUNT
48	CSR48	HI ADDRESS INIT BLOCK
50	CSR50	CURRENT RX STATUS
52	CSR52	CURRENT TX STATUS
54	CSR54	NEXT RX/TX STATUS
56	CSR56	RUNT BACKUP BYTE COUNT
58	CSR58	RETRY BACKUP BYTE COUNT
59	CSR59	REVISIN REGISTER

CONTROL AND STATUS REGISTER 0 (CSR0). RAP = 0

The ILACC updates CSR0 by logical "ORing" the previous and present values.



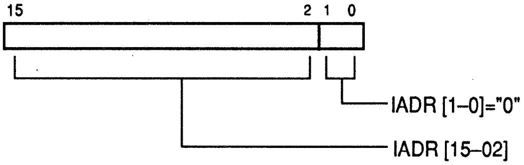
10594-012A

Bit	Name	Description
15	ERR	ERROR summary is set by the "OR" 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 packet. BABL indicates excessive length in the transmit buffer. It will be set after 1519 data bytes have been transmitted. The chip will continue to transmit until the byte count equals zero. When BABL is set, an INTR interrupt will be generated providing INEA = 1 and the mask bit BABLM (CSR3 bit 14) is clear. BABL is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by RESET or by setting the STOP bit.
13	CERR	COLLISION ERROR indicates that the collision input to the ILACC (CDT or CI±) failed to activate within 40 network bit times after a transmission was completed (the 40 bit times window commences upon CRS going inactive). This collision after transmission is a transceiver test feature (SQE Test). CERR will not cause an interrupt to be generated (INTR = 0, INTR line unaffected). CERR is READ/CLEAR only. It is set by the chip, and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by RESET or by setting the stop bit.
12	MISS	MISSED PACKET is set when the receiver loses a packet, because it does not own a receive buffer and the FIFO has overflowed, 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 INTR interrupt will be generated providing INEA = 1, and the mask bit MISSM (CSR3 bit 12) is clear. MISS is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by RESET or by setting the STOP bit.
11	MERR	MEMORY ERROR is set when the chip is the Bus Master and has not received a "ready" indication (READYL Low) from memory within 512 XCLK counts (i.e., 25.6µs for 10-MHz network data rate) after asserting DAS. When a Memory Error is detected, the receiver and transmitter are turned off. In addition, the INTR pin will be asserted LOW provided the INEA bit =1 and the MERRM bit (CSR3 bit 11) =0. MERR is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by RESET or by setting the STOP bit.
10	RINT	RECEIVER INTERRUPT is set after the ILACC has completed a received packet and toggled the OWN bit in the last Receive Descriptor Ring in the chain. When RINT is set, the RINTR interrupt will be generated, providing INEA = 1 and the mask bit RINTM (CSR3 bit 10) is clear. The interrupt summary bit, INTR, will also be set, but the condition of the external INTR line is unaffected by the state of RINT. RINT is READ/CLEAR ONLY and set only by the chip. It can be cleared by writing a "1" into the bit, (writing a "0" has no effect), by the application of RESET or by setting the STOP bit.
09	TINT	TRANSMITTER INTERRUPT is set after the ILACC has completed a transmitted packet and toggled the OWN bit in the last Transmit Descriptor Ring in the chain. When TINT is set, an INTR interrupt will be generated providing INEA = 1 and the mask bit TINTM (CSR3 bit 09) is clear. TINT is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by RESET or by setting the STOP bit.
08	IDON	INITIALIZATION DONE indicates that the chip has completed the initialization procedure started by setting the INIT bit. When IDON is set, the chip has read the Initialization Block from memory and stored the new parameters. When IDON is set, an INTR interrupt will be generated, providing INEA = 1 and the mask bit IDONM (CSR3 bit 8) is clear. IDON is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has not effect. It is also cleared by RESET or by setting the STOP bit.

Bit	Name	Description
07	INTR	INTERRUPT FLAG indicates that one or more of the following interrupt-causing conditions has occurred: BABL, MISS, MERR, RINT, TINT, IDON, LBE or TXSTRT; and its associated mask bit is clear. If INEA = 1 and INTR = 1, the $\overline{\text{INTR}}$ output pin will become active when any one or more of the above interrupt flags is set, with the exception of RINT. INTR is READ ONLY; writing this bit has no effect. INTR is cleared by $\overline{\text{RESET}}$, by setting the STOP bit, or by clearing the relevant interrupt bit(s).
06	INEA	INTERRUPT ENABLE will cause an external $\overline{\text{INTR}}$ to be generated for any of the following interrupt bits set in CSR0 (providing the associated mask bit in CSR3 is clear): BABL, MISS, MERR, TINT or IDON, as well as LBE and TXSTRT in CSR4 (providing the mask bits in CSR4 are clear). Note that RINT will not cause the $\overline{\text{INTR}}$ line to be asserted. If INEA = 1 and INTR = 1 (caused by any interrupt flag except RINT), 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 can be cleared by the host, by $\overline{\text{RESET}}$ or by setting the STOP bit.
05	RXON	RECEIVER ON indicates that the receiver is enabled. RXON is set when STRT is set, if DRX = 0 in the MODE register of the initialization block (and the initialization block has been read by the chip by setting the INIT bit). RXON is cleared when IDON is set and DRX = 1 in the MODE register, or if 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 of the initialization block and the INIT bit has been set. TXON is cleared when IDON is set and DTX = 1 in the MODE register, if a memory error (MERR) has occurred, or if transmitter underflow occurs (UFLO in transmit descriptor entry). TXON is READ ONLY; writing this bit has no effect. TXON is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
03	TDMD	TRANSMIT DEMAND, when set, causes the chip 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 chip's response to a Transmit Descriptor Ring entry insertion by the host. TDMD is WRITE WITH ONE ONLY and 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 $\overline{\text{RESET}}$ or by setting the STOP bit. Writing a "0" in this bit has no effect.
02	STOP	STOP disables the chip from all external activity when set and clears the internal logic. The chip 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 $\overline{\text{RESET}}$. Writing a "0" to this bit has no effect. STOP is cleared by setting either INIT or STRT.
01	STRT	START enables the chip to send and receive packets, perform direct memory access and do buffer management. Setting STRT clears the STOP bit. If STRT and INIT are set together, the INIT function will be executed first. STRT is READ/WRITE WITH ONE ONLY. Writing a "0" into this bit has no effect. STRT is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
00	INIT	INITIALIZE, when set, causes the chip to begin the initialization procedure and access the Initialization Block. Setting INIT clears the STOP bit. If STRT and INIT are set together, the INIT function will be executed first. INIT is READ/WRITE WITH ONE ONLY. Writing a "0" into this bit has no effect. INIT is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit. The ILACC does not internally clear the INIT bit on completion of the initialization procedure.

CONTROL AND STATUS REGISTER 1 (CSR1). RAP = 1

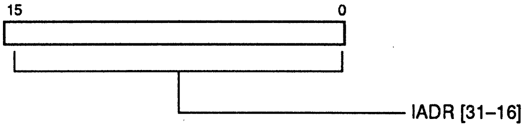
READ/WRITE accessible only when the STOP bit in CSR0 is set. Unaffected by $\overline{\text{RESET}}$.



Bit	Name	Description
15-00	IADR	The low order 16 bits of the address of the first word (lowest address) in the Initialization Block. Bits 00 and 01 must be zero. Used with CSR2 to form the base address of the Initialization Block in host memory.

CONTROL AND STATUS REGISTER 2 (CSR2). RAP = 2

READ/WRITE accessible only when the STOP bit in CSR0 is set. CSR2 is unaffected by $\overline{\text{RESET}}$.



10594-015A

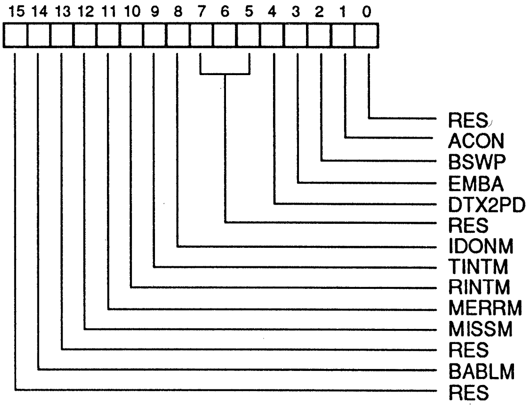
Bit	Name	Description
15-00	IADR	The high order 16 bits of the base address for the Initialization Block in host memory. Used with CSR1 to form the full 32 bit base address of theInitialization Block.

CONTROL AND STATUS REGISTER 3 (CSR3). RAP = 03

Accessible regardless of the state of the STOP bit. CSR3 is cleared by $\overline{\text{RESET}}$.

CSR3 allows redefinition of the Bus Master interface and masking of selected interrupts.

All other bits will be read as zeroes, regardless of data during write operations.

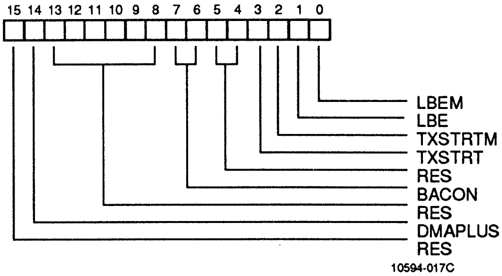


10594-016B

Bit	Name	Description
15	RES	RESERVED, written and read as zero.
14	BABLM	BABBLE MASK. If BABLM is set, the INTR bit and the external INTR will remain inactive, regardless of the state of BABL in CSR0.
13	RES	RESERVED, written and read as zero.
12	MISSM	MISSED PACKET MASK. If MISSM is set, the INTR bit and the external INTR will remain inactive, regardless of the state of MISS in CSR0.
11	MERRM	MEMORY ERROR MASK. If MERRM is set, the INTR bit and the external INTR will remain inactive, regardless of the state of MERR in CSR0.
10	RINTM	RECEIVE INTERRUPT MASK. If RINTM is set, the INTR bit and the external RINTR will remain inactive, regardless of the state of RINT in CSR0.
09	TINTM	TRANSMIT INTERRUPT MASK. If TINTM is set, the INTR bit and the external INTR will remain inactive, regardless of the state of TINT in CSR0.
08	IDONM	INITIALIZATION DONE MASK. If IDONM is set, the INTR bit and the external INTR will remain inactive, regardless of the state of IDON in CSR0.
07-05	RES	RESERVED, written and read as zeroes.
04	DTX2PD	DISABLE TRANSMIT TWO PART DEFER. When this bit is set, the optional transmit two part deferral mechanism is disabled and the deferral time or interpacket gap (IPG) following ILACC *transmit* activity is calculated according to the deferral process in IEEE 802.3, i.e. the IPG following ILACC transmit activity will be counted as 96 network bit times, regardless of carrier activity on the network. When DT2PD is cleared, the optional transmit two part deferral mechanism is in effect and the IPG following ILACC *transmit* activity will also be counted in two parts (regardless of the state of DT2PD, the IPG following network activity where the ILACC was *not* transmitting is always counted in two parts). Whenever the ILACC counts IPG in two parts, the first part of IPG will have a nominal duration of 6.0 microseconds, and the second part will have a duration of 3.6 microseconds. When DT2PD=0, carrier activity during the first part of the IPG following ILACC transmit activity will cause the IPG counter in the ILACC to be reset. Note that carrier activity is always masked out (ignored) for the first 40 network bit times following *transmit* activity to allow for SQE test, and will not reset the ILACC IPG counter. Hence the IPG counter will be reset if carrier activity occurs anytime between 4.0 µs to 6.0 µs following ILACC transmit activity. DT2PD is cleared upon RESET. This optional two part deferral mechanism is not currently part of the IEEE 802.3 specification.
03	EMBA	ENABLE MODIFIED BACKOFF ALGORITHM. When this bit is set, an alternative to the IEEE 802.3 truncated binary exponential backoff algorithm is invoked in the retry logic. This alternate algorithm is different from the IEEE 802.3 algorithm only in that the counting of the retry interval period is suspended whenever a carrier is present (DI± active or the CRS pin asserted, depending on the value of the PORTSEL bit) and resumes when a carrier is not present. EMBA is cleared upon RESET. This modified backoff algorithm is not currently a part of the IEEE 802.3 specification.
02	BSWP	BYTE SWAP. BSWP is read only and indicates whether the chip is programmed for systems that consider bits 00-07 on the data bus to be the most significant byte. When BSWP = 1, the chip will swap high-order bytes for low-order bytes on DMA data transfers between the FIFO and bus memory. Only data from the FIFO transfer is swapped; Initialization Block data and Descriptor Ring entries are not swapped. Byte swapping and the state of BSWP are controlled by the BACON bits; writing BSWP will have no effect.
01	ACON	ALE CONTROL. Defines the assertive state of ALE/AS when the chip is the Bus Master. ACON = 0: ALE (falling edge latches address). ACON = 1: AS (rising edge latches address). ACON is READ/WRITE and cleared by RESET.
00	RES	RESERVED, written and read as zero.

CONTROL AND STATUS REGISTER 4 (CSR4). RAP = 04

CSR4 controls selected diagnostic functions, microprocessor bus acquisition and DMA signaling. CSR4 can be accessed regardless of the state of the STOP bit. CSR4 is cleared by RESET.



Bit	Name	Description
15	RES	RESERVED, written and read as zero.
14	DMAPLUS	DMAPLUS = 1, allows a DMA burst transfer to continue until the internal FIFO is empty or full, or until preempted. When DMAPLUS = 0, the 16-byte burst mode is selected. Cleared by RESET, unaffected by STOP. Note that DMAPLUS affects the minimum transmit buffer size, for the first buffer in a chain. The first buffer should be a minimum of 100 bytes with DMAPLUS = 0, and increased to 116 bytes minimum with DMAPLUS = 1. This guarantees the ILACC can retransmit the packet if a collision occurs within the slot time.
13-08	RES	RESERVED and read as zeroes only.
07-06	BACON	BUS ACQUISITION CONTROL is used to optimally mate the ILACC with various microprocessor and system buses.The BACON bits will override the programming of BSWP (see table below). BACON is READ/WRITE and cleared by RESET, but unaffected by the STOP bit in CSR0.

BACON	Bus Configuration	BSWP*
00	32 bit 80X86	0
01	32 bit 680X0	1
10	Reserved	X
11	Reserved	X

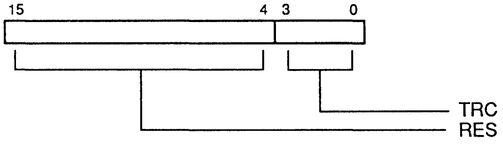
** NOTE: BSWP is read only. The swapping function is programmed in accordance with the Bus Acquisition Control (BACON) bits in CSR4.*

Bit	Name	Description
05-04	RES	RESERVED, written and read as zeroes only.
03	TXSTRT	TRANSMIT START status bit is set each time the ILACC attempts to begin a transmission (at the start of preamble). The INTR bit in CSR0 will be set if TXSTRTM is clear, and an INTR interrupt will be generated if INEA is set. TXSTRT is cleared by writing a "1" to its bit position, or by activating RESET or STOP. Unaffected by writing a "0".
02	TXSTRTM	TRANSMIT START MASK enables or disables the generation of the INTR bit and the external INTR interrupt associated with TXSTRT. Cleared by RESET or STOP.
01	LBE	LOOPBACK EXCESSIVE is set when the transmit packet size exceeds the recommended 41-byte limit (45 if DTCR = 1). In this case, the FIFO will overflow while filling. When LBE is set, STOP and INTR are also set. Cleared by writing a "1" or by hardware reset.
00	LBEM	LOOPBACK ERROR MASK enables or disables the generation of the INTR bit and the external INTR interrupt associated with LBE. Cleared by RESET or STOP.

USER READABLE REGISTERS

TRANSMIT RETRY COUNT (CSR5).

Contains the number of retry attempts to transmit the current buffer. Written into the transmit descriptor table entry when either a successful transmission has been completed or the transmission was aborted due to excessive retries. Read only access when STOP = 1. See the description of the transmit descriptor table for further details.

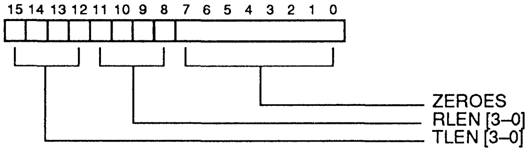


10594-018B

RX/TX DESCRIPTOR TABLE LENGTH (CSR6).

Contains a copy of the RLEN and TLEN bits read from user memory during the initialization sequence. Read only access when STOP = 1.

See the description of the Initialization Block for further details.

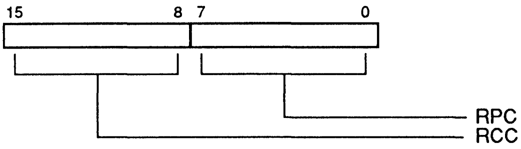


10594-019A

RECEIVE COLLISION/RUNT PACKET COUNT (CSR7)

The Receive Collision Count indicates the number of collisions detected on the network since the last received packet. The value is written into the receive descriptor table entry when a successful reception has been completed. The RCC will be reset immediately after the OWN bit for the descriptor is cleared by the ILACC, or upon a host read. Read only access when STOP = 1. See the description of the receive descriptor table entries for further details.

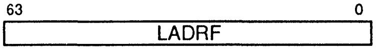
The Runt Packet Count indicates the number of runt packets (less then 64 bytes) addressed to the node since the last successfully received packet. The value is written to the receive descriptor table entry when a successful reception has been completed. The RCC will be reset immediately after the OWN bit for the descriptor is cleared by the ILACC or upon a host read. Read only access when STOP = 1. See the description of the receive descriptor table entries for more details.



10594-020A

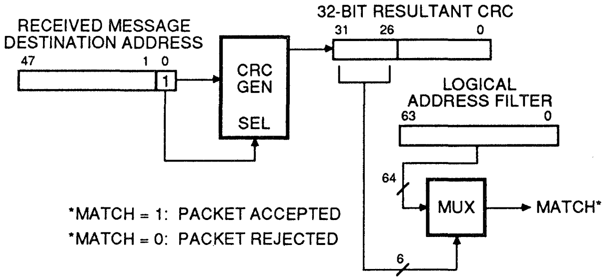
LOGICAL ADDRESS FILTER (CSR8-11)

READ ONLY access, when the STOP bit is set.



10594-021A

Bit	Name	Description
63-00	LADRF	The 64-bit mask that is used to accept incoming Logical Addresses. The Logical Address Filter is a read only copy of the entry of the same name in the Initialization Block. The value is updated only when the Initialization Block is read (INIT = 1). If the least significant address bit of a received message is set (PADR bit 00 = 1), then the address is deemed logical, and passed through the CRC generator. After processing the 48-bit destination address, a 32-bit resultant CRC is produced and strobed into a register. The high order 6 bits of this resultant CRC 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.



LOGICAL ADDRESS FILTER OPERATION

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 physical address that was loaded through the Initialization Block.

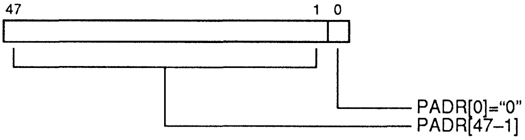
The Logical Address Filter is used in multicast addressing schemes. The acceptance of the incoming frame based on the filter value indicates that the message may be intended for the node. It is the user's responsibility to determine if the message is actually intended for the node by comparing the destination 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. If the Logical Address Filter is loaded with all zeroes (and PROM = 0), all incoming logical addresses except Broadcast will be rejected.

The multicast addressing in external loopback is operational only when DTCR = 1 in the Mode Register.

PHYSICAL ADDRESS REGISTER (CSR12-14)

READ ONLY access when the STOP bit is set.



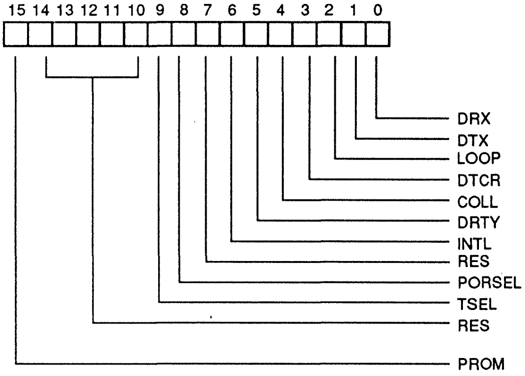
10594-023A

Bit	Name	Description
47-00	PADR	PHYSICAL ADDRESS is the unique 48-bit physical address assigned to the chip. PADR[0] must be zero. The Physical Address value is a read only copy of the entry of the same name in the Initialization Block. This internal copy is updated only when the Initialization Block is read (INIT = 1).

MODE REGISTER (CSR15)

The MODE Register is a read only copy of the entry of the same name in the Initialization Block. This internal copy is updated only when the Initialization Block is read (INIT = 1) and will be cleared by $\overline{\text{RESET}}$.

The MODE Register entry in the Initialization Block (located in user memory) permits alteration of the chip's operating parameters and provides the programming options for the internal SIA and/or general purpose interface port. Normal operation is with the Mode Register clear.



10594-024B

Bit	Name	Description
15	PROM	PROMISCUOUS Mode. When PROM = 1, all incoming packets are accepted.
14-10	RES	RESERVED, written and read as zeroes only.
09	TSEL	TRANSMIT MODE SELECT. TSEL = 0: In the idle transmit state, DO+ and DO- are equal, providing "zero" differential to operate transformer coupled loads. Delay and output return to zero are controlled internally. TSEL High: In Idle transmit state DO+ is positive with respect to DO-.
08	PORTSEL	PORT SELECT allows the chip to select between the IEEE standard Attachment Unit Interface (AUI) using the internal SIA, or the general purpose serial interface (GPSI) to drive an independent device. When PORTSEL = 1, the general purpose port is selected. During the selection of the internal SIA, the GPSI outputs should be left unconnected and inputs should be tied to ground. Cleared by $\overline{\text{RESET}}$, unaffected by STOP.If Internal Loopback is enabled (MODE Register, bits 2 and 6) the state of PORTSEL is ignored, and the serial bit stream is looped back prior to the Manchester Encoder section of the ILACC.
07	RES	RESERVED, written and read as zero.
06	INTL	INTERNAL LOOPBACK (INTL = 1) selects Internal Loopback and 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. The condition of PORTSEL does not effect the operation of Internal Loopback and the data stream is looped back prior to the integral SIA. Since this represents full duplex operation, the packet size is limited to 41 bytes if DTCR = 0, or 45 bytes if DTCR = 1. Only packets which the node addresses to itself will be accepted. Packets addressed to the node from other nodes will not be accepted.
		EXTERNAL LOOPBACK (INTL = 0) permits the ILACC to transmit a packet out to the physical medium, either using the internal SIA, or via the general purpose serial interface (determined by PORTSEL, CSR15 bit 8). If PORTSEL = 0, the internal SIA will used, and the transmit/receive path will be tested to the physical medium (via external transceiver). If PORTSEL = 1, the general purpose interface will be exercised. Multicast addressing in External Loopback is valid only when DTCR = 1 (user must append 4 byte CRC to packet). Packets addressed to the node from other nodes will be accepted.
		INTL is only valid if LOOP = 1; otherwise it is ignored.
05	DRTY	DISABLE RETRY. When DRTY = 1, the chip 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 the TDTE. The RTRY error flag appears in TMD2.
04	COLL	FORCE COLLISION. This bit allows the collision logic to be tested. The chip must be in internal loopback for COLL to be valid. If COLL = 1, a collision willbe forced during the subsequent transmission attempt. This will result in 16 total transmission attempts with a retry error reported in TMD2.
03	DTCR	DISABLE TRANSMIT CRC. When DTCR = 0, the transmitter will generate and append a CRC to the transmitted packet, provided the NCRC bit in TMD1 is cleared. When DTCR = 1, the CRC logic is allocated to the receiver and no CRC is generated or sent with the transmitted packet, even if the NCRC bit is cleared.
02	LOOP	LOOPBACK allows the chip to operate in full duplex mode for test purposes. When LOOP = 1, loopback is enabled.
		The loopback packet size is user selectable (see description of INTL bit). The last 4 bytes will contain the CRC. If DTCR = 0, the 4 byte CRC generated and appended at transmit time will be stored in user memory after the data field. If DTCR = 1, a user-calculated CRC may be contained in the last 4 bytes of the transmit buffer. The receiver will generate a comparison to check against the transmitted CRC, which will be loaded into the last 4 bytes of the received buffer.
01	DTX	DISABLE THE TRANSMITTER causes the chip 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 chip to reject all incoming packets and not access the Receive Descriptor Ring. DRX = 1 will clear the RXON bit in CSR0 when initialization is complete.

INITIALIZATION BLOCK ADDRESS (CSR16-17)

A read only copy of the start address of the Initialization Block located in user memory, as programmed in CSR1 and CSR2.

CURRENT RX BUFFER ADDRESS (CSR18-19)

Contains the current receive buffer address the ILACC will use to dump an incoming packet. If the ILACC was stopped while receiving a packet, it will be the address of the incompleted buffer the ILACC was using. The address is 32 bits wide.

CURRENT TX BUFFER ADDRESS (CSR20-21)

Contains the address of the transmit buffer the ILACC transmitted last. In the event the ILACC was stopped while transmitting, it will contain the address of the incompleted buffer the ILACC was using. The address is 32 bits wide.

NEXT RX/TX BUFFER ADDRESS (CSR22-23)

Since the ILACC can only operate in half duplex, this register is shared depending on the condition of the ILACC at any time (i.e., transmit or receive). When the ILACC is transmitting, the register contains the address of the transmit buffer the ILACC will attempt to transmit next. When the ILACC is receiving, it will contain the address of the next receive buffer in which the ILACC will dump incoming packet data. In both cases, the ILACC will read the value during buffer lookahead while it is dealing with the current buffer. The address is 32 bits wide.

BASE ADDRESS OF RX RING (CSR24-25)

Contains the base address of the receive descriptor table entries in user memory. The value is a read only copy of the value obtained from the Initialization Block at initialization time. The address is 32 bits wide.

NEXT ADDRESS OF RX RING (CSR26-27)

Contains the address of the next receive descriptor that the ILACC will use. The ILACC will calculate this address based on the current descriptor address and the descriptor entry length. It will poll the next descriptor during lookahead to determine if a receive buffer is available and its location. The address is 32 bits wide.

CURRENT ADDRESS OF RX RING (CSR28-29)

Contains the address of the receive descriptor that the ILACC will use for the next incoming packet. If the ILACC was stopped during reception, it will contain the descriptor address of the incompleted message. The ILACC uses the address to examine the descriptor and determine if a receive buffer is available, locate it, and indicate the condition and length of the received data. The address is 32 bits wide.

BASE ADDRESS OF TX RING (CSR30-31)

Contains the base address of the transmit descriptor table entries in user memory. The value is a read only copy of the value obtained from the Initialization Block at initialization time. The address is 32 bits wide.

NEXT ADDRESS OF TX RING (CSR32-33)

Contains the address of the next transmit descriptor that the ILACC will use. The ILACC will calculate this address based on the current descriptor address and the descriptor entry length. It will poll the next descriptor during lookahead to determine if a transmit buffer is available and its location. The address is 32 bits wide.

CURRENT ADDRESS OF TX RING (CSR34-35)

Contains the address of the transmit descriptor that the ILACC will use for the next outgoing packet. If the ILACC was stopped during transmission it will contain the descriptor address of the incompleted message. The ILACC uses the address to examine the descriptor to determine if a transmit buffer is available, locate it, and indicate the condition of the transmitted data. The address is 32 bits wide.

RUNT BACKUP BUFFER ADDRESS (CSR36-37)

Contains a copy of the current receive buffer address. In the event that the receive packet is identified as a runt, the Runt Backup Buffer Address is written back into the Current Rx Buffer Address. The address is 32 bits wide.

RETRY BACKUP BUFFER ADDRESS (CSR38-39)

CURRENT RX BYTE COUNT (CSR40)

Contains a copy of the current receive descriptor byte count as read from the current receive descriptor.

CURRENT TX BYTE COUNT (CSR42)

Contains a copy of the current transmit descriptor byte count as read from the current transmit descriptor.

NEXT RX/TX BYTE COUNT (CSR44)

Since the ILACC can only operate in half duplex, this register is shared depending on the condition of the ILACC at any time (i.e., transmit or receive). During transmission, it contains the transmit buffer byte count as read from the next transmit descriptor entry during lookahead. During reception, it contains the buffer byte count as read from the next receive descriptor entry during lookahead.

POLL TIME COUNT (CSR46)

The Poll Time Count is a copy of the internal roll over counter which determines the frequency at which the ILACC will inspect the transmit descriptor ring entry.

HI ADDRESS INIT BLOCK (CSR48)

Contains a copy of the contents of CSR2 at initialization time.

CURRENT RX STATUS (CSR50)

Contains a copy of the current receive descriptor status byte as read from the current receive descriptor.

CURRENT TX STATUS (CSR52)

Contains a copy of the current transmit descriptor status byte as read from the current transmit descriptor.

NEXT RX/TX STATUS (CSR54)

Since the ILACC can only operate in half duplex, this register is shared depending on the condition of the ILACC at any time (i.e., transmit or receive). During transmission, it contains the transmit status byte as read from the next transmit descriptor entry during lookahead. During reception, it contains the status byte as read from the next receive descriptor entry during lookahead.

RUNT BACKUP BYTE COUNT (CSR56)

Contains a copy of the current receive buffer byte count. In the event that the receive packet is identified as a runt, the Runt Backup Byte Count is written back into the Current Rx Byte Count.

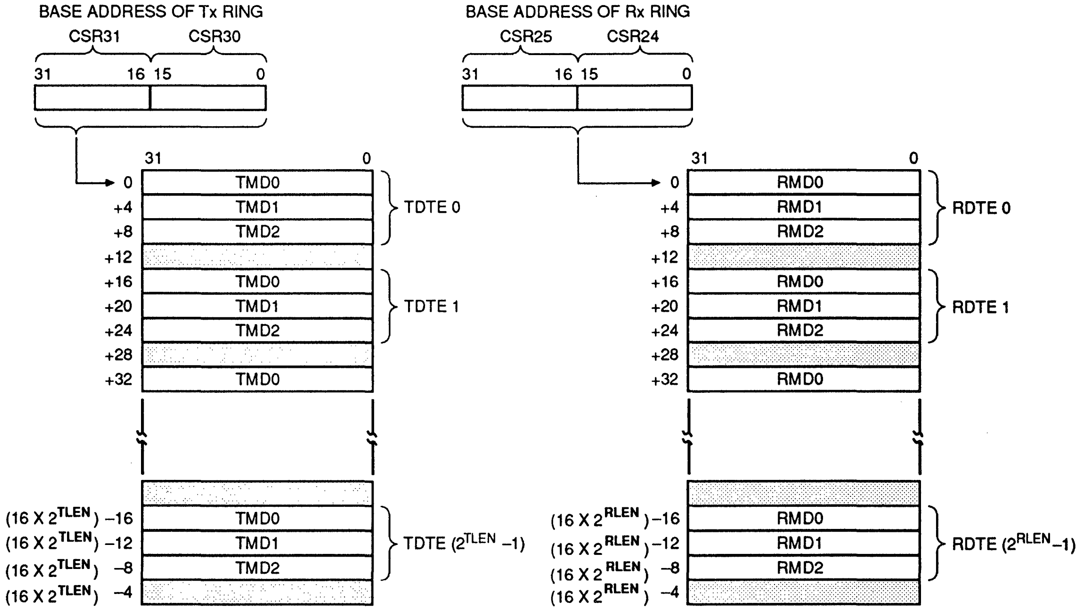
RETRY BACKUP BYTE COUNT (CSR58)

Contains a copy of the current transmit buffer byte count. In the event that the transmit packet suffers a collision, the Retry Backup Byte Count is written back into the Current Tx Byte Count.

REVISION REGISTER (CSR59)

Contains a value that identifies the silicon revision of the ILACC.

DESCRIPTOR TABLE CONTENT



10594-026A

INITIALIZATION

PROCEDURE

ILACC 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. On completion of the read operation and after internal registers have been updated, the IDON will be set in CSR0, and an interrupt generated if INEA is set.

The Initialization Block is vectored by the contents of CSR1 (least significant word of address) and CSR2 (most significant word of address). The block is resident in host memory, and contains the user defined conditions for ILAC operation together with the address and length information to allow linkage of the transmit and receive descriptor rings.

INITIALIZATION BLOCK

<table>
<tr><th></th><th colspan="4">31</th><th>0</th></tr>
<tr><td>IADR +24</td><td colspan="5">TDRA31–00</td></tr>
<tr><td>IADR +20</td><td colspan="5">RDRA31–00</td></tr>
<tr><td>IADR +16</td><td colspan="5">LADRF63–32</td></tr>
<tr><td>IADR +12</td><td colspan="5">LADRF31–00</td></tr>
<tr><td>IADR +8</td><td colspan="4">RESERVED</td><td>PADR47–32</td></tr>
<tr><td>IADR +4</td><td colspan="5">PADR31–00</td></tr>
<tr><td>IADR 0</td><td>TLEN</td><td>RES</td><td>RLEN</td><td>RES</td><td>MODE 15–00</td></tr>
</table>

MODE REGISTER

The Mode Register defines the transmit/receive operation of the ILACC. At initialization, this user-defined value is stored in CSR15.

LOGICAL ADDRESS FILTER (LADRF)

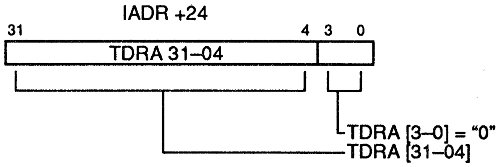
The filter value used for multicast addressing. Stored in CSR8-11 during initialization.

PHYSICAL ADDRESS REGISTER (PADR)

The individual node address assigned to the ILACC and stored in CSR12-14 during initialization.

TRANSMIT DESCRIPTOR RING ADDRESS (TDRA)

The base address (lowest address) of the user area where the transmit descriptor ring is located. This is a full 32-bit address and is stored in the ILACC during initialization.



10594-029A

The least significant 4 bits of the ring base address must be zero.

TRANSMIT DESCRIPTOR RING LENGTH (TLEN)

TLEN defines the number of TDTEs which will be used in the ring. TLEN is located at the base location of the Initialization Block (IADR + 0) with the Mode Register and the RLEN entry. A maximum of 512 transmit descriptor entries is permitted.

TLEN has a four bit field. The user is free to write any 4-bit code into this field. Binary values greater than 9 (e.g., TLEN = 1111b) will be stored as written, and the ILACC will expect 512 TDTEs.

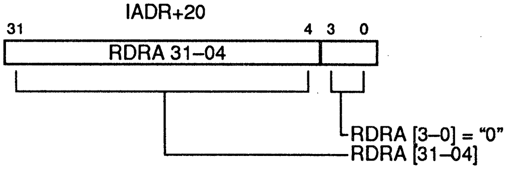
TLEN is expressed as a power of two, as follows:

TLEN				No. of TDTEs
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	512

Note that the field is stored in CSR6 during initialization. For details, refer to the definition within the Description of User Accessible Resources.

RECEIVE DESCRIPTOR RING ADDRESS (RDRA)

The base address of the user area where the receive descriptor ring is located. This is a full 32-bit address and is stored in the ILACC during initialization.



10594-033A

The least significant four bits of the ring base address must be zero.

RECEIVE DESCRIPTOR RING LENGTH (RLEN)

RLEN defines the number of RDTEs which will be used in the ring. RLEN is located at the base location of the Initialization Block (IADR+0) with the MODE Register and the TLEN entry. Maximum of 512 transmit descriptor entries are permitted.

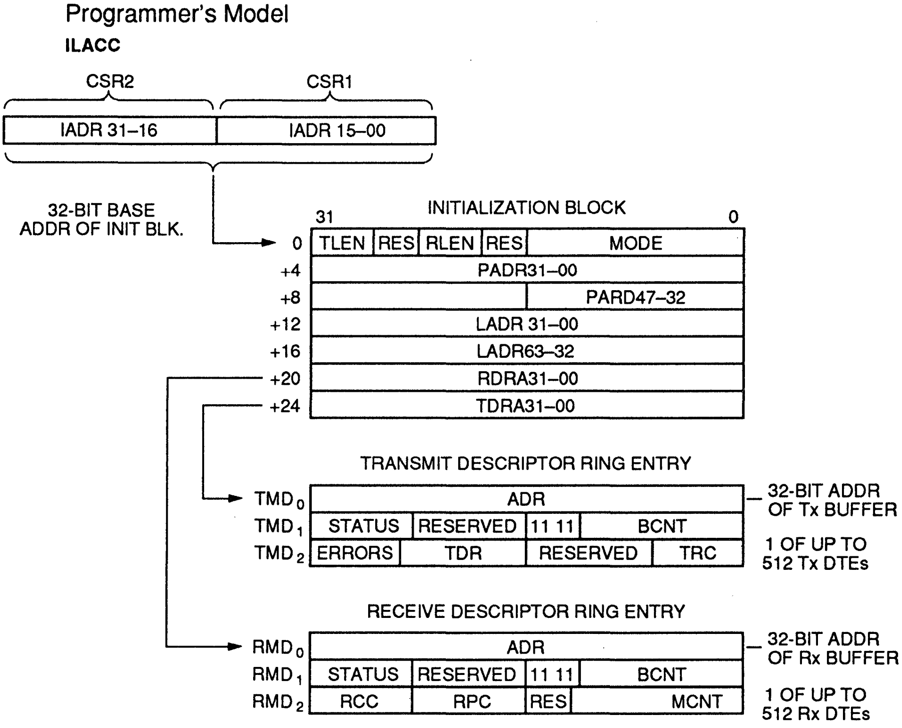
RLEN is expressed as a power of two as follows:

RLEN				No. of RDTEs
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	512

RLEN has a 4-bit field. The user is free to write any 4-bit code into this field. Binary values greater than 9 (i.e., RLEN = 1111b) will be stored as written, and the ILACC will expect 512 RDTEs.

Note that the field is stored in CSR6 during initialization. For details, refer to the definition within the Description of User Programmable Registers.

INITIALIZATION BLOCK LAYOUT



10594-036A

Reinitialization

The transmitter and receiver section of the ILACC are turned on via the initialization block (MODE Register: DRX, DTX bits). The state of the transmitter and receiver can be monitored through CSR0 (RXON, TXON bits). The ILACC must be reinitialized if the transmitter and/or the receiver were not turned on during the original initialization, and it is subsequently required to activate them. Alternatively, the ILACC may require reinitialization if either section shuts off due to the detection of an error condition (MERR, UFLO, TX BUFF error). Care must be taken when the ILACC is reinitialized.

Prior to reinitialization of the ILACC, the user must set the STOP bit in CSR0.The user should also reinitialize the descriptor pointers in the software and reinitialize all descriptor OWN bits in memory prior to re-enabling the transmit and receive functions. This is necessary since the ILACC's transmit and receive descriptor pointers are reloaded with their respective base addresses upon initialization. The ILACC can then be reinitialized by setting the INIT bit in CSR0.

Buffer Management

Buffer management is accomplished through message descriptors organized as ring structures in memory. There are two rings; a receive ring and a transmit ring. Each message descriptor entry requires three double words (6 words or 12 bytes).

To simplify the maintenance of pointers for the rings, the space allocated for the transmit/receive descriptor table entries is as follows:

Descriptor Table Entries are 8 words long, located on 16-byte boundaries (bits 0-3 of pointer address must be zero).

Descriptor Rings

Each descriptor ring must be organized in a contiguous area of memory. At initialization time, the ILACC reads the user-defined base address for the transmit and receive descriptor rings as well as the number of entries contained. Maximum of 512 ring entries is permitted.

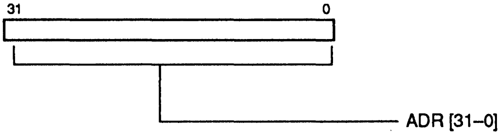
Each ring entry contains the following information:

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 ILACC or the host. The OWN bit within the descriptor status information 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 Memory Allocation

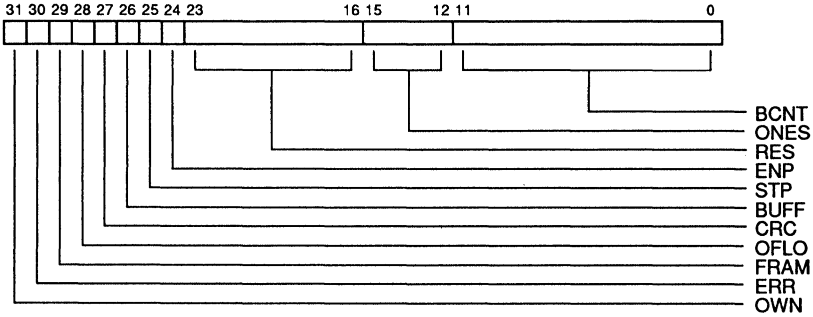
RECEIVE MESSAGE DESCRIPTOR 0 (RMD0).



10594-048B

Bit	Name	Description
31-00	ADR	Address of the buffer pointed to by this descriptor. ADR is written by the host and unchanged by the ILACC. The buffer can be located on an arbitrary byte boundary.

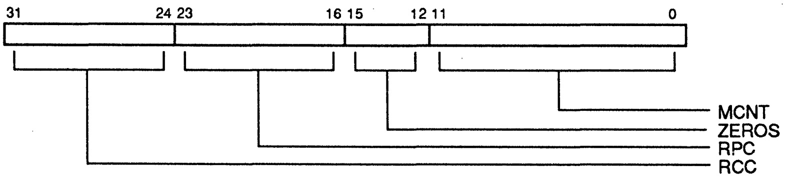
RECEIVE MESSAGE DESCRIPTOR 1 (RMD1).



10594-049A

Bit	Name	Description
31	OWN	This bit indicates that the descriptor entry is owned by the host (OWN=0) or by ILACC (OWN=1). The ILACC 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 ILACC or host has relinquished ownership of a buffer, it must not change any field in the three words that comprise the descriptor entry.
30	ERR	ERROR summary is the "OR" of FRAM, OFLO, CRC or BUFF. ERR is set by the ILACC and cleared by the host.
29	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. FRAM is set by the ILACC and cleared by the host.
28	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 FIFO overflowed. OFLO is valid only when ENP is not set. OFLO is set by the ILACC and cleared by the host.
27	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. CRC is set by the ILACC and cleared by the host.
26	BUFF	BUFFER ERROR is set any time the ILACC 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. 2. FIFO overflow occurred before the ILACC received the next STATUS. 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 ILACC and cleared by the host.
25	STP	START OF PACKET indicates that this is the first buffer used by the ILACC for this packet. It is used for data chaining buffers. STP is set by the ILACC and cleared by the host.
24	ENP	END OF PACKET indicates that this is the last buffer used by the ILACC 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 ILACC and cleared by the host.
23-16	RES	RESERVED.
15-12	ONES	MUST BE ONES. This field is written by the host and unchanged by the ILACC.
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 ILACC. Minimum buffer size is 64 bytes.

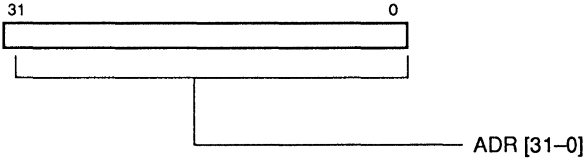
RECEIVE MESSAGE DESCRIPTOR 2 (RMD2)



10594-050A

Bit	Name	Description
31-24	RCC	RECEIVE COLLISION COUNT indicates the number of collisions on the network since the last successfully received packet. The internal value of RCC in CSR7 is written to RMD2 (in the last descriptor of the chain) prior to the OWN bit being flipped (i.e., after a successful reception). The collision count in CSR7 is reset immediately after the OWN bit for the descriptor is reset or upon a host read. The count does not roll over when more than 255 collisions are detected but will freeze at the maximum count.
23-16	RPC	RUNT PACKET COUNT indicates the number of runt packets addressed to this node since the last successfully received packet. The internal value of RPC in CSR7 is written to RMD2 (in the last descriptor of the chain) prior to the OWN bit being flipped (i.e., after a successful reception). The runt packet counter in CSR7 is reset immediately after the OWN bit for the descriptor is reset or upon a host read. The count will be frozen at 255 if runt packets in excess of this are detected.
15-12	ZEROS	ZEROS. This field is written by the ILACC.
11-00	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 chip and cleared by the host.

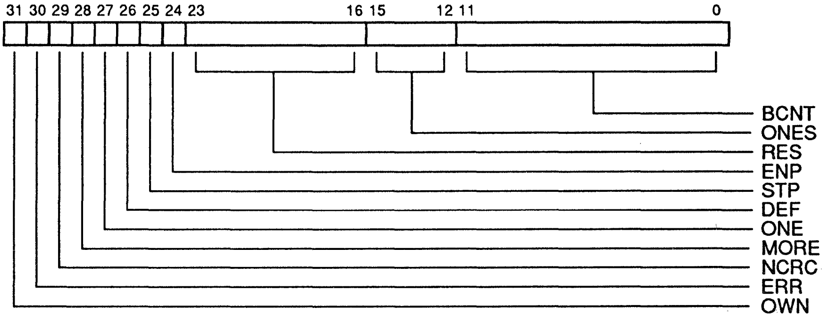
TRANSMIT MESSAGE DESCRIPTOR 0 (TMD0).



10594-051B

Bit	Name	Description
31-00	ADR	The 32-bit address of the buffer pointed to by this descriptor. ADR is written by the host and unchanged by the ILACC. The buffer can be located on an arbitrary byte boundary.

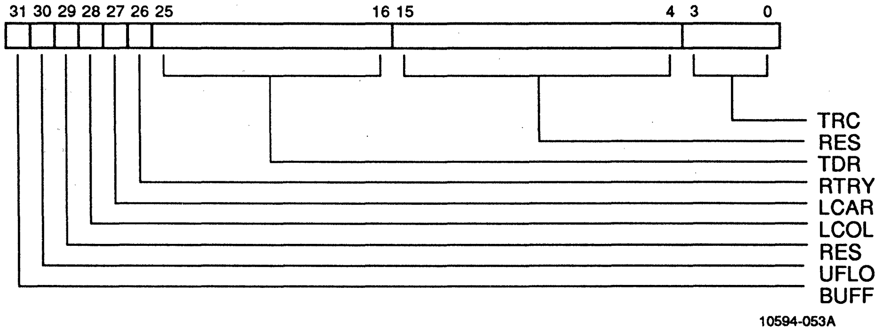
TRANSMIT MESSAGE DESCRIPTOR 1 (TMD1).



10594-052A

Bit	Name	Description
31	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = 0) or by the ILACC (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by this descriptor. The ILACC clears the OWN bit after transmitting the contents of the buffer. Both the host and the ILACC must not alter a descriptor entry after relinquishing ownership.
30	ERR	ERROR summary is the "OR" of UFLO, LCOL, LCAR, or RTRY. ERR is set by the ILACC and cleared by the host.
29	NCRC	NO CRC dynamically controls the generation of CRC on a packet by packet basis as long as the DTCR bit in CSR15 = 0. It is valid in the last descriptor of the packet to be transmitted. When NCRC=1, CRC generation is inhibited. When NCRC=0, CRC generation is activated. NCRC is set by the host and unchanged by the ILACC.
28	MORE	MORE indicates that more than one retry was needed to transmit a packet. MORE is set by the ILACC and cleared by the host.
27	ONE	ONE indicates that exactly one retry was needed to transmit a packet. The ONE flag is not valid when LCOL is set. ONE is set by the ILACC and cleared by the host.
26	DEF	DEFERRED indicates that the ILACC had to defer while trying to transmit a packet. This condition occurs if the channel is busy when the ILACC is ready to transmit. DEF is set by the ILACC and cleared by the host.
25	STP	START OF PACKET indicates that this is the first buffer to be used by the ILACC for this packet. It is used for data chaining buffers. STP is set by the host and unchanged by the ILACC. The STP bit must be set in the first buffer of the packet or the ILACC will skip over this 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 ILACC.
24	ENP	END OF PACKET indicates that this is the last buffer to be used by the ILACC 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 unchanged by the ILACC.
23-16	RES	RESERVED
15-12	ONES	MUST BE ONES. Written by host.
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 buffer byte length. This is the number of bytes from this buffer that will be transmitted by the ILACC. This field is written by the host and unchanged by the ILACC. The first buffer of a packet must be a minimum of 116 bytes (DMAPLUS = 1) or 100 bytes (DMAPLUS = 0) when data chaining, and 64 bytes (DTCR=1) or 60 bytes (DTCR=0) when not data chaining.

TRANSMIT MESSAGE DESCRIPTOR 2 (TMD2).



10594-053A

Bit	Name	Description
31	BUFF	BUFFER ERROR is set by the ILACC during transmission when the ILACC does not find the ENP flag in the current buffer and does not own the next buffer. This can occur if either: 1. The OWN bit of the next buffer is zero. 2. FIFO underflow occurred before the ILACC was able to read the next STATUS byte. BUFF is set by the ILACC 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 transmit data chaining. BUFF is set by the ILACC 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 FIFO has emptied before the end of the packet was reached. Upon UFLO error, the transmitter is turned off (CSR0, TXON =O). UFLO is set by the ILACC and cleared by the host.
29	RES	RESERVED bit. The ILACC will write this bit with a "0."
28	LCOL	LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The ILACC does not retry on late collisions. LCOL is set by the ILACC and cleared by the host.
27	LCAR	LOSS OF CARRIER is set when the carrier is lost during an ILACC-initiated transmission. The loss is detected internally if the ILACC's integral SIA isbeing used or via the CRS line becoming deasserted if the general purpose serial interface is utilized. The ILACC 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. LCAR is set by the ILACC and cleared by the host.
26	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. RTRY is set by the ILACC and cleared by the host.
25-16	TDR	TIME DOMAIN REFLECTOMETRY reflects the state of an internal ILACC counter that counts from the start of a transmission to the occurence 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 ILACC and is valid only if the RTRY bit is set. The TDR counter is incremented by the 20 MHz clock at XTAL1 (for PORTSEL=0) or the 10MHz clock at TxC (for PORTSEL=1).
15-04	RES	RESERVED
03-00	TRC	TRANSMIT RETRY COUNT indicates the number of transmit retries of the associated packet. The maximum count is 15. Written by the ILACC into the last transmit descriptor for the message (ENP = 1), and valid only when OWN = 0.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature:	-65 to +150°C
Supply Voltage (AV_{DD}, V_{DD}) referenced to AV_{SS} or V_{SS}:	-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 absolutely maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A):	0 to 70°C
Supply Voltages (AV_{DD}, V_{DD}):	5 V ±5%

DC CHARACTERISTICS
CAPACITANCE

Parameter Symbol	Parameter Description	Typ.	Unit
C_{IN}	Input Pins	10	pF
C_{IO}	Bidirectional Pins	20	pF
C_{OUT}	Output Pins	10	pF

DC CHARACTERISTICS

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}	X_{TAL1} Input LOW Voltage (External Clock Signal)	V_{SS} = 0.0 V	-0.5	0.8	V
V_{IHX}	X_{TAL1} Input HIGH Voltage (External Clock Signal)	V_{SS} = 0.0 V	3.5	V_{DD}+0.5	V
V_{OL}	Output LOW voltage	I_{OL1} = 20 mA, I_{OL2} = 6 mA I_{OL3} = 4 mA (Note 1)		0.4	V
V_{OH}	Output HIGH voltage	I_{OH} = -0.4 mA (Note 2)	2.4		V
I_{IX}	Input leakage current	0 V < V_{IN} < V_{DD} (Note 3)	-10	10	µA
I_{ILX}	X_{TAL1} Input LOW Current	V_{IN} = V_{SS}	-10		µA
I_{IHX}	X_{TAL1} Input HIGH Current	V_{IN} = V_{DD}		10	µA
I_{IAXD}	DI± and CI± Input Current	$AV_{SS} \leq V_{IN} \leq AV_{DD}$	-500	500	µA
I_{OZL}	Output Leakage Current	V_{OUT} = 0 V (Note 4)	-10		µA
I_{OZH}	Output Leakage Current	V_{OUT} = V_{DD} (Note 4)		10	µA
V_{AOD}	AUI Differential Output Voltage |(DO+) - (DO-)|	R_L = 78 Ω	630	1100	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	-512	512	µA
V_{AOCM}	DO± Common Mode Output Voltage	R_L = 78 Ω	AV_{DD}-3.0	AV_{DD}-1.0	V
V_{AODO}	DO± Differential Output Voltage Imbalance	R_L = 78 Ω |DO+| - |DO-|	-25	25	mV
V_{ATH}	DI± Differential Input Switching Threshold		-35	35	mV
V_{ASQ}	DI±, CI± Differential Input Squelch Threshold		-275	-175	mV
V_{AIDV}	DI± and CI±, Differential Mode Input Voltage Range		-2.5	2.5	V
V_{AICM}	DI± and CI± Input Common Mode Open Circuit Voltage (Bias)	I_{IN} = 0 mA	AV_{DD}-3.0	AV_{DD}-1.0	V

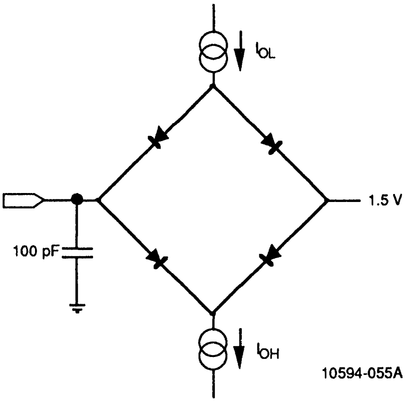
DC CHARACTERISTICS (Continued)

Parameter	Description	Test Conditions	Min.	Max.	Unit
V_{AODP}	DO± Undershoot Voltage at zero differential on transmit return to zero (end of message)	AUI load = 27 µH ± 1% and 73 Ω or 83 Ω ± 1% (See IEEE 802.3)		-100	mV
I_{DD}	Power Supply Current	f_{XTAL} = 20 MHz f_{BCLK} = 16.67 MHz		150	mA

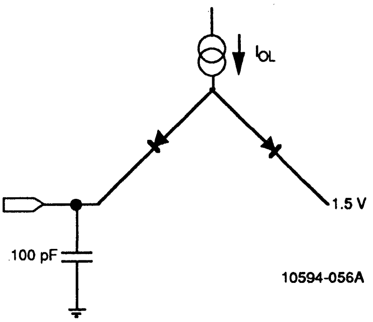
Notes:

1. I_{OL1} = 20 mA: READYL
   I_{OL2} = 6 mA: BE3/DAL[1], BE2/DAL[0], R/W, ALE/AS, DAS, BGACK/MASTER, HOLD/BUSREQ, BE0/SIZ0, BE1/SIZ1
   I_{OL3} = 4 mA: DAL[31:2], DALI, DALO, INTR, RINTR, XCLK, TxD, RTS.
2. V_{OH} does not apply to open-drain output pins.
3. I_{IX} applies to pins: HLDA/BUSACK, C/D, CS, BCLK, and RESET.
4. I_{OZH} and I_{OZL}apply to pins: DAS, DAL0, DALI, BGACK/MASTER, HOLD/BUSREQ, TXC, RXC, RXD.TXD, RTS, XCLK, BE0-1/SIZ0-1, BE2/DAL0, BE3/DAL1, DAL2-31, R/W, RINTR, INTR, READYL, ALE/AS.

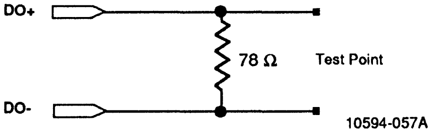
SWITCHING TEST LOADS



Normal and Three-State Outputs



Open Drain Outputs



AUI DO Switching Test Circuit

AC TIMING PARAMETERS
CLOCK SIGNALS

Clock and Reset

#	Parameter	Description	Test Conditions	Min.	Max.	Unit
1	t_{BCLK}	BCLK period:		60	250	ns
2	t_{BCLKL}	BCLK LOW pulse width		0.47*t_{BCLK}	0.53*t_{BCLK}	ns
3	t_{BCLKH}	BCLK HIGH pulse width		0.47*t_{BCLK}	0.53*t_{BCLK}	ns
4	t_{BCLKR}	BCLK Rise Time	(Note 3) (Note 11)		3	ns
5	t_{BCLKF}	BCLK Fall Time	(Note 3) (Note 11)		3	ns
6	t_{RST}	$\overline{\text{RESET}}$ Pulse Width		9*t_{BCLK}		ns

Internal SIA Configuration Clock Parameters (driven by external clock source)

#	Parameter	Description	Test Conditions	Min.	Max.	Unit
9	t_{X1}	XTAL1 period	(Note 2)	49.995	50.005	ns
11	t_{X1H}	XTAL1 HIGH pulse width		20		ns
12	t_{X1L}	XTAL1 LOW pulse width		20		ns
13	t_{X1R}	XTAL1 Rise Time	(Note 3)		5	ns
14	t_{X1F}	XTAL1 Fall Time	(Note 3)		5	ns

General Purpose Serial Interface (GPSI) Configuration Clock Parameters

#	Parameter	Description	Test Conditions	Min.	Max.	Unit
17	t_{TxC}	TxC period		99	101	ns
18	t_{TxCL}	TxC low pulse width		45		ns
19	t_{TxCH}	TxC high pulse width		45		ns
20	t_{TxCR}	TxC rise time	(Note 3)		5	ns
21	t_{TxCF}	TxC fall time	(Note 3)		5	ns
22	t_{RxC}	RxC period		85	117	ns
23	t_{RxCH}	RxC HIGH pulse width		38		ns
24	t_{RxCL}	RxC LOW pulse width		38		ns
25	t_{RxCR}	RxC rise time	(Note 3)		5	ns
26	t_{RxCF}	RxC fall time	(Note 3)		5	ns

Bus Slave

#	Parameter	Description	Test Conditions	Min.	Max.	Unit
30	t_{30}	$\overline{\text{DAS}}$ LOW and $\overline{\text{CS}}$↓ setup to BCLK↑		10		ns
31	t_{31}	Data in valid hold time from $\overline{\text{READYL}}$ ↓		0		ns
32	t_{32}	R/$\overline{\text{W}}$ setup time to $\overline{\text{DAS}}$↓		5		ns
33	t_{33}	$\overline{\text{CS}}$ hold time from $\overline{\text{DAS}}$↑	(Note 12)	0		ns
34	t_{34}	C/$\overline{\text{D}}$ setup time to $\overline{\text{DAS}}$↓		5		ns
35	t_{35}	C/$\overline{\text{D}}$ hold time from $\overline{\text{DAS}}$↓		15		ns
36	t_{36}	R/$\overline{\text{W}}$ hold time from $\overline{\text{DAS}}$↓		15		ns

AC TIMING PARAMETERS

Bus Slavė (Continued)

#	Parameter	Description	Test Conditions	Min.	Max.	Unit
37	t_{37}	$\overline{\text{DAS}}$↑ delay to $\overline{\text{DALO}}$↑			45	ns
38	t_{38}	$\overline{\text{DAS}}$↓ delay to $\overline{\text{DALO}}$↓			45	ns
39	t_{39}	$\overline{\text{DAS}}$↑ delay to $\overline{\text{DALI}}$↑			40	ns
40	t_{40}	$\overline{\text{DAS}}$↓ delay to $\overline{\text{DALI}}$↓			45	ns
41	t_{41}	$\overline{\text{DAS}}$↑ delay to $\overline{\text{READYL}}$↑ (de-assertion)			35	ns
42	t_{42}	$\overline{\text{DAS}}$↓ delay to $\overline{\text{READYL}}$↓	(Note 1)		$5^{*}t_{BCLK}$	ns
43	t_{43}	$\overline{\text{DAS}}$↓ with $\overline{\text{CS}}$ LOW delay to Data In valid			t_{BCLK}-20	ns
44	t_{44}	Data Out valid setup to $\overline{\text{READYL}}$↓		15		ns
45	t_{45}	Data In valid hold time from $\overline{\text{DAS}}$↑		0		ns
46	t_{46}	Data Out valid hold time from $\overline{\text{DAS}}$↑		16		ns
47	t_{47}	BCLK↑ delay to $\overline{\text{READYL}}$↓		5	30	ns
48	t_{48}	$\overline{\text{DAS}}$↑ hold time from $\overline{\text{READYL}}$↓	(Note 6)	0		ns
49	t_{49}	$\overline{\text{CS}}$ LOW AND $\overline{\text{DAS}}$↓ setup to BCLK↑		10		ns
50	t_{50}	BCLK↑ delay to Data In Valid		t_{BLCK}-20		ns
51	t_{51}	BCLK↑ delay to Data Out Valid			60	ns
52	t_{52}	Data In valid hold time from BCLK↑		20		ns
53	t_{53}	$\overline{\text{DAS}}$↑ to Data Out High Impedance			40	ns

Bus Acquisition, Relinquish, and Preemption

#	Parameter	Description	Test Conditions	Min.	Max.	Unit
54	t_{54}	HLDA / $\overline{\text{BUSACK}}$↓ setup to BCLK↓	(Note 9)	5		ns
55	t_{55}	BCLK↓ after $\overline{\text{BUSACK}}$ LOW delay to $\overline{\text{BUSREQ}}$↑ (de-assertion)			t_{BCLK}+35	ns
56	t_{56}	HLDA/$\overline{\text{BUSACK}}$ LOW hold time from BCLK↓	(Note 9)	15		ns
57	t_{57}	$\overline{\text{BGACK}}$↑ setup to BCLK↓	(Note 10)	5		ns
58	t_{58}	$\overline{\text{BUSACK}}$ LOW Pulse Width	(Note 9)	t_{BCLK}+30		ns
59	t_{59}	BCLK↓ delay to $\overline{\text{HOLD}}$↑			35	ns
60	t_{60}	BCLK↑delay to $\overline{\text{HOLD/BUSREQ}}$↓			40	ns
61	t_{61}	$\overline{\text{HLDA}}$↑ setup to BCLK↓ (preemption)	(Note 10)	15		ns
62	t_{62}	BCLK↓ delay to $\overline{\text{BGACK}}$↑			35	ns
63	t_{63}	BCLK↓ (S1) delay to $\overline{\text{HOLD}}$↑ (preemption)	(Note 8)		$8.5^{*}t_{BCLK}$ + 40	ns
64	t_{64}	$\overline{\text{BUSACK}}$ LOW hold time from $\overline{\text{BUSREQ}}$↑	(Note 6) (Note 9)	0		ns
65	t_{65}	BCLK↑ delay to $\overline{\text{BGACK}}$↓			40	ns

Bus Acquisition, Relinquish, and Preemption (Continued)

#	Parameter	Description	Test Conditions	Min.	Max.	Unit
66	t_{66}	$\overline{\text{HOLD}}$↑ or $\overline{\text{HLDA}}$↑ delay to $\overline{\text{MASTER}}$↑			30	ns
67	t_{67}	$\overline{\text{HOLD}}$ LOW and $\overline{\text{HLDA}}$↓ delay to $\overline{\text{MASTER}}$↓			40	ns
68	t_{68}	BCLK↑ delay to memory control, DAL[31:0] outputs high impedance (Bus Relinquish)			40	ns
72	t_{72}	BCLK↓ delay to ALE, $\overline{\text{AS}}$, $\overline{\text{DAS}}$, $\overline{\text{BE0}}$, $\overline{\text{BE1}}$, SIZ0, SIZ1 driven		0		ns

AC TIMING PARAMETERS

Bus Master

#	Parameter	Description	Test Conditions	Min.	Max.	Unit
73	t_{73}	BCLK↑ delay to ALE/$\overline{\text{AS}}$ valid			30	ns
74	t_{74}	BCLK↑ delay to address valid	(Note 6)	15	42	ns
75	t_{75}	$\overline{\text{DAS}}$↑ delay to address driven		t_{BCLKL}-5		ns
76	t_{76}	BCLK↑ delay to R/$\overline{\text{W}}$ Valid			30	ns
77	t_{77}	Data in valid setup to $\overline{\text{DAS}}$↑	(Note 4)	35		ns
78	t_{78}	Data in valid hold from $\overline{\text{DAS}}$↑	(Note 4)	0		ns
79	t_{79}	BCLK↓ to DAL[31:0] (Address) high impedance			30	ns
80	t_{80}	Address out valid hold from ALE↓ or $\overline{\text{AS}}$↑		t_{BCLKH}		ns
81	t_{81}	BCLK↓ delay to $\overline{\text{DAS}}$	(Note 6)	0	35	ns
83	t_{83}	BCLK↓ delay to $\overline{\text{DAS}}$↓	(Note 6)	0	35	ns
84	t_{84}	$\overline{\text{READYL}}$ valid setup to BCLK↑	(Note 5)	3		ns
85	t_{85}	$\overline{\text{DAS}}$↓ delay to $\overline{\text{READYL}}$↓	(0 wait states) (Note 5) (Note 7)		1.5*t_{BCLK} -35	ns
86	t_{86}	$\overline{\text{READYL}}$ valid hold from BCLK↑	(Note 5)	5		ns
87	t_{87}	BCLK↓ delay to $\overline{\text{DALO}}$↓	(Note 6)	0	30	ns
88	t_{88}	$\overline{\text{READYL}}$ pulse width	(Note 5)	2*t_{BCLK}		ns
89	t_{89}	BCLK↑ delay to $\overline{\text{DALI}}$↓	(Note 6)	0	35	ns
90	t_{90}	BCLK↓ delay to Data Out valid			40	ns
91	t_{91}	Data Out valid hold from $\overline{\text{DAS}}$↑		t_{BCLKL}-20		ns
92	t_{92}	DALO↑ hold time from $\overline{\text{DAS}}$↑		t_{BCLKL}-5		ns
93	t_{93}	BCLK↑ delay to $\overline{\text{DALO}}$↑	(Note 6)	0	40	ns
94	t_{94}	BCLK↓ delay to $\overline{\text{DALI}}$↑	(Note 6)	0	35	ns
95	t_{95}	Address Out valid setup to ALE↓ or $\overline{\text{AS}}$↑		22		ns
96	t_{96}	Data Out valid setup to $\overline{\text{DAS}}$↑	(Note 6)	2*t_{BCLK} -40		ns
97	t_{97}	Data In valid setup to BCLK↓	(Note 4)	0		ns
98	t_{98}	Data In valid hold after BCLK↓	(Note 4)	28		ns

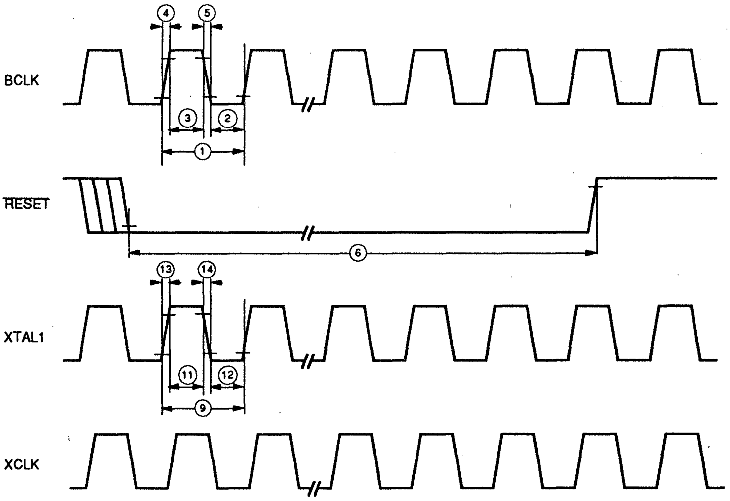
AC TIMING PARAMETERS

Serial Timing

#	Parameter	Description	Test Conditions	Min.	Max.	Unit
104	t_{DOTR}	DO± Rise Time (10% to 90%)		2.5	5.0	ns
105	t_{DOTF}	DO± Fall Time (90% to 10%)		2.5	5.0	ns
106	t_{DORM}	DO± Rise and Fall Time mismatch			1.0	ns
107	t_{DOETD}	DO± End of Transmission		200	375	ns
108	t_{PWRDI}	DI± pulse width to reject	input<VASQ		15	ns
109	t_{PWODI}	DI± pulse width to turn on internal DI carrier sense	input<VASQ	45		ns
110	t_{PWMDI}	DI± pulse width to maintain internal DI carrier sense on	input<VASQ	45	136	ns
111	t_{PWKDI}	DI± pulse width to turn internal DI carrier sense off	input<VASQ	165		ns
112	t_{PWRCI}	CI± pulse width to reject	input<VASQ		18	ns
112	t_{PWOCI}	CI± pulse width to turn on internal SQE sense	input<VASQ	26		ns
114	t_{PWMCI}	CI± pulse width to maintain internal SQE sense on	input<VASQ	26	90	ns
115	t_{PWKCI}	CI± pulse width to turn internal SQE sense off	input<VASQ	160		ns
116	t_{PTxCR}	TxC↑ delay to RTS↑			70	ns
117	t_{RTSHT}	RTS hold time from TxC↑		5		ns
118	t_{PTxCT}	TxC↑ delay to TxD change			70	ns
119	t_{TxDHT}	TxD hold time from TxC↑		5		ns
120	t_{RxDR}	RxD rise time	(Note 3)		8	ns
121	t_{RxDF}	RxD fall time	(Note 3)		8	ns
122	t_{RxDHT}	RxD hold time (RxC↑ to RxD change)		12		ns
123	t_{RxDST}	RxD setup time (RxD stable to RxC↑)		20		ns
124	t_{CRSL}	CRS low time	(Note 6)	t_{TxC}+20		ns
125	t_{CDTH}	CDT high time	(Note 6)	t_{TxC}+30		ns
126	t_{PCDTR}	CDT↑ delay to RTS de-asserted	(Note 12)	$32*t_{TxC}$	$99*t_{TxC}$	ns
127	t_{PRxCC}	CRS hold time from RXC↑	(Note 6)	0		ns

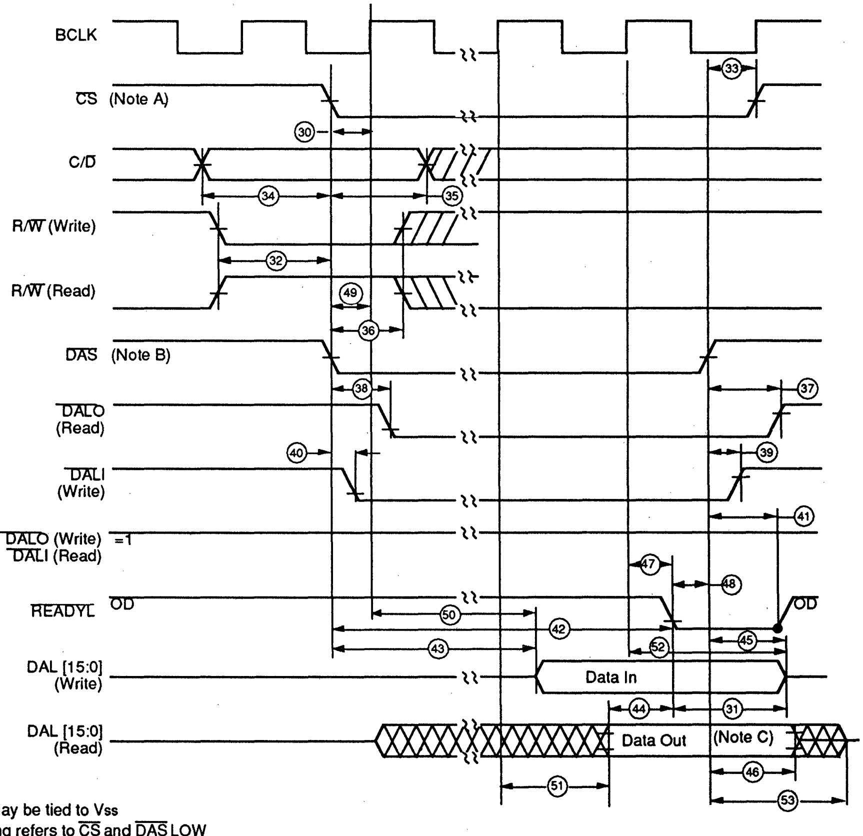
Notes:

1. CSR0, 3 ,4, and the RAP register can be accessed within 5 BCLK periods. All other CSRs take 15 BCLK periods max.
2. IEEE 802.3 requirement.
3. Not tested.
4. Asynchronous parameters 77 and 78 must be met or synchronous parameters 97 and 98 must be met.
5. Asynchronous parameters 85 and 88 must be met or synchronous parameters 84 and 86 must be met.
6. Parameter minimum guaranteed by design—not tested.
7. Parameter shown for 0 wait states. For n wait states, max delay is $(1.5+n)*t_{BCLK}-35$ (ns)
8. Shown for 0 wait states. For n wait states, max delay is $(8.5+2n)*t_{BCLK}+40$ (ns)
9. Asynchronous parameters 58 and 64 must be met or synchronous parameters 54 and 56 must be met.
10. This parameter needs to be met only to produce guaranteed synchronous timing.
11. For BCLK frequencies below 10MHz, the rise and fall times max value is 10 ns.
12. Guaranteed by design—not tested.



10594-058A

BCLK and RESET Timing



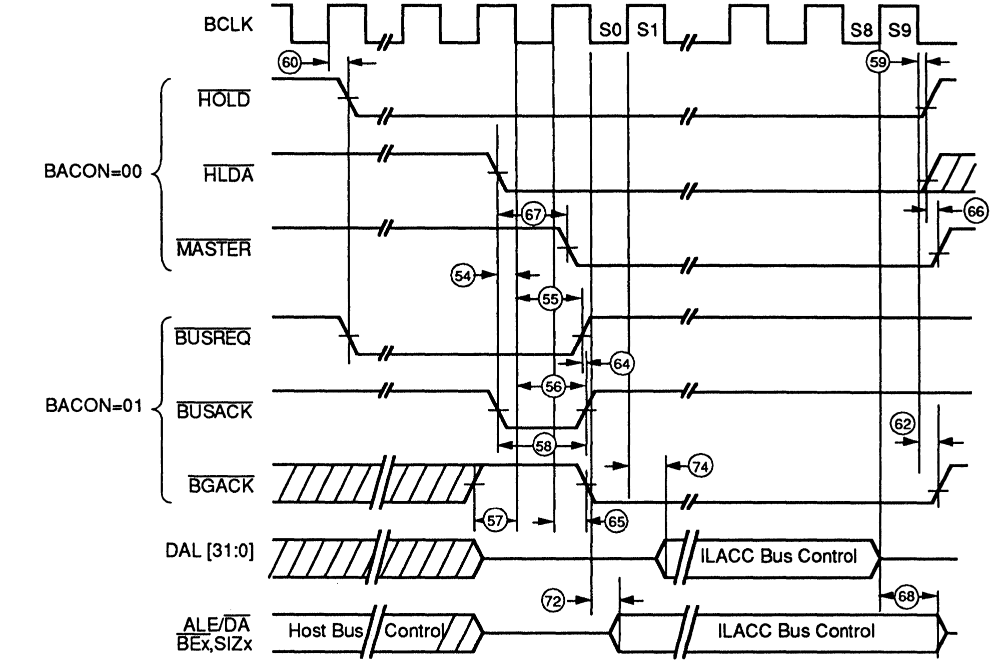
Notes:

A. CS May be tied to Vss
B. Timing refers to CS and DAS LOW
C. DAL[31:16] are active but undefined during 16-bit slave transfers

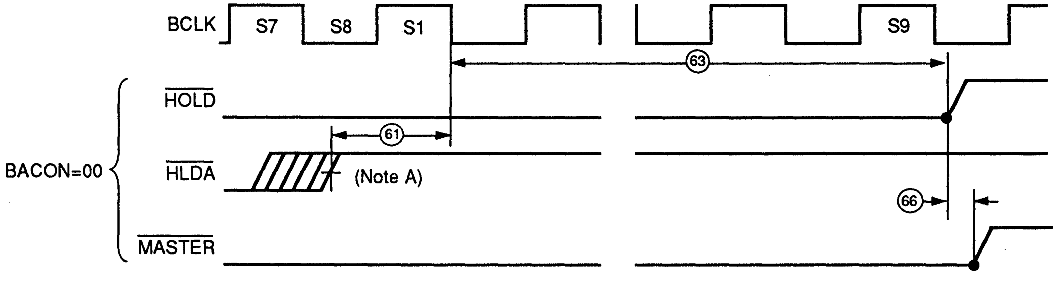
10594-059A

Bus Slave Timing (BACON = 00 or 01)
32-bit 80X86 and 680X0

BUS INTERFACE TIMING DIAGRAMS



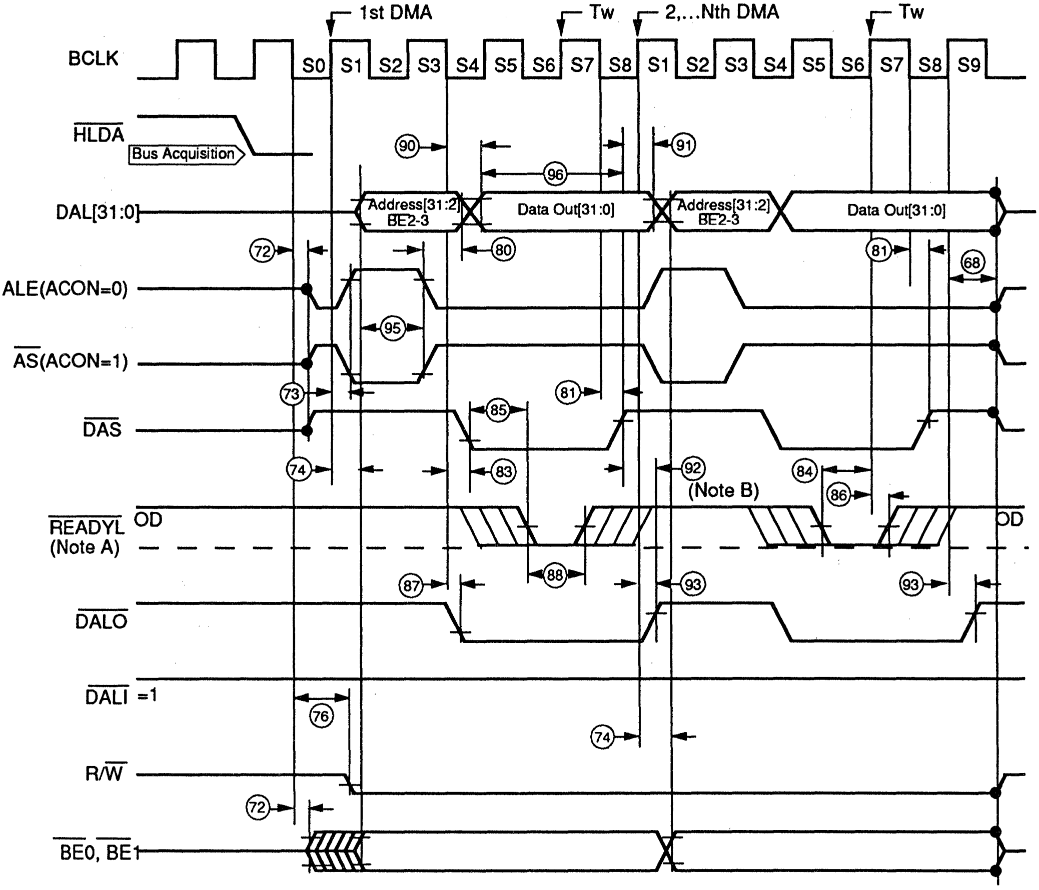
Bus Acquisiton and Relinquish Timing



Note:
A. HLDA is sampled on the falling edge of the S1 state of BCLK

Bus Pre-emption Timing (80X86 Mode Only)

BUS INTERFACE TIMING DIAGRAMS



Notes:

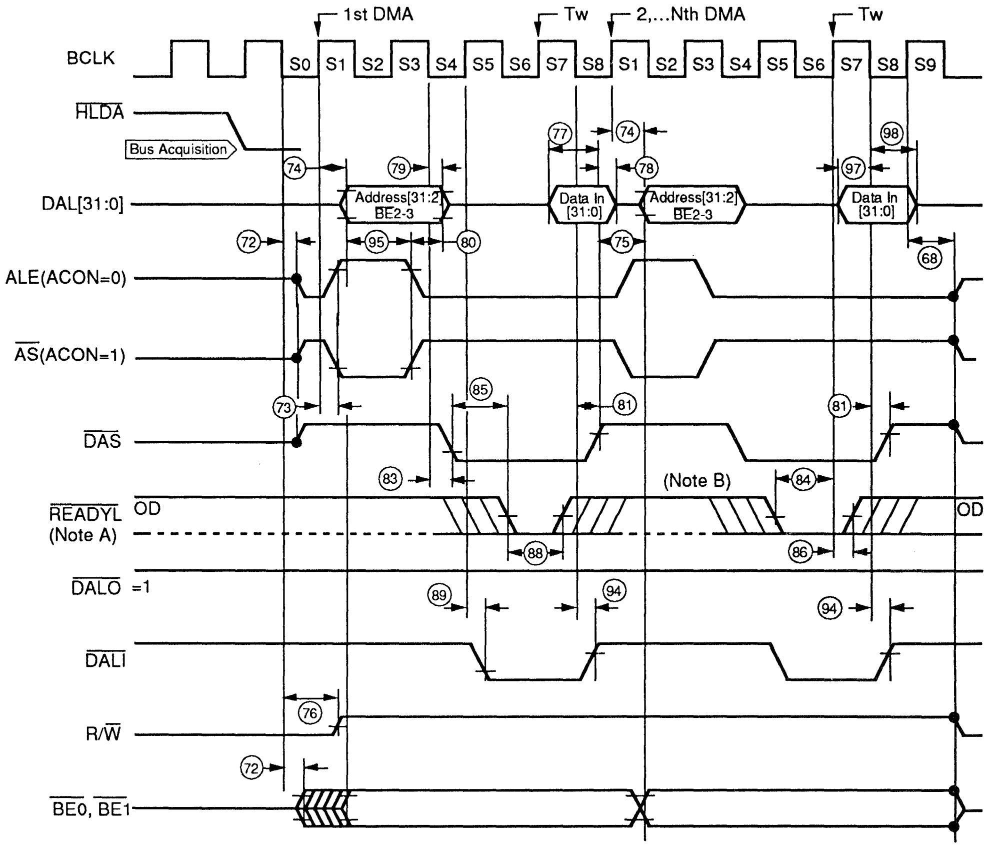
10594-063A

A: Timing diagram shows READYL for 0 wait states. For n wait states,
min = (n + 0.5) * tBCLK - 5 (ns)
max = (n + 1.5) * tBCLK - 35 (ns)

B: If READYL is not returned high prior to sampling in next cycle, a 0 wait state cycle will occur

Bus Master Write Timing (80X86, BACON = 00)

BUS INTERFACE TIMING DIAGRAMS



Notes:

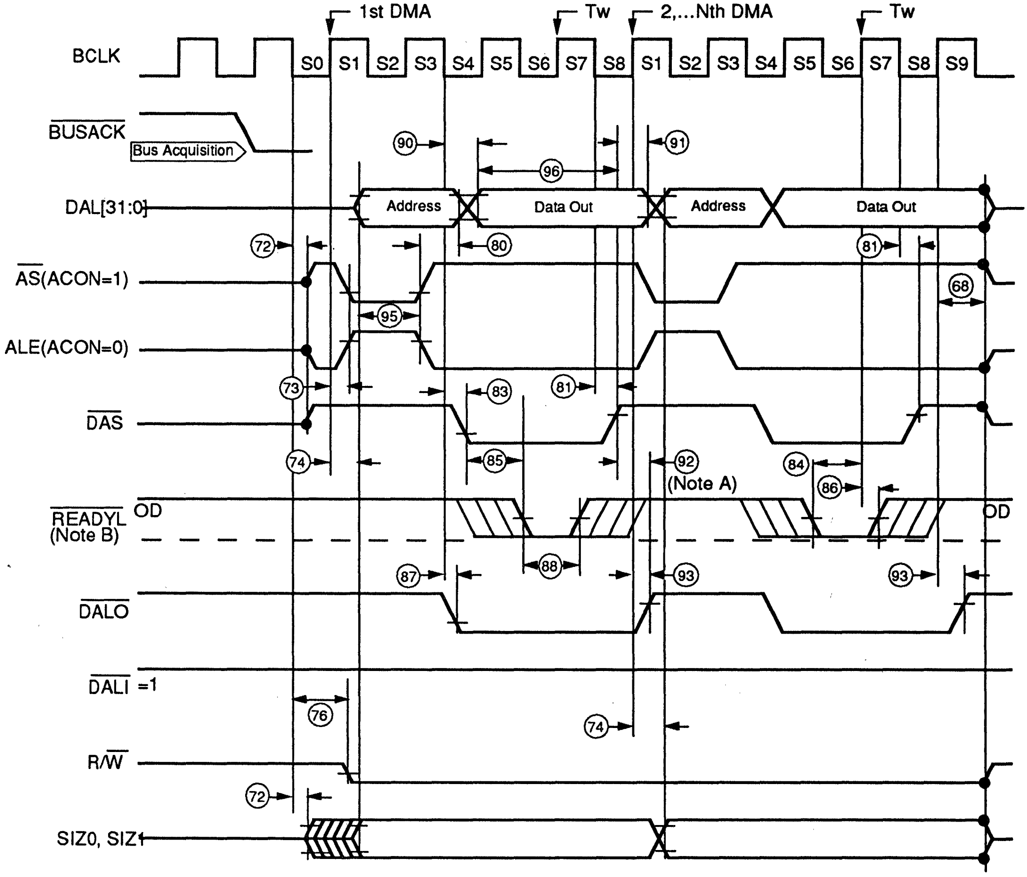
A: Timing diagram shows $\overline{\text{READYL}}$ for 0 wait states. For n wait states,
min = (n + 0.5) * tBCLK - 5 (ns)
max = (n + 1.5) * tBCLK - 35 (ns)

B: If $\overline{\text{READYL}}$ is not returned high prior to sampling in next cycle, a 0 wait state cycle will occur

10594-062A

Bus Master Read Timing (80X86, BACON = 00)

BUS INTERFACE TIMING DIAGRAMS



Notes:

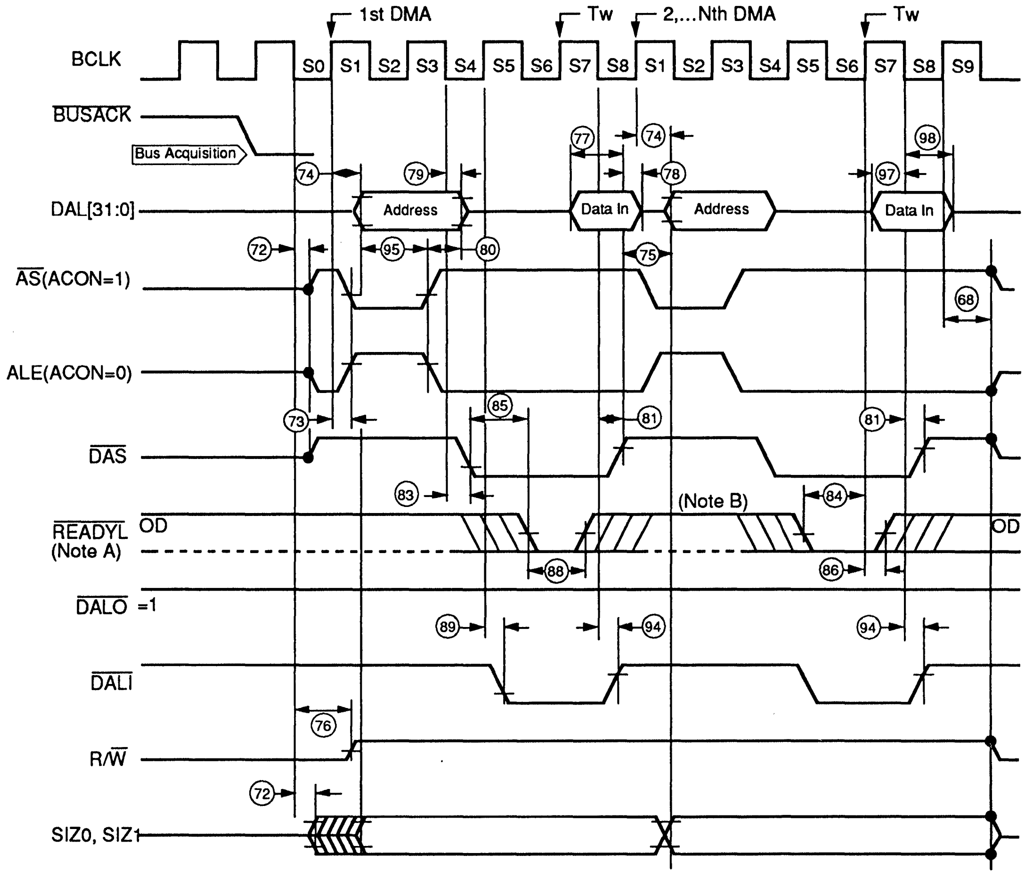
A: Timing diagram shows READYL for 0 wait states. For n wait states,
min = (n + 0.5) * tBCLK - 5 (ns)
max = (n + 1.5) * tBCLK - 35 (ns)

10594-065

B: If READYL is not returned high prior to sampling in next cycle, a 0 wait state cycle will occur

Bus Master Write Timing (680X0, BACON = 01)

BUS INTERFACE TIMING DIAGRAMS



Notes:

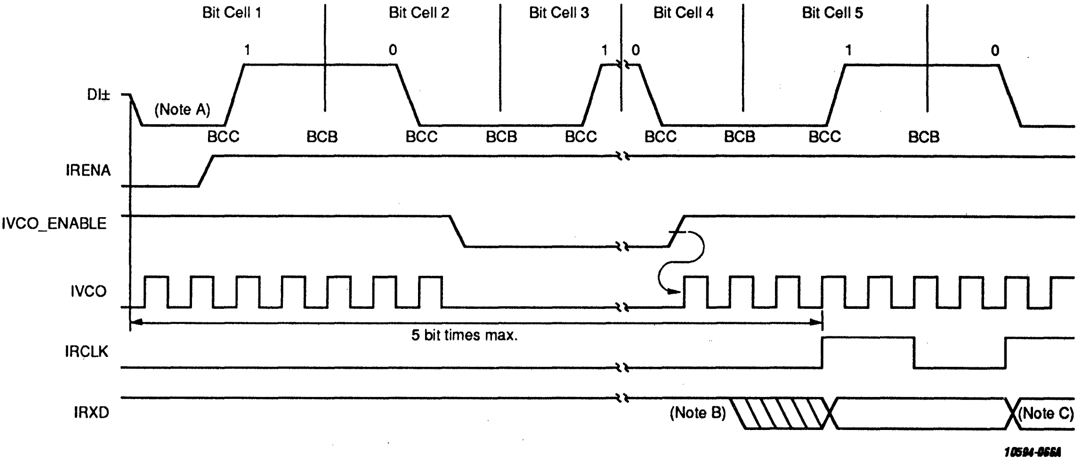
A: Timing diagram shows $\overline{\text{READYL}}$ for 0 wait states. For n wait states,
min = (n + 0.5) * t_{BCLK} - 5 (ns)
max = (n + 1.5) * t_{BCLK} - 35 (ns)

10594-064

B: If $\overline{\text{READYL}}$ is not returned high prior to sampling in next cycle, a 0 wait state cycle will occur

Bus Master Read Timing (680X0, BACON = 01)

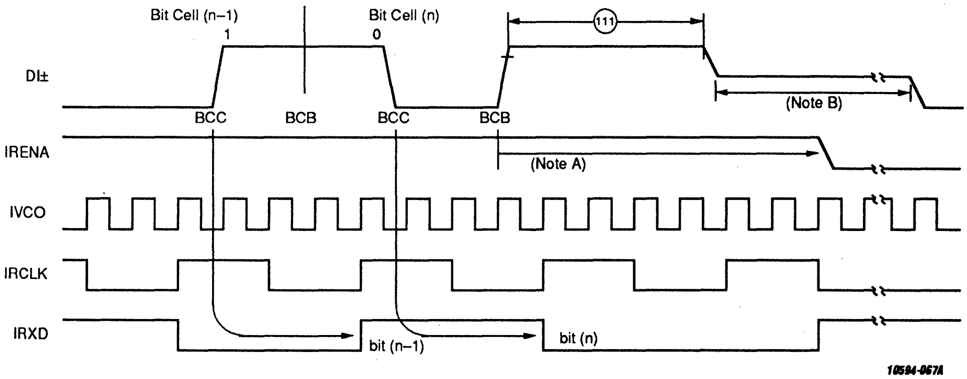
SERIAL INTERFACE TIMING DIAGRAMS



Notes:

A. Min. width > 45 ns
B. IRXD first decoded bit is not defined until bit time 5
C. First Valid Data

**Internal SIA Serial Receive Timing
Start of Reception & Clock Acquisition**

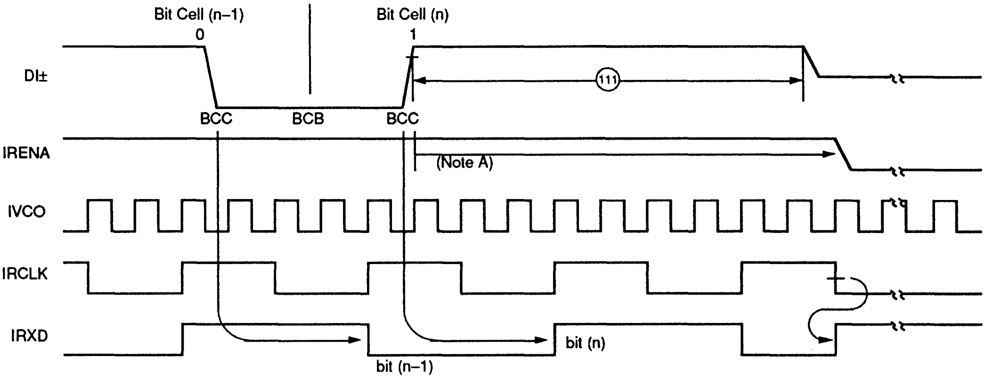


Notes:

A. IRENA deasserts in less than 3 bit times after last DI± rising edge
B. Start of next packet reception (2 bit times)

**Internal SIA Serial Receive Timing
End of Reception(Last Bit = 0)**

SERIAL INTERFACE TIMING DIAGRAMS

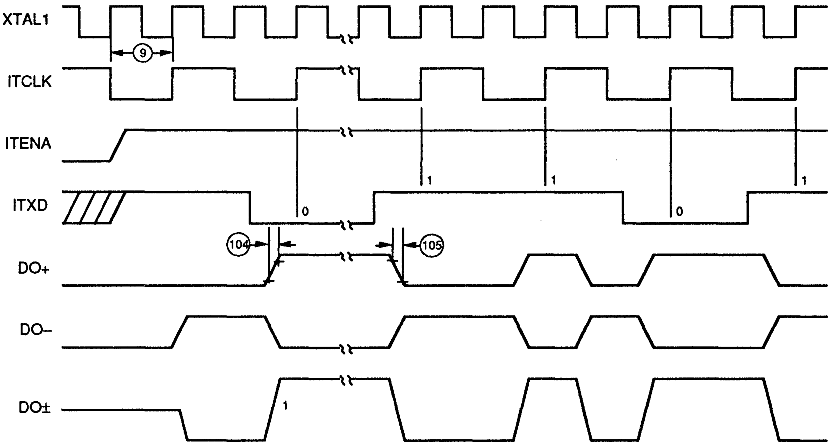


10594-068A

Note:

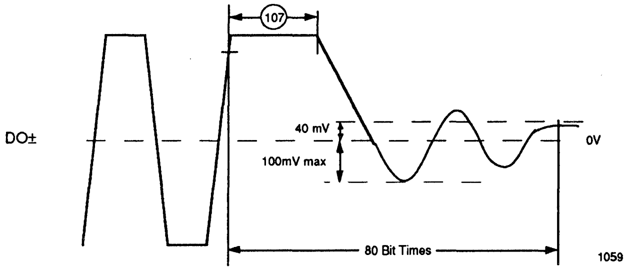
A. IRENA deasserts in less than 3 bit times after last DI± rising edge

Internal SIA Serial Timing
End of Reception (Last Bit = 1)



10594-069A

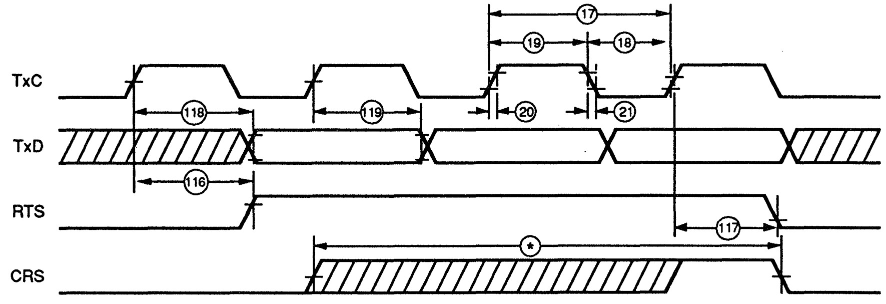
Internal SIA Transmit Timing
Start of Packet



10594-070A

AUI Port DO ETD Waveform

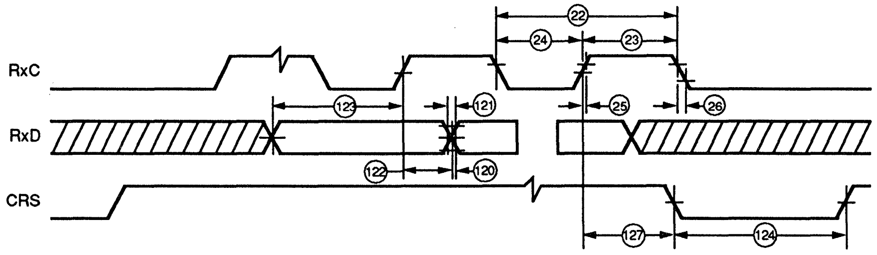
SERIAL INTERFACE TIMING DIAGRAMS



10584-071A

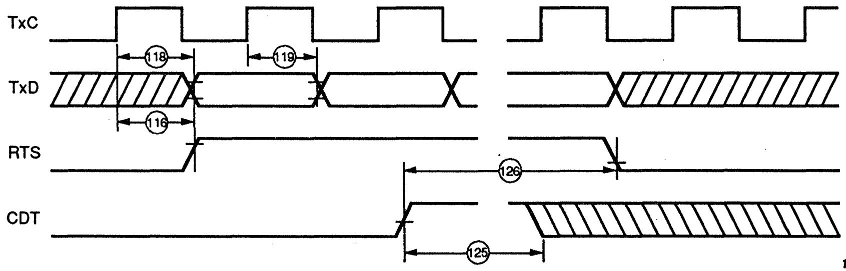
* During transmit, CRS input must be asserted (HIGH) and remain active HIGH after RTS goes inactive (LOW). If CRS is deasserted before RTS is deasserted, LCAR will be reported in TMD2 after the transmission is completed by the ILACC.

Transmit Timing General Purpose Serial Interface Port



10584-072A

Receive Timing General Purpose Serial Interface Port



10584-073A

Transmit Timing During Collision General Purpose Serial Interface Port

ADVANCE INFORMATION CONDENSED

Advanced Micro Devices

Am79C940-16/25, Am79C945-16

Media Access Controller for Ethernet (MACE)

DISTINCTIVE CHARACTERISTICS

- **Am79C940 integrated with 10BASE-T transceiver and AUI port**
- **Am79C945 optimized for use with external transceivers using AUI**
- **Supports IEEE 802.3/ANSI 8802-3 and Ethernet standards**
- **Low power, CMOS design with sleep mode allows reduced power consumption for critical battery powered applications**
- **84-pin PLCC Package**
- **Modular architecture allows easy tuning to specific applications**
- **High speed, 16-bit synchronous host system interface with 2 or 3 cycles/transfer**
- **Individual 128 byte transmit and receive 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
- **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; no requirement for external Serial Interface Adaptor (SIA)**
- **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, 10BASE-T or 10BASE-F MAU
  - DAI to external 10BASE-T or 10BASE-F MAU
  - GPSI to external encoding/decoding scheme
  - Internal 10BASE-T transceiver (Am79C940 only) with automatic selection of 10BASE-T or AUI port
- **Speed grades available:**
  - Am79C940 with 16 and 25 MHz system clock
  - Am79C945 with 16 MHz system clock

GENERAL DESCRIPTION

The Media Access Controller for Ethernet (MACE) is an 84-pin CMOS VLSI device designed to provide flexibility in customized LAN design. The MACE 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 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 provides an IEEE 802.3 interface tailored to a specific application. Its superior modular architecture and versatile system interface allow the MACE to be configured as a stand-alone device or as a connectivity cell incorporated into a larger, integrated system.

Two versions of the MACE are available. The standard version, the Am79C940, provides a complete Ethernet node solution with an integrated 10BASE-T transceiver, and supports 16 MHz and 25 MHz system clocks. The condensed version of the MACE, the Am79C945, is similar to the standard version but without the integrated 10BASE-T transceiver, and supports a 16 MHz system clock. Both the Am79C940 and the Am79C945 embody the Media Access Control (MAC) and Physical Layer Signaling (PLS) sub-layers of the IEEE 802.3 standard, and provide an IEEE defined Attachment Unit Interface (AUI) for coupling to an external Medium Attachment Unit (MAU). The MACE is compliant with 10BASE2, 10BASE5, 10BASE-T, and 10BASE-F transceivers.

Publication# 16235 Rev. A Amendment/0
Issue Date: January 1992

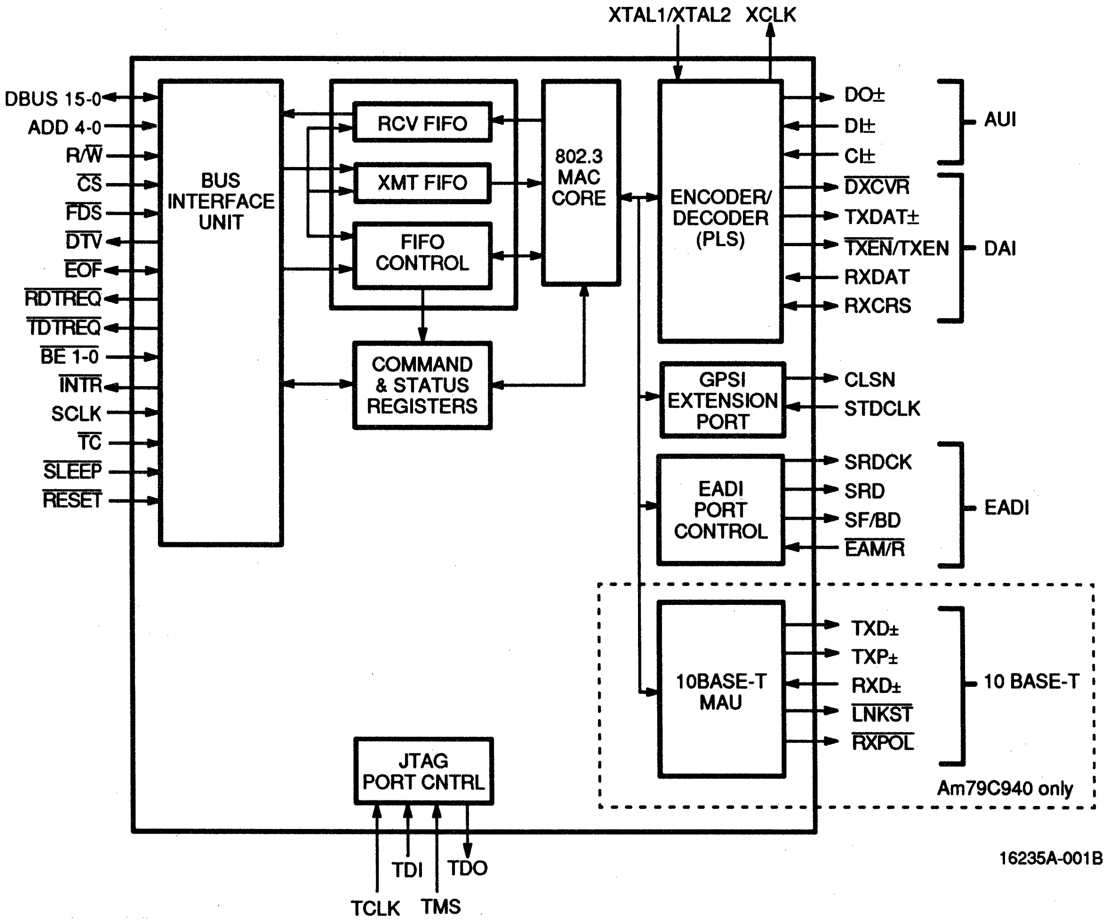
This document contains information on a product under development at Advanced Micro Devices, Inc. The information is intended to help you to evaluate this product. AMD reserves the right to change or discontinue work on this proposed product without notice.

Both versions of the MACE have additional features that enhance over-all system design. The individual 128 byte 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 can also be used to indicate transmit, receive, or collision status by connecting LEDs to the port. The MACE also provides an External Address Detection Interface (EADI) to allow external hardware address filtering in internetworking applications.

RELATED PRODUCTS

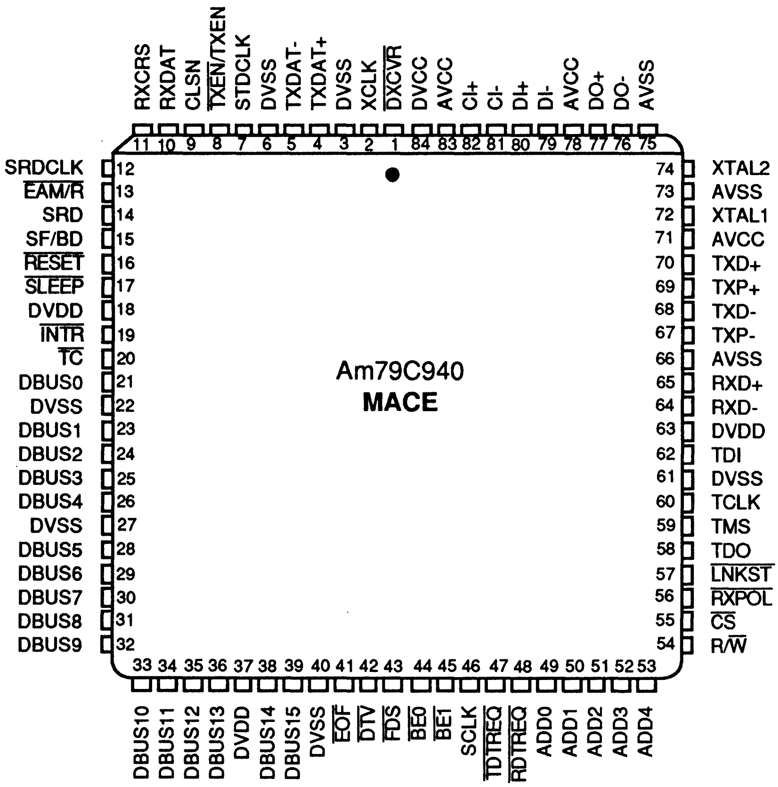
Part No.	Description
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am79C100	Enhanced Twisted Pair Ethernet Transceiver (TPEX Plus)
Am7996	IEEE 802.3/Ethernet/Cheapernet Transceiver
Am7997	IEEE 802.3 10BASE2/5 Compliant Tap Transceiver
Am79C980	Integrated Multiport Repeater (IMR)

BLOCK DIAGRAM



16235A-001B

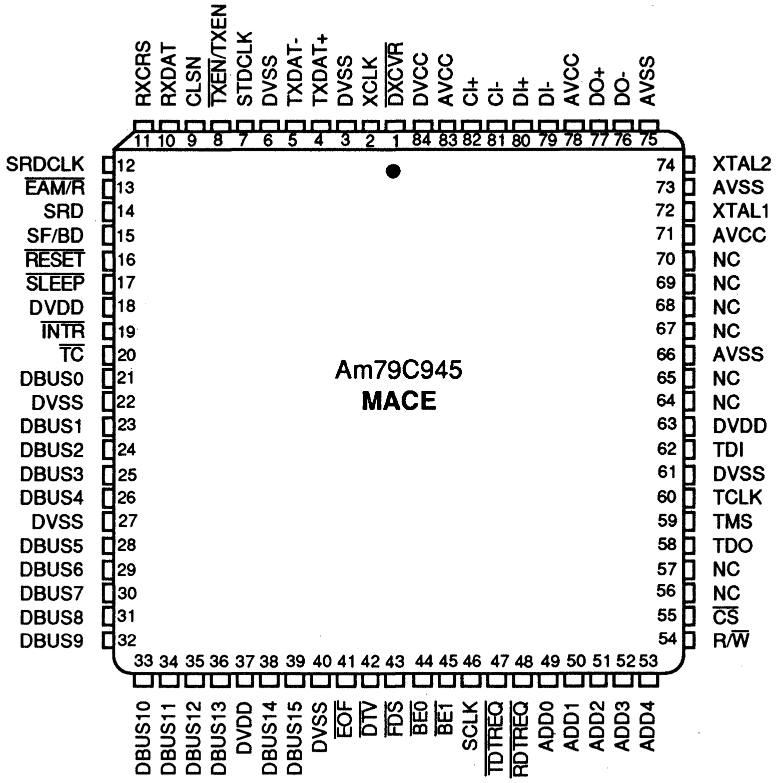
CONNECTION DIAGRAM



16235A-002A

WARNING: The pin-out is subject to change prior to product release. AMD reserves the right to change without notice.

CONNECTION DIAGRAM



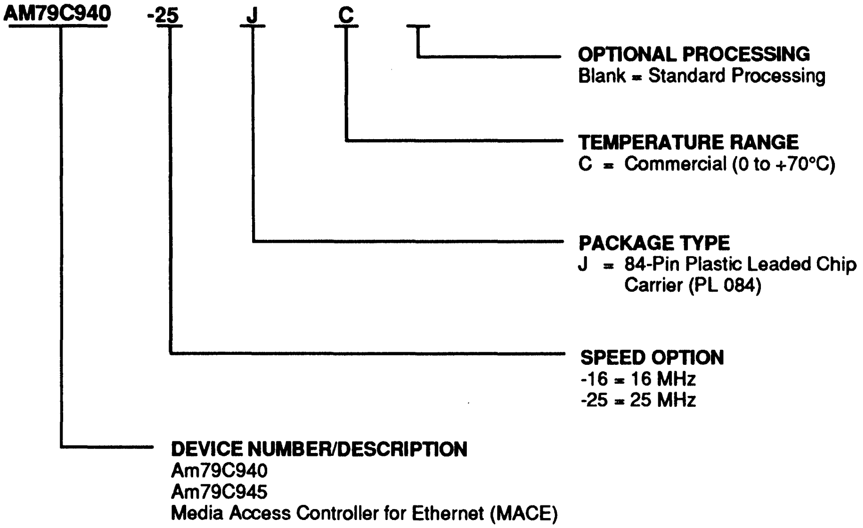
16235A-003A

WARNING: The pin-out is subject to change prior to product release. AMD reserves the right to change without notice.

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:



AM79C940 -25 J C

OPTIONAL PROCESSING
Blank = Standard Processing

TEMPERATURE RANGE
C = Commercial (0 to +70°C)

PACKAGE TYPE
J = 84-Pin Plastic Leaded Chip Carrier (PL 084)

SPEED OPTION
-16 = 16 MHz
-25 = 25 MHz

DEVICE NUMBER/DESCRIPTION
Am79C940
Am79C945
Media Access Controller for Ethernet (MACE)

Valid Combinations	
AM79C940-16 AM79C940-25 AM79C945-16	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, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN SUMMARY

Pin Name	Pin Function	Type
Attachment Unit Interface (AUI)		
DO+/DO–	Data Out	O
DI+/DI–	Data In	I
CI+/CI–	Control In	I
Digital Attachment Interface (DAI)		
TXDAT+	Transmit Data +	O
TXDAT–	Transmit Data –	O
$\overline{\text{TXEN}}$/TXEN	Transmit Enable	O
RXDAT	Receive Data	I
RXCRS	Receive Carrier Sense	I/O
$\overline{\text{DXCVR}}$	Disable Transceiver	O
General Purpose Serial Interface (GPSI) Extension		
STDCLK	Serial Transmit Data Clock	O
CLSN	Collision	I/O
External Address Detection Interface (EADI)		
SF/BD	Start Frame/Byte Delimiter	O
SRD	Serial Receive Data	O
$\overline{\text{EAM}}/\overline{\text{R}}$	External Address Match/Reject	I
SRDCLK	SRD Clock	O
System Interface		
DBUS$_{15-0}$	Data Bus	I/O
ADD$_{4-0}$	Address	I
R/$\overline{\text{W}}$	Read/Write	I
$\overline{\text{RDTREQ}}$	Receive Data Transfer Request	O
$\overline{\text{TDTREQ}}$	Transmit Data Transfer Request	O
$\overline{\text{DTV}}$	Data Transfer Valid	O
$\overline{\text{FDS}}$	FIFO Data Strobe	I
$\overline{\text{EOF}}$	End Of Frame	I/O
$\overline{\text{BE0}}$	Byte Enable 0	I
$\overline{\text{BE1}}$	Byte Enable 1	I
$\overline{\text{CS}}$	Chip Select	I
$\overline{\text{INTR}}$	Interrupt	O
$\overline{\text{RESET}}$	Reset	I
SCLK	System Clock	I
$\overline{\text{TC}}$	Timing Control	I
IEEE 1149.1 Test Access Port (TAP) Interface		
TCLK	Test Clock	I
TMS	Test Mode Select	I
TDI	Test Data Input	I
TDO	Test Data Out	O

PIN SUMMARY (Continued)

Pin Name	Pin Function	Type
General Interface		
XTAL1	Crystal Input	I
XTAL2	Crystal Output	O
XCLK	Crystal Clock	O
$\overline{\text{SLEEP}}$	Sleep Mode	I
DV_{DD}	Digital Power (4 pins)	P
DV_{SS}	Digital Ground (6 pins)	P
AV_{DD}	Analog Power (3 pins)	P
AV_{SS}	Analog Ground (3 pins)	P
Twisted Pair Transceiver Interface (10BASE-T) Am79C940 only		
TXD+/TXD–	Transmit Data	O
TXP+/TXP–	Transmit Pre-Distortion	O
RXD+/RXD–	Receive Data	I
$\overline{\text{LNKST}}$	Link Status	O
$\overline{\text{RXPOL}}$	Receive Polarity	O

Am7992B

Serial Interface Adapter (SIA)

Advanced Micro Devices

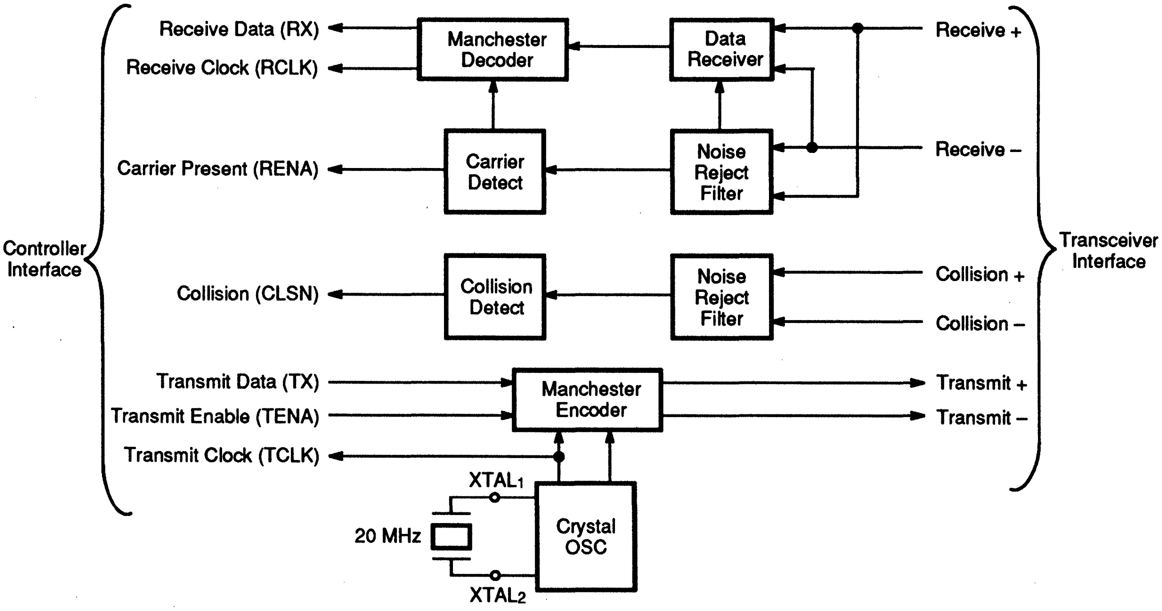
DISTINCTIVE CHARACTERISTICS

- **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



03378H-001A

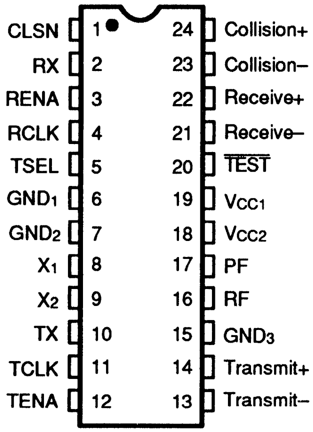
Publication# 03378 Rev. H Amendment/0
Issue Date: January 1992

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)

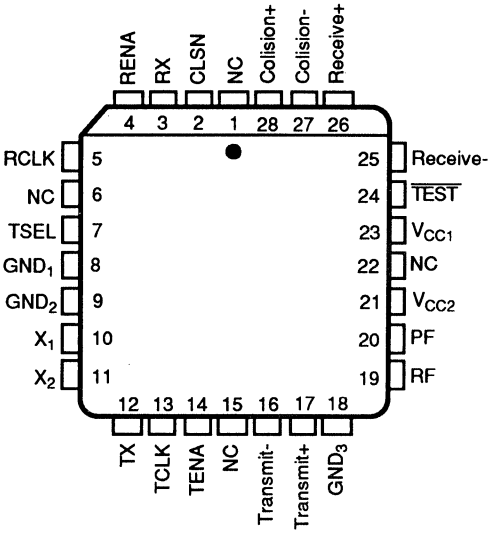
CONNECTION DIAGRAMS

DIP



03378H-002A

PLCC



03378H-003A

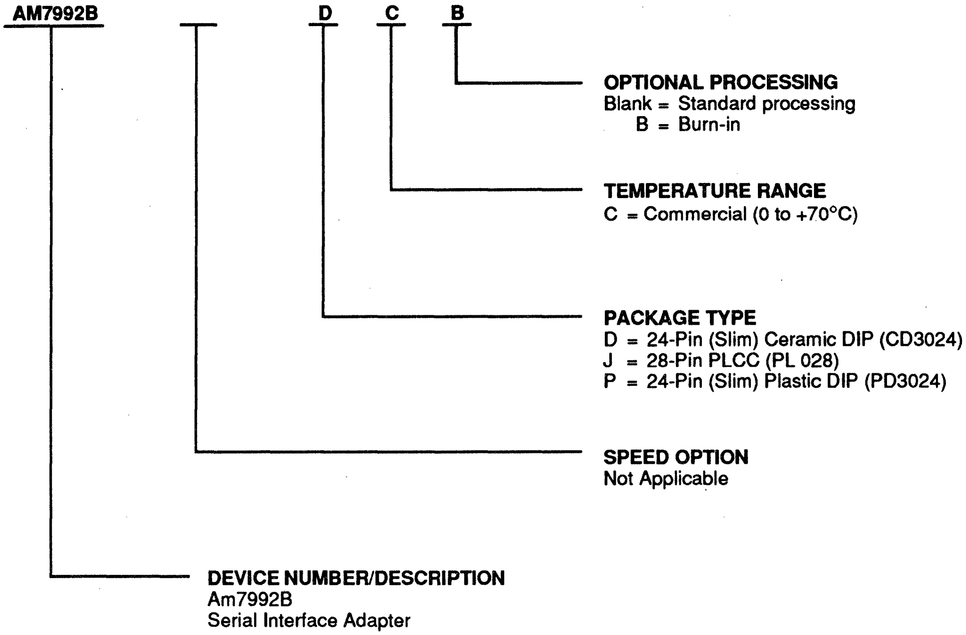
Note:

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:



OPTIONAL PROCESSING
Blank = Standard processing
B = Burn-in

TEMPERATURE RANGE
C = Commercial (0 to +70°C)

PACKAGE TYPE
D = 24-Pin (Slim) Ceramic DIP (CD3024)
J = 28-Pin PLCC (PL 028)
P = 24-Pin (Slim) Plastic DIP (PD3024)

SPEED OPTION
Not Applicable

DEVICE NUMBER/DESCRIPTION
Am7992B
Serial Interface Adapter

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 or to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

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 $\overline{\text{TEST}}$ 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 $\overline{\text{TEST}}$ 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 $\overline{\text{TEST}}$ 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 end of message. When $\overline{\text{TEST}}$ 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.

X_1, X_2

Biased Crystal Oscillator (Input)

X_1 is the input and X_2 is the bypass port. When connected for crystal operation, the system clock which appears at TCLK is half the frequency of the crystal oscillator. X_1 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.

$\overline{\text{TEST}}$

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 $\overline{\text{TEST}}$ is grounded, RX is enabled and RCLK is enabled except during Clock acquisition when RCLK is HIGH.

GND_1

High Current Ground

GND_2

Logic Ground

GND_3

Voltage Controlled Oscillator Ground

VCC_1

High Current and Logic Supply

VCC_2

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.

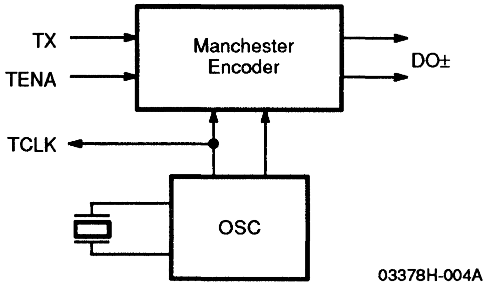


Figure 1. Transmit Section

Transmitter Timing and Operation

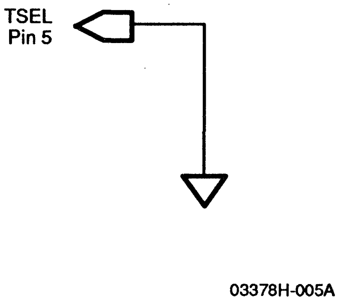
A 20 MHz fundamental mode crystal oscillator provides the basic timing reference in the SIA. It is divided by two to create the Transmit Clock reference (TCLK). Both 20 MHz and 10 MHz clocks are fed into the Manchester Encoder to generate the transitions in the encoded data stream. The 10 MHz clock, 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:

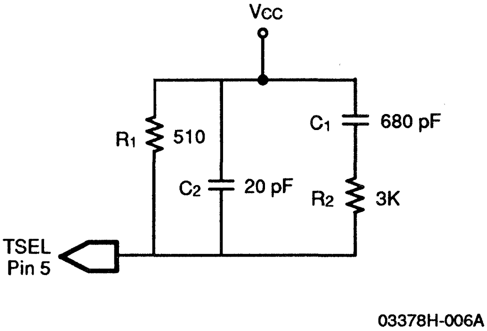
TSEL HIGH: The idle state of Transmit± yields "zero" differential to operate transformer-coupled loads (see Figure 2, Transmitter Timing – End of Transmission waveform diagram and Typical Performance Curve diagram).

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±.

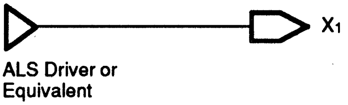


A. TSEL LOW



B. TSEL HIGH

Figure 2. Transmit Mode Select (TSEL) Connection



03378H-007A

Figure 3. TTL Clock Driver Circuit for X_1

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.	Nom.	Max.	Units
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, X_2 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 X_1 Driven from External Source waveform diagram and the TTL Clock Driver Circuit for X_1, Figure 3):

Clock Frequency: 20 MHz ±0.01%

Rise/Fall Time (t_R/t_F): < 4 ns, monotonic

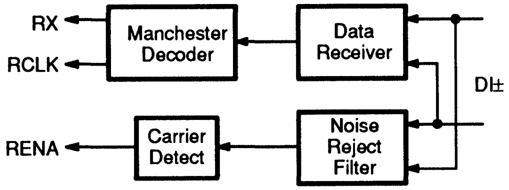
X_1 HIGH/LOW Time (t_{HIGH}/t_{LOW}): > 20 ns

X_1 Falling Edge to Falling Edge Jitter:
< ±0.2 ns at 1.5 V input

Receiver Path

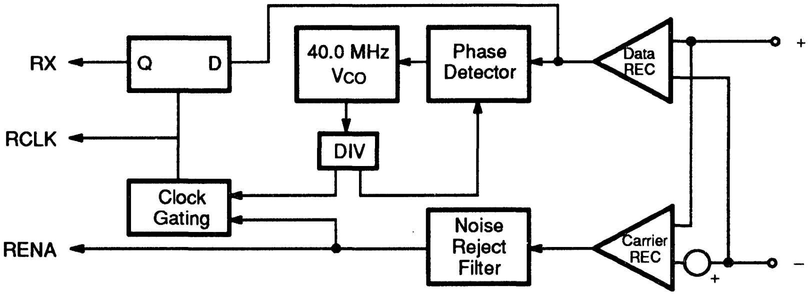
The principle functions of the Receiver are to signal the LANCE that there is information on the receive pair, and separate the incoming Manchester-encoded data stream into clock and NRZ data.

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 to 5.5 volts.



03378H-008A

Figure 4. Receiver



03378H-009A

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 presents 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 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

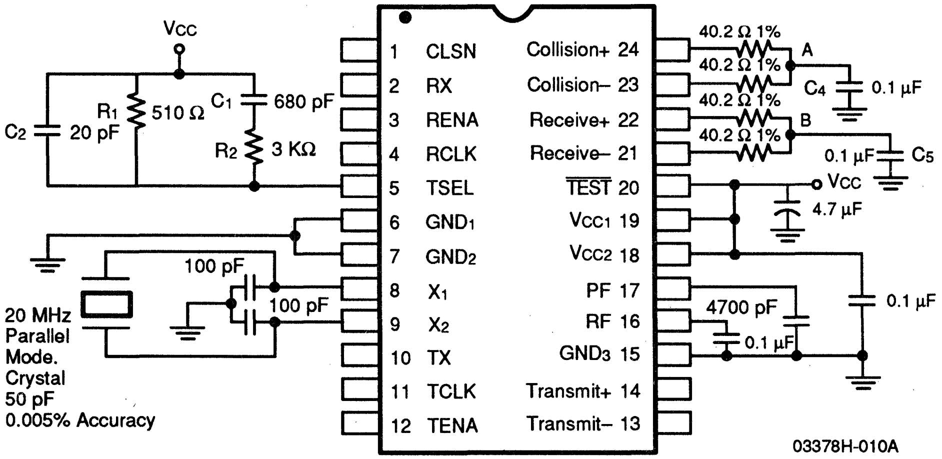
The data receiver is a comparator with clocked output to minimize noise sensitivity to the Receive± inputs. Input error (VIRD) is less than ±35 mV to minimize sensitivity to input rise and fall time. RCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit and 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 Vco.

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 R_1, R_2, C_1, C_2 for 0 differential nontransmit. Connect to ground for logic 1 differential nontransmit.
2. Pin 20 shown for normal device operation.
3. The inclusion of C_4 and C_5 is necessary to reduce the common-mode loading on certain transceivers which are direct coupled.
4. C_2 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

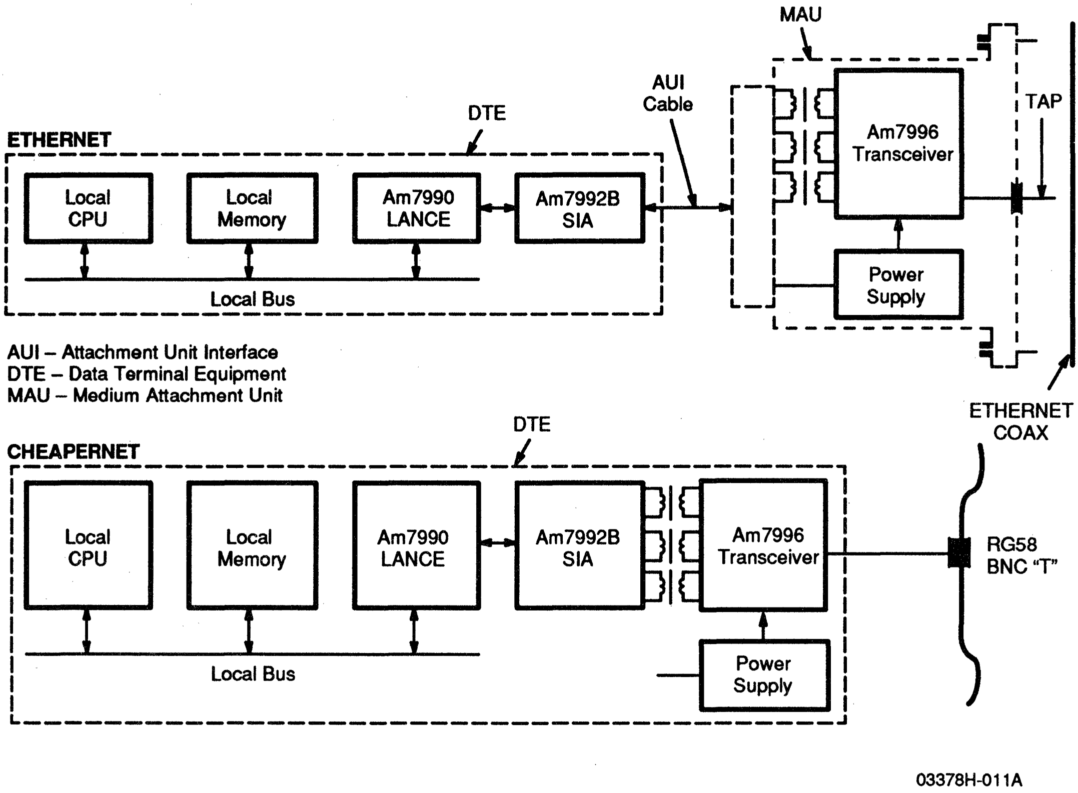


Figure 7. Typical ETHERNET Node

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	−65°C to +150°C
Ambient Temperature with Power Applied	0 to +70°C
Supply Voltage Continuous	+7.0 V
DC Voltage Applied to Outputs	−0.5 V to V_{CC} Max.
DC Input Voltage (Logic Inputs)	+5.5 V
DC Input Voltage (Receive±/Collision±)	− 6 to +16 V
Transmit± Output Current	−50 to +25 mA
DC Output Current, Into Outputs	100 mA
DC Input Current (Logic Inputs)	±30 mA
Transmit± Applied Voltage	0 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 (T_C)	0°C to +70°C
Supply Voltage (V_{CC})	+5.0 V ±5%

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		Unit
				Min.	Max.	
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 RCLK, TSEL, TCLK, RENA, RX, CLSN	I_{OL} = 16 mA, V_{CC} = Min.			0.5	V
		I_{OL} = 1 mA, V_{CC} = Min.			0.4	
V_{OD}	Differential Output Voltage (Transmit+) – (Transmit–) TX+ > TX– for V_O	R_L = 78 Ω		550	770	mV
	TX+ < TX– for $\overline{V_O}$			–550	–770	
$V_{OD\ OFF}$	Transmit Differential Output Idle Voltage	V_{CC} = Min., R_L = 78 Ω, TSEL = HIGH	(Note 1)	–20	20	mV
$I_{OD\ OFF}$	Transmit Differential Output Idle Current		(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 $\lvert\lvert V_O\rvert - \lvert\overline{V_O}\rvert\rvert$		(Note 1)		20	mV
V_{IH}	Input HIGH Voltage TX, TENA			2.0		V
I_{IH}	Input HIGH Current TX, TENA, $\overline{\text{TEST}}$	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, $\overline{\text{TEST}}$	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	
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, $\overline{\text{TEST}}$)	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

DC CHARACTERISTICS (Continued)

Parameter Symbol	Parameter Description	Test Conditions	COM'L		Unit
			Min.	Max.	
I_{IHZ}	Receive and Collision Input HIGH Current Power Off	V_{CC} = 0, V_{IN} = +6 V		1.86	mA
I_{IHX}	Oscillator (X_1) Input HIGH Current	V_{IN} = 2.4 V, V_{CC} = Max.		+800	µA
I_{ILX}	Oscillator (X_1) Input LOW Current	V_{IN} = 0.4 V, V_{CC} = Max.		−1.2	mA
V_{IHX}	Oscillator (X_1) Input HIGH Voltage	(Note 3)	2.0		V
V_{ILX}	Oscillator (X_1) Input LOW Voltage	(Note 3)		0.8	V

Notes:

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-On Delay (V_{IDC} Min. on Collision ± to $CLSN_L$)			160	ns

SWITCHING CHARACTERISTICS (Continued)

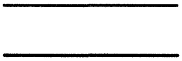
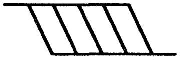
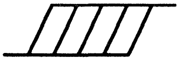
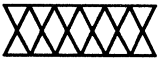
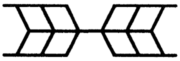
No.	Parameters	Description	Test Conditions	Min.	Max.	Unit
Transmitter Specification						
24	t_{TCL}	TCLK LOW Time	(Note 11)	45		ns
25	t_{TCH}	TCLK HIGH Time		45		ns
26	t_{TCR}	TCLK Rise Time			8	ns
27	t_{TCF}	TCLK Rise Time			8	ns
28	t_{TDS}, t_{TES}	TX and TENA Setup Time to TCLK	(Note 1)	5		ns
29	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}	X_1 to TCLK Propagation Delay for HIGH	(Notes 7 & 12)	5	18	ns
35	t_{XTCL}	X_1 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:

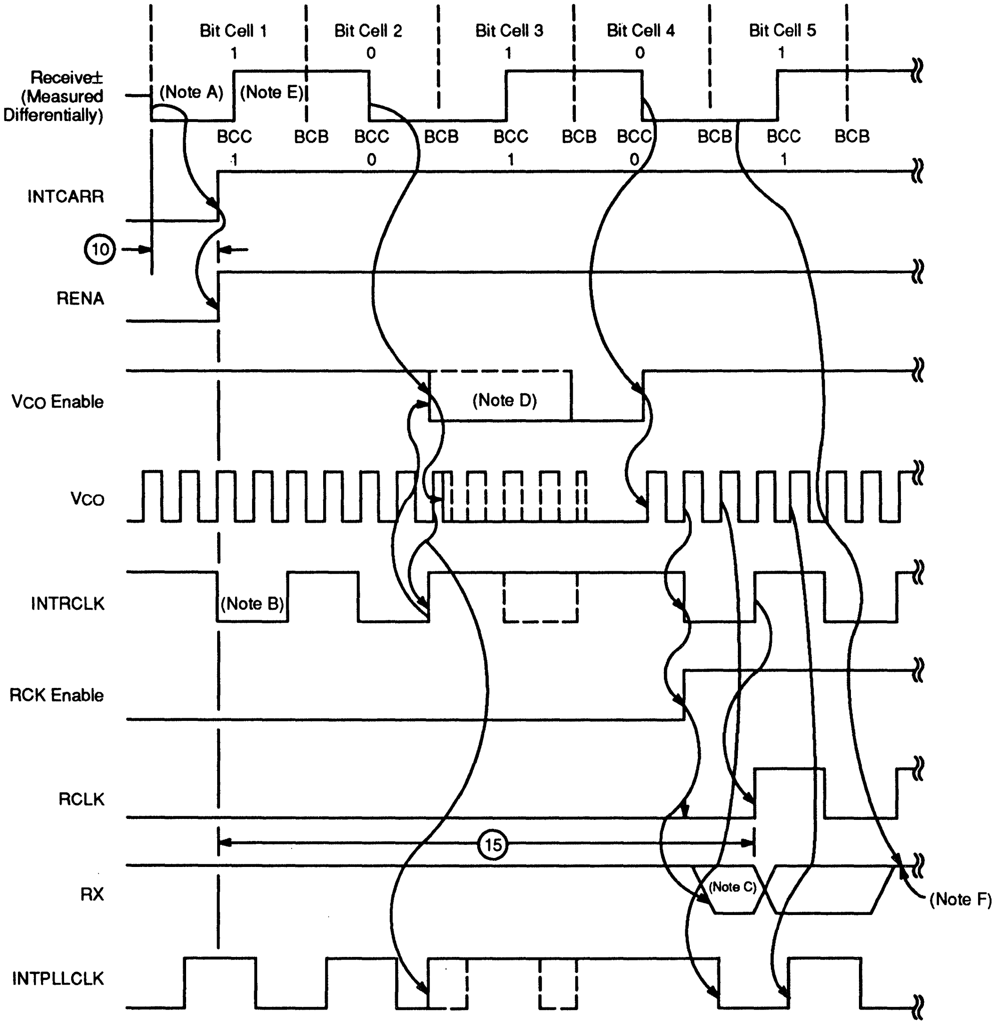
1. Tested but to values in excess of limits. Test accuracy not sufficient to allow screening guardbands.
2. Correlated to other tested parameter: I_{OD} OFF = V_{OD} OFF/R_L.
3. Not tested.
4. Test done by monitoring output functionally.
5. Receive, Collision and Transmit functions are inactive: X_1 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 X_1 rising edge, but initial state of TCLK is not defined. When TENA is High, TX data is Manchester encoded on the falling edge of X_1 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 volts of falling to 2.0 volts of rising edge.
11. Test correlated to T_{TCH}.
12. Measured from 50% point of X_1 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



03378H-012A

Notes:

A. Minimum Width > 45 ns

B. RCLK = INTRCLK when $\overline{\text{TEST}}$ LOW

C. RX undefined until bit time 5 (1st decoded bit)

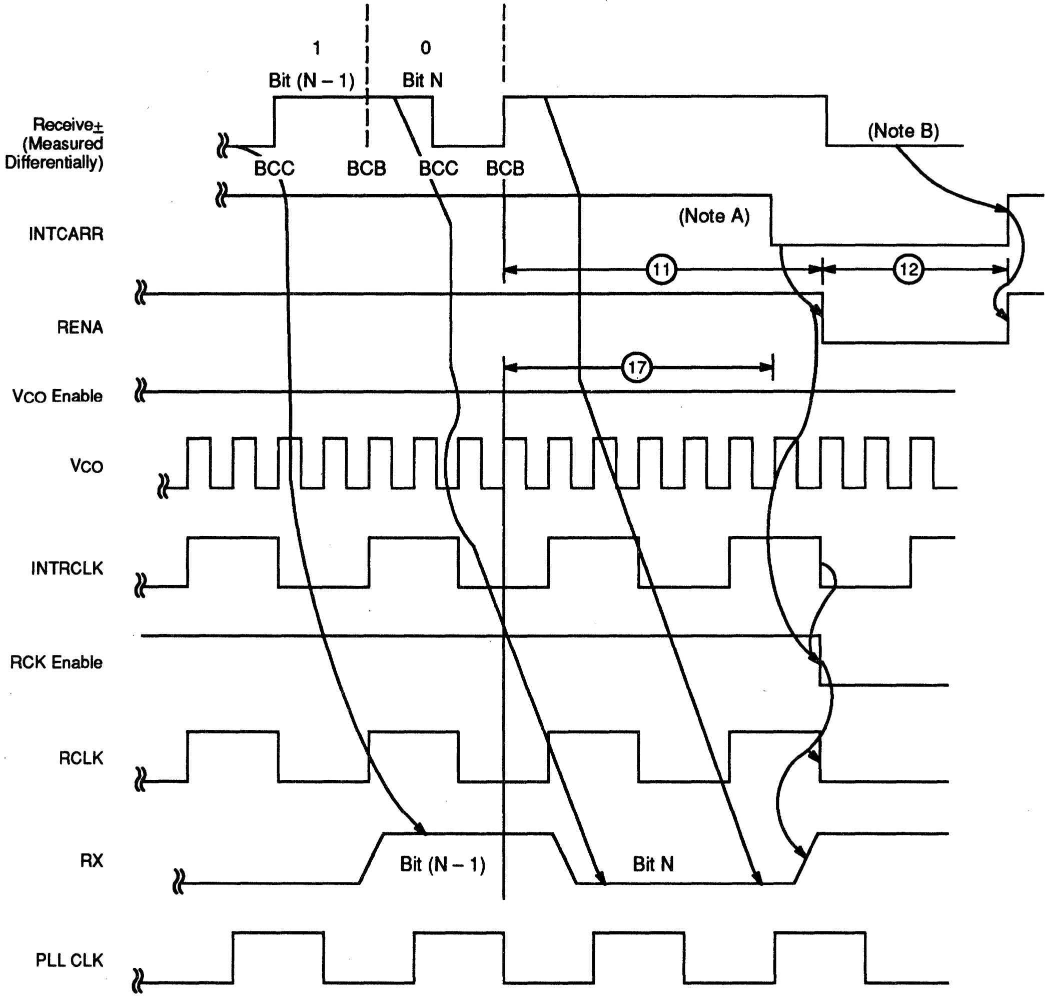
D. Oscillator Interrupt may occur at 2nd INTRCLK after Bit 2 Clock Transition

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



03378H-013A

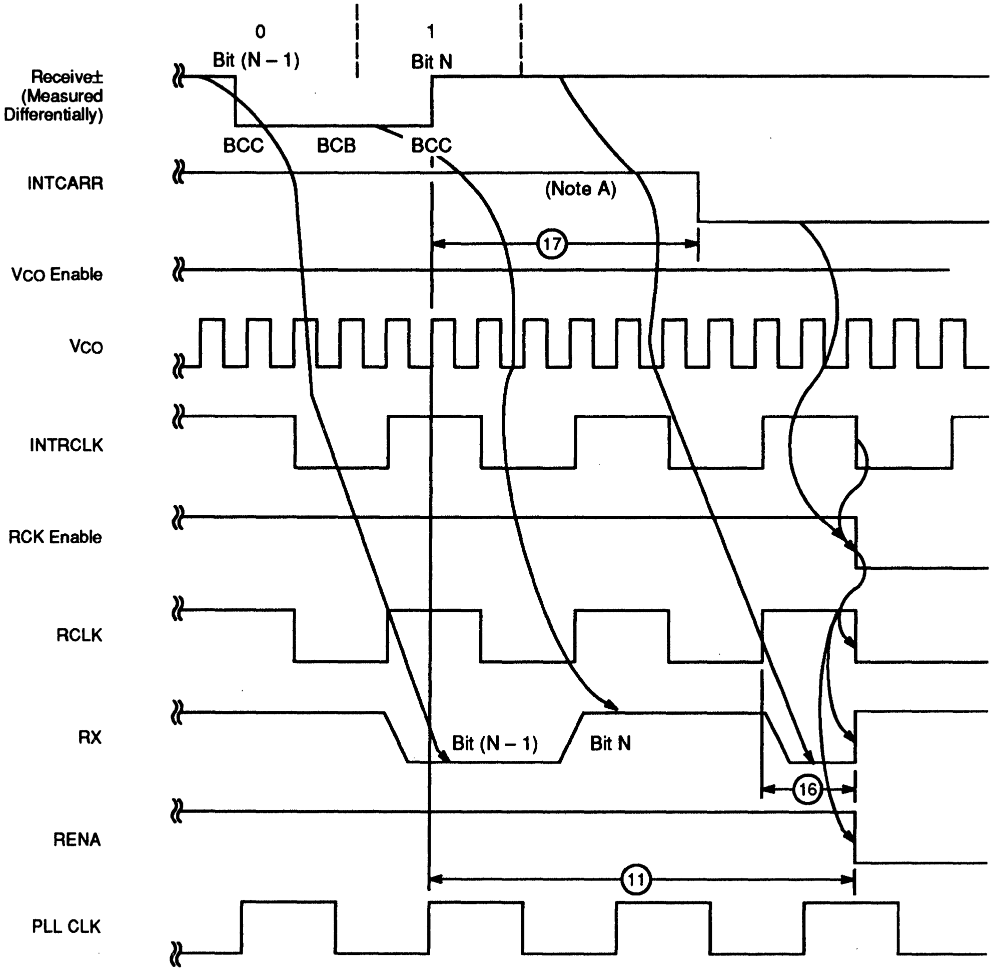
Notes:

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

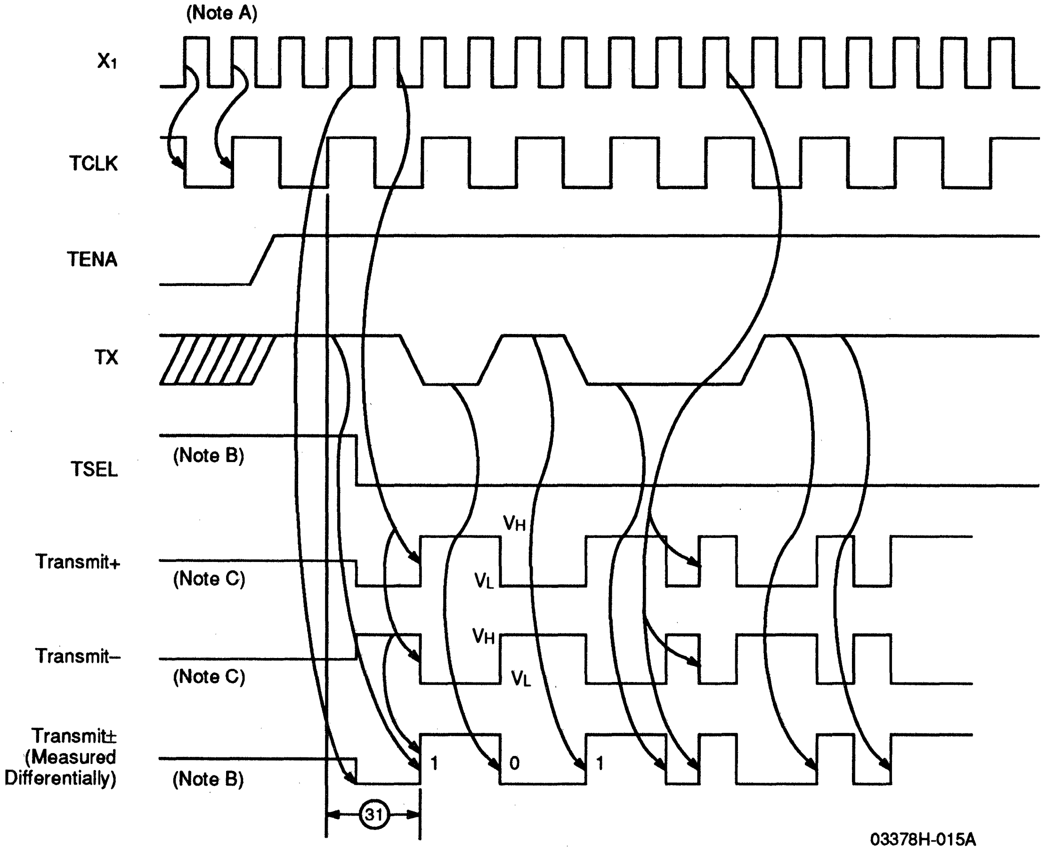


Note:

A. INTCARR deasserts 1.55 bit times after last Receive± Rising Edge

Receive Timing – End of Reception (Last Bit = 1)

SWITCHING WAVEFORMS



Notes:

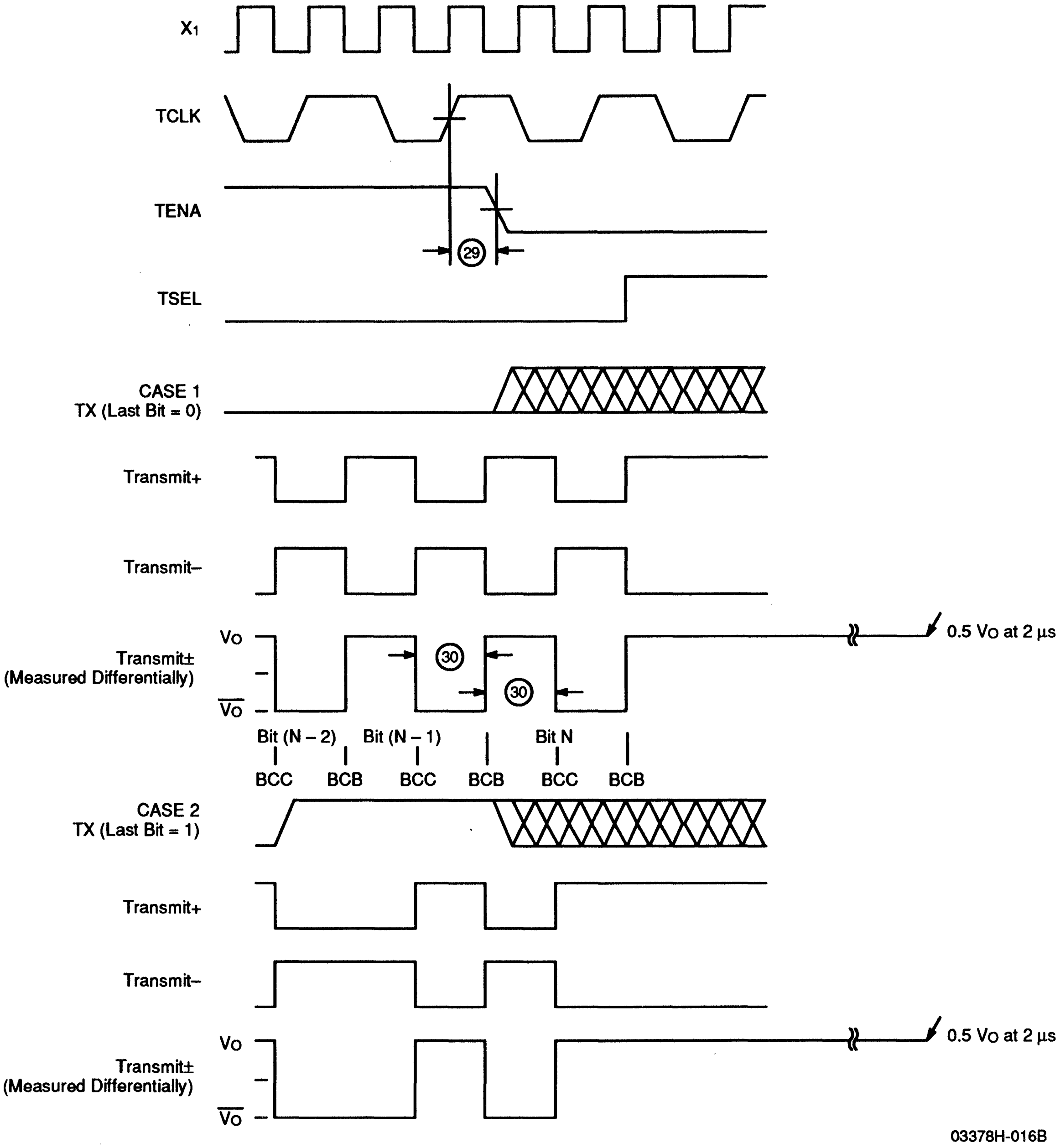
A. X_1 20 MHz Sine Wave from Crystal Oscillator or driven with X_1 driven from External Source Waveform.

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

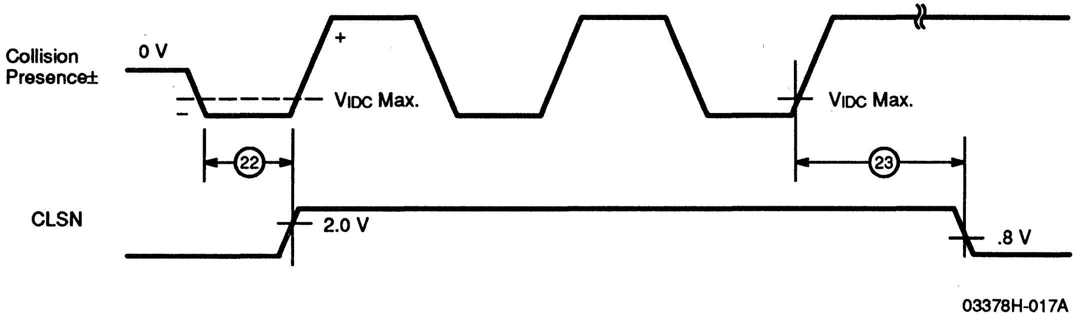


Transmit Timing – End of Transmission*

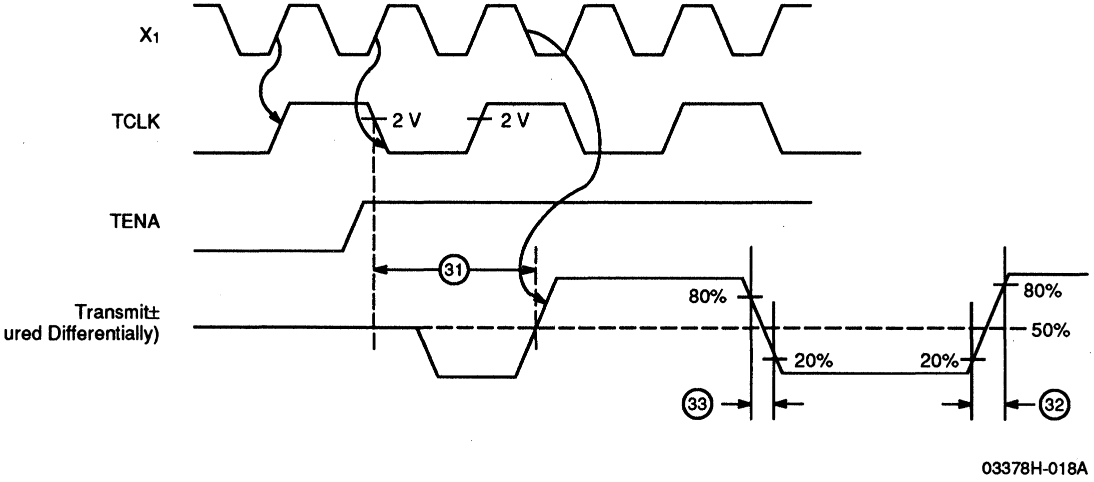
*TSEL Components (see Figure 2B)

See Typical Performance Curve for Response at End of Transmission with Inductive Loads

SWITCHING WAVEFORMS

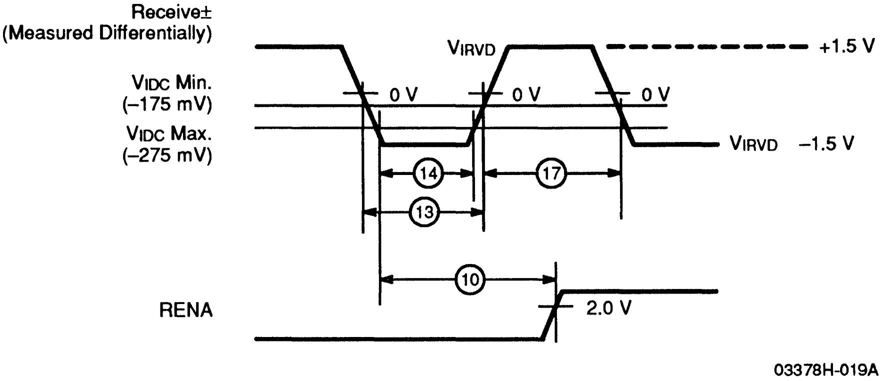


Collision Timing

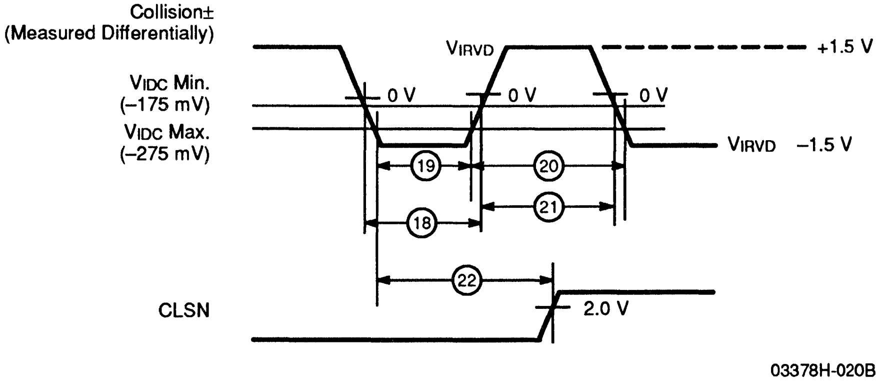


Transmit Timing (at start of packet)

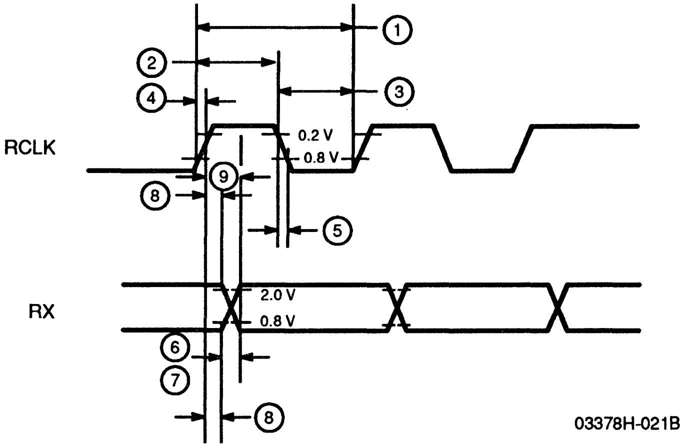
SWITCHING WAVEFORMS



Receive± Input Pulse Width Timing

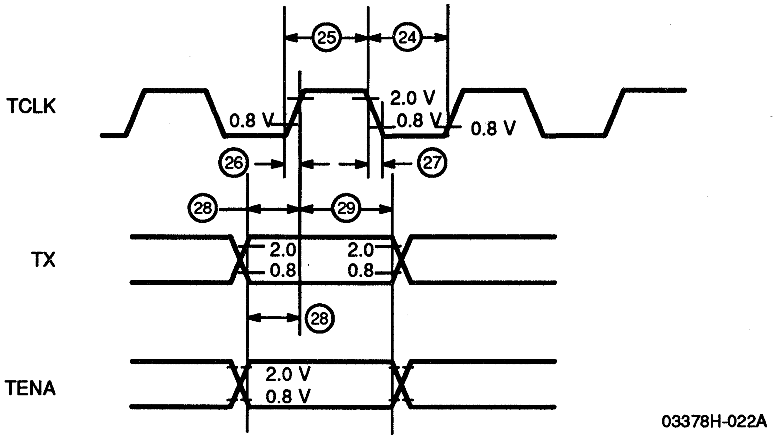


Collision± Input Pulse Width Timing

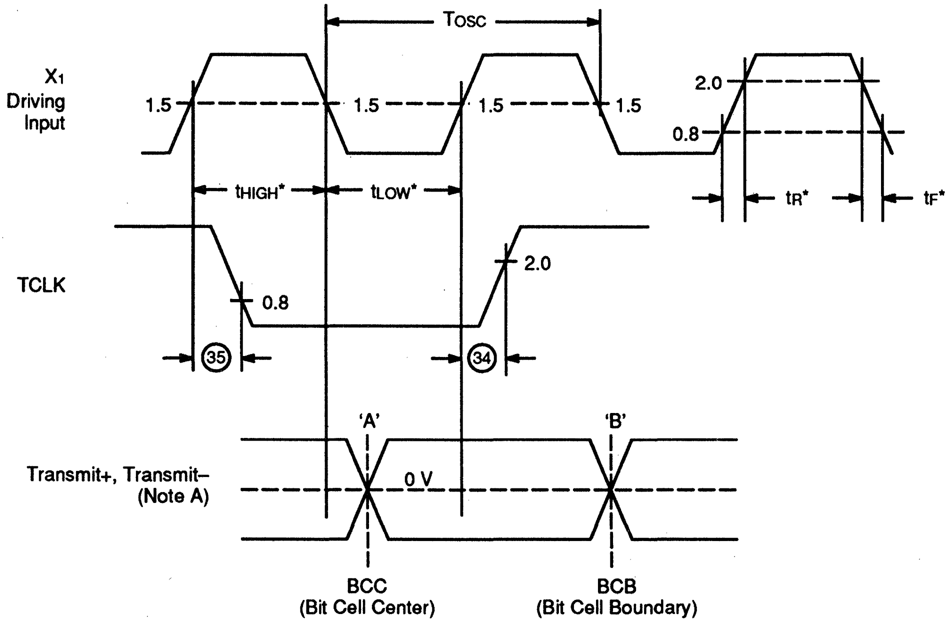


RCLK and RX Timing

SWITCHING WAVEFORMS



TCLK and TX Timing



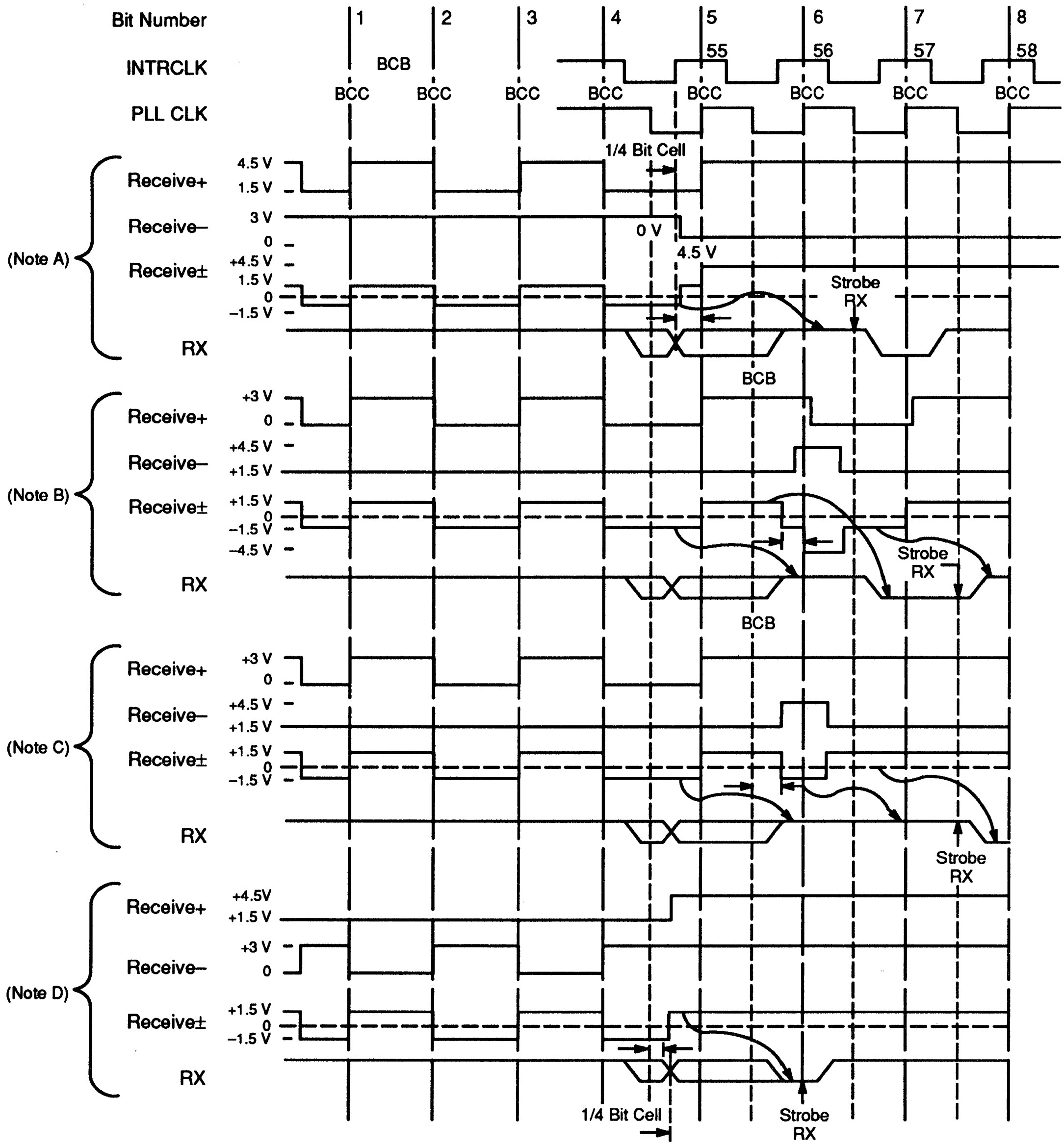
Note:

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



03378H-024A

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.

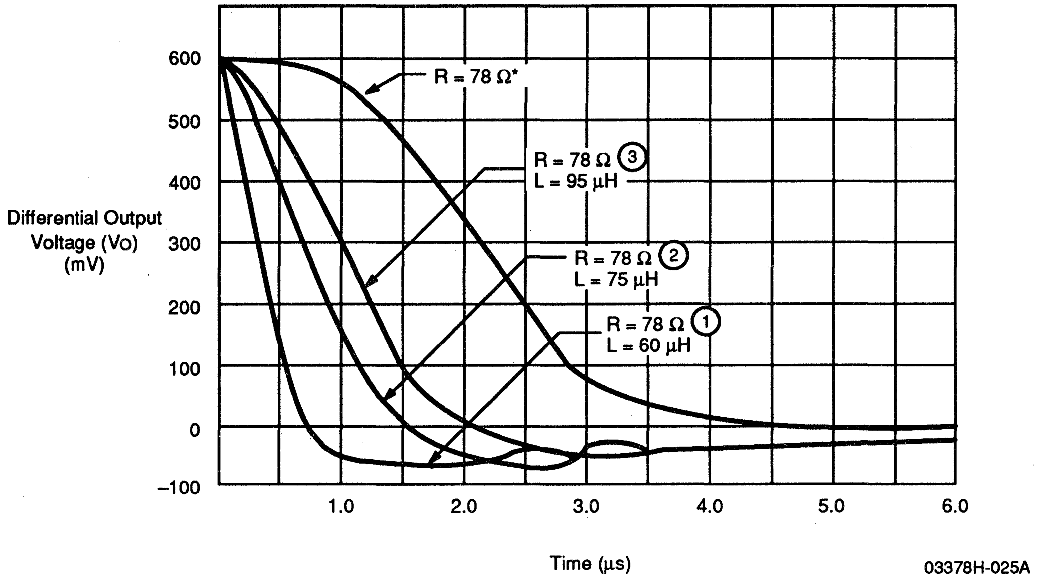
B. **Case 2, 6 Data Bit Pattern 1, 0**
Falling clock edge moved toward 1/4 bit cell RCLK data strobe. Case 2 uses bit 6, Case 6 uses bit 56.

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.

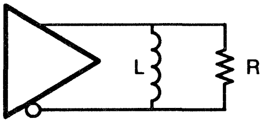
Input Jitter Timing

TYPICAL PERFORMANCE CURVE



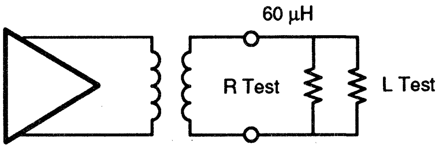
End of Transmission – Differential Output Voltage*

*Equivalent Load:

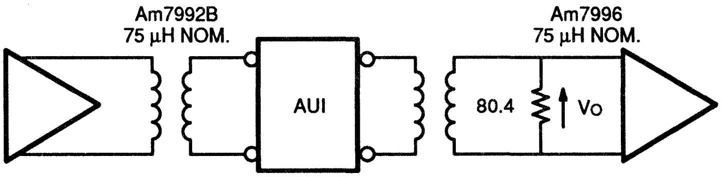


Notes:

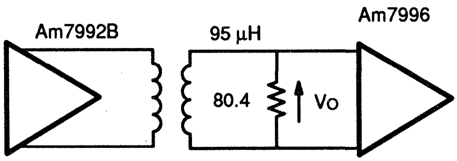
1. 802.3 Test Load:



2. 802.3 10BASE5 Network Connection:

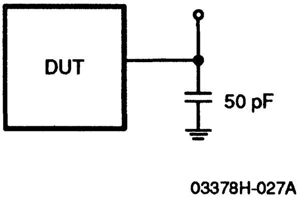


3. 802.3 10BASE2 Network Connection:

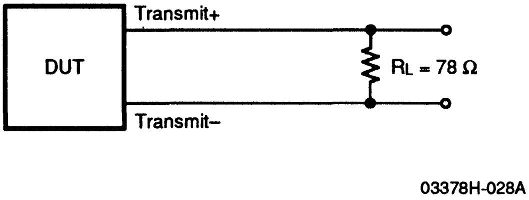


03378H-026A

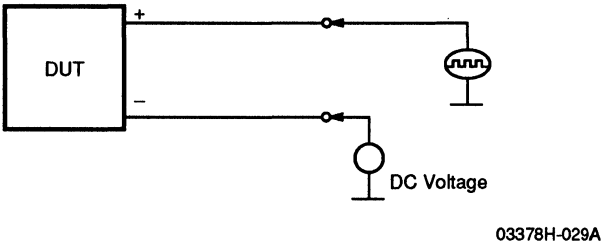
SWITCHING TEST CIRCUITS



A. Test Load for RX, RENA, RCLK, TCLK, CLSN



B. Transmit± Output



C. Receive± and Collision± Input

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

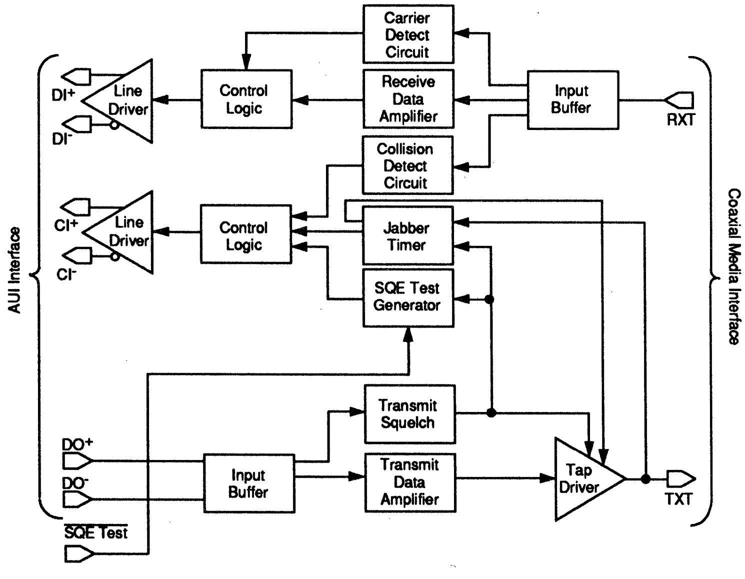
The Am7996 IEEE 802.3/Ethernet/Cheapernet Transceiver supports Ethernet Version 2, IEEE 802.3 (10BASE5), and IEEE 802.3 (10BASE2 – Cheapernet) transceiver applications. Transmit, receive, and collision detect functions at the coaxial media interface to the Data Terminal Equipment (DTE) are all performed 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



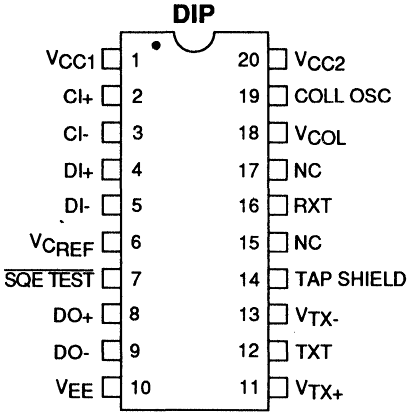
07506D-001B

Publication# 07506 Rev. D Amendment/0
Issue Date: January 1992

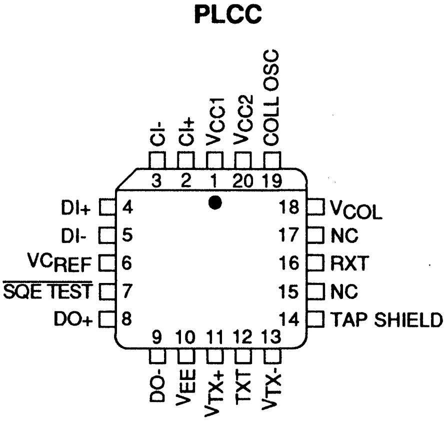
RELATED PRODUCTS

Part No.	Description
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am7992B	Serial Interface Adapter (SIA)
Am79C900	Intergrated Local Area Communications Controller
Am79C940	Media Access Controller for Ethernet (MACE)
Am79C980	Integrated Multiport Repeater (IMR)

CONNECTION DIAGRAMS



07506D-002A



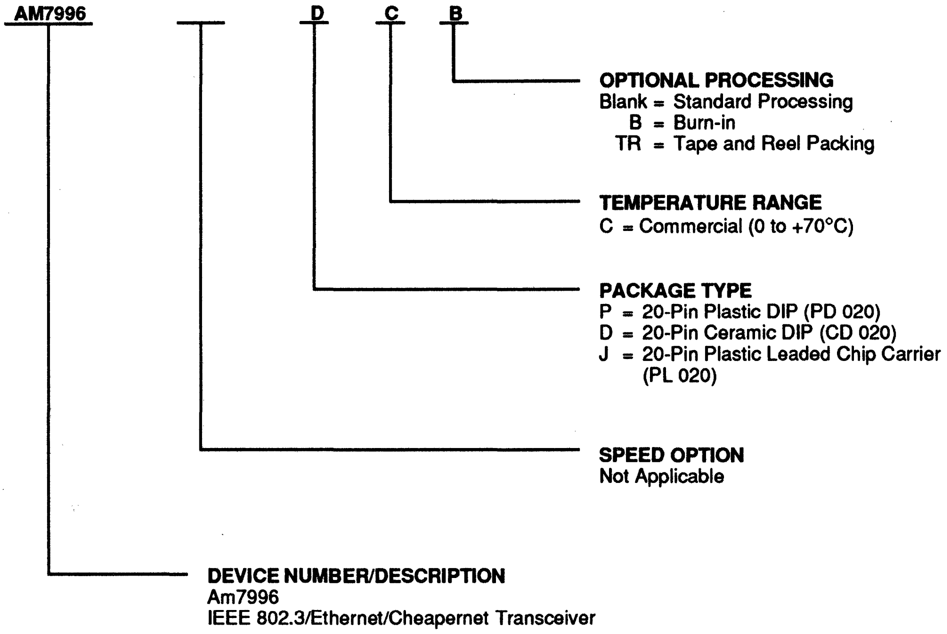
07506D-003A

Note:
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:



AM7996 ___ D C B

OPTIONAL PROCESSING
Blank = Standard Processing
B = Burn-in
TR = Tape and Reel Packing

TEMPERATURE RANGE
C = Commercial (0 to +70°C)

PACKAGE TYPE
P = 20-Pin Plastic DIP (PD 020)
D = 20-Pin Ceramic DIP (CD 020)
J = 20-Pin Plastic Leaded Chip Carrier (PL 020)

SPEED OPTION
Not Applicable

DEVICE NUMBER/DESCRIPTION
Am7996
IEEE 802.3/Ethernet/Cheapernet Transceiver

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 or to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

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 Reterence (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 counts 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 reset 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/IEEE802.3–10BASE5). 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.

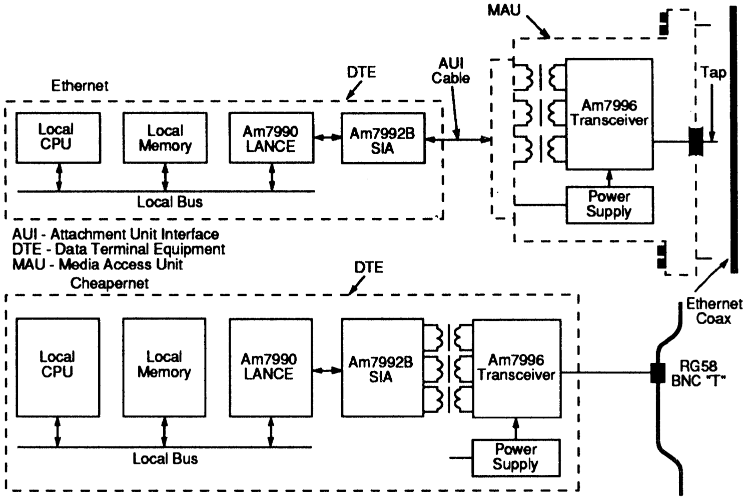
SQE Test

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)



07506D-004B

Figure 1. Typical Ethernet Node

Table 1. Transmit Mode Collision Detect Function Table

MAU	Number of Transmitters		
Mode of Operation	< 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

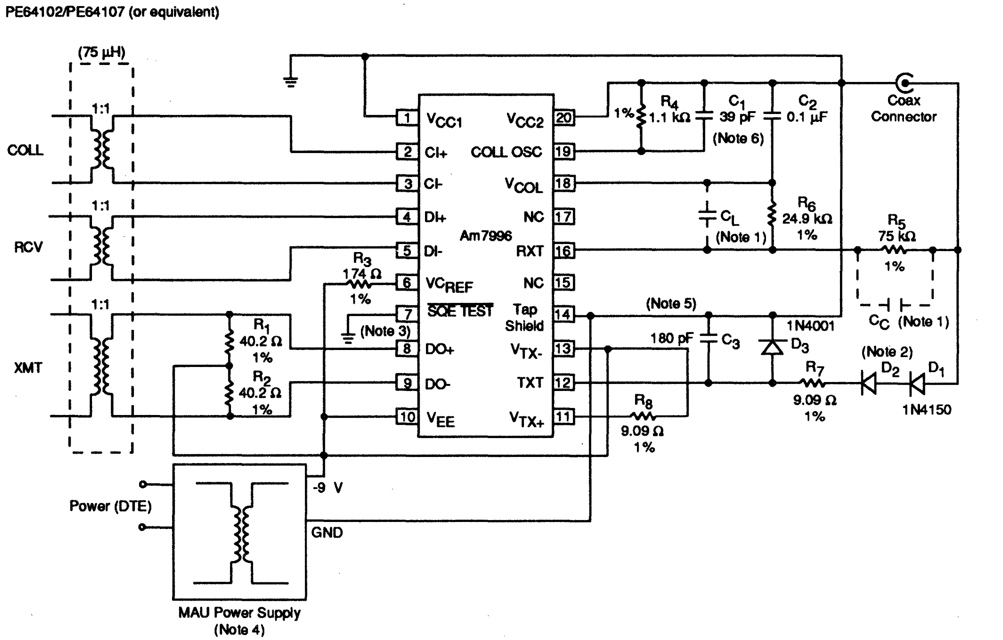
Table 3. Receive Mode Collision Detect Function Table

MAU	Number of Transmitters		
Mode of Operation	< 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 R_9, R_{10} and C_4. 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.

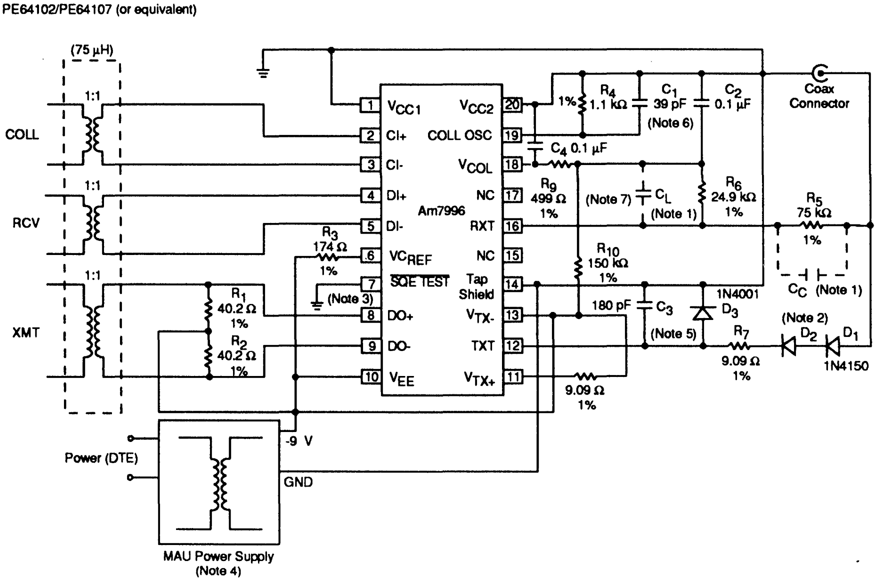


07506D-005A

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 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 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. Figure 2 used for production testing of all parameters that are tested.

Figure 2. Am7996 External Component Diagram for Transmit Mode Collision Detect



07506D-006A

Notes:

1. CL is the effective load capacitance across R6; CC is the compensation capacitance (CC = 1/3 CL).
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

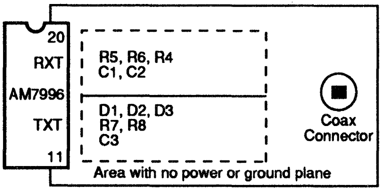
LAYOUT CONSIDERATIONS

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.

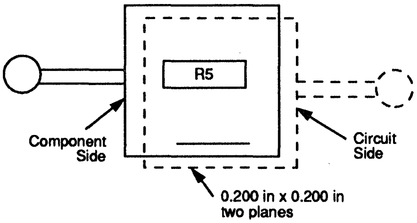


07506D-007B

Figure 4-1.

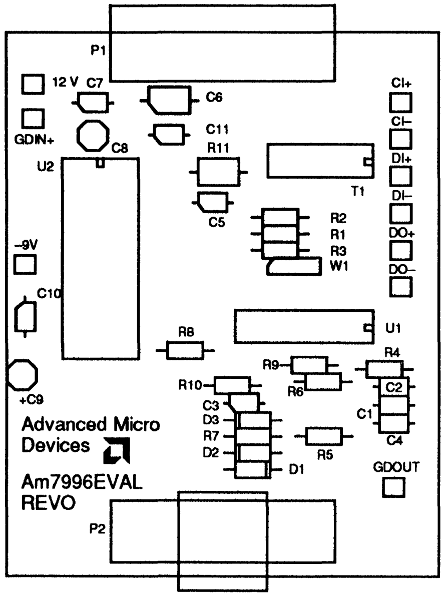
If the above layout rules are followed, the parasitic capacitance in parallel with R6 will be about 6 pF. This parasitic capacitance is shown in the schematics as C_L (Note that C_L is a parasitic capacitance. Do not add a discrete capacitor in parallel with R6). The capacitor labeled C_C in the schematics 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 inch by 0.200 inch 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 inch 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 4 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.



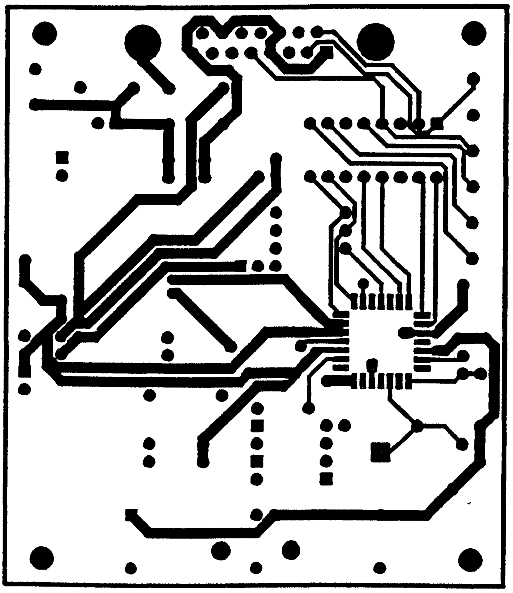
07506D-008A

Figure 4-2.



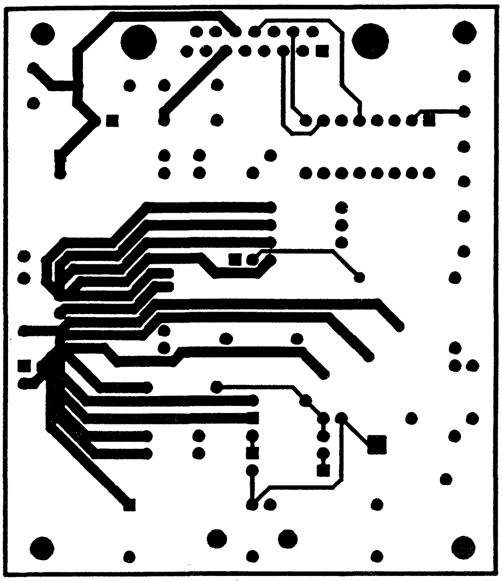
07506D-009B

Component Side Silkscreen



12473-021A

Component Side



12473-020A

Solder Side

C11	CAP-0.01UF
C2, C4, C7, C10	CAP-0.1UF
C8	CAP-4.7UF
C3	CAP-150PF
C6	GAP CAP-0.001UF
C1	CAP-39PF
C5	CAP-1000PF
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 4 Layer PCB Application

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	–65 to +150°C
Ambient Temperature Under Bias	0 to +70°C
Supply Voltages (V_{EE}, V_{TX-})	–12.0 to +0.5 V
DC Input Voltage (D0+, D0–)	–12.0 to +0.5 V
DC Input Voltage (RXT)	–6 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 to +70°C
Supply Voltage (V_{EE})	–8.1 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 (Note10)	Commercial			Units
			Min.	Typ.	Max.	
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–) $\|\|V_{OD}\| - \|V_{OD}\|\|$ (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 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; Pin 15, 17 – No Connections)

Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Units
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

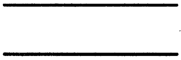
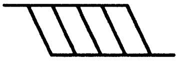
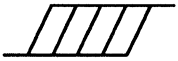
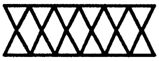
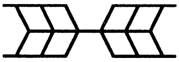
No.	Parameters	Description	Test Conditions	Commercial			Units
				Min.	Typ.	Max.	
Receiver Specification							
1	t_{PWREJ}	DO± Input Pulse Width to Reject (DO± > V_{IDC}, Min.)	(Note 1)		15	7	ns
2	t_{PWTON}	DO± Input Pulse Width to Turn On (DO± > V_{IDC}, Max.)	(Note 1)	20	15		ns
3	t_{PWSON}	DO± Input Pulse Width to Stay On (DO± < V_{IDC}, Max.)	(Note 1)			105	ns
4	t_{PWOFF}	DO± Input Pulse Width to Turn Off (DO± < V_{IDC}, Min.)	(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 $\|t_{TXTR}-t_{TXTF}\|$	(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%, $R_L = 78\ \Omega$			7	ns
20	t_{RF}	DI± and CI± Fall Time	80–20%, $R_L = 78\ \Omega$			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.

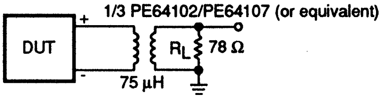
*Notes listed correspond to the respective references made in DC Characteristics and Switching Characteristics tables.

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

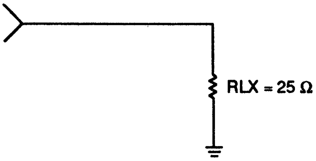
KS000010

SWITCHING TEST CIRCUIT



07506D-010A

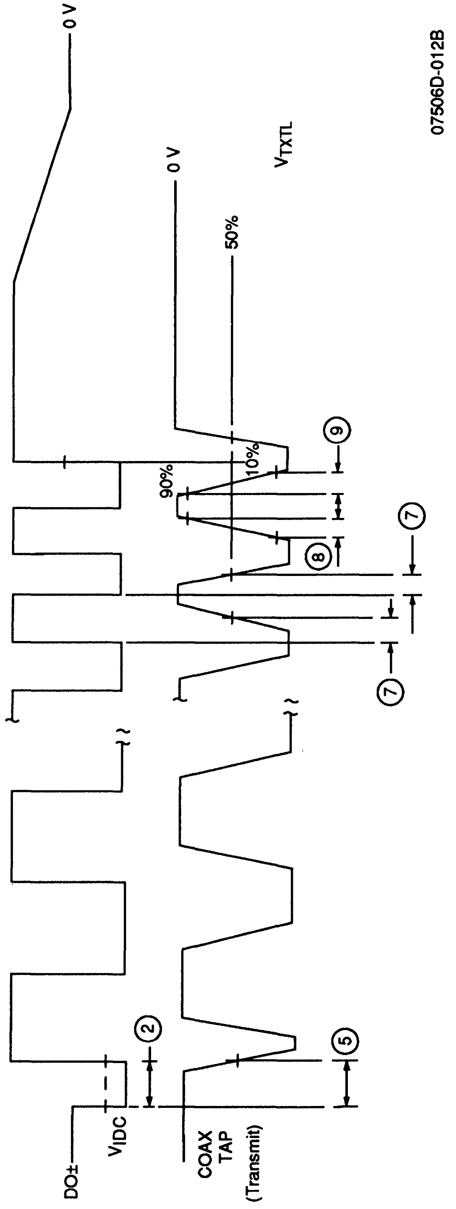
A. AUI Transmit (DI+, DI–,; CI+, CI–)



07506D-011A

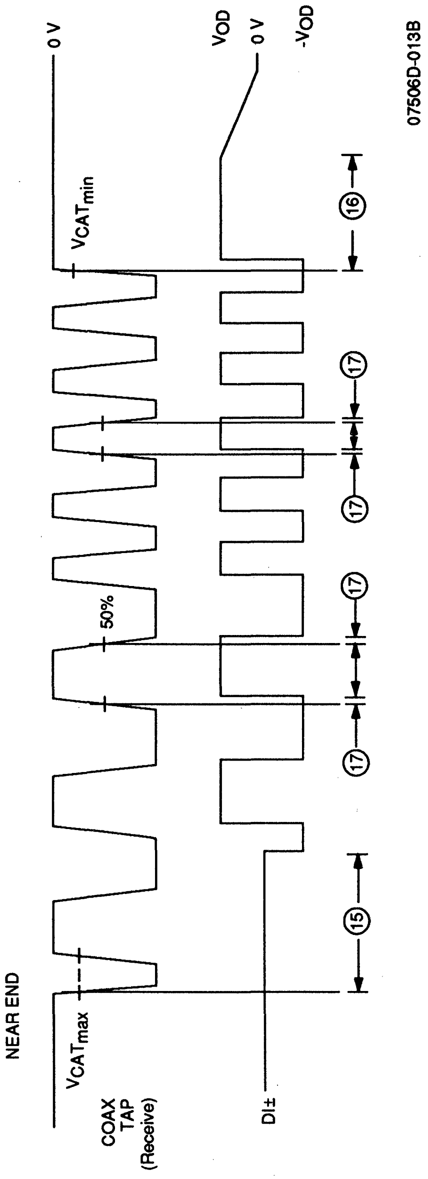
B. Test Load (TXT)

SWITCHING WAVEFORMS



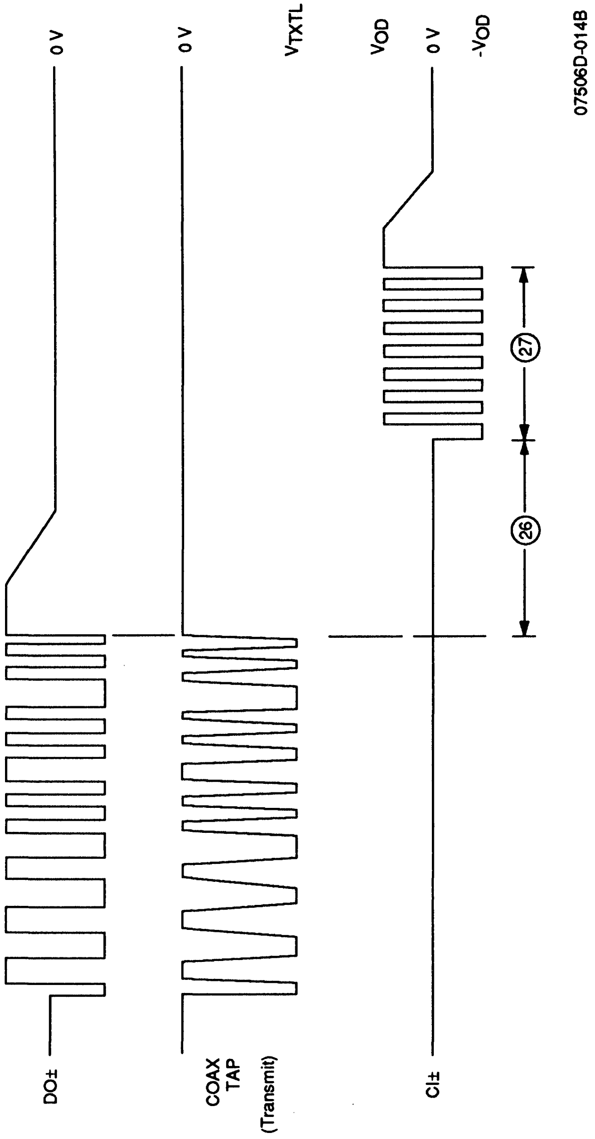
Transmit Function

SWITCHING WAVEFORMS



Receive Function

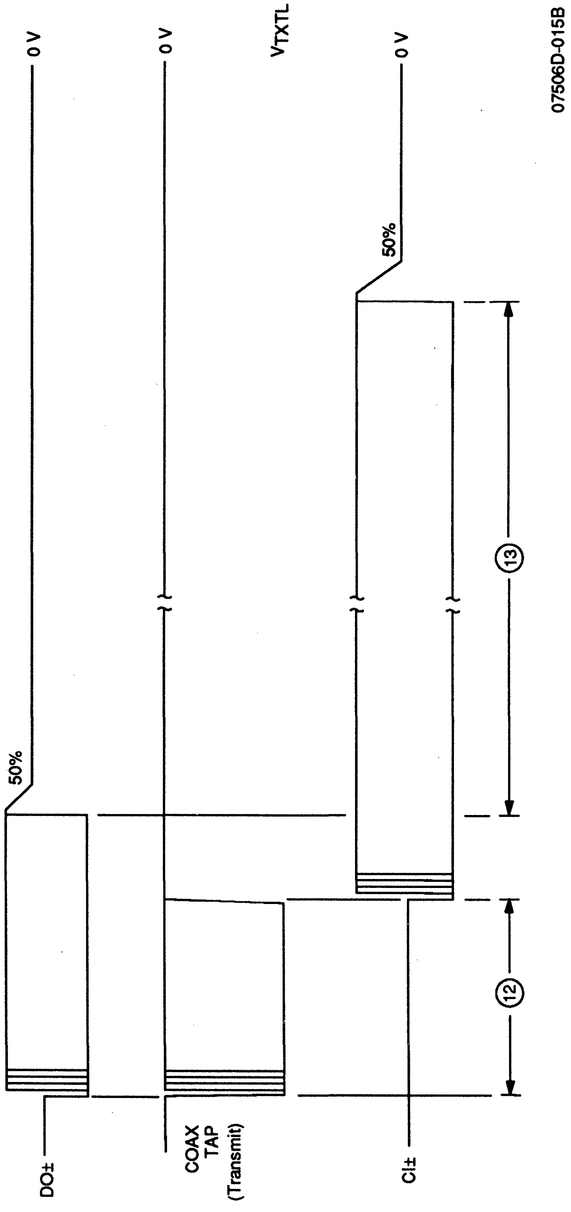
SWITCHING WAVEFORMS



SQE Test*

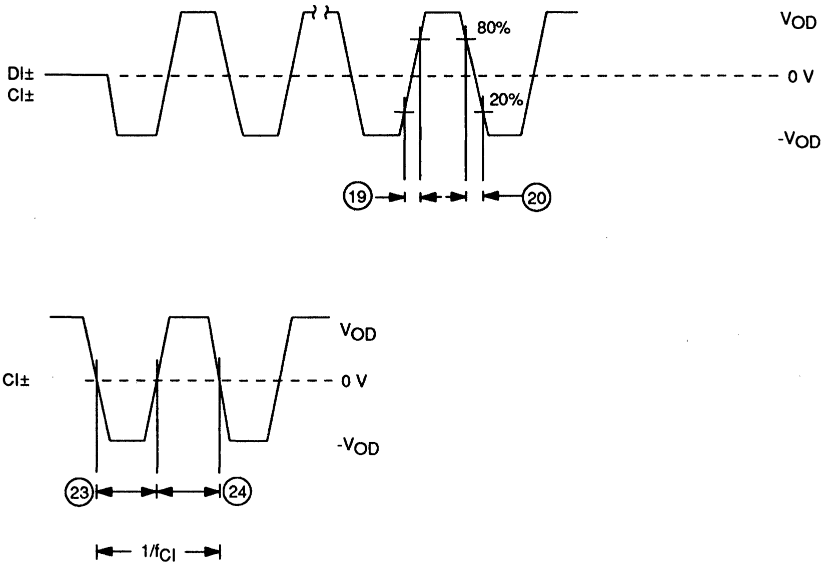
* $\overline{\text{SQE TEST}}$ pin connected to V_{EE}

SWITCHING WAVEFORMS



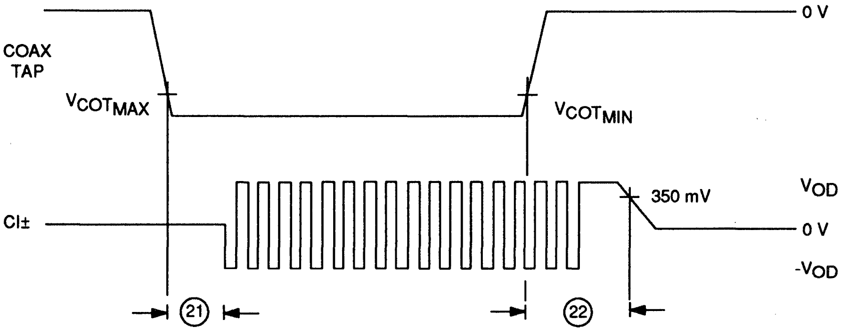
Jabber Function

SWITCHING WAVEFORMS



07506D-016B

DI±/CI± Parameters



07506D-017A

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.

PRELIMINARY

Advanced Micro Devices

Am7997

IEEE 802.3 10BASE5/2 Coaxial Tap Transceiver

DISTINCTIVE CHARACTERISTICS

- **Compliant to ISO 8802-3 ANSI/IEEE-802.3 10BASE5 and 10BASE2 specifications**
- **Pin-selectable SQE Test (Heartbeat) enable/disable**
- **Internal jabber control timer prevents excessive transmission time**
- **Implements noise rejection on AUI DO pair to ensure only valid data is transmitted onto the network**
- **Implements receive mode collision detection**
- **Transmitter average AC output level and collision threshold voltage derived from same reference for close thermal tracking**
- **External component configuration for high reliability network operation**

GENERAL DESCRIPTION

The Am7997 supports the IEEE-802.3 10BASE5/Ethernet® Version 2 and 10BASE2/Cheapernet standards for transceiver applications. This device performs the Medium Attachment Unit (MAU) functions necessary to allow the Data Terminal Equipment (DTE) to perform transmit, receive and collision detection functions over the coaxial media interface.

In IEEE-802.3 10BASE5/Ethernet Version 2 applications, the Am7997 interfaces the "thick" coax media to the DTE remotely through the use of the Attachment Unit Interface (AUI) which permits the MAU to be located up to 50m away from the DTE. To satisfy the MAU electrical isolation requirement, IEEE-802.3 Section 8.3.2.1 specification, the interface between the Am7997 and the AUI is transformer isolated at the MAU. Additional isolation may be implemented by placing a transformer between the DTE and the AUI.

In IEEE-802.3 10BASE2/Cheapernet applications, the Am7997 typically resides within the DTE with its signals transformer isolated and the connection to the "thin" coaxial media can be made directly at the DTE.

In IEEE-802.3 10BASE5 and 10BASE2 applications, the Am7997 power and ground are isolated from that of the DTE.

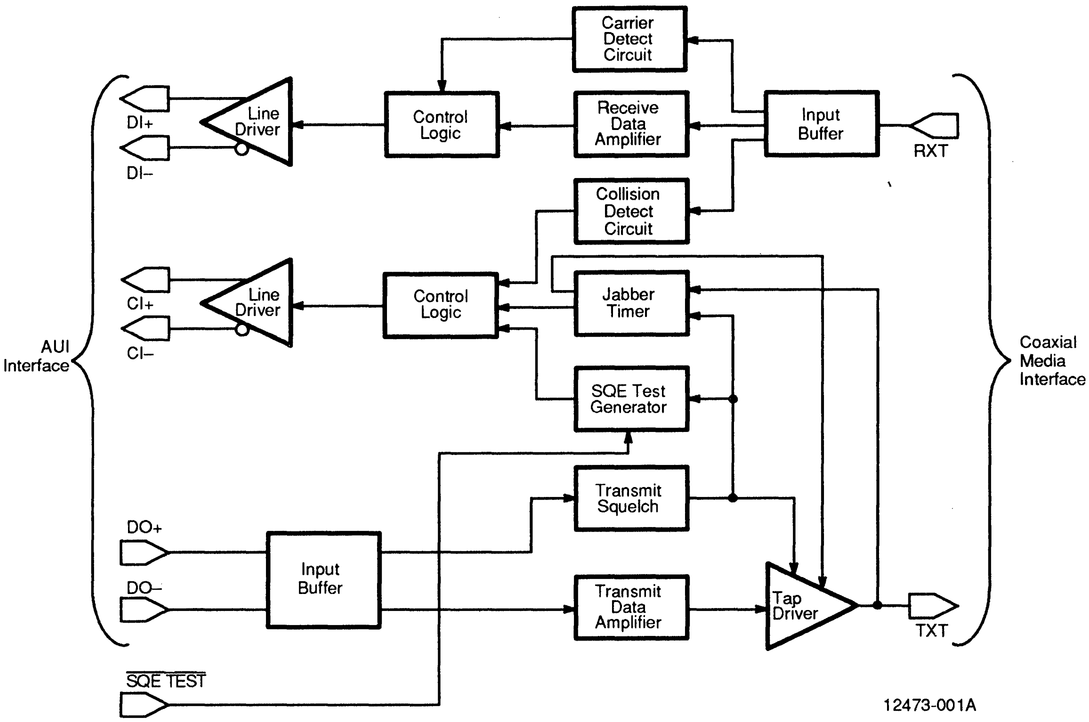
The Am7997 provides controlled transition time skew and current drive for data signaling on the coaxial media and a Jabber Controller that prevents the node from transmitting excessively. Collisions on the media are detected if two or more stations are transmitting.

The Am7997 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 acknowledgment to the node that packet transmission onto the coax was completed.

The external component configuration improves network reliability by safe-guarding the Am7997 from high voltage transients on the coax media.

Publication# 12473 Rev. B Amendment/0
Issue Date: January 1992

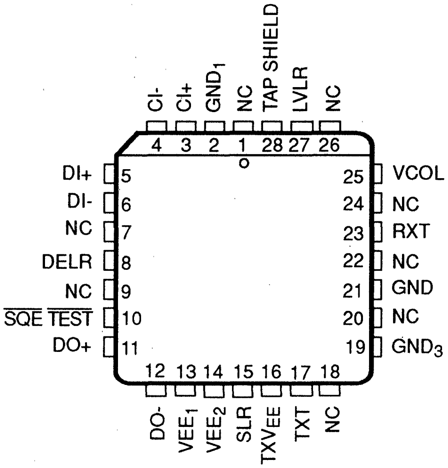
BLOCK DIAGRAM



12473-001A

CONNECTION DIAGRAM
Top View

PLCC



12473-003A

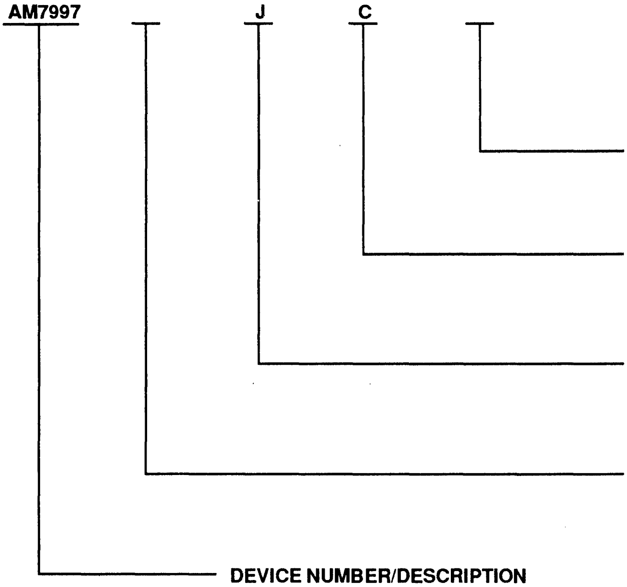
RELATED PRODUCTS

Part No.	Description
Am7990	Local Area Network Controller For Ethernet (LANCE)
Am7992B	Serial Interface Adapter (SIA)
Am79C900	32-bit Integrated Local Area Communications Controller (ILACC)
Am79C940	Media Access Controller for Ethernet (MACE)
Am79C98	Twisted Pair Ethernet Transceiver
Am79C980	Integrated Multiport Repeater

ORDERING INFORMATION

Standard Products

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



OPTIONAL PROCESSING
Blank = Standard processing
TR = Tape and Reel packing

TEMPERATURE RANGE
C = Commercial (0 to + 70°C)

PACKAGE TYPE
J = 28-Pin Plastic Leaded Chip Carrier (PL 028)

SPEED OPTION
Not Applicable

DEVICE NUMBER/DESCRIPTION
Am7997
IEEE-802.3 10BASE5/2 Coaxial Tap Transceiver

Valid Combinations	
AM7997	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, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Attachment Unit Interface (AUI)

DI+, DI–

Data In (Differential Outputs)

These outputs drive transformer-coupled, terminated 78-ohm transmission lines. Received data from the coax tap which meets signal level and pulse width requirements is forwarded to the DTE over DI±. Signaling at DI± meets the requirements of IEEE-802.3.

CI+, CI–

Collision In (Differential Outputs)

These outputs drive transformer-coupled, terminated 78-ohm transmission lines. The 10 MHz signal from the collision oscillator appears at CI± when there is a collision, abnormally long transmission (jabber), or during the Signal Quality Error test at the end of a transmission. Signaling at CI± meets the requirements of IEEE-802.3.

DO+, DO–

Data Out (Differential Inputs)

Data to be transmitted to the network is applied at these inputs. Data signals meeting squelch requirements are wave shaped and output at TXT.

Coaxial Media (Tap) Interface

RXT

Tap Receiver (Input)

The tap drives this input through a 4:1 attenuator composed of a 25 kΩ and a 75 kΩ resistor. Tap signals meeting receiver signal qualification requirements are filtered, amplified, and presented at the DI± outputs.

TXT

Tap Transmitter (Output)

This is a current output driver with controlled, matched rise and fall times. Signals at DO± meeting squelch requirements are wave shaped and presented at TXT. This output drives the media through an isolation network and a low-pass filter.

Tap Shield

Tap Ground Return

This input is the ground return for the tap interface and the collision reference. It makes a single-point connection to the positive power supply.

Circuit Bias and Control

LVLR

Tap Transmitter Output Current Set (Input)

Connecting a 1.96 kΩ resistor between LVLR and Tap Shield sets the tap transmitter for the IEEE-802.3 specified output levels.

$\overline{\text{SQE TEST}}$

Signal Quality Error Test Enable (Input)

The SQE test function is enabled by connecting this input to V_{EE} and disabled by connecting this input to GND.

VCOL

Collision Threshold Reference (Output)

This is the collision sense reference. It is designed to provide a collision threshold level for 802.3 10BASE5 and 10BASE2.

SLR

TXT Waveform Edge Slew Rate Timing Resistor (Input)

An external 2.74K ohm resistor connected to this pin gives TXT waveform 10%–90% transition times between 20 ns and 30 ns.

DELR

Receiver/Transmitter/Collision Oscillator Timing/ Delay Resistor (Input)

An external 237 ohm resistor connected to this pin sets RXT and DO± input pulse width qualification, and CI± and DI± output start-of-idle timing. The internal oscillator frequency is also controlled by this input.

Power Supply Pins

GND_1

Digital Positive Supply

GND_2

Analog Positive Supply

GND_3

Transmit Output Positive Supply

V_{EE2}

Negative Supply

NC

No Connect

No connection of any kind is required at pins labled NC.

TXV_{EE}

Transmitter Output Negative Supply

Driving this pin between V_{EE} + 2.4 V and V_{EE} + 4.0 V disables TXT and causes the 10 MHz internal oscillator signal to appear at CI±.

V_{EE2}

Negative Supply

FUNCTIONAL DESCRIPTION

The Am7997 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 DTE transmissions that are excessive in length.

Transmit

The Am7997 receives differential signals from the DTE (in the case of Am7990 Family applications, from the Am7992B-Serial Interface Adapter-SIA or from the Am79C900-ILACC). For IEEE-802.3 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.

Differential data is received through a squelch network that rejects signals with pulse widths less than 7 ns, or with levels more positive than –175 mv peak. Only signals with pulse widths wider than 20 ns and levels more negative than –275 mv peak from the DTE are guaranteed to be enabled. This minimizes false starts due to noise and ensures no valid packets are missed.

The Am7997's Tap Driver provides the driving capability to ensure adequate signal level at the end of the maximum length network segment (500m 10BASE5, 185m 10BASE2) under the worst-case number of connections (100 nodes 10BASE5, 30 nodes 10BASE2). The required rise and fall times of data transmitted on the network are maintained by the Am7997 Tap Driver. The Tap Driver's output is connected to the medium through external isolating diodes. To safeguard network integrity, the driver is disabled whenever power falls below the minimum operation voltage.

During transmission, the Am7997 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" transmitter. Once disabled, the driver remains disabled for an additional 250–750 ms after the DO pair is idle and there is no carrier sensed from the tap on the TXT pin. During the disable time, the 10 MHz internal oscillator signal is sent on the CI± pair to the DTE.

When $\overline{\text{SQE TEST}}$ is tied to V_{EE}, the Am7997 generates a SQE message at the end of every transmission. This signal is a self-test indication to the DTE that the Medium Attachment Unit (MAU) collision pair is operational.

Receive and Carrier Detect

Received signals are 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 Am7997 receives the signal. The received signal passes through a high-pass filter to minimize inter-symbol distortion, and then through a data slicer. The Am7997 Carrier Detect compares received signals to a reference. Signals meeting carrier squelch criteria are passed 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 in IEEE 10BASE5/Ethernet applications. In IEEE-802.3 10BASE2/Cheapernet implementations the AUI cable is optional. Following the last transition in a packet, the DI± pair is held HIGH for two bit times and then decreases to the idle level within eighty bit times.

Collision Detect

The Am7997 detects collisions if two or more stations are transmitting on the network.

The average DC level of received signals is 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 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, an SQE Test is in progress, or the Jabber Controller is activated. The oscillator is also utilized internally to clock the Jabber and SQE Test timers.

The CI± output meets the drive requirements for the AUI interface. The output stays HIGH for two bit times at the end of the packet, decreasing to the idle level within eighty bit times.

Jabber Function

The Am7997 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 (20–150 ms). A 10 MHz internal oscillator signal is enabled on the CI± pair for the fault duration after the jabber time is exceeded.

After the fault is removed, the Jabber Timer counts the unjab time of 250–750 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 TXV_{EE}. The Am7997 senses this condition and forces a 10 MHz internal oscillator signal on the CI pair during the disable time.

SQE Test

A SQE Test will occur at the end of every transmission if the $\overline{\text{SQE TEST}}$ pin is tied to V_{EE}. A SQE Test signal is a 10 MHz signal gated 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 GND.

APPLICATIONS

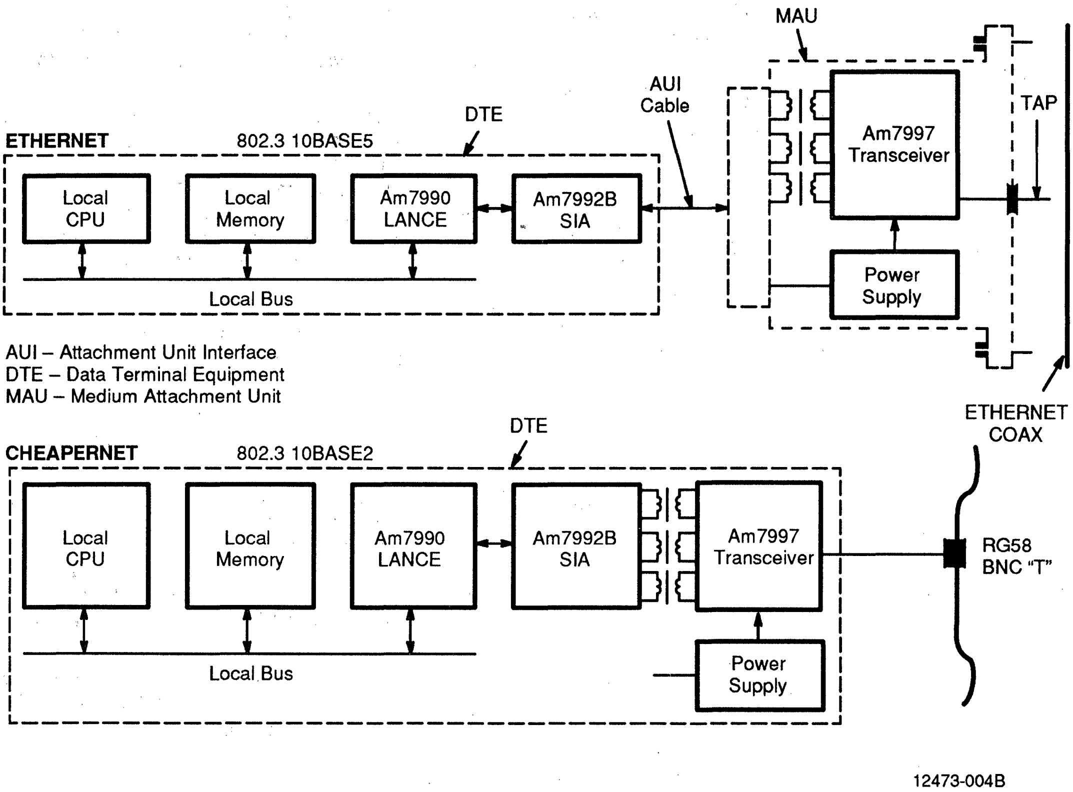


Figure 1.
Typical Ethernet Node and Cheapernet (IEEE-802.3 10BASE5 and 10BASE2)

APPLICATIONS

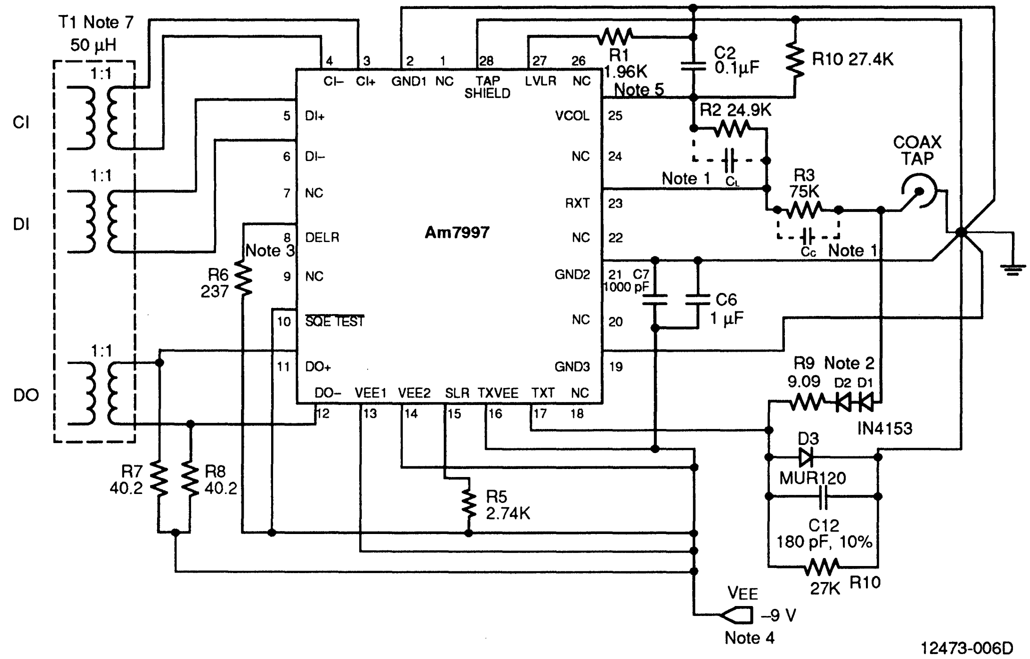


Figure 2.
Am7997 PLCC Package External Component Diagram for an IEEE-802.3 10BASE5/2 Application

Notes:

1. C_L is the effective load capacitance across R_2: C_C is the compensation capacitance ($C_C = 1/3\ C_L$).
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, 10BASE5):	**Cheapernet (IEEE-802.3, 10BASE2):**
Reliability Inc: 2E12R9	Reliability Inc: 2VP5U9
Valor Electronics: PM1001	Valor Electronics: PM7102

5. Minimizes excess stray capacitance in PC layouts with noisy power supplies.
6. All capacitors have 20% tolerance except C_{12} which has 10% or better. All resistors are 1% tolerance.
7. Isolation transformer module manufacturers include:

   Valor Electronics: LT6031
   Pulse Engineering: PE64101
8. For Heat Sink Design using the Am7997 in Repeater Applications. From the Am7997 characterization data, we have the following:

   θ_{JA} (non-heat sinked) = 52°C/W
   $\theta_{JC} = 11°C/W$

   Worst case Junction Temperature occurs under the following condition:

   $I_{EE} = -190$ mA $T_A = 70°C$
   $V_{EE} = -9.9\ V_{DC}$

   This implies: (Power Dissipation) $P_D = I_{EE} * V_{DC}$
   $= (-190\text{ mA}) * (-9.9\ V_{DC})$
   $= 1881$ mW

Now (Junction Temperature) $T_J = [(\theta_{JA}) * (P_D)] + T_A$
Therefore $T_J = [(52°C/W) * (1.881\text{ W})] * 70°C$
$= 169°C$

At this point, the goal is to reduce the T_J to less than 150°C. To do this, a heat dissipation plane is recommended. The effect of this plane is to reduce the value of θ_{JA} which is defined to be ($\theta_{JC} + \theta_{HS}$), where θ_{JC} is the Thermal Resistance from the Junction to the Case and θ_{HS} defined as Thermal Resistance from Case to Ambient (θ_{CA} or θ_{HS}).

Substituting, we have the following equation:

$$T_J = [(\theta_{JA}) * (P_D)] + T_A \le 150°C$$

or

$$[(\theta_{JC}) + \theta_{HS}) * P_D] + T_A \le 150°C$$

$$\theta_{JC} + \theta_{HS} \le \frac{150 - T_A}{P_D}$$

$$\theta_{HS} \le \frac{150 - T_A}{P_D} - \theta_{JC}$$

Thus under worst case Junction Temperature conditions and substituting the values above, we have:

$$\theta_{HS} \le \frac{150 - 70}{1.881} - 11$$

$$\theta_{HS} \le 31.5°C/W$$

Therefore, the heat dissipation plane is recommended to have a $\theta_{HS\ max}$ (or θ_{CA}) rating of 31.5°C/W to ensure $T_J \le 150°C$. In conclusion, θ_{HS} determines the size of the heat sink to be selected. As θ_{HS} decreases, the size of the heat sink increases. Hence, the above value determines the minimum size of the heat sink to be selected.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	−65 to +150°C
Ambient Temperature Under Bias, Package	0 to 70°C
Lead Temperature, Soldering For 10 Seconds	+300°C
Supply Voltage	−12.0 to +0.5 V
DC Input Voltage (RXT)	−6.0 to +0.5 V
DC Input Voltage (All other inputs)	−12.0 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° 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 COMMERCIAL operating range unless otherwise specified

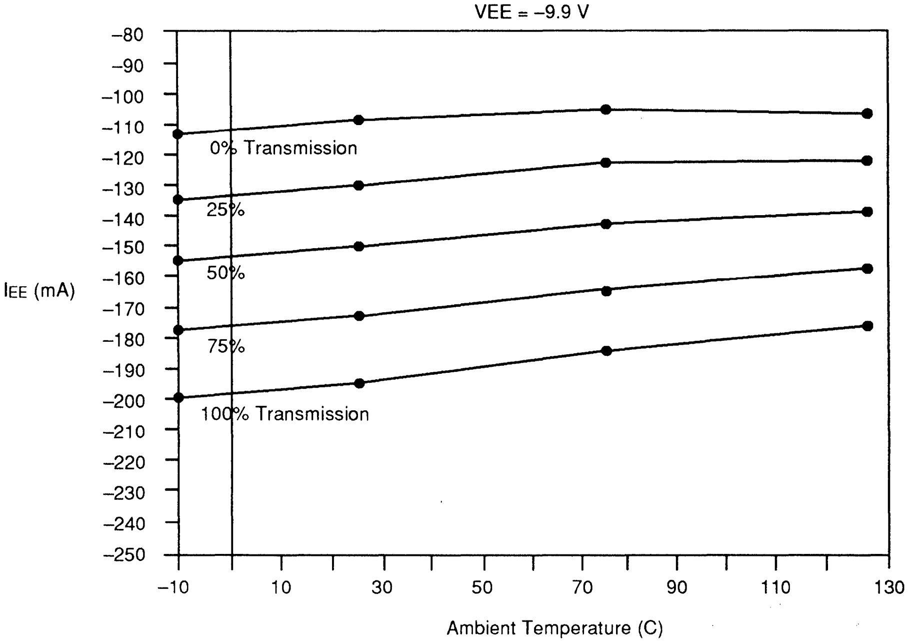
Parameter Symbol	Parameter Description	Test Conditions		Commercial		Unit
				Min.	Max.	
Transmit Signals						
V_{TH}	DO± differential input switching threshold			−160	+160	mV*
V_{SQ}	DO± differential input squelch threshold			−275	−175	mV
V_{ICM}	DO± open circuit input common mode bias voltage	I_{IN} = 0		V_{EE} +0	V_{EE} +4.0	V
I_{TXTL}	TXT output LOW current	R_L = 25.0 Ω (Note 2)		−90	−65	mA*
I_{TXTH}	TXT output HIGH current	R_L = 25.0 Ω (Note 2)		−17	0.0	mA*
I_{TXTA}	TXT output average DC current	R_L = 25.0 Ω (Notes 1, 2)		−45	−37	mA*
I_{ID}	Input current (DO+ or DO−)	$V_{EE} < V_{IN} < 0$ V All V_{EE} = −9.9 V		−2.0	+2.5	mA
I_{T10}	TXT output current for VTAP = −10 V (While transmitting)	(Note 4)		−250	+250	µA*
I_{Tx}OFF	TXT idle-state leakage current			−23	0.0	µA*
R_{DO}	DO± differential input resistance	VDO+ = −6.95 V VDO− = −7.05 V		6		kΩ
Receive/Collision Signals						
V_{OD}	DI± & CI± differential output voltage	Load R_L = 78 Ω L = 50 µH	V_{OD+}	+450	+1315	mV*
			V_{OD-}	−1315	−450	
V_{ODI}	DI± & CI± differential output imbalance voltage	Load R_L = 78 Ω L = 50 µH		−25	+25	mV
V_{OD}OFF	DI± & CI± differential idle output voltage	Load R_L = 78 Ω L = 50 µH		−40	+40	mV*

DC CHARACTERISTICS (Continued)

Parameter Symbol	Parameter Description	Test Conditions	Commercial		Unit
			Min.	Max.	
V_{OCM}	DI± & CI± common mode output voltage	R_L = 2 x 39 Ω series resistors	–2.5	–1.3	V
V_{US}	DI± & CI± undershoot at start-of-idle	R_L = 78 Ω L = 50 µH		–100	mV*
V_{CAT}	RXT carrier sense	Notes 4, 5	–600	–380	mV
V_{COT}	Collision threshold voltage	Note 15	–1582	–1448	mV*
V_{IH}	SQE TEST input HIGH voltage		–0.5	0.0	V
V_{IL}	SQE TEST input LOW voltage		V_{EE}	V_{EE}+0.5	V
I_{RXT}	RXT input bias current	V_{RXT} = –1 to –2.5 V	–0.5	+0.5	µA*
I_{OD}OFF	DI± & CI± differential idle output current threshold	R_L=78 Ω L = 50 µH	–4	+4	mA*
Global					
I_{EE}	Supply current – Non-Transmitting	total current @ all V_{EE} pins @ 70°C		–121	mA
	Supply current – Transmitting	R_{LX} = 25 Ω with all V_{EE} @ –9.9 V		–211	mA

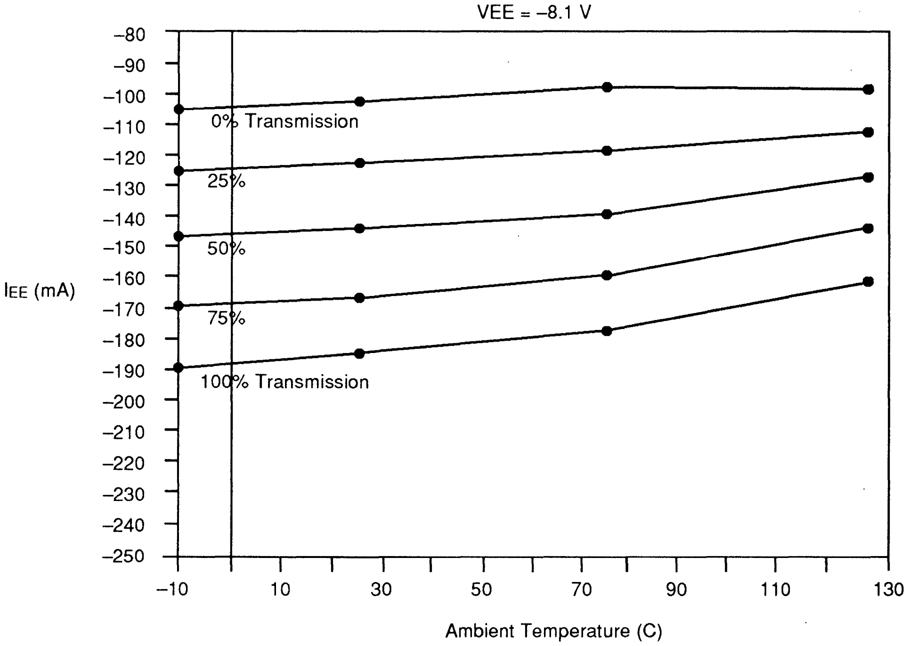
CAPACITANCE
UNTESTED; TYPICAL values at T_A = 25°C, V_{EE} = 0, All NC pins unconnected

Parameter Symbol	Parameter Description	Test Conditions	Typ	Unit
C_{RXT}	RXT input capacitance		1.1	pF



12473-018B

Typical IEE vs. Temperature at % of Transmission



12473-022A

Typical IEE vs. Temperature at % of Transmission

AC CHARACTERISTICS over COMMERCIAL operating ranges unless otherwise specified

Transmit

Parameter			Commercial		
Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
t_{PWREJ}	Transmit reject pulse pulse width (DO±)	(Note 6)		7	ns
t_{PWTON}	Transmit turn-on pulse width (DO±)	DO± = 600 mVp-p (Note 6)	20		ns
t_{PWOFF}	Transmit turn-off pulse width (DO±)	DO± = 600 mVp-p (Note 6)	130	200	ns
t_{TON}	Transmit start up delay (DO± to TXT)	DO± = 600 mVp-p (Notes 6, 7, 13)		200	ns*
t_{TSD}	Transmit static propagation delay (DO± to TXT)	DO± = 600 mVp-p (Notes 6, 7, 14)		50	ns*
t_{TR}	Transmitter rise time (10% to 90% at TXT)	(Notes 6, 7, 14)	20	30	ns*
t_{TF}	Transmitter fall time (90% to 10% at TXT)	(Notes 6, 7, 14)	20	30	ns*
t_{TM}	Transmitter rise and fall time mismatch (TXT; $t_{TR} - t_{TF}$)	(Notes 6, 7, 14)	–2.0	+2.0	ns*
t_{TS}	Transmitter skew (TXT waveform symmetry)	(Notes 6, 7, 14)	–2.0	+2.0	ns*
t_{JA}	Transmit jabber activation time		20.0	150	ms*
t_{JR}	Transmit jabber reset time	(Note 18)	250	750	ms*
t_{JREC}	Transmit jabber recovery time (minimum time gap between transmitted packets to prevent jabber activation)		1.0		μs
L2, 3	2nd and 3rd harmonics of TXT Output	(Note 17)	–20		dB*
L4, 5	4th and 5th harmonics of TXT Output	(Note 17)	–30		dB*
L6, 7	6th and 7th harmonics of TXT Output	(Note 17)	–40		dB*
L8	All higher harmonics of TXT Output	(Note 17)	–50		dB*

AC CHARACTERISTICS (Continued)

Receive/Collision

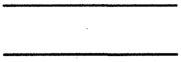
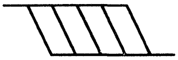
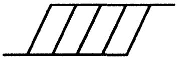
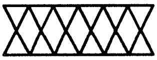
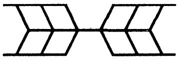
Parameter Symbol	Parameter Description	Test Conditions	Commercial Min.	Commercial Max.	Unit
t_{RON}	Receiver start up delay (coax tap to DI ±)	(Notes 6, 9, 16)		500	ns*
t_{RVB}	First validly timed bit on DI ±	(Notes 6, 9, 14, 16)		t_{RON}	ns*
t_{RSD}	Receiver static propagation delay (coax tap to DI ±)	(Notes 6, 9, 15, 16)		50	ns*
t_{ROFF}	Receiver turnoff delay (coax tap to DI ±)	(Notes 6, 10)	200	1000	ns*
t_{RR}	DI ± and CI ± rise time (20% to 80% of full differential swing)	(Note 6)		7.0	ns*
t_{RF}	DI ± and CI ± fall time (80% to 20% of full differential swing)	(Note 6)		7.0	ns*
t_{RS}	Receiver skew (DI ± waveform symmetry)	(Notes 6, 15, 16)	–3.5	+3.5	ns*
t_{SETL}	DI ± and CI ± settling time to idle state (± 40 mV)	(Note 6)		8	µs*
t_{RM}	DI ± & CI ± rise and fall time mismatch (t_{RR} – t_{RF})	(Note 6)	–3.5	+3.5	ns*
t_{RJIT}	Receiver jitter	(Notes 6, 8)	–8.0	+8.0	ns*
t_{CON}	Collision turn-on delay (CI ±)			900	ns*
t_{COFF}	Collision turn-off delay (CI ±)			2000	ns*
t_{PER}	Collision period (CI ±)	(Note 11)	80	117	ns*
t_{CPW}	Collision output pulse width (CI ±)	(Note 11)	40	60	ns*
t_{SQED}	SQE test delay time			900	ns*
t_{SQEL}	SQE test length		500	1500	ns*

Notes:

* Indicates 802.3 10BASE5/2 specification requirement.

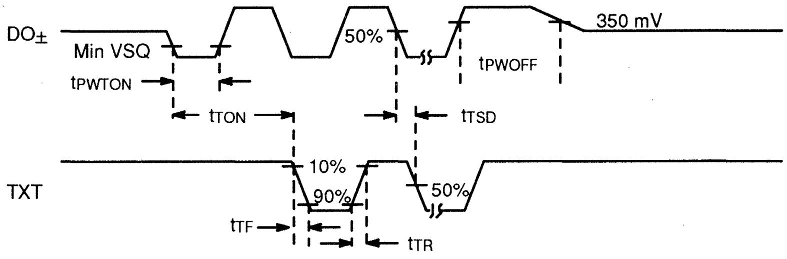
1. Measured at 50% point of output waveform.
2. Measured at both 5 MHz and 10 MHz TXT output frequencies.
4. Measured at the coax tap using Tap Application Circuit (Figure 2).
5. Measured with transmitter idle.
6. Measured using Tap Application Circuit (Figure 2).
7. Measured using Tap Application Circuit and a forcing source at DO± with 1200 mV peak-to-peak amplitude, less than 0.1 ns of skew or jitter, and maximum 10%–90% edge transition times of 7.0 ns.
8. Forcing source at tap has a 600 mV peak-to-peak swing centered at –1.0 V, between 7.0 ns and 8.0 ns of edge jitter, and 30.0 ns +0, –2 ns 10%–90% edge transition times.
9. Measured from the 50% point of the forcing source waveform at the tap to the differential zero crossing at DI±.
10. Receiver turnoff point is at 70% of VOD on last high-to-low transition of DI±.
11. Period and pulse width will fall in these ranges using a 1% resistor shown in Tap Application Circuit (Figure 2).
12. Forcing source frequency is 10 MHz.
13. Forcing source frequency is 5 MHz.
14. Measured with forcing source frequencies of both 5 MHz and 10 MHz.
15. V_{COT} is the average DC level of a 5 MHz 400 mVp-p pulse train applied at the coax tap that causes CI± to go active.
16. Forcing source at tap has a 1.625 Vp-p swing centered at –1.0 V, no more than ±0.1 ns of edge jitter and 25.0 ns ± 2.0 ns 10% – 90% edge transition times.
17. To be met while transmitting 10 MHz signals. Correlated to t_{TR} and t_{TF}; not tested.
18. Reset time starts only after both D0± and the coax tap become inactive (idle).

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

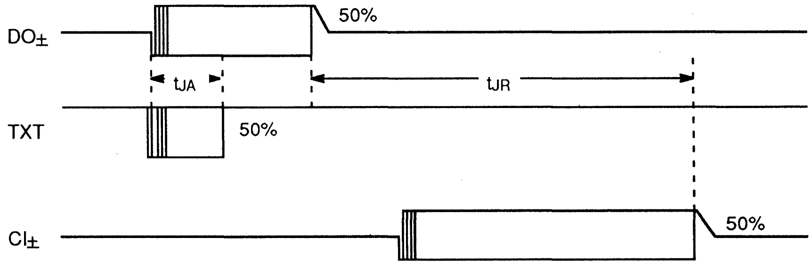
KS000100

SWITCHING WAVEFORMS



12473-007A

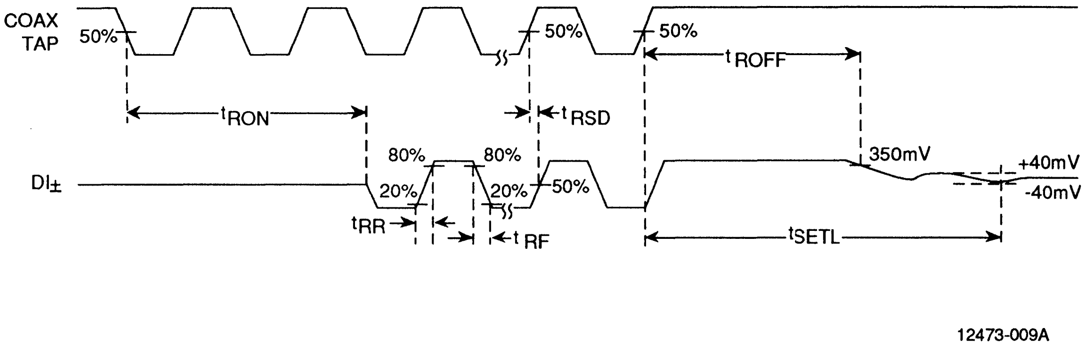
Transmitter Timing



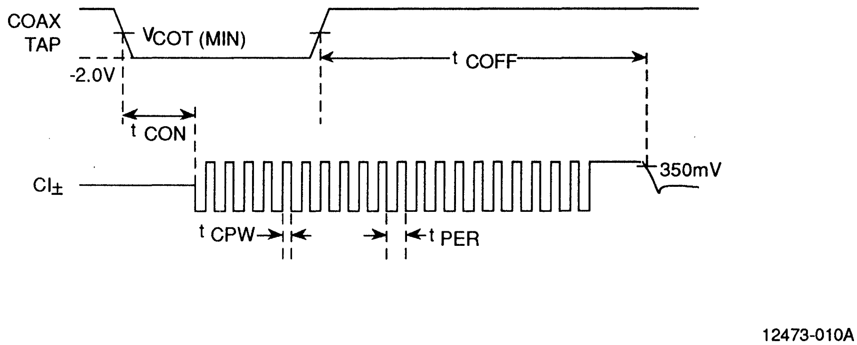
12473-008A

Transmitter Jabber Timing

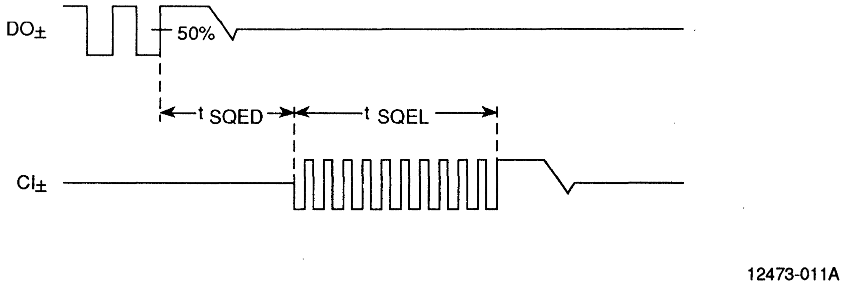
SWITCHING WAVEFORMS



Receiver Timing

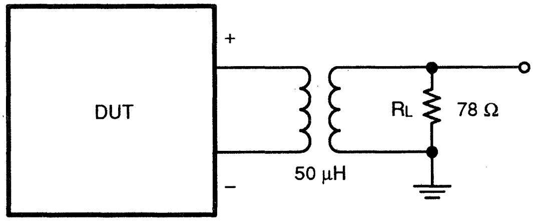


Collision Timing



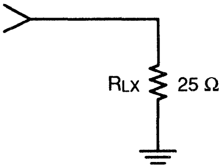
SQE Test Timing (SQE Test pin connected to V_{EE})

SWITCHING TEST CIRCUITS



12473-012A

A. AUI Transmit Load (DI +, DI–; CI+, CI–)



12473-013A

B. Test Load (TXT)

LAYOUT CONSIDERATIONS

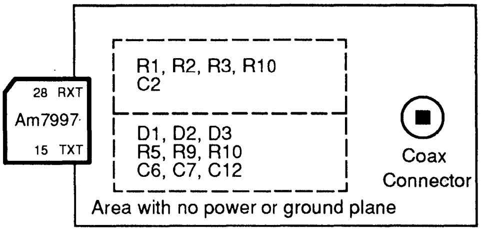
To protect the transceiver from the environment and to achieve optimum performance, the Am7997 is designed to be used with two sets of external components: the transmitter circuit consisting of components D1, D2, D3, R5, R9, R10, C6, C7, and C12, and the receiver circuit consisting of components R1, R2, R3, C2, C_L, and C_C (C_L is a parasitic capacitance rather than a discrete component). These combined circuits are shown in Figure 2. The resistor tolerances for these circuits are specified as 1%.

The primary layout restriction for the transmitter circuit is that the longest current path from the TXT pin (PLCC pin 17) to the coaxial cable's center conductor must be as short as possible. Also, the transmitter circuitry should be physically isolated from power and ground planes, i.e. no power or ground planes under the transmitter circuit components.

The layout of the receiver circuit is even more critical. To minimize parasitic capacitance that can degrade the received signal, the external receiver circuit should also be physically isolated from power and ground planes. There must be no power or ground plane under the area of the P.C. board that includes PLCC pins 22 through 1, R2, R3, 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 R2–R3 attenuator. Also, the RXT pin should be as close as possible to the coaxial cable connector.

The layout of the receiver and transmitter circuits can be simplified by omitting power and ground planes from the whole area on the right side of the Am7997 as shown in Figure 3-1. Also, the TAP SHIELD pin (PLCC pin 28) should be connected directly to the shield of the coax connector without any other P.C. board ground trace hanging off of it.

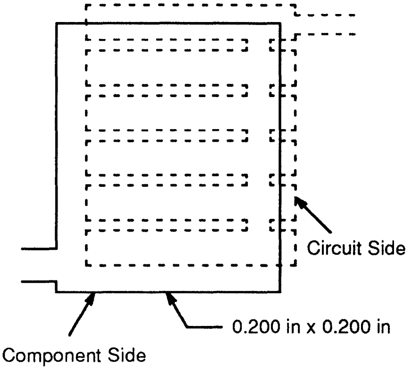
If the above layout rules are followed, the parasitic capacitance in parallel with R2 will be about 6 pF. This parasitic capacitance is shown in the schematics as C_L (Note that C_L is a parasitic capacitance. Do not add a discrete capacitor in parallel with R2). The capacitor labeled C_C in the schematics is the total capacitance in parallel with R3 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 R3 to R2, which is 3:1. This means that an additional 1 pF of capacitance must be added in parallel with R3.



12473-014C

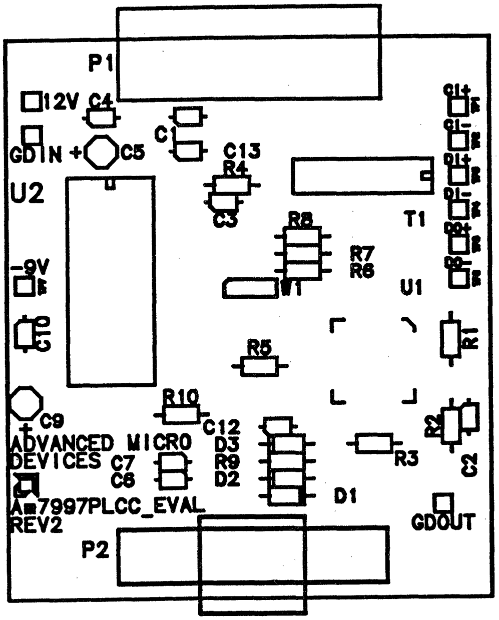
Figure 3-1

This extra capacitance can easily be added by building a capacitor out of P.C. board traces connected to R3. Since the amount of extra parasitic capacitance required is affected by board layout, components, and P.C. board material, it is recommended that during prototyping a 'trimmable' capacitor be used. One example of this is a 0.200 x 0.200 inch parallel plate capacitor on two layers of the P.C. board as shown in Figure 3-2. Note that one of the parallel plates is segmented so that sections of the plate may be trimmed off until the correct amount of capacitance remains. The resulting capacitor configuration may then be reproduced on the production P.C. boards.



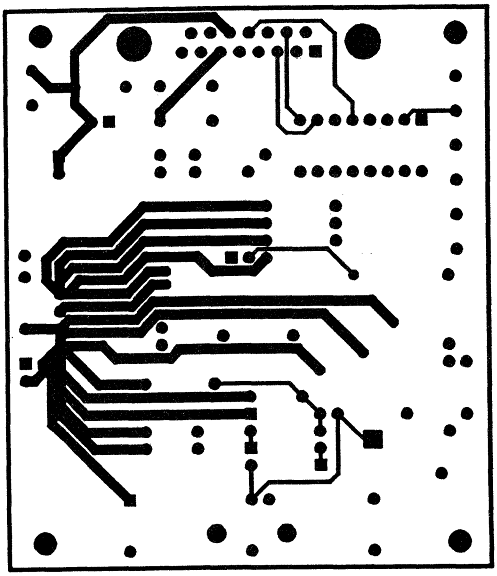
12473B-023A

Figure 3-2



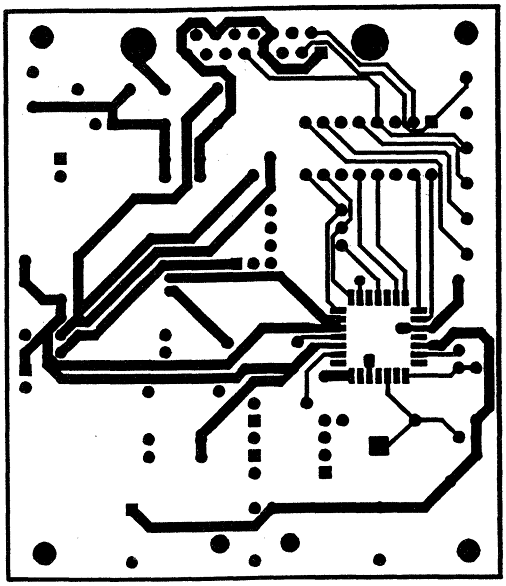
12473-019A

Component Side Silkscreen



12473-020A

Solder Side



12473-021A

Component Side

C1	.001µF GAPKAP
C2	0.1µF
C3	1000pF 3KV
C4	0.1µF
C5	4.7µF
C6	1µF
C7	1000pF
C9	4.7µF
C10	0.1µF
C11	NOT USED
C12	180pF
C13	0.01µF 50V
D1	1N4153
D2	1N4153
D3	MUR120
P1	15-Pin D Connector Male
P2	BNC Connector
R1	1.96KΩ 1%
R2	24.9KΩ 1%
R3	75.0KΩ 1%
R4	1MΩ 5% 0.5W
R5	2.74KΩ 1%
R6	237Ω 1%
R7	40.2Ω 1%
R8	40.2Ω 1%
R9	9.09Ω 1%
R10	27.4KΩ 1%
T1	LT6031 Pulse Transformer
U1	Am7997JC
U2	PM7005 DC-DC Converter
W1	3-Pin Jumper Block

(All Resistors are 1% except where noted)

Figure 4. Suggested Printed Circuit Board Layout for a Double-Sided PCB Application

PRELIMINARY

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 (Draft 10)**

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.

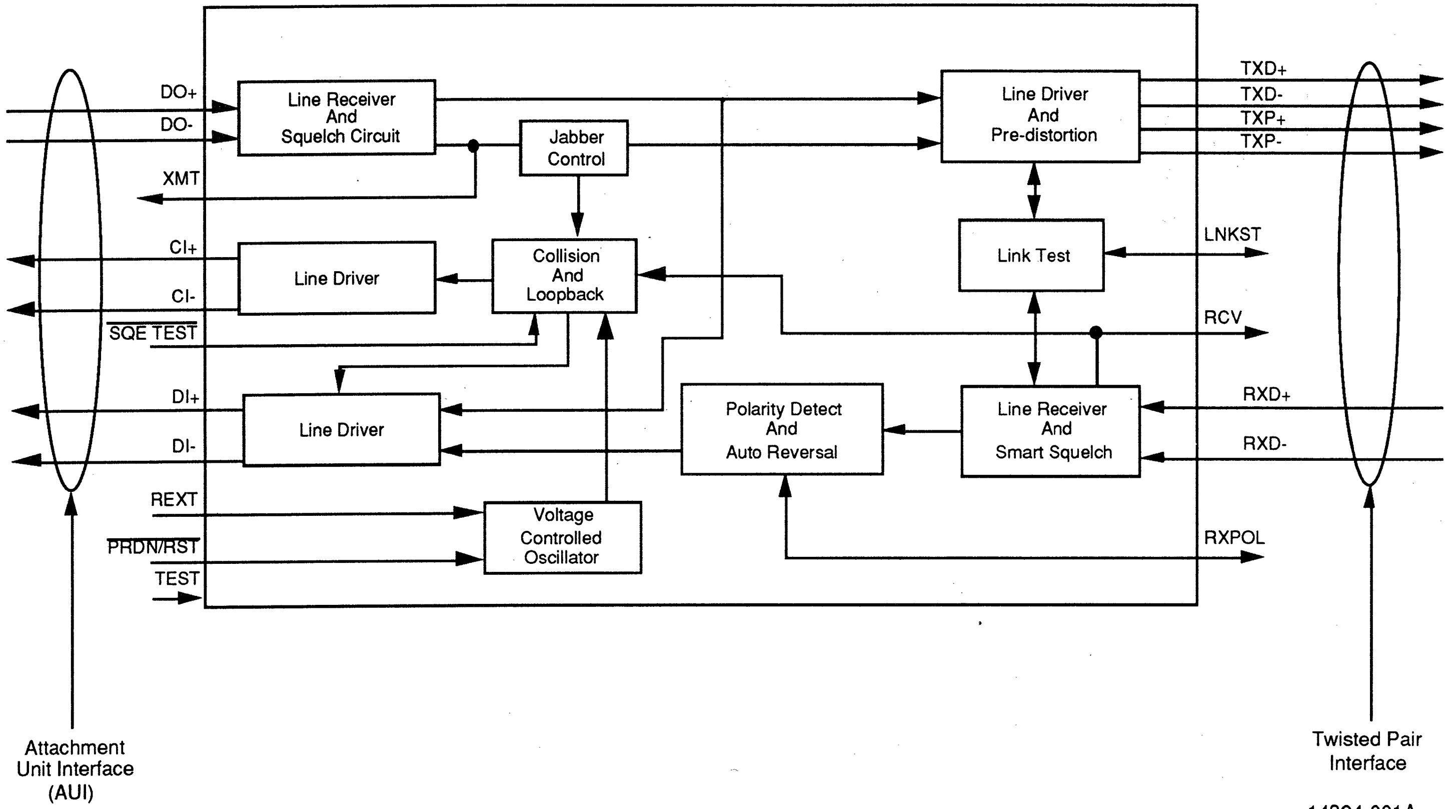
A network based on the 10BASE-T standard can use unshielded twisted pair cables, therefore providing an economical solution to networking by allowing the use of existing telephone wiring. The Am79C98 provides a minimal component count and cost effective solution to the design and implementation of 10BASE-T 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 +5V supply. The device is available in 24 pin SKINNYDIP® Plastic Dual-in-Line and 28 pin Plastic Leaded Chip Carrier (PLCC).

Publication # 14395 Rev. C Amendment/0
Issue Date: January 1992

BLOCK DIAGRAM


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

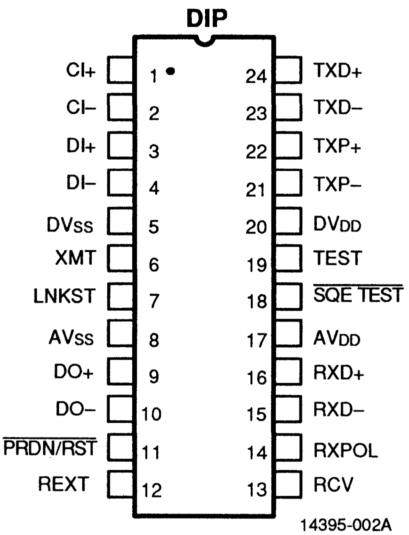
14394-001A

RELATED AMD PRODUCTS

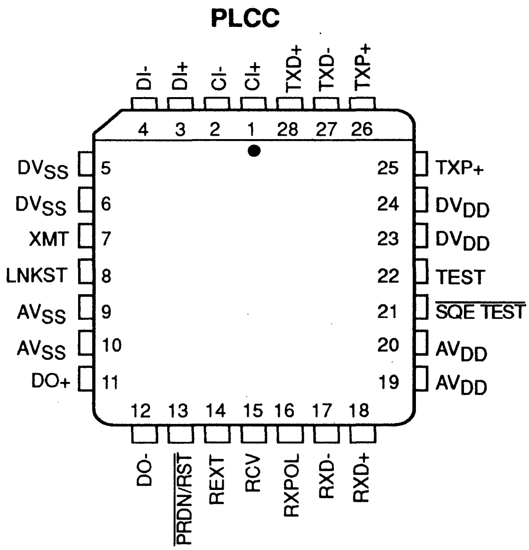
Part No.	Description
Am79C900	Integrated Local Area Communications Controller (ILACC)
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am7992B	Serial Interface Adapter (SIA)
Am79C980	Integrated Multiport Repeater for 10BASE-T (IMR)
Am7996	IEEE-802.3/Ethernet/Cheapernet Transceiver
Am79C940	Media Access Controller for Ethernet (MACE)

CONNECTION DIAGRAM

Top View



14395-002A

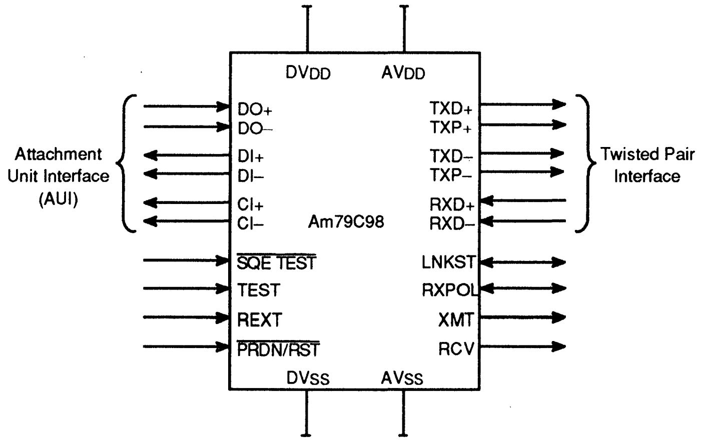


Note:

Pin 1 is marked for orientation

14395-003A

LOGIC SYMBOL

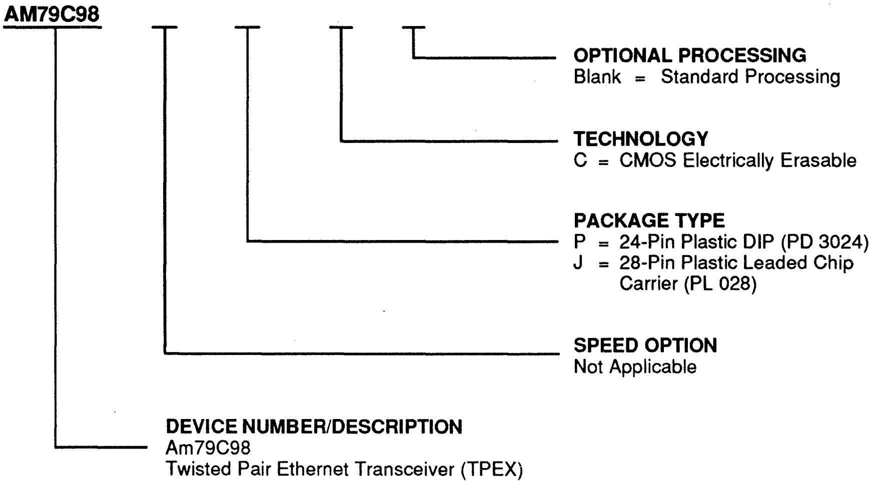


14395-004A

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:



AM79C98

OPTIONAL PROCESSING
Blank = Standard Processing

TECHNOLOGY
C = CMOS Electrically Erasable

PACKAGE TYPE
P = 24-Pin Plastic DIP (PD 3024)
J = 28-Pin Plastic Leaded Chip Carrier (PL 028)

SPEED OPTION
Not Applicable

DEVICE NUMBER/DESCRIPTION
Am79C98
Twisted Pair Ethernet Transceiver (TPEX)

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, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

AM79C98	PC, JC

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.

$\overline{\text{PRDN/RST}}$
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 AVDD, 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 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 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 5 to 6 consecutive Link Test pulses of identical polarity. On entry to the Link Pass state, the polarity of the last 5 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 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 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 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:

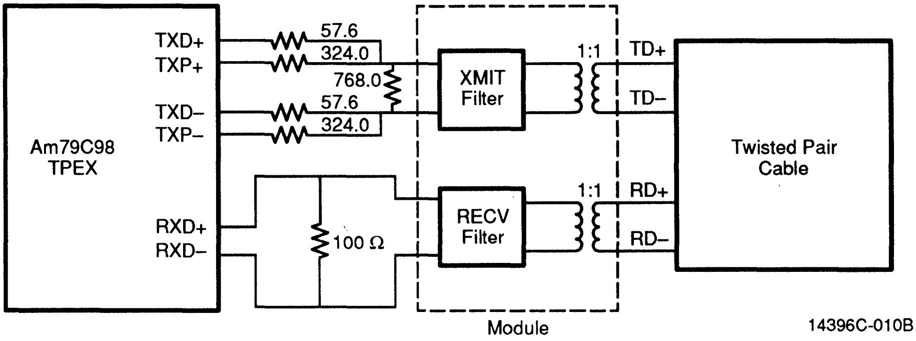
1. **Station (DTE):** $\overline{\text{SQE TEST}}$ pin LOW. Data on DO+/– pair is transmitted onto the TXD+/– and TXP+/– pairs and data on the RXD+/– input pair is transmitted onto the DI+/– output pair. The jabber function and collision detection function are disabled.

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.

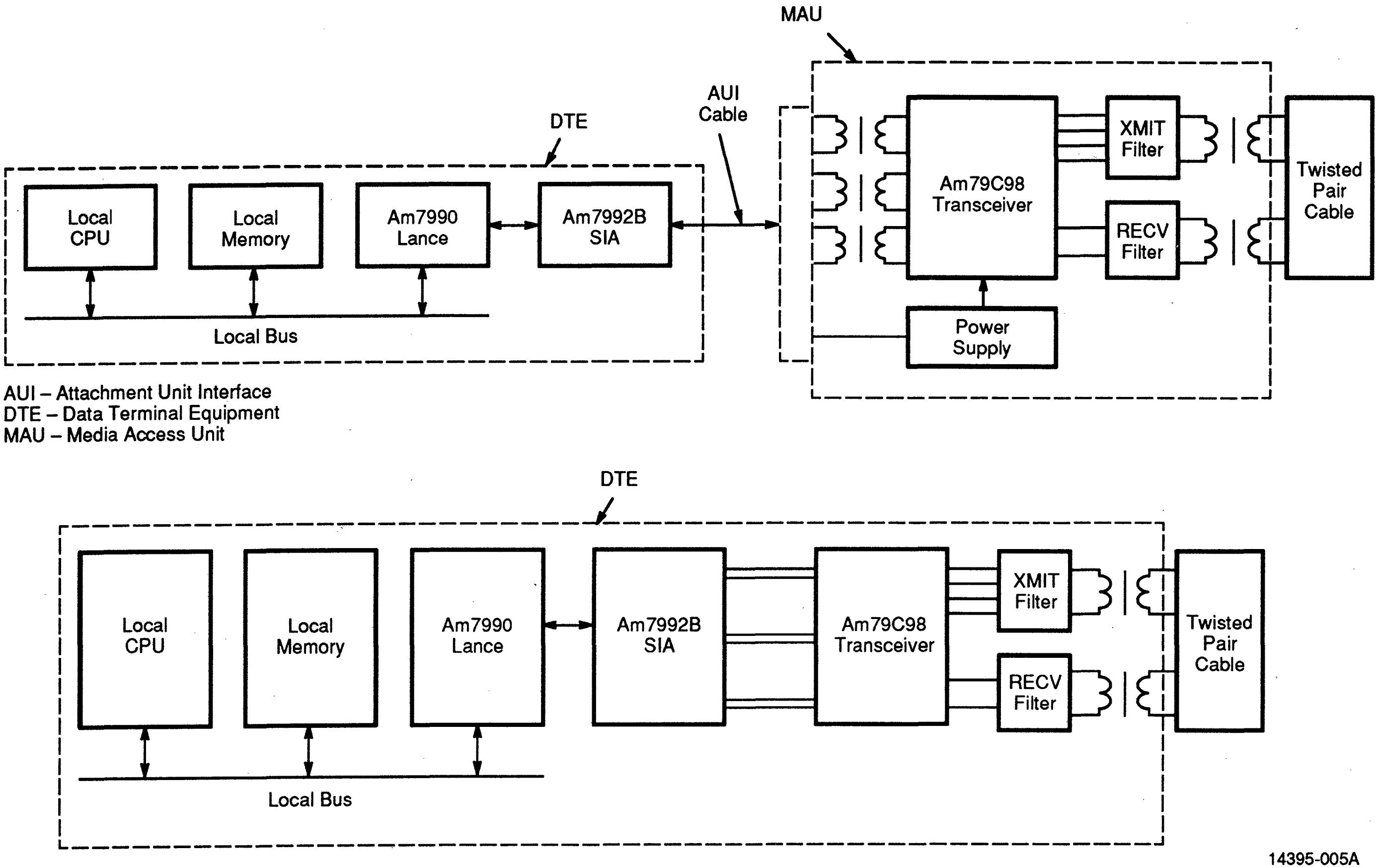


The Filter/Transformer Module shown is available from the following manufacturers:

Belfuse
Pulse Engineering
Valor Electronics

TDK
PCA
Nano Pulse

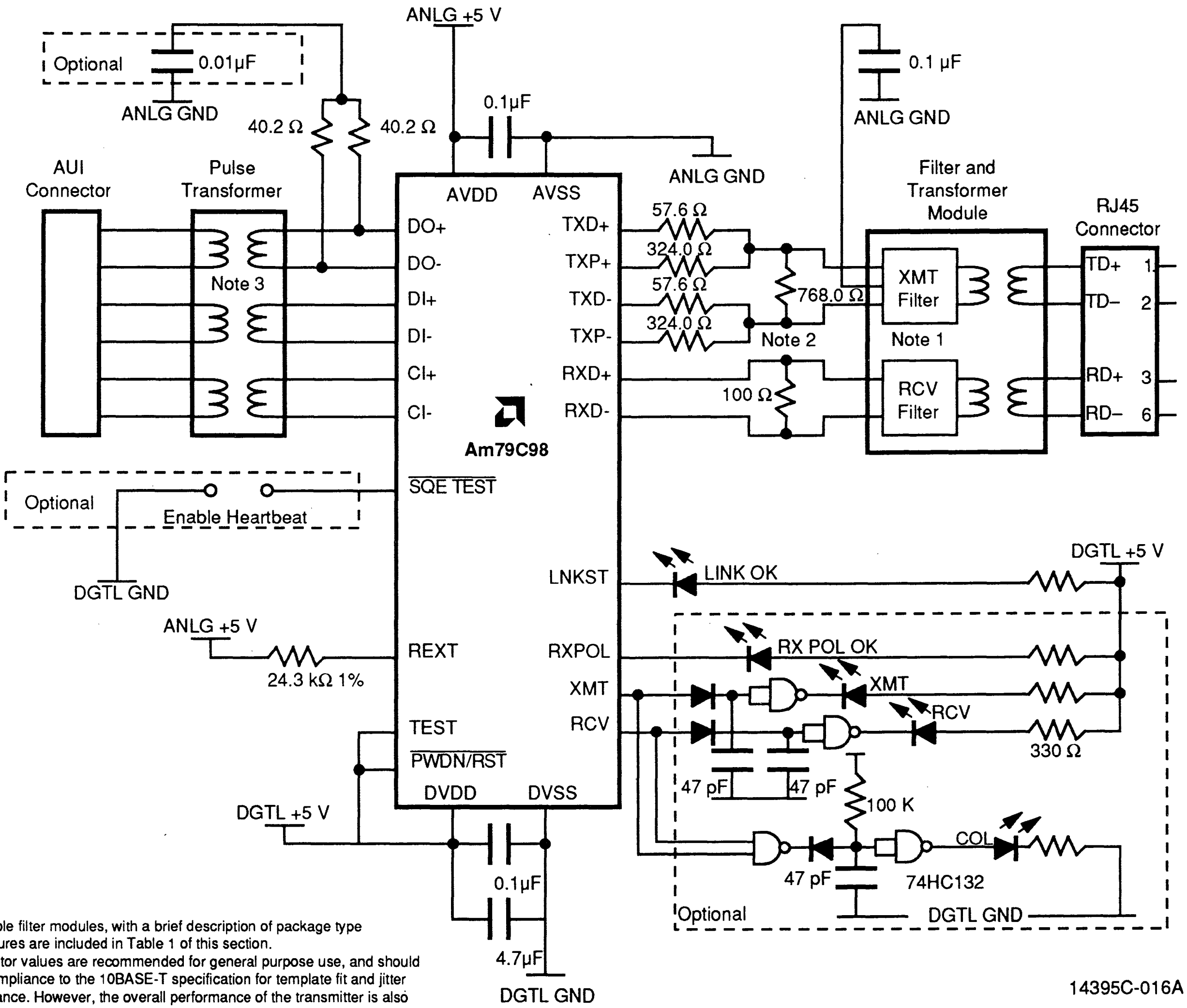
Figure 1. Typical TP Port External Components



AUI – Attachment Unit Interface
DTE – Data Terminal Equipment
MAU – Media Access Unit

14395-005A

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 to +150°C
Ambient Temperature Under Bias:	0 to +70°C
Supply Voltage to AV_{SS} or DV_{SS} (AV_{DD}, DV_{DD}):	–0.3 to +6V

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 Voltages (AV_{DD}, DV_{DD}):	+5V ± 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	uA
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_{ddl} = 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_{AOD}OFF	DI+/- & CI+/- Differential Idle Output Voltage	R_L = 78 Ω	-40	+40	mV
I_{AOD}OFF	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_{RXD}	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_{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 operating ranges

Parameter Symbol	Parameter Description		Min.	Max.	Unit
Transmit Timing					
t_{PWODO}	DO Pulse Width Accept/Reject Threshold	$\lvert V_{IN}\rvert > \lvert V_{ASQ}\rvert$ (Note 3)	15	35	ns
t_{PWKDO}	DO Pulse Width Maintain/Turn-Off Threshold	$\lvert V_{IN}\rvert > \lvert V_{ASQ}\rvert$ (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

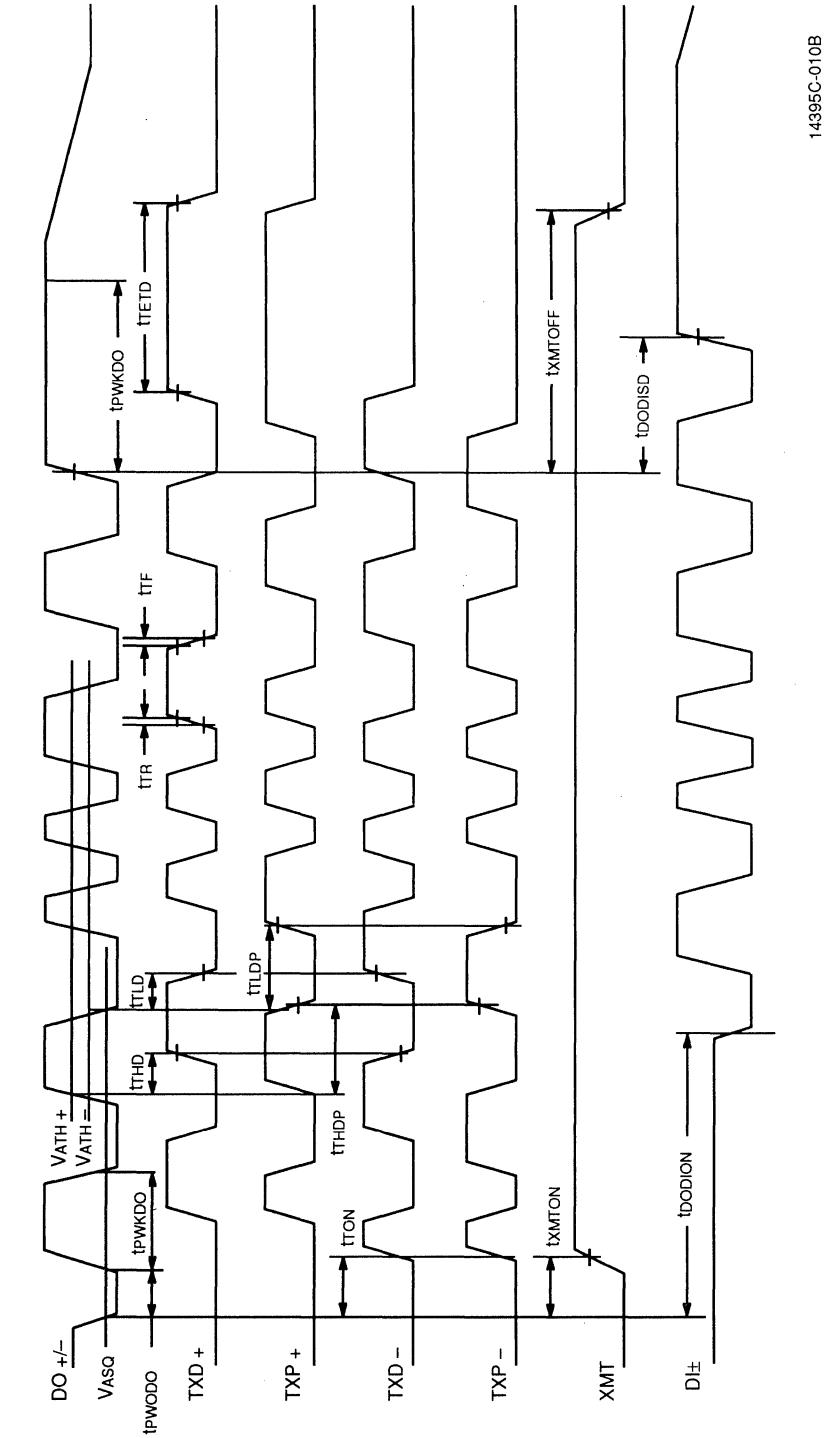
SWITCHING CHARACTERISTICS (Continued)

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:

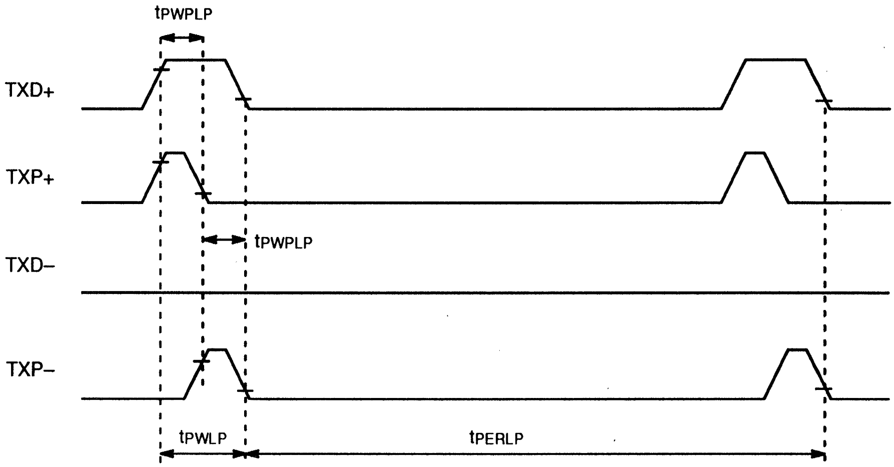
1. Parameter not tested.
2. Uses switching test load.
3. DO pulses narrower than t_{PWODO} (min) will be rejected; pulses wider than t_{PWODO} (max) will turn internal DO carrier sense on.
4. DO pulses narrower than t_{PWKDO} (min) will maintain internal DO carrier sense on; pulses wider than t_{PWKDO} (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

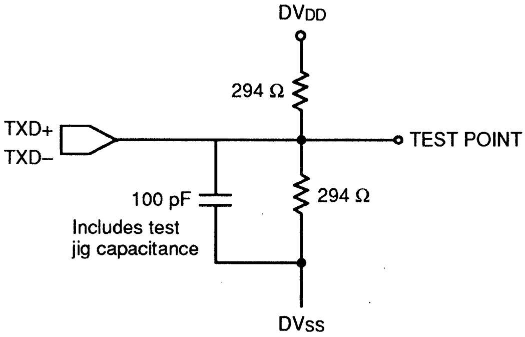
SWITCHING WAVEFORMS



14396-027A

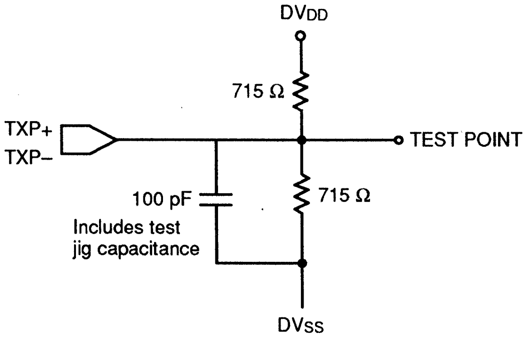
TP Idle Link Test Pulse

SWITCHING TEST CIRCUITS



14395C-014A

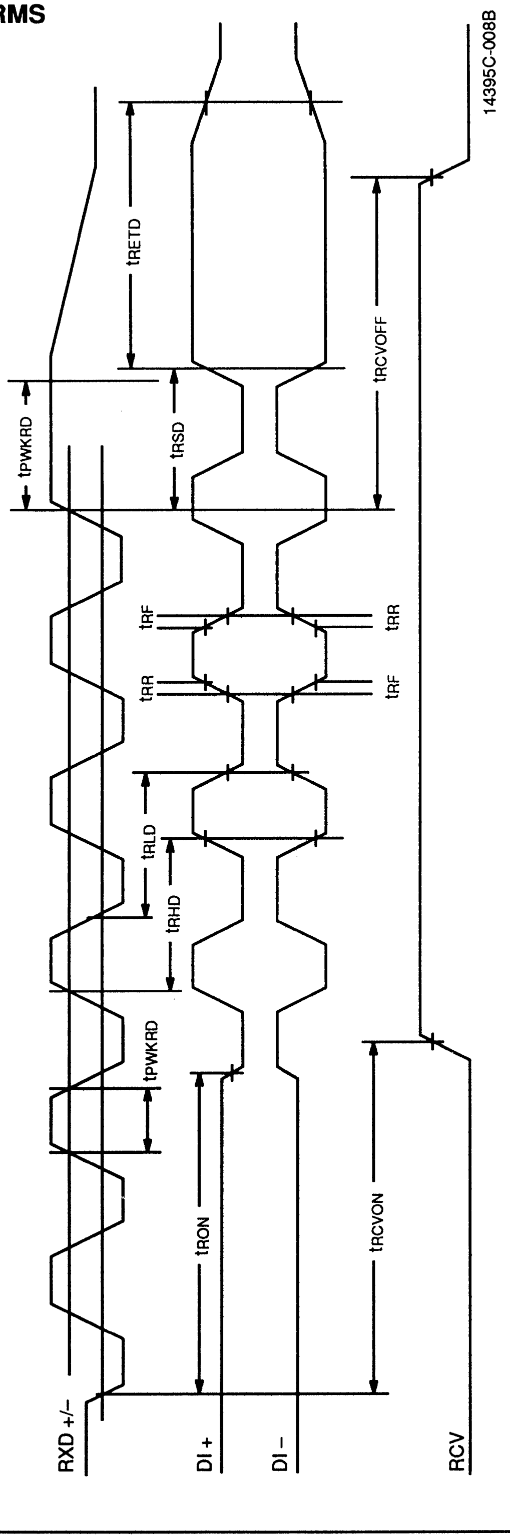
TXD Switching Test Circuit



14395C-015A

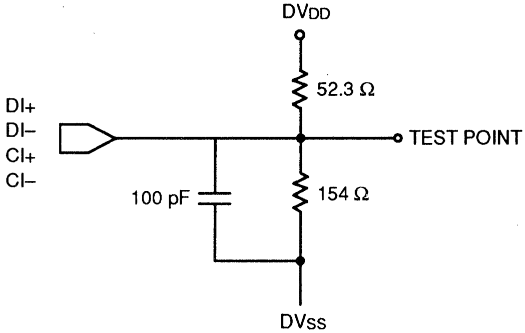
TXP Switching Test Circuit

SWITCHING WAVEFORMS



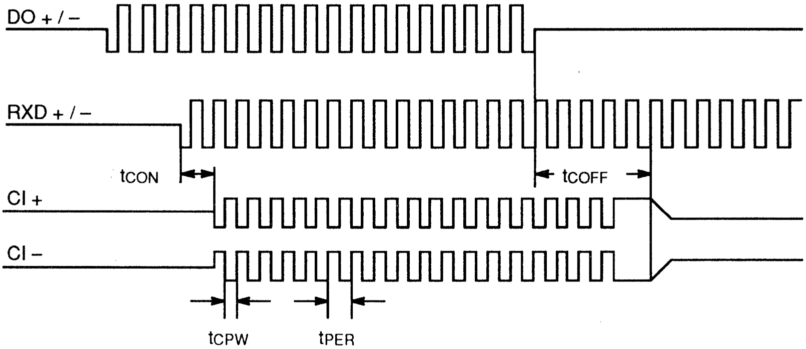
Receive Timing

RECEIVE TEST CIRCUIT

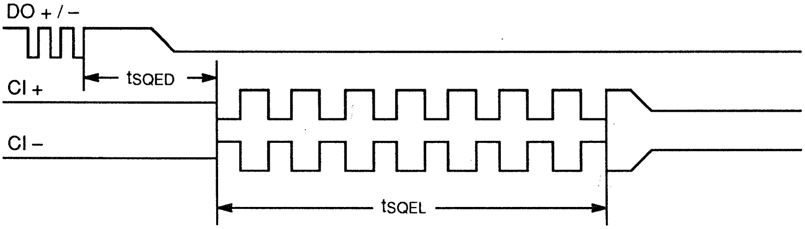


14395C-013A

AUI DI, CI Switching Test Circuit



Collision Timing



14395C-007B

SQE Test Timing ($\overline{\text{SQE Test}}$ Pin Connected to V_{SS})

PRELIMINARY

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

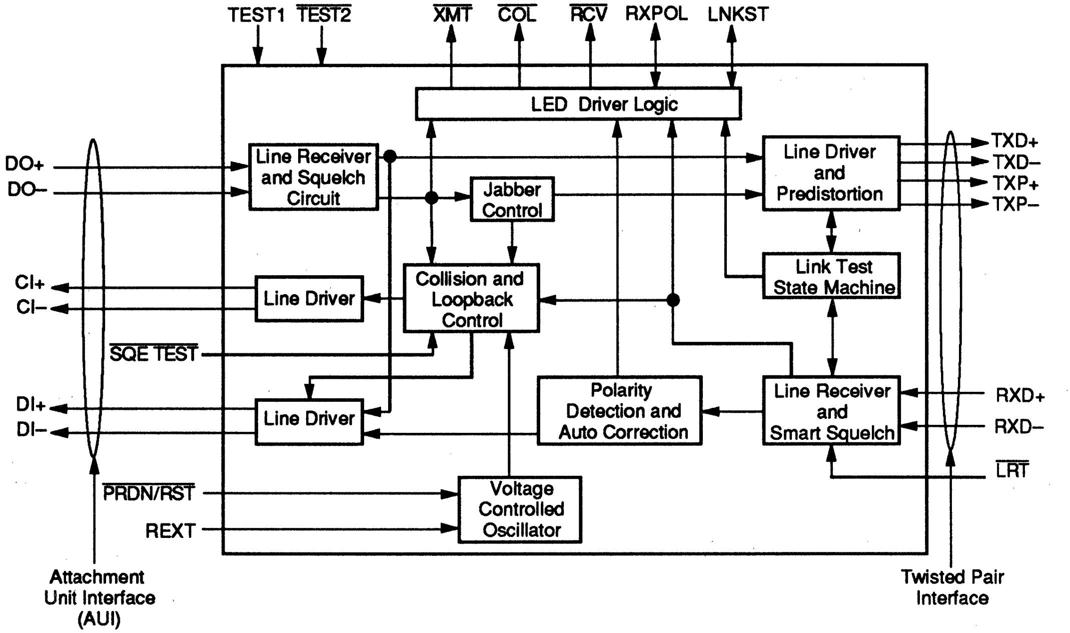
The Am79C100 Twisted Pair Ethernet Transceiver Plus (TPEX Plus) is an integrated circuit that implements the Medium Attachment Unit (MAU) functions for the Twisted Pair Medium, as specified by the supplement to IEEE 802.3 standard (Type 10BASE-T). This device provides the necessary electrical and functional 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. A Amendment/0
Issue Date: January 1992

BLOCK DIAGRAM


TEST1
TEST2
XMT
COL
RCV
RXPOL
LNKST
LED Driver Logic
DO+
DO–
Line Receiver and Squelch Circuit
Jabber Control
Line Driver and Predistortion
TXD+
TXD–
TXP+
TXP–
Link Test State Machine
CI+
CI–
Line Driver
Collision and Loopback Control
SQE TEST
DI+
DI–
Line Driver
Polarity Detection and Auto Correction
Line Receiver and Smart Squelch
RXD+
RXD–
LRT
PRDN/RST
REXT
Voltage Controlled Oscillator
Attachment Unit Interface (AUI)
Twisted Pair Interface

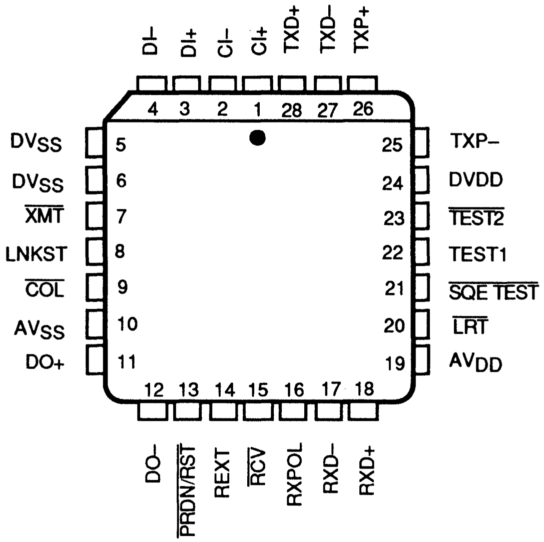
16511B-001B

RELATED AMD PRODUCTS

Part No.	Description
Am79C900	Integrated Local Area Communications Controller (ILACC)
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am7992B	Serial Interface Adapter (SIA)
Am79C980	Integrated Multiport Repeater for 10BASE-T (IMR)
Am7996	IEEE-802.3/Ethernet/Cheapernet Transceiver
Am79C940	Media Access Controller for Ethernet (MACE)

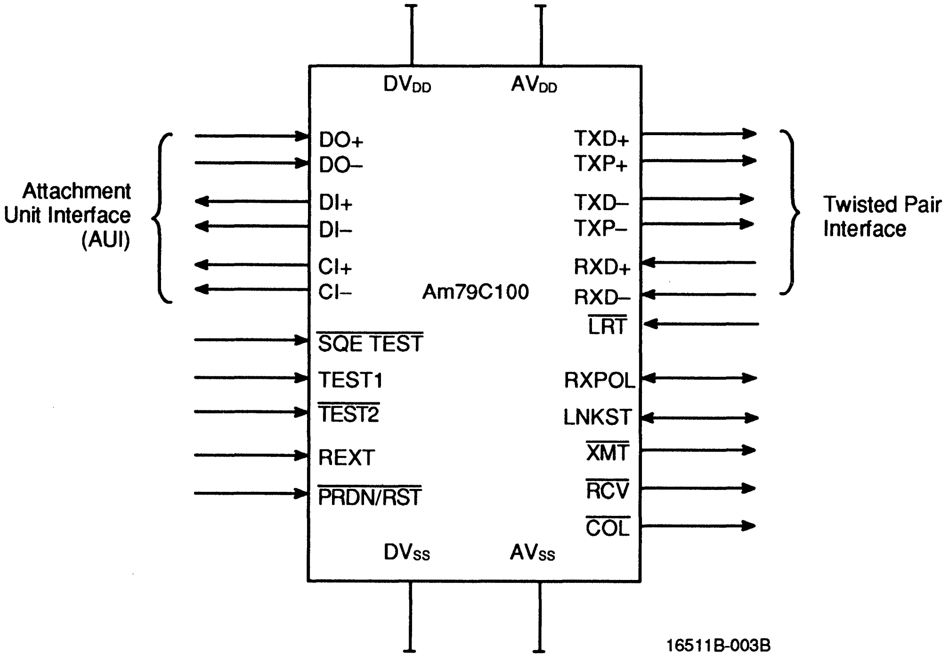
CONNECTION DIAGRAM

PLCC



16511A-002A

LOGIC SYMBOL

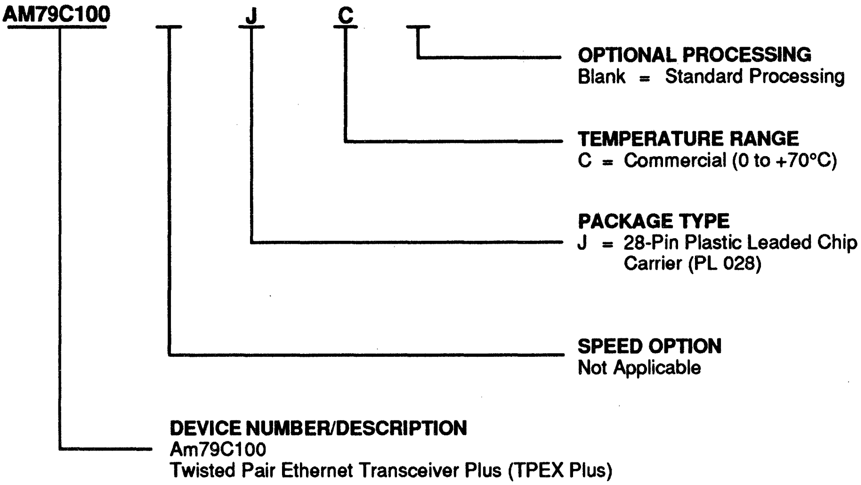


16511B-003B

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:



AM79C100 _ J C _

OPTIONAL PROCESSING
Blank = Standard Processing

TEMPERATURE RANGE
C = Commercial (0 to +70°C)

PACKAGE TYPE
J = 28-Pin Plastic Leaded Chip Carrier (PL 028)

SPEED OPTION
Not Applicable

DEVICE NUMBER/DESCRIPTION
Am79C100
Twisted Pair Ethernet Transceiver Plus (TPEX Plus)

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, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

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.

$\overline{\text{COL}}$
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 $\overline{\text{COL}}$ returns inactive. $\overline{\text{COL}}$ 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 $\overline{\text{COL}}$ 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.

$\overline{\text{LRT}}$
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 $\overline{\text{LRT}}$ in the HIGH state, the unsquelch threshold for the RXD± circuit will be 300 to 520 mV peak. With $\overline{\text{LRT}}$ in the LOW state, the unsquelch threshold for the RXD± circuit will be 180 to 312 mV peak. In either case, the RXD± circuit post unsquelch threshold will be approximately one half of the initial unsquelch threshold.

$\overline{\text{PRDN/RST}}$
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).

$\overline{\text{RCV}}$
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 $\overline{\text{COL}}$ 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 $\overline{\text{RCV}}$ output is pulse stretched for 20 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.

$\overline{\text{SQE TEST}}$

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. $\overline{\text{SQE TEST}}$ 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 $\overline{\text{TEST2}}$ is maintained HIGH), TPEX Plus will enter the Loopback Test mode. The type of loopback is determined by the state of the $\overline{\text{SQE TEST}}$ pin. If $\overline{\text{SQE TEST}}$ 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 $\overline{\text{SQE TEST}}$ 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.

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 $\overline{\text{COL}}$ 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.

$\overline{\text{TEST2}}$

Test
Input, Active LOW

This pin should be tied HIGH for normal operation. $\overline{\text{TEST2}}$ 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.

$\overline{\text{XMT}}$

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 $\overline{\text{COL}}$ 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 $\overline{\text{XMT}}$ 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 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 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 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 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 to 1500 ns). When enabled, a SQE Test will occur at the end of every transmission, starting 6 to 16 bit times (600 to 1600 ns) after the last transition of the transmitted signal. For repeater applications, the SQE Test function can be disabled by tying the $\overline{\text{SQE TEST}}$ pin HIGH or by leaving it disconnected. The $\overline{\text{COL}}$ 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 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 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 $\overline{\text{COL}}$ driver, disable the $\overline{\text{XMT}}$ driver (regardless of DO± activity), and allow the $\overline{\text{RCV}}$ driver to indicate the current state of the RXD± pair. If there is no receive activity on RXD±, only $\overline{\text{COL}}$ will be active during jabber detect. If there is RXD± activity, both $\overline{\text{COL}}$ and $\overline{\text{RCV}}$ will be active.

Power Down

In addition to on board power-on-reset circuitry, the $\overline{\text{PRDN/RST}}$ pin is used as the master reset for TPEX Plus. $\overline{\text{PRDN/RST}}$ must be driven LOW for a minimum of 2 µs for reset to occur. The $\overline{\text{PRDN/RST}}$ 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 $\overline{\text{PRDN/RST}}$ 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 $\overline{\text{XMT}}$, $\overline{\text{RCV}}$, and $\overline{\text{COL}}$ pins are in the high impedance state. TPEX Plus will remain in idle mode as long as $\overline{\text{PRDN/RST}}$ is asserted.

Following the rising edge of the signal on $\overline{\text{PRDN/RST}}$, 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 $\overline{\text{XMT}}$, $\overline{\text{RCV}}$ and $\overline{\text{COL}}$ outputs low for 20 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 $\overline{\text{TEST2}}$ 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):** $\overline{\text{SQE TEST}}$ 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:** $\overline{\text{SQE TEST}}$ 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).

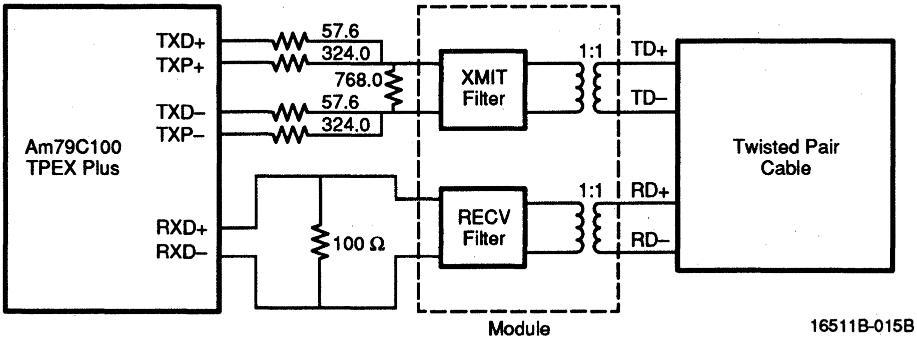
In both modes TPEX Plus will be forced into the Link Pass state, and will not enter the Link Fail state regardless of RXD± inactivity. The following functions are disabled: jabber circuit, collision detection, and collision oscillator. The functions which remain enabled are: the DO± and RXD± squelch circuits, $\overline{\text{XMT}}$ and $\overline{\text{RCV}}$ outputs, Link Beat pulse generation and polarity detection/correction. In addition, in both modes, the $\overline{\text{COL}}$ 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

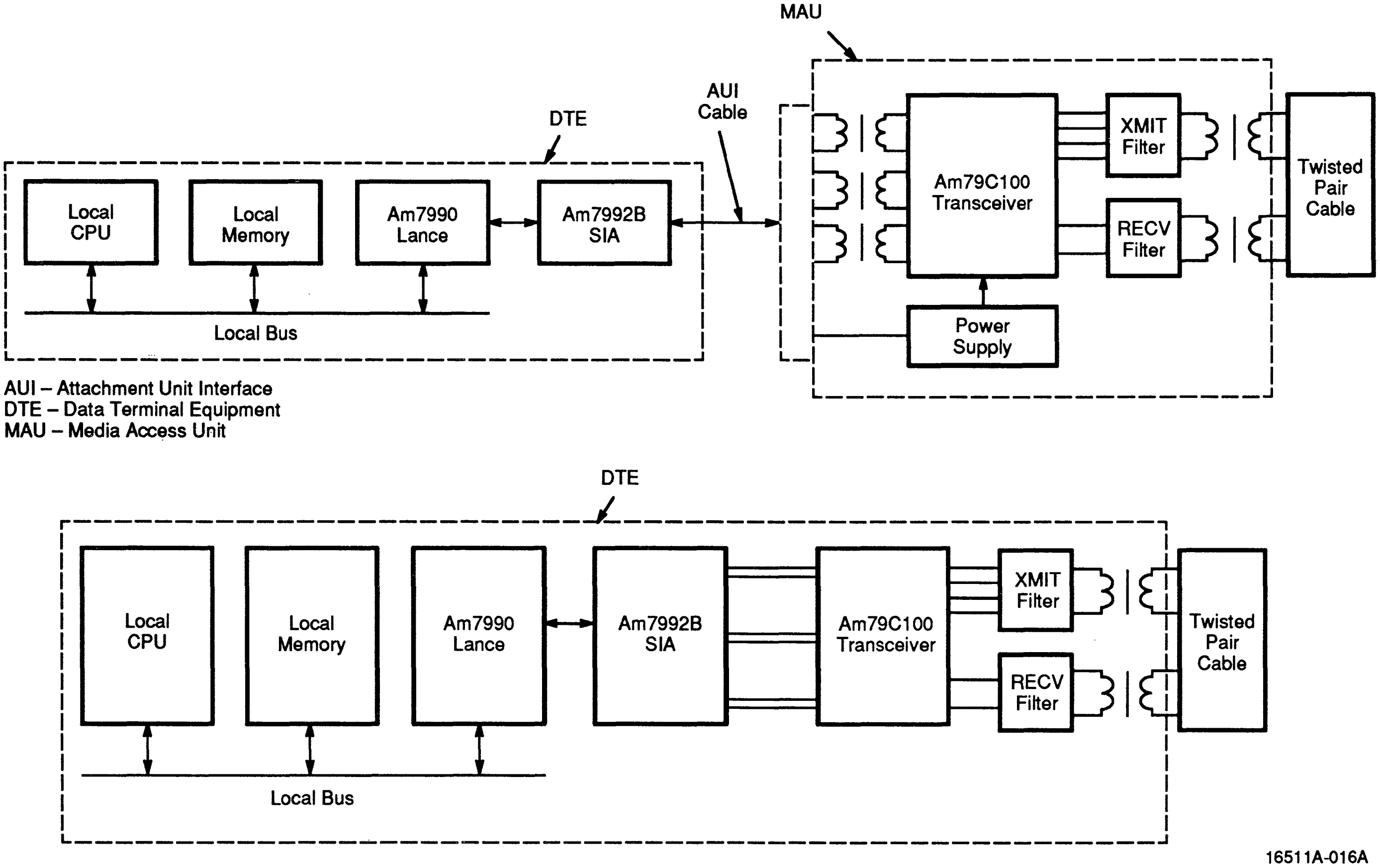
Figure 1 shows a typical twisted pair port external components schematic. The resistors used should have a ±1% tolerance to ensure interoperability with 10BASE-T compliant networks. The filters and pulse transformers are necessary devices that have a major influence on the performance and compliance 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:

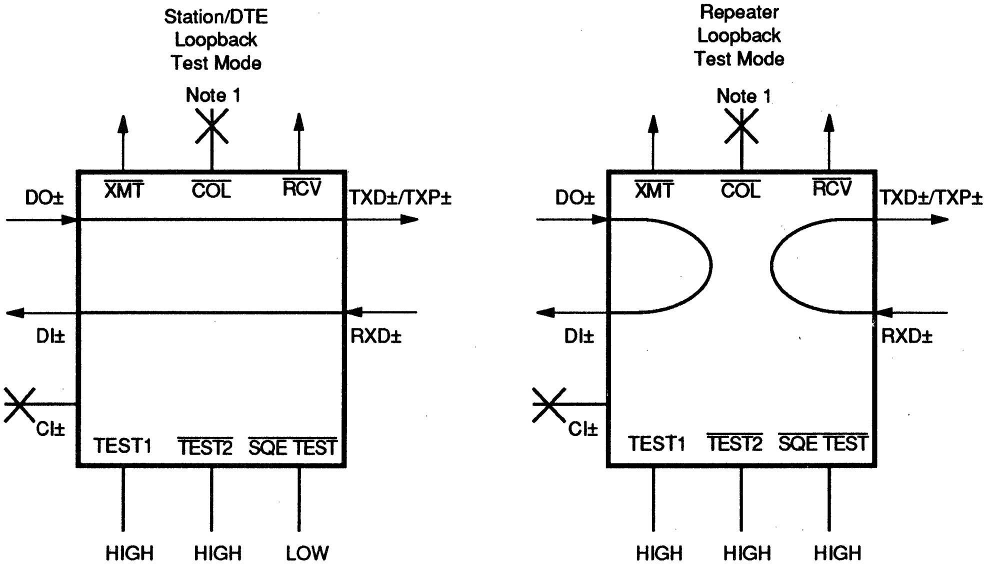
Belfuse	TDK
Pulse Engineering	PCA
Valor Electronics	Nano Pulse

Figure 1. Typical Twisted Pair Port External Components



AUI – Attachment Unit Interface
DTE – Data Terminal Equipment
MAU – Media Access Unit

Figure 2. Typical Twisted Pair Ethernet Node

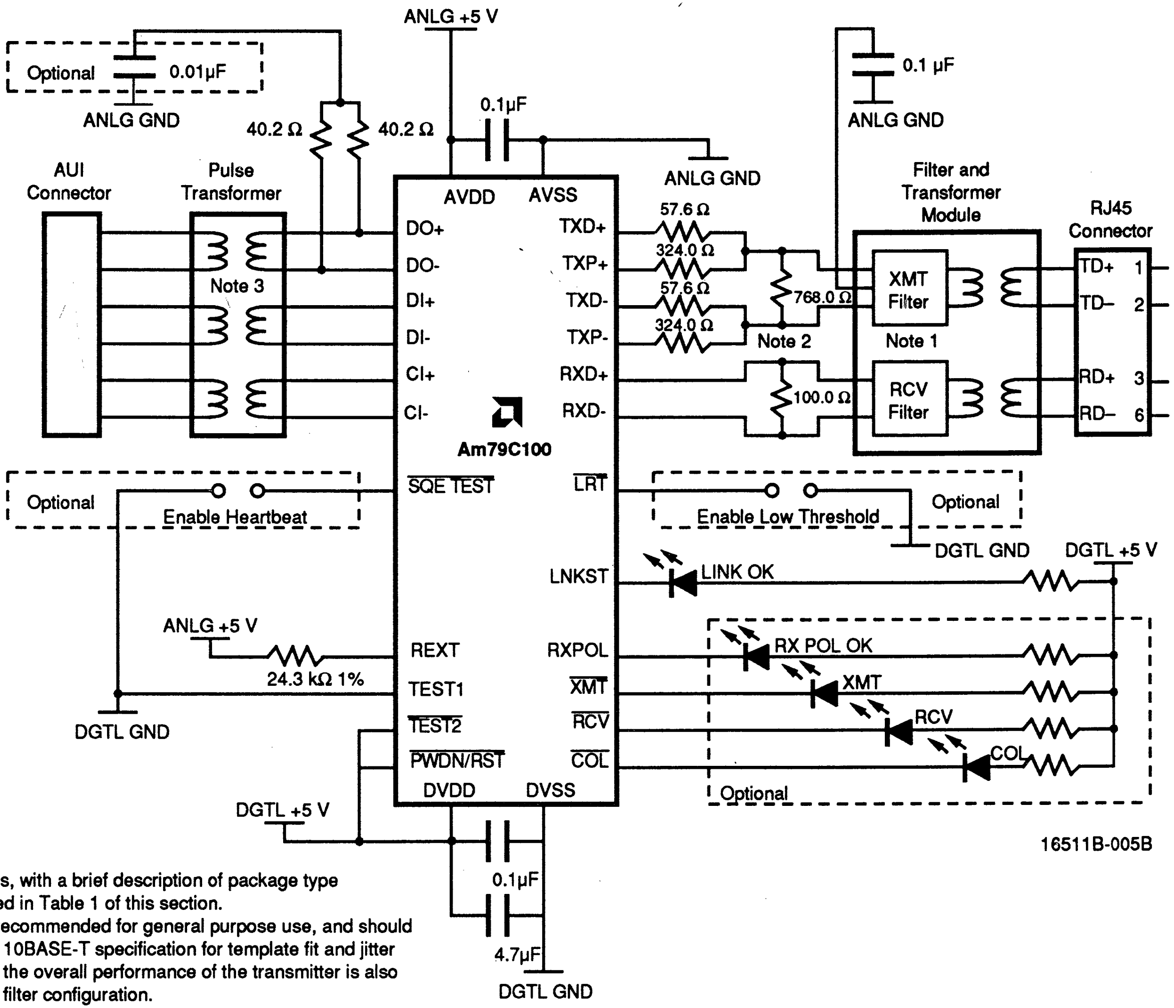


16511A-004A

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 to +150°C
Ambient Temperature Under Bias:	0 to +70°C
Supply Voltage 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 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_{AOD}OFF	DI± & CI± Differential Idle Output Voltage	R_L = 78 Ω	–40	+40	mV
I_{AOD}OFF	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

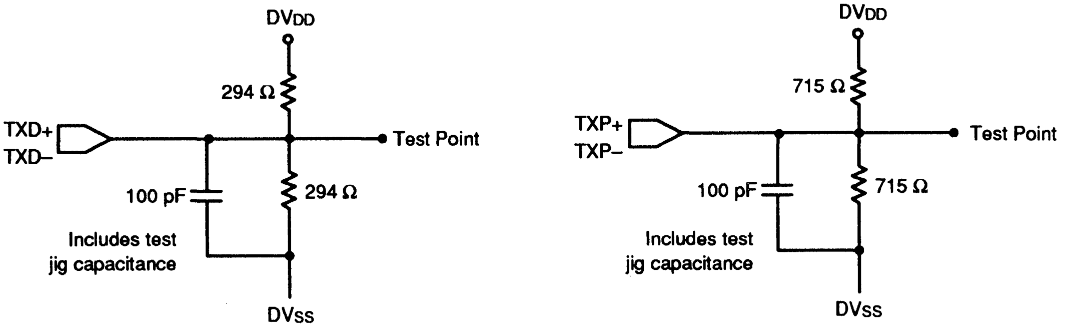
SWITCHING CHARACTERISTICS (Continued)

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.

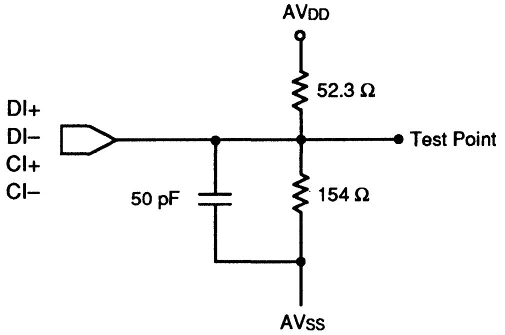
SWITCHING TEST CIRCUITS



16511A-006A

16511A-007A

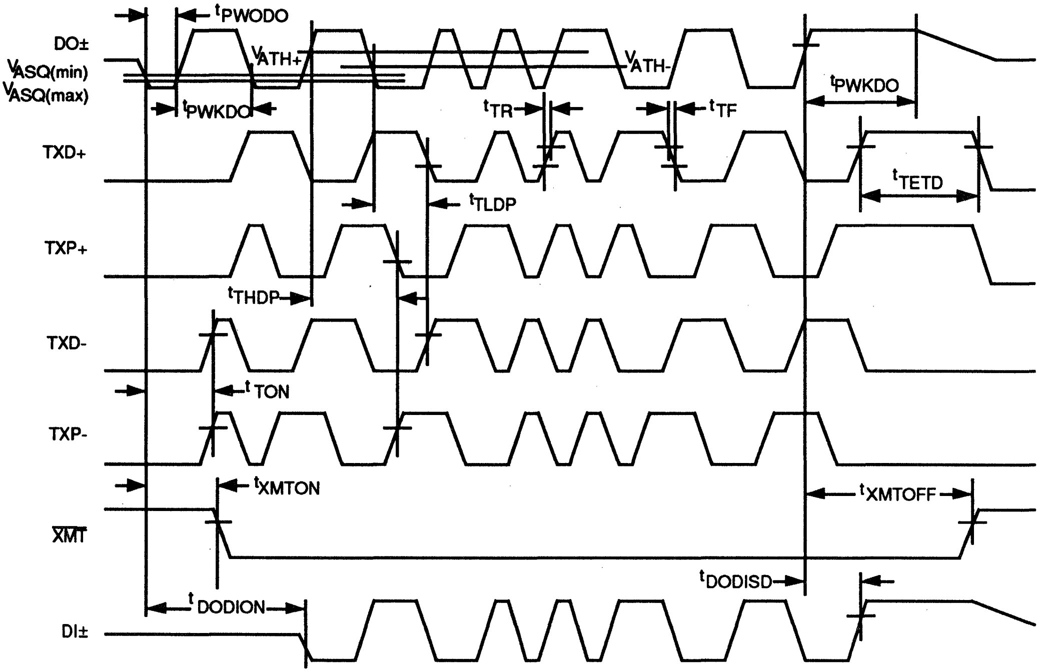
Twisted Pair Transmit Test Circuit



16511A-008A

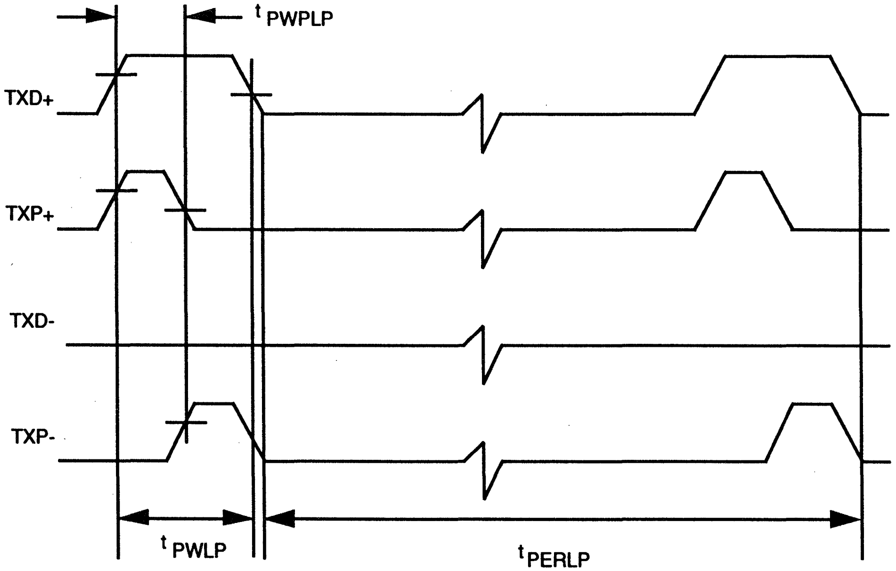
AUI Transmit Test Circuit

SWITCHING TEST WAVEFORMS



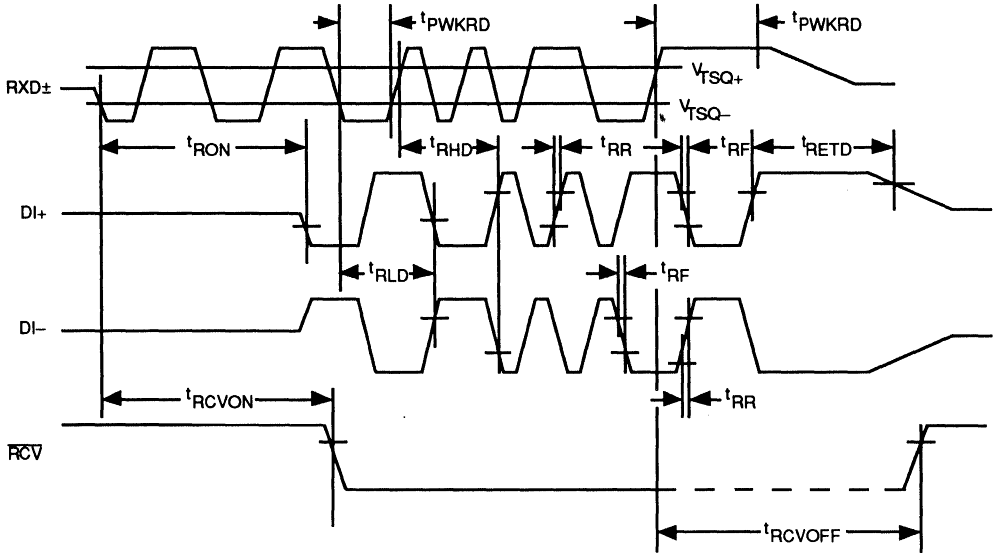
Transmit Timing

16511A-009A



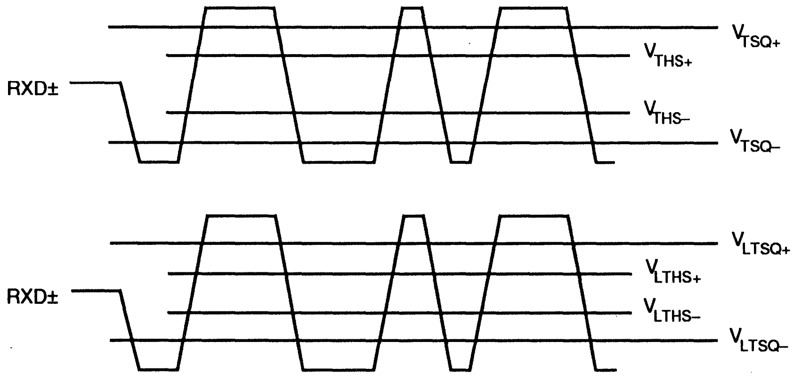
Transmit Link Beat Pulse

16511A-010A



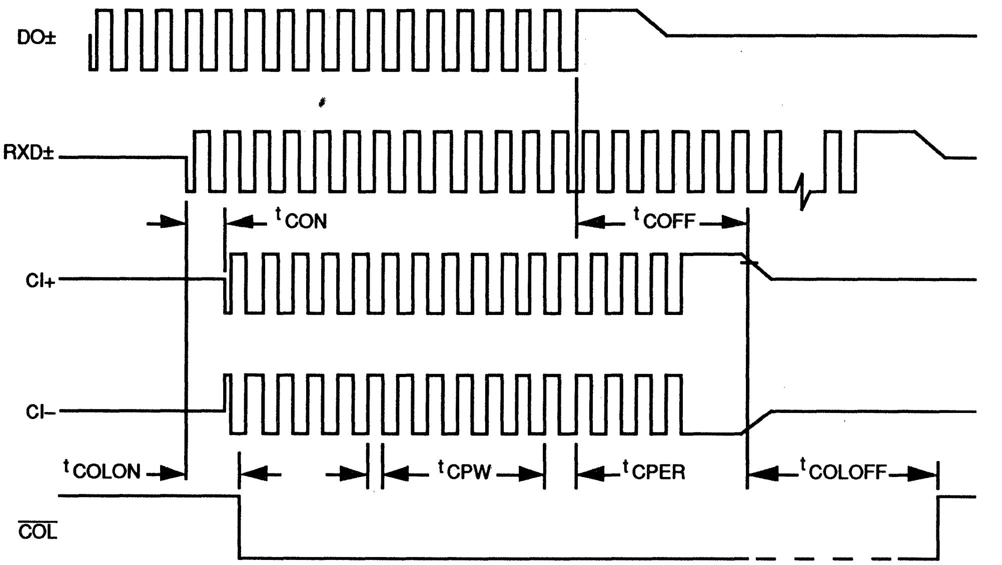
Receive Timing

16511A-011A



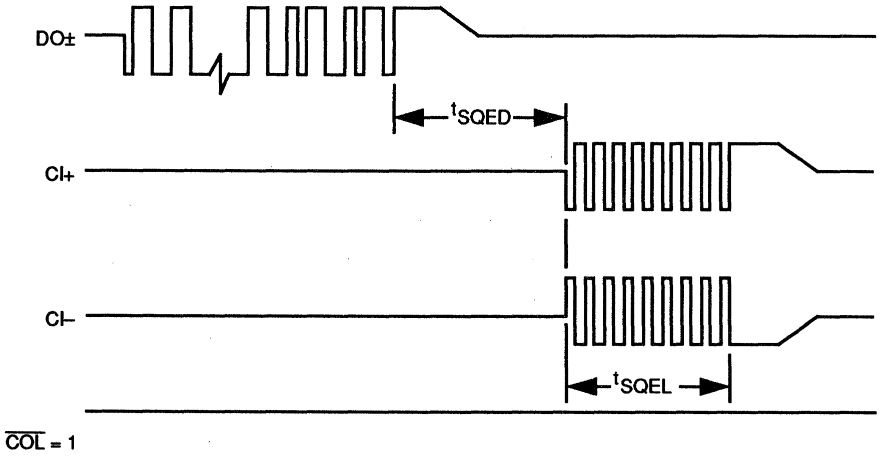
16511A-012A

Receive Thresholds



16511A-013A

Collision Timing



16511A-014A

SQE Test Timing

PRELIMINARY

Advanced Micro Devices

Am79C980

Integrated Multiport Repeater (IMR)

DISTINCTIVE CHARACTERISTICS

- **CMOS device features high integration and low power with a single +5 V supply**
- **Repeater functions conform to 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 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 re-transmitted waveforms**
- **Preamble loss effects eliminated by deep FIFO**

GENERAL DESCRIPTION

The Integrated Multiport Repeater (IMR) 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 conforming to the 10BASE-T standard. The Am79C980 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).

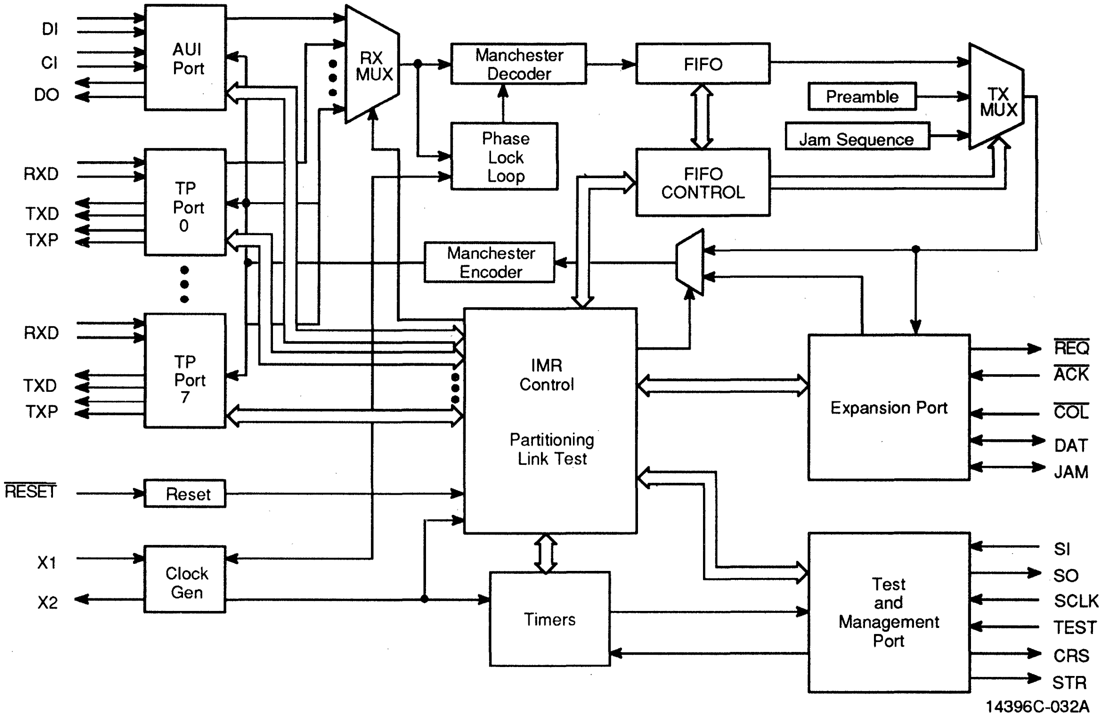
A network based on the 10BASE-T standard uses unshielded twisted pair cables, therefore providing an economical solution to networking by allowing the use of existing telephone wiring.

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 10BASE-5 (Ethernet) and 10BASE-2 (Cheapernet) networks, as well as Fiber Optic Inter-repeater Link (FOIRL) segments. Network management and test functions are provided through TTL compatible I/O pins.

The device is fabricated in CMOS technology and requires a single +5 V supply.

Publication# 14396 Rev. C Amendment/0
Issue Date: January 1992

BLOCK DIAGRAM



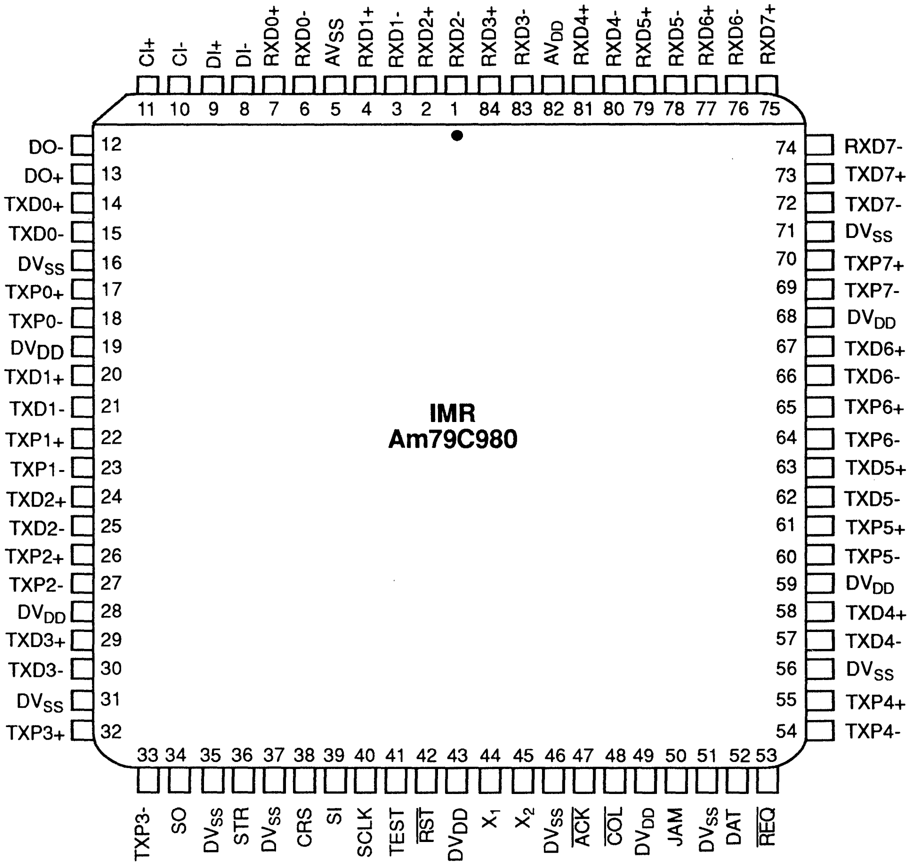
14396C-032A

RELATED AMD PRODUCTS

Part No.	Description
Am79C900	32-Bit Integrated Local Area Communications Controller (ILACC)
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am7992B	Serial Interface Adapter (SIA)
Am79C98	Twisted Pair Ethernet Transceiver (TPEX)
Am7996	IEEE 802.3/Ethernet/Cheapernet Transceiver
Am7997	IEEE 802.3 Compliant Tap Transceiver

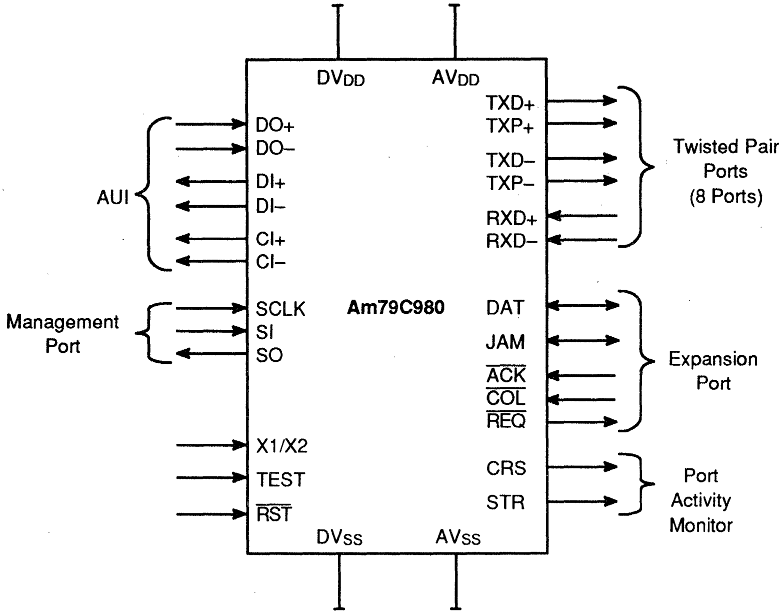
CONNECTION DIAGRAM

PLCC



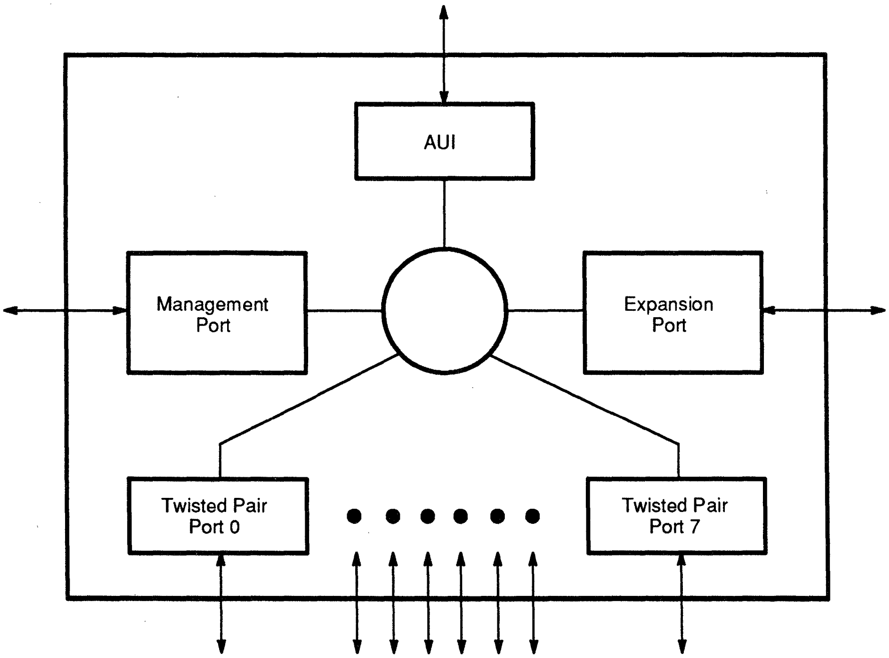
14396C-001B

LOGIC SYMBOL



14396C-035A

LOGIC DIAGRAM

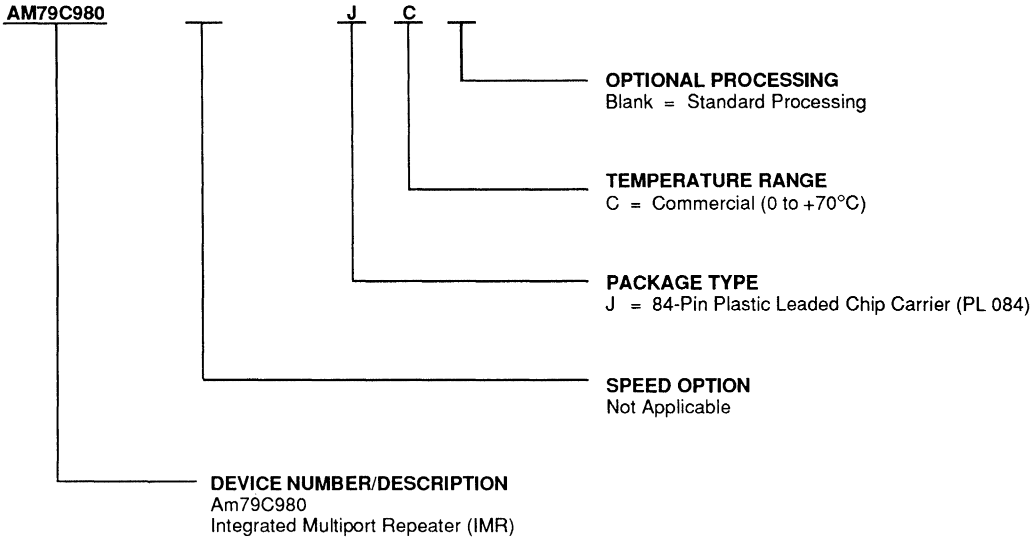


14396-002A

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:



AM79C980 __ J C __

OPTIONAL PROCESSING
Blank = Standard Processing

TEMPERATURE RANGE
C = Commercial (0 to +70°C)

PACKAGE TYPE
J = 84-Pin Plastic Leaded Chip Carrier (PL 084)

SPEED OPTION
Not Applicable

DEVICE NUMBER/DESCRIPTION
Am79C980
Integrated Multiport Repeater (IMR)

Valid Combinations	
AM79C980	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

$\overline{\text{ACK}}$

Acknowledge

Input, Active LOW

When this input is asserted, it signals to the requesting IMR that it may control the DAT and JAM pins. If the IMR is not requesting control of the DAT line ($\overline{\text{REQ}}$ pin HIGH), then the assertion of the $\overline{\text{ACK}}$ 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. These power pins should be decoupled with a 47 µF capacitor 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}.

$\overline{\text{COL}}$

Expansion Collision

Input, Active LOW

When this input is asserted by an external arbiter, it signifies that more than one IMR is active and that each IMR should generate Collision Jam Sequence independently.

CI+, CI-

Control In

Input

AUI port differential receiver.

CRS

Carrier Sense

Output

The states of the internal carrier sense signals for the AUI port and the eight twisted pair ports is serially output on this pin continuously. The output serial bit stream is synchronized to the X_1 clock.

DAT

Data

Input/Output

When there is a single IMR active (in a multiple IMR design), the IMR with both $\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ pins asserted drives the DAT line with NRZ data. The pin is an input when only the $\overline{\text{ACK}}$ signal is asserted, and is in a high impedance state if neither $\overline{\text{REQ}}$ or $\overline{\text{ACK}}$ is asserted.

DI+, DI-

Data In

Input

AUI port differential receiver.

DO+, DO-

Data Out

Output

AUI port differential driver.

DV_{DD}

Digital Power

Power Pin

These pins supply the +5 V to the logic portions of the IMR 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 #	DV_{SS} 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

This pin is an output on the single active IMR ($\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ both asserted). The active IMR drives the JAM pin HIGH to indicate that it is in a Collision state. The pin is an input when only the $\overline{\text{ACK}}$ signal is asserted, and is in a high impedance state if neither $\overline{\text{REQ}}$ or $\overline{\text{ACK}}$ is asserted.

$\overline{\text{REQ}}$

Request

Output, Active LOW

This pin is driven LOW when the IMR is active. An active IMR is defined as an IMR which 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 is requesting the use of the DAT and JAM lines for the transfer of repeated data or collision status to other IMRs.

$\overline{\text{RST}}$

Reset

Input, Active LOW

Driving this pin LOW resets the internal logic of the IMR. Reset should be synchronized to 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

Serial data (input or output) is clocked (in or out) on the rising edge of the signal on this pin.

SI

Serial In

Input

Test/Management serial input port. Management commands are clocked in on this pin synchronous to the SCLK input.

SO

Serial Out

Output

Test/Management serial output port. Management results are clocked out on this pin synchronous to the SCLK input.

STR

Store

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 (one microsecond).

TEST

Test Pin

Input, Active HIGH

This pin should be tied LOW for normal operation. If this pin is driven HIGH, then the IMR can be programmed for Loopback Test Mode.

$TXD+_{0-7}$, $TXD-_{0-7}$

Transmit Data

Output

10BASE-T port differential drivers (8 ports).

$TXP+_{0-7}$, $TXP-_{0-7}$

Transmit Pre-distort

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 Am79C980 Integrated Multiport Repeater 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 also provides the hooks necessary for complex network management and evaluation. The IMR is also expandable, enabling the implementation of repeaters based on several IMRs.

The IMR 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 will re-transmit the received data to all other enabled network ports. The repeated data will also be presented on the DAT line to facilitate multiple-IMR repeater applications.

Signal Regeneration

When re-transmitting a packet, the IMR ensures that the outgoing packet complies to the 802.3 specification in terms of preamble structure, voltage amplitude, and timing characteristics. Specifically, data packets repeated by the IMR 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 spec. Finally, signal symmetry is restored to data packets repeated by the IMR, removing jitter and distortion caused by the network cabling.

Jabber Lockup Protection

The IMR 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 for 96 bit times if the IMR has been transmitting continuously for more than 65,536 bit times.

Collision Handling

The IMR will detect and respond to collision conditions as specified in 802.3. A multiple-IMR repeater implementation also conforms to the 802.3 spec due to the inter-IMR status communication provided by the expansion port of the IMR. Specifically, a repeater based on one or more IMRs will handle the transmit collision and one-port-left collision conditions correctly as specified in section 9 of the 802.3 spec.

Fragment Extension

If the total packet length received by the IMR is less than 96 bits, including preamble, the IMR will extend the repeated packet length to 96 bits by appending a Jam sequence to the original runt packet.

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 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 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 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 network port will cause the IMR to partition that port:

a. A collision condition exists continuously for a time between 1024 and 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 will reconnect that port if the following is met:

a. (Standard reconnection algorithm) A data packet longer than 488 bit times (nominal) is transmitted or received by the partitioned port without a collision

b. (Alternate reconnection algorithm) A data packet longer than 488 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 will transmit Link Test pulses to any TP port after that port's transmitter has been inactive for more than 16 milliseconds (nominal). Conversely, if a TP port does not receive any data packets or Link Test pulses for more than 50 to 150 milliseconds 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 (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 management port on a port-by-port basis to allow the IMR to interoperate with pre-10BASE-T twisted pair networks that do not implement the Link Test function. This interoperability is possible because the IMR 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 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 Management Port.

Reset

The IMR enters reset state when the $\overline{\text{RESET}}$ pin is driven LOW. The RESET pin should be held LOW for a minimum of 150 µs (3000 X1 clock cycles). This allows the IMR to reset the internal logic and permits the internal PLL to stabilize. During reset, the output signals are placed in their inactive states. That is, all analog signals are placed in their idle states, all bidirectional signals are not driven, active LOW signals are driven HIGH, and all active HIGH signals are driven LOW.

In a multiple IMR repeater the $\overline{\text{RESET}}$ signal should be applied simultaneously to all IMRs and should be synchronized to the external X_1 CLOCK. Reset synchronization is also required when accessing PAM (Port Activity Monitor).

SI should be held HIGH for at least 500 ns following the rising edge of $\overline{\text{RST}}$.

The following table summarizes the state of the IMR following reset.

Table 1. IMR after Reset

Function	State after Reset
Active LOW outputs	HIGH
Active HIGH outputs	LOW
SO Output	HIGH
Bidirectional Pins	HI-IMPEDANCE
Transmitters (TP and AUI)	IDLE
Receivers (TP and AUI)	ENABLED
AUI Partitioning/Reconnection Algorithm	STANDARD ALGORITHM
TP Port Partitioning/Reconnection Algorithm	STANDARD ALGORITHM
Link Test Function for TP Ports	ENABLED, TP PORTS IN LINK FAIL
Automatic Receiver Polarity Reversal Function	DISABLED

Expansion Port

The IMR Expansion Port is comprised of five pins; two are bi-directional signals (DAT and JAM), two are input signals ($\overline{\text{ACK}}$ and $\overline{\text{COL}}$), and one is an output signal ($\overline{\text{REQ}}$). These signals are used when a multiple-IMR repeater application is employed. In this configuration, all IMRs must be clocked synchronously with a common clock connected to the X1 inputs of all IMRs.

The IMR expansion scheme allows the use of multiple IMRs in a single board repeater or a modular multiport repeater with a backplane architecture. As many as three IMRs can be connected together without using external bus transceivers. The DAT pin is a bidirectional I/O pin which can be used with external bus transceivers to transfer data between the IMRs in a multiple-IMR 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 to communicate its internal status to the remaining (inactive) IMRs. When JAM is asserted HIGH, it indicates that the active IMR 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 is detecting collision on one port only or on more than one port. When DAT is driven HIGH by the IMR (while JAM is asserted by the IMR), then the active IMR is detecting a collision condition on one port only. This 'one-port-left' signaling is necessary for a multiple-IMR repeater to function correctly as a single multiport repeater unit. The IMR 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 nanoseconds) before deasserting $\overline{\text{REQ}}$.

The arbitration for access to the bussed bi-directional signals (DAT and JAM) is provided by one output ($\overline{\text{REQ}}$) and two inputs ($\overline{\text{ACK}}$ and $\overline{\text{COL}}$). The IMR asserts the $\overline{\text{REQ}}$ pin to indicate that it is active and wishes to drive the DAT and JAM pins. An external arbiter senses the $\overline{\text{REQ}}$ lines from all the IMRs and asserts the $\overline{\text{ACK}}$ line when one and only one IMR is asserting its $\overline{\text{REQ}}$ line. If more than one IMR is asserting its $\overline{\text{REQ}}$ line, the arbiter must assert the $\overline{\text{COL}}$ signal, indicating that more than one IMR is active. More than one active IMR at a time constitutes a collision condition, and all IMRs are notified of this occurence via the $\overline{\text{COL}}$ line of the Expansion Port.

Note that a transition from multiple IMRs arbitrating for the DAT and JAM pins (with $\overline{\text{COL}}$ asserted, $\overline{\text{ACK}}$ deasserted) to a condition when only one IMR is arbitrating for the DAT and JAM pins (with $\overline{\text{ACK}}$ asserted, $\overline{\text{COL}}$ deasserted) involves one expansion port bus cycle (100 nanoseconds). During this transitional bus cycle, $\overline{\text{COL}}$ is deasserted, $\overline{\text{ACK}}$ is asserted, and the DAT and JAM pins are not driven. However, each IMR will remain in the collision state (transmitting jam sequence) during this transitional bus cycle. In subsequent expansion port bus cycles ($\overline{\text{REQ}}$ and $\overline{\text{ACK}}$ still asserted), the IMRs will return to the 'master and slaves' condition where only one IMR 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 bus after a multi-IMR collision must assert the JAM line until it too drops its request for the expansion bus.

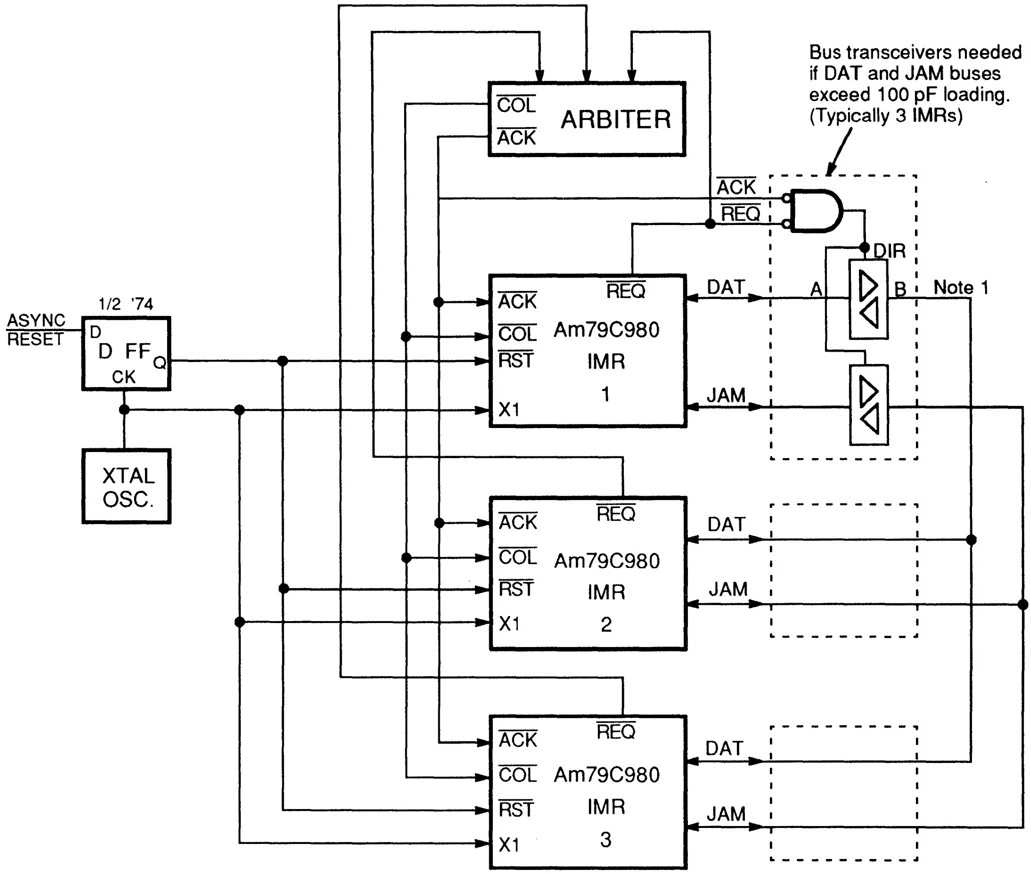
External Arbiter

A simple arbitration scheme is required when multiple IMRs are connected together to increase the total number of repeater ports. The arbiter should have one input ($\overline{\text{REQ1}}$...$\overline{\text{REQn}}$) for each of the n IMRs to be used, and two global outputs ($\overline{\text{COL}}$ and $\overline{\text{ACK}}$). This function is easily implemented in a PAL® device, with the following logic equations:

$$
\begin{array}{lcl}
\text{ACK} & = & \text{REQ1} \;\&\; \overline{\text{REQ2}} \;\&\; \overline{\text{REQ3}} \;\&\; ....\overline{\text{REQn}} \\
 & + & \overline{\text{REQ1}} \;\&\; \text{REQ2} \;\&\; \overline{\text{REQ3}} \;\&\; ....\overline{\text{REQn}} \\
 & & \bullet \\
 & & \bullet \\
 & & \bullet \\
 & + & \overline{\text{REQ1}} \;\&\; \overline{\text{REQ2}} \;\&\; \overline{\text{REQ3}} \;\&\; ....\ \text{REQn} \\
\text{COL} & = & \overline{\text{ACK}} \;\&\; (\text{REQ1} + \text{REQ2} + \text{REQ3} + ...\ \text{REQn})
\end{array}
$$

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 eight IMRs.



14396C-003B

Note 1:

Direction	DIR
B → A	LOW
A → B	HIGH

Figure 1. Multiple IMRs

Modular Repeater Design

The expansion port of the IMR 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 IMRs on that module, and provides signals to the backplane for use by a global arbiter.

Figure 2 shows an expandable repeater based on 3 modules that each use 3 IMRs. In this design, each module provides 24 10BASE-T ports and requires a single PALCE16V8-15 to perform the local arbitration function. The backplane portion of the modular repeater consists only of the global arbiter, also a single PALCE16V8-15.

Local Arbiter Logic Equations (for n IMRs per module):

$$
\begin{aligned}
\mathrm{Rm} = {} & \mathrm{REQ1} * \overline{\mathrm{REQ2}} * \overline{\mathrm{REQ3}} * \ldots \overline{\mathrm{REQn}} \\
& + \overline{\mathrm{REQ1}} * \mathrm{REQ2} * \overline{\mathrm{REQ3}} * \ldots \overline{\mathrm{REQn}} \\
& + \overline{\mathrm{REQ1}} * \overline{\mathrm{REQ2}} * \mathrm{REQ3} * \ldots \overline{\mathrm{REQn}} \\
& \bullet \\
& \bullet \\
& \bullet \\
& + \overline{\mathrm{REQ1}} * \overline{\mathrm{REQ2}} * \overline{\mathrm{REQ3}} * \ldots \mathrm{REQn}
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Cm} &= \overline{\mathrm{Rm}} * (\mathrm{REQ1} + \mathrm{REQ2} + \mathrm{REQ3} + \ldots \mathrm{REQn}) \\
\mathrm{DIR} &= \mathrm{Rm} * \mathrm{GLOBALACK}
\end{aligned}
$$

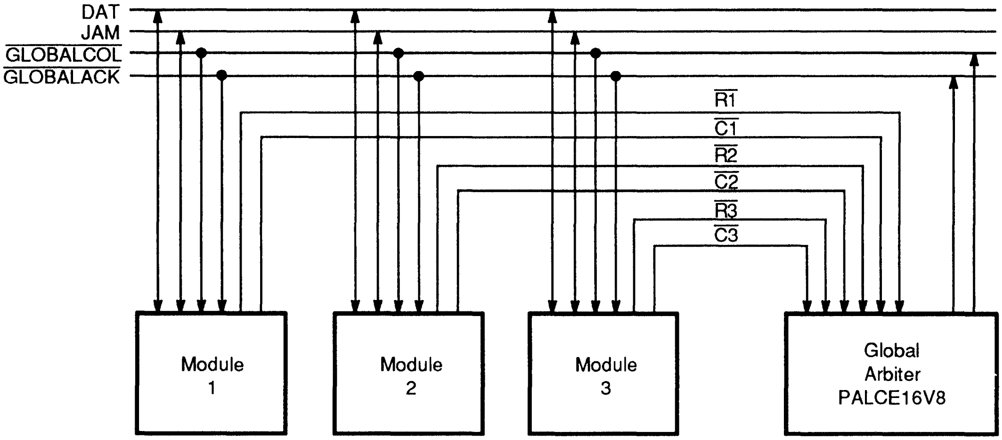
The DIR signal is HIGH when the module drives the DAT and JAM backplane bus signals. A single PALCE16V8 can support up to 8 IMRs per module.

Global Arbiter Logic Equations (for m modules):

$$
\begin{aligned}
& \mathrm{GLOBALACK} \\
& = \overline{\mathrm{C1}} * \overline{\mathrm{C2}} * \overline{\mathrm{C3}} * \ldots \overline{\mathrm{Cm}} * \mathrm{R1} * \overline{\mathrm{R2}} * \overline{\mathrm{R3}} * \ldots \overline{\mathrm{Rm}} \\
& + \overline{\mathrm{C1}} * \overline{\mathrm{C2}} * \overline{\mathrm{C3}} * \ldots \overline{\mathrm{Cm}} * \overline{\mathrm{R1}} * \mathrm{R2} * \overline{\mathrm{R3}} * \ldots \overline{\mathrm{Rm}} \\
& + \overline{\mathrm{C1}} * \overline{\mathrm{C2}} * \overline{\mathrm{C3}} * \ldots \overline{\mathrm{Cm}} * \overline{\mathrm{R1}} * \overline{\mathrm{R2}} * \mathrm{R3} * \ldots \overline{\mathrm{Rm}} \\
& \bullet \\
& \bullet \\
& \bullet \\
& + \overline{\mathrm{C1}} * \overline{\mathrm{C2}} * \overline{\mathrm{C3}} * \ldots \overline{\mathrm{Cm}} * \overline{\mathrm{R1}} * \overline{\mathrm{R2}} * \overline{\mathrm{R3}} * \ldots \mathrm{Rm}
\end{aligned}
$$

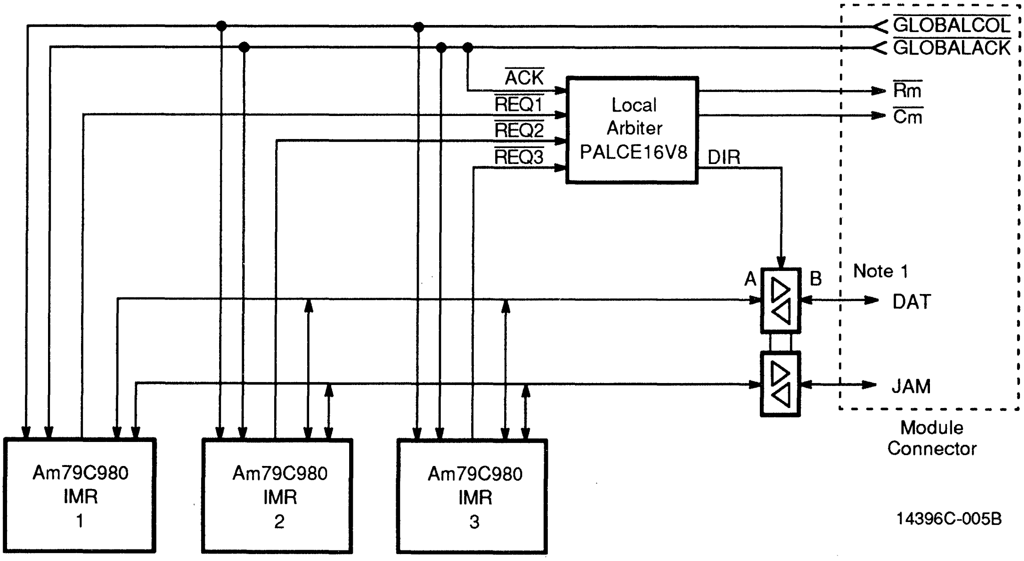
$$
\begin{aligned}
\mathrm{GLOBALCOL} = {} & \overline{\mathrm{GLOBALACK}} * (\mathrm{R1} + \mathrm{R2} + \ldots \mathrm{Rm}) \\
& + (\mathrm{C1} + \mathrm{C2} + \mathrm{C3} + \ldots \mathrm{Cm})
\end{aligned}
$$

A single PALCE22V10 can perform the global arbitration for up to 8 modules.



14396C-004B

Modular Repeater



14396C-005B

Repeater Module with 3 IMRs

Note 1:

Direction	DIR
B → A	LOW
A → B	HIGH

Figure 2. Expandable Modular Repeater

In-Band Hub Management

Because all repeated data in the IMR or multi-IMR 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 or Am79C900. 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 3 shows how to interface a repeater based on two IMRs to an Ethernet controller such as the Am79C900 ILACC or the Am7990 LANCE.

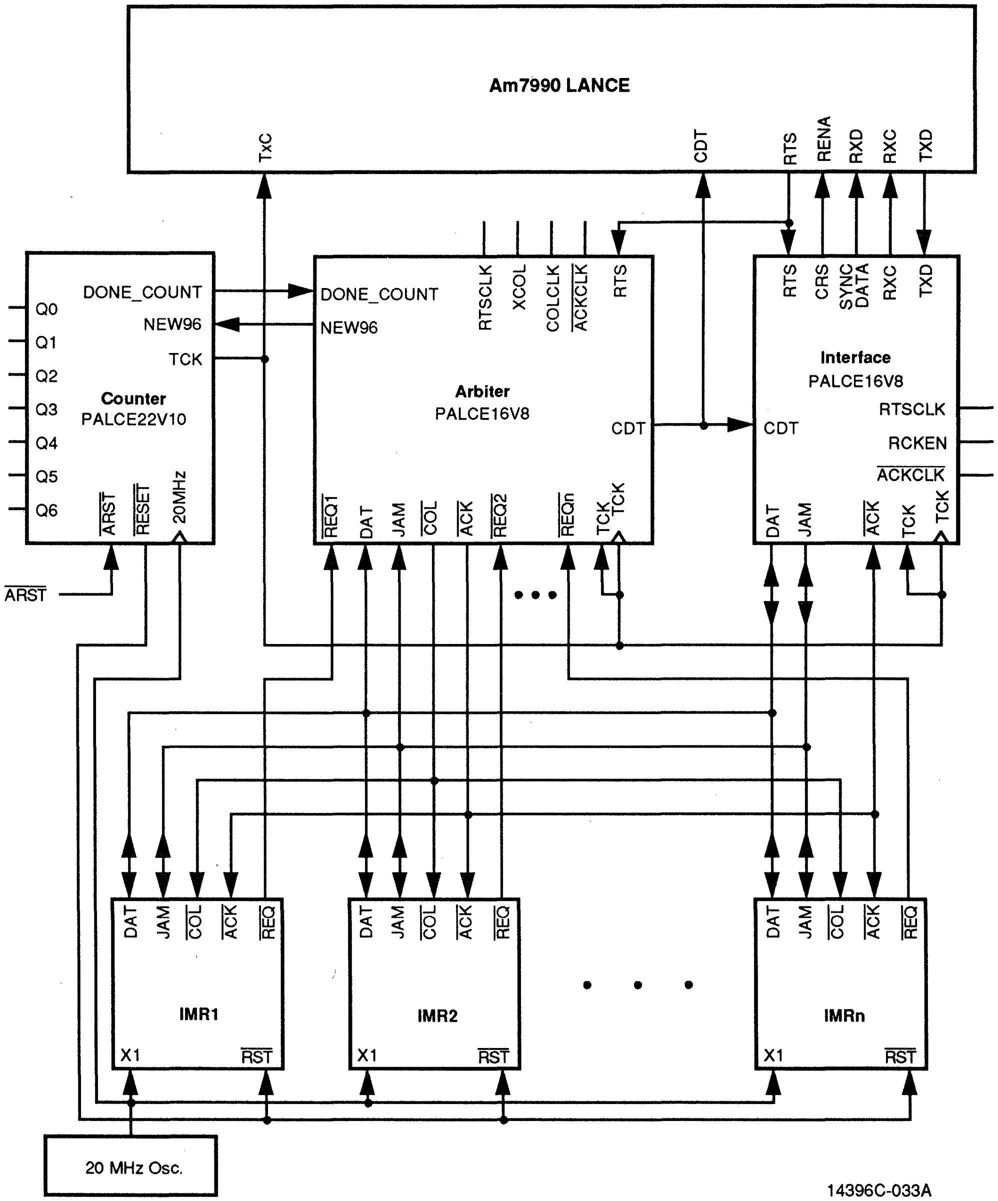


Figure 2. Expandable Modular Repeater

ARBITER LOGIC EQUATIONS

$\text{ACK} = \text{REQ1} * \overline{\text{REQ2}} * \ldots \overline{\text{REQn}} * \overline{\text{RTS}}$	ACK is asserted when one and only one
$+ \overline{\text{REQ1}} * \text{REQ2} * \ldots \overline{\text{REQn}} * \overline{\text{RTS}}$	expansion bus request is active
$+ \quad \bullet$	
$+ \overline{\text{REQ1}} * \overline{\text{REQ2}} * \ldots \text{REQn} * \overline{\text{RTS}}$	
$+ \overline{\text{REQ1}} * \overline{\text{REQ2}} * \ldots \overline{\text{REQn}} * \text{RTS}$	
$+ \overline{\text{REQ1}} * \overline{\text{REQ2}} * \ldots \overline{\text{REQn}} * \text{RTSCLK} * \text{ACKCLK} * \overline{\text{CDT}}$	Term to extend ACK when MAC device finishes sourcing data onto expansion bus
$\text{ACKCLK} := \text{ACK}$	Clock delayed ACK signal (ACK*ACKCLK defines first clock period with valid DAT and JAM)
$\text{COL} = \overline{\text{ACK}} * (\text{REQ1} + \text{REQ2} + \ldots \text{REQn} + \text{RTS})$	COL is asserted when more than one expansion bus request is active
$\text{COLCLK} := \text{COL}$	Clock delayed COL signal
$\text{CDT} :=$	CDT causes the MAC device to back off/stay off the expansion bus
COL	Arbiter detects collision
$+ \text{ACK} * \text{ACKCLK} * \text{JAM}$	Single IMR collision
$+ \text{COLCLK} * \text{ACK}$	Holds CDT active during 'dead' clock cycle
$+ \overline{\text{DONE\_COUNT}} * \text{CDT}$	Maintains CDT (suppresses new MAC requests) until 96 bit timeout
$\text{XCOL} :=$	Present transmit collision state
$\text{ACK} * \text{ACKCLK} * \text{JAM} * \overline{\text{DAT}}$	Ongoing single IMR transmit collision
$+ \text{COL}$	Ongoing multiple IMR transmit collision
$+ \text{COLCLK} * \text{ACK}$	Ongoing multiple IMR transmit collision ('dead' clock cycle)
$\text{RTSCLK} := \text{RTS}$	Clock delayed RTS signal (RTS*RTSCLK defines first recognized RTS from MAC)
$\text{NEW96} :=$	Triggers or re–triggers counter (96 bit times)
$\text{ACK} * \overline{\text{ACKCLK}} * \overline{\text{CDT}}$	New repeater data (start of repeated packet, ACK∠ with no existing collision) (Trigger only)
$+ \text{ACK} * \underline{\text{ACKCLK}} * \text{JAM} * \overline{\text{DAT}} * \overline{\text{XCOL}}$	New transmit collision (Single IMR)
$+ \text{COL} * \overline{\text{XCOL}}$	New transmit collision (Multi-IMR)

INTERFACE LOGIC EQUATIONS

DAT.TRST = ACK * ACKCLK * RTSCLK	Enabled if MAC request gets arbiter ACK (delayed)
DAT = SYNCDATA * $\overline{\text{CLK}}$	Synchronized and latched MAC data (if enabled)
+ DAT * CLK	guarantees hold time for slave IMRs
+ DAT * SYNCDATA	
+ CDT	Overriding HIGH if collision exists (if enabled)
JAM.TRST = ACK * ACKCLK * RTSCLK	Enabled if MAC request gets arbiter ACK (delayed)
JAM = CDT	HIGH if collision exists
ACKCLK := ACK	Clock delayed ACK signal (ACK*ACKCLK defines first clock period with valid DAT and JAM)
CRS = ACK * ACKCLK * $\overline{\text{JAM}}$ * $\overline{\text{CDT}}$	Single active IMR or MAC sourcing data
+ RCKEN	Term to extend CRS at end of MAC receive
SYNCDATA :=	
$\overline{\text{RTS}}$ * DAT * ACK * ACKCLK	Synchronized receive data for the MAC
+ TXD * RTS	Synchronized TxD data for MAC transmit
RXC = RCKEN * $\overline{\text{CLK}}$	Inverted clock for synchronized data for MAC receive
RCKEN := ACK * ACKCLK * $\overline{\text{RTS}}$ * $\overline{\text{RTSCLK}}$ * $\overline{\text{JAM}}$	RCLK enabled if non–MAC data transfer (MAC is slave device)
+ ACK * RTS * $\overline{\text{CDT}}$	RCLK enabled on transmit to allow data loopback to MAC
RTSCLK := RTS	Clock delayed RTS signal (RTS*RTSCLK defines first recognized RTS from MAC)

COUNTER LOGIC EQUATIONS

Upon RESET, the counter is initialized to the terminal value of 60H (96), and remains there until triggered. If (re)triggered by NEW96, the counter is reset to 00H. The counter will count at 10 MHz until it reaches the terminal value, and will stop there until triggered.TCK and $\overline{\text{RESET}}$ are also generated here.

TCK	$:= \overline{\text{TCK}} * \overline{\text{RESET}} * \overline{\text{ARST}}$	Generates TCK for MAC use
RESET	$:= \text{ARST}$	Synchronizes the async reset signal
Q0	$:= \overline{\text{Q0}} * \overline{\text{TCK}} * \overline{\text{DONE\_COUNT}} * \overline{\text{RESET}} * \overline{\text{NEW96}}$	Toggle if still counting LOW at reset, LOW when triggered by NEW96
	$+ \text{Q0} * (\text{TCK} + \text{DONE\_COUNT}) * \overline{\text{RESET}} * \overline{\text{NEW96}}$	Stop at terminal value
Q1	$:= \overline{\text{Q1}} * \text{Q0} * \overline{\text{TCK}} * \overline{\text{DONE\_COUNT}} * \overline{\text{RESET}} * \overline{\text{NEW96}}$	LOW at reset, LOW when (re)triggered by NEW96
	$+ \text{Q1} * (\overline{\text{Q0}} + \text{TCK} + \text{DONE\_COUNT}) * \overline{\text{RESET}} * \overline{\text{NEW96}}$	
Q2	$:= \overline{\text{Q2}} * \text{Q1} * \text{Q0} * \overline{\text{TCK}} * \overline{\text{DONE\_COUNT}} * \overline{\text{RESET}} * \overline{\text{NEW96}}$	LOW at reset, LOW when (re)triggered by NEW96
	$+ \text{Q2} * (\overline{\text{Q1}} + \overline{\text{Q0}} + \text{TCK} + \text{DONE\_COUNT}) * \overline{\text{RESET}} * \overline{\text{NEW96}}$	
Q3	$:= \overline{\text{Q3}} * \text{Q2} * \text{Q1} * \text{Q0} * \overline{\text{TCK}} * \overline{\text{DONE\_COUNT}} * \overline{\text{RESET}} * \overline{\text{NEW96}}$	LOW at reset, LOW when (re)triggered by NEW96
	$+ \text{Q3} * (\overline{\text{Q2}} + \overline{\text{Q1}} + \overline{\text{Q0}} + \text{TCK} + \text{DONE\_COUNT}) * \overline{\text{RESET}} * \overline{\text{NEW96}}$	
Q4	$:= \overline{\text{Q4}} * \text{Q3} * \text{Q2} * \text{Q1} * \text{Q0} * \overline{\text{TCK}} * \overline{\text{DONE\_COUNT}} * \overline{\text{RESET}} * \overline{\text{NEW96}}$	LOW at reset, LOW when (re)triggered by NEW96
	$+ \text{Q4} * (\overline{\text{Q3}} + \overline{\text{Q2}} + \overline{\text{Q1}} + \overline{\text{Q0}} + \text{TCK} + \text{DONE\_COUNT}) * \overline{\text{RESET}} * \overline{\text{NEW96}}$	
Q5	$:= \overline{\text{Q5}} * \text{Q4} * \text{Q3} * \text{Q2} * \text{Q1} * \text{Q0} * \overline{\text{TCK}} * \overline{\text{DONE\_COUNT}} * \overline{\text{NEW96}}$	Toggle if all lower bits HIGH and still counting LOW if (re)triggered by NEW96
	$+ \text{Q5} * (\overline{\text{Q4}} + \overline{\text{Q3}} + \overline{\text{Q2}} + \overline{\text{Q1}} + \overline{\text{Q0}} + \text{TCK} + \text{DONE\_COUNT}) * \overline{\text{NEW96}}$	Stay if any lower bit is LOW or if done counting "
	$+ \text{RESET}$	HIGH at reset
Q6	$:= \overline{\text{Q6}} * \text{Q5} * \text{Q4} * \text{Q3} * \text{Q2} * \text{Q1} * \text{Q0} * \overline{\text{TCK}} * \overline{\text{DONE\_COUNT}} * \overline{\text{NEW96}}$	LOW when (re)triggered by NEW96
	$+ \text{Q6} * (\overline{\text{Q5}} + \overline{\text{Q4}} + \overline{\text{Q3}} + \overline{\text{Q2}} + \overline{\text{Q1}} + \overline{\text{Q0}} + \text{TCK} + \text{DONE\_COUNT}) * \overline{\text{NEW96}}$ "	
	$+ \text{RESET}$	HIGH at reset
DONE_COUNT	$= \text{Q6} * \text{Q5} * \overline{\text{Q4}} * \overline{\text{Q3}} * \overline{\text{Q2}} * \overline{\text{Q1}} * \overline{\text{Q0}}$	Terminal count (96)

Management Port

The IMR 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 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 to be compatible with 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. 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.

AUI Port Disable

SI data: 0 0 1 0 1 1 1 1
SO data: None

The AUI port will be disabled upon receiving this command. Subsequently, the IMR 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.

AUI Port Enable

SI data: 0 0 1 1 1 1 1 1
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.

TP Port Disable

SI data: 0 0 1 0 0 b b b
(b b b is TP port #)
SO data: None

The TP port designated in the command byte will be disabled upon receiving this command. Subsequently, the IMR 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 enter the Link Fail state and to have its partitioning state machine returned to its idle state.

TP Port Enable

SI data: 0 0 1 1 0 b b b
(b b b is TP port #)
SO data: None

This command enables a previously disabled TP port. Note that to force a TP port into the Link Fail state and/or to reconnect a partitioned TP port, the port can first be disabled (TP Port Disable Command) and then re-enabled with this command.

AUI Port Partitioning Status

SI data: 1 0 0 0 1 1 1 1
SO data: P 0 0 0 0 0 0 0

P = 0 – Partitioned
P = 1 – Connected

The Partitioning Status of the AUI port is accessed by this command. If a port is disabled, reading its partitioning status will indicate that it is connected.

TP Port Partitioning Status

SI data: 1 0 0 0 0 0 0 0
SO data: $P_7\ P_6\ P_5\ P_4\ P_3\ P_2\ P_1\ P_0$

$P_n = 0$ TP port n partitioned
$P_n = 1$ TP port n connected

The Partitioning Status of all eight TP ports are accessed by this command. If a port is disabled, reading its partitioning status will indicate that it is connected.

Alternate Reconnection Algorithm (AUI Port)

SI data: 0 0 0 1 1 1 1 1
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.

Alternate Reconnection Algorithm (TP Ports)

SI data:	0 0 0 1 0 0 0 0
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.

Link Test Status of TP Ports

SI data:	1 1 0 1 0 0 0 0
SO data:	L_7 L_6 L_5 L_4 L_3 L_2 L_1 L_0
$L_n = 0$	TP Port n in Link Test Fail
$L_n = 1$	TP Port n in Link Test Pass

The Link Test Statuses of all eight TP ports are accessed by this command. If a twisted pair port is disabled, reading its Link Test Status indicates it being in Link Test Pass state. Upon reenabling, the port will be forced into Link Fail.

Disable Link Test Function

SI data: 0 1 0 0 0 b b b
(b b b is TP port #)
SO data: None

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 Test 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

SI data: 0 1 0 1 0 b b b
(b b b is TP port #)
SO data: None

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.

Polarity Status of TP Ports

SI data:	1 1 1 0 0 0 0 0
SO data:	P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0
$P_n = 0$	TP Port n polarity correct
$P_n = 1$	TP Port n polarity reversed

The statuses of all eight TP port polarities are accessed with this command. The IMR 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 will set the appropriate bit in this command's results byte only if the Polarity Reversal Function is enabled for that port.

Enable Automatic Receiver Polarity Reversal

SI data: 0 1 1 1 0 b b b
(b b b is TP port #)
SO data: None

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 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.

Disable Automatic Receiver Polarity Reversal

SI data: 0 1 1 0 0 b b b
(b b b is TP port #)
SO data: None

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 reverse polarity packets would be repeated to all other network ports in the IMR as inverted data.

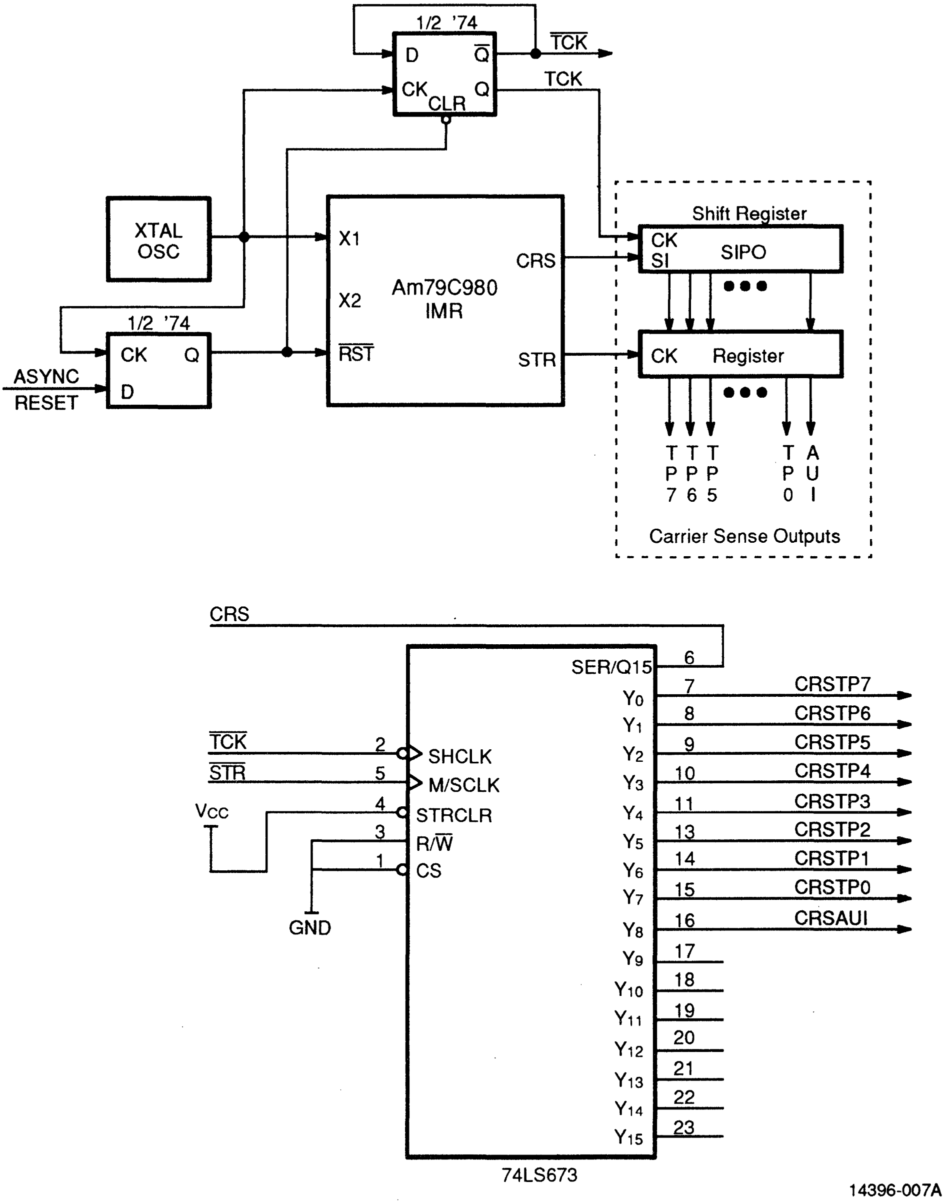
Table 2. Management Port Command Summary

Command	SI data	SO data
AUI Port Disable	00101111	–
AUI Port Enable	00111111	–
TP Port Disable	00100bbb	–
TP Port Enable	00110bbb	–
AUI Port Partitioning Status	10001111	P0000000
TP Port Partitioning Status	10000000	$P_7 - P_0$
Alternate Reconnection Algorithm (AUI)	00011111	–
Alternate Reconnection Algorithm (TP)	00010000	–
Link Test Status of TP Ports	11010000	$L_7 - L_0$
Disable Link Test Function	01000bbb	–
Enable Link Test Function	01010bbb	–
Polarity Status of TP Ports	11100000	$P_7 - P_0$
Enable Automatic Receiver Polarity Reversal	01110bbb	–
Disable Automatic Receiver Polarity Reversal	01100bbb	–

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 diagram below 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 microsecond). 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.



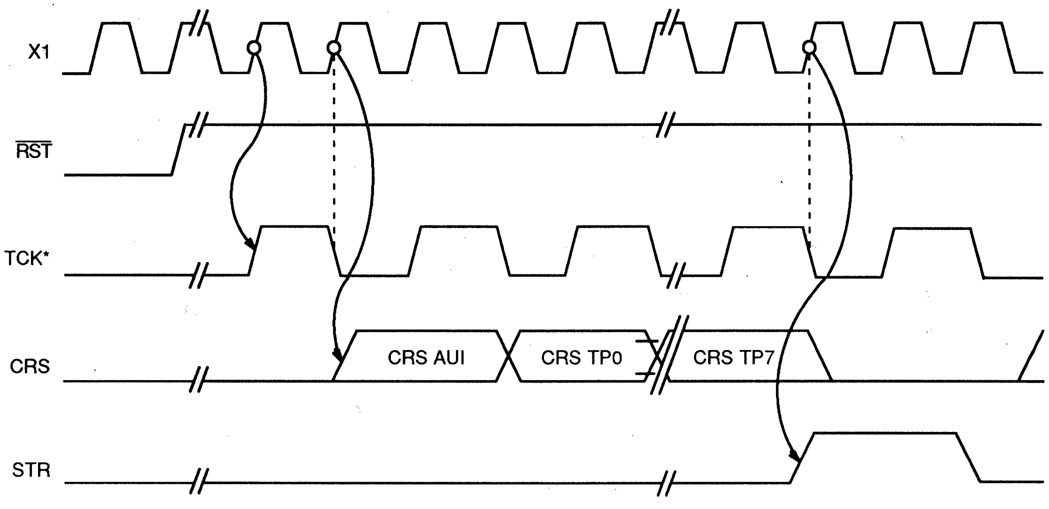
Single IC Implementation of the SIPO and Output Register.

Figure 4a. Port Activity Monitor Implementation

Loopback Test Mode

The IMR 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 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.

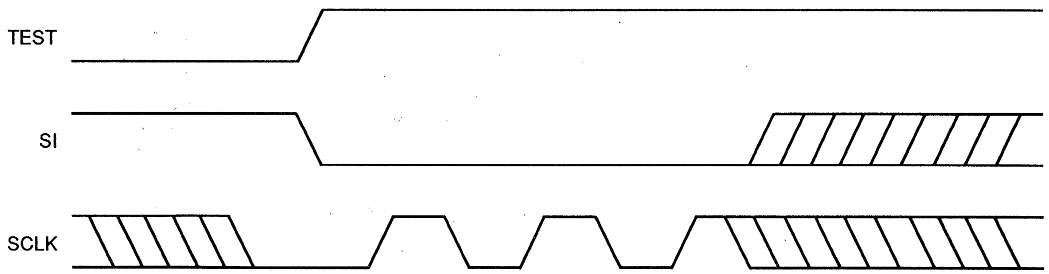
The IMR will return to normal operation when the TEST pin is again driven LOW.



*Externally generated signal illustrates internal IMR clock phase relationship.

14396C-008B

Figure 4b. Port Activity Monitor Implementation (Continued)



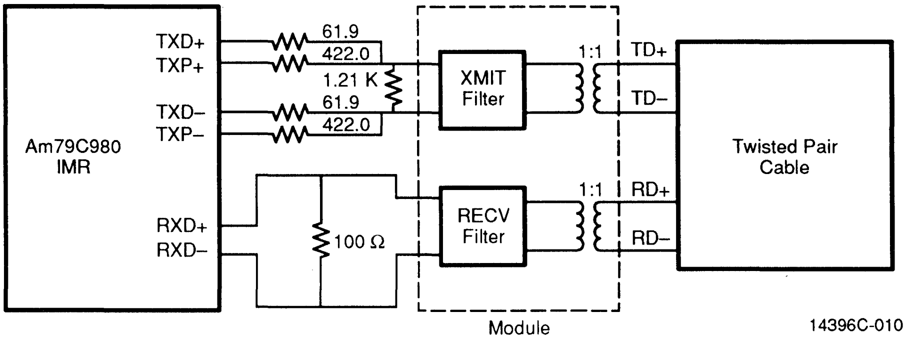
14396-009B

Figure 5. Programming the IMR for Loopback Mode

IMR External Components

Fig. 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 modulĕ that combines the functions of the transmit and receive filters and the pulse transformers into one package.

The AUI port, if used in a repeater design, should comply with IEEE 802.3. This is accomplished through the use of standard AUI pulse transformers and drop cable termination networks.



The Filter/Transformer Module shown is available from the following manufacturers:

Belfuse	TDK
Pulse Engineering	PCA
Valor Electronics	Nano Pulse

Figure 6a. Typical TP Port External Components

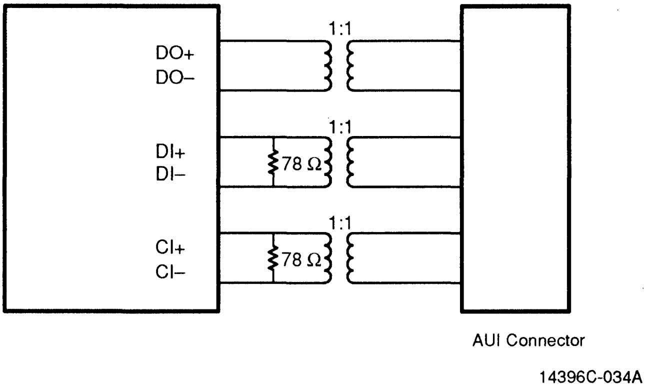
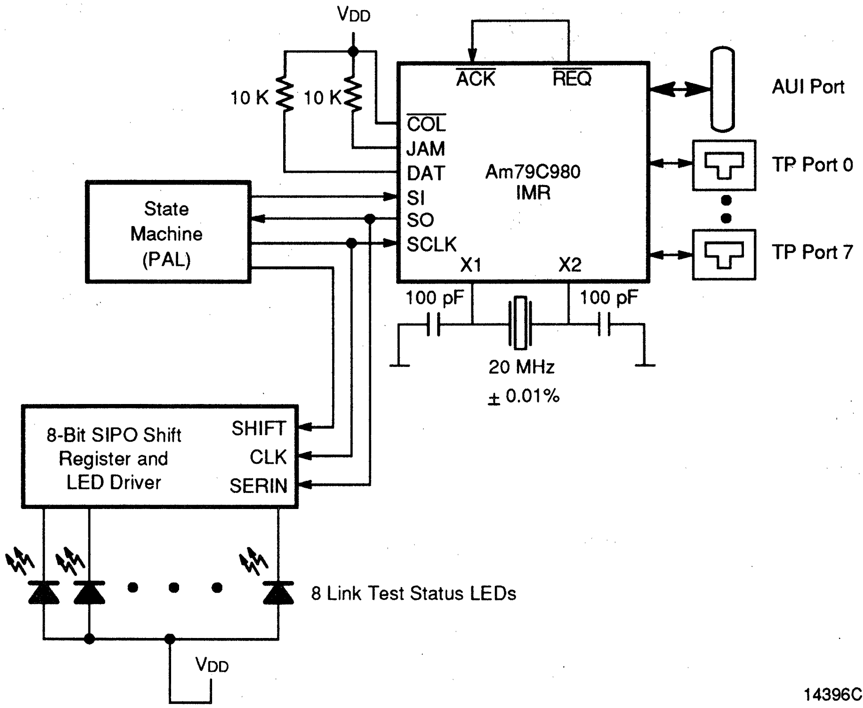


Figure 6b. Typical AUI Port Components

APPLICATIONS



14396C-011B

Note: Unused receiver pairs (DI +/–, CI +/–, RXDn +/–) should be shorted together.

Figure 7. Low Cost 10BASE-T Repeater

APPLICATIONS (Continued)

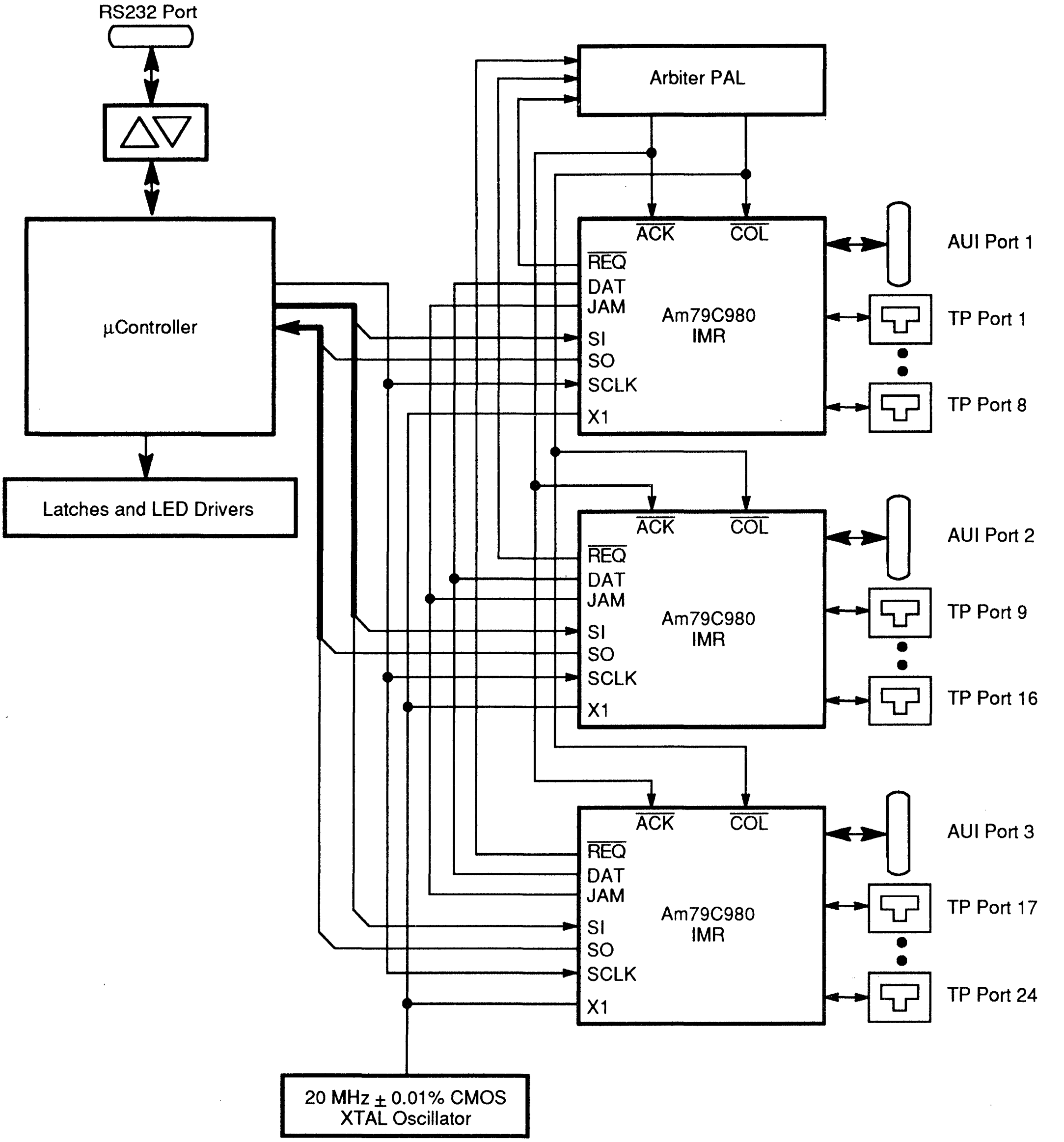


Figure 8. Intelligent Multi-IMR Based Repeater

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 functionality 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_{ADDI}	DO Differential Output Voltage Imbalance	R_L = 78 Ω	−25	+25	mV
V_{AOD}OFF	DO Differential Idle Output Voltage	R_L = 78 Ω	−40	+40	mV
I_{AOD}OFF	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_{RST}	Reset pulse width ($\overline{\text{RST}}$ pin LOW)		150	–	µs
t_{RSTSET}	$\overline{\text{RST}}$ HIGH setup time with respect to X1 Clock		20	–	ns
t_{RSTHLD}	$\overline{\text{RST}}$ 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	–	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
Expansion Port					
t_{X1HRL}	X1 rising edge to $\overline{\text{REQ}}$ driven LOW	C_L = 100 pF	14	40	ns
t_{X1HRH}	X1 rising edge to $\overline{\text{REQ}}$ 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
t_{DJSET}	DAT/JAM setup time		10	–	ns
t_{DJHOLD}	DAT/JAM hold time		14	–	ns
t_{CASET}	$\overline{\text{COL/ACK}}$ setup time		5	–	ns
t_{CAHOLD}	$\overline{\text{COL/ACK}}$ hold time		14	–	ns

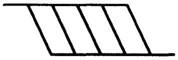
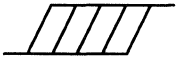
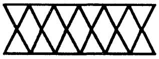
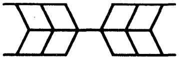
SWITCHING CHARACTERISTICS (Continued)

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
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	$\lvert V_{IN}\rvert > \lvert V_{ASQ}\rvert$ Note 3	15	45	ns
t_{PWKDI}	DI pulse width maintain/turn-off threshold	$\lvert V_{IN}\rvert > \lvert V_{ASQ}\rvert$ Note 4	136	200	ns
t_{PWOCI}	CI pulse width accept/reject threshold	$\lvert V_{IN}\rvert > \lvert V_{ASQ}\rvert$ Note 5	10	26	ns
t_{PWKCI}	CI pulse width maintain/turn-off threshold	$\lvert V_{IN}\rvert > \lvert V_{ASQ}\rvert$ Note 6	90	160	ns
Twisted Pair Ports					
t_{TXTD}	X1 rising edge to TXD+,TXP+ TXD-,TXP- transition delay		–	45	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	$\lvert V_{IN}\rvert > \lvert V_{THS}\rvert$ Note 7	136	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

Notes:

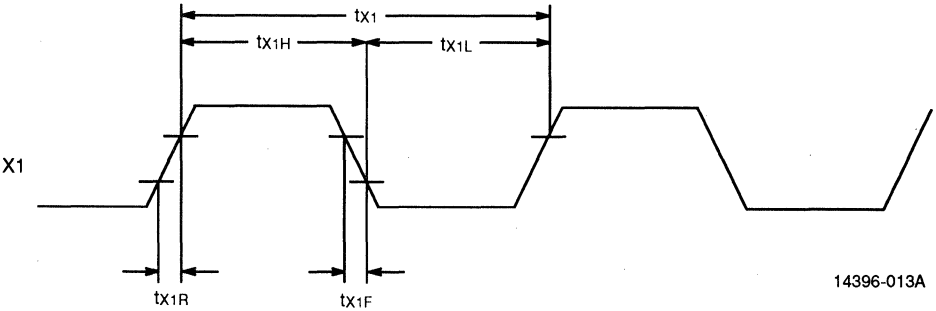
1. Parameter not tested.
2. Uses switching test load.
3. DI pulses narrower than t_{PWODI} (min) will be rejected; pulses wider than t_{PWODI} (max) will turn internal DI carrier sense on.
4. DI pulses narrower than t_{PWKDI} (min) will maintain internal DI carrier sense on; pulses wider than t_{PWKDI} (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 6, 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; 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

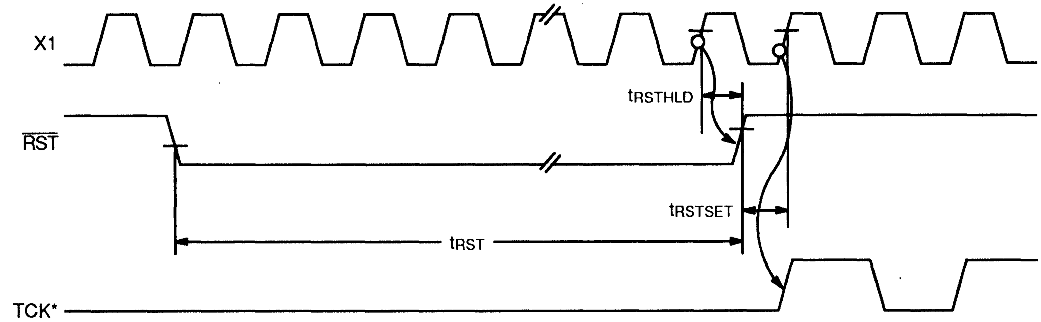
SWITCHING WAVEFORMS



14396-013A

Clock Timing

SWITCHING WAVEFORMS (Continued)



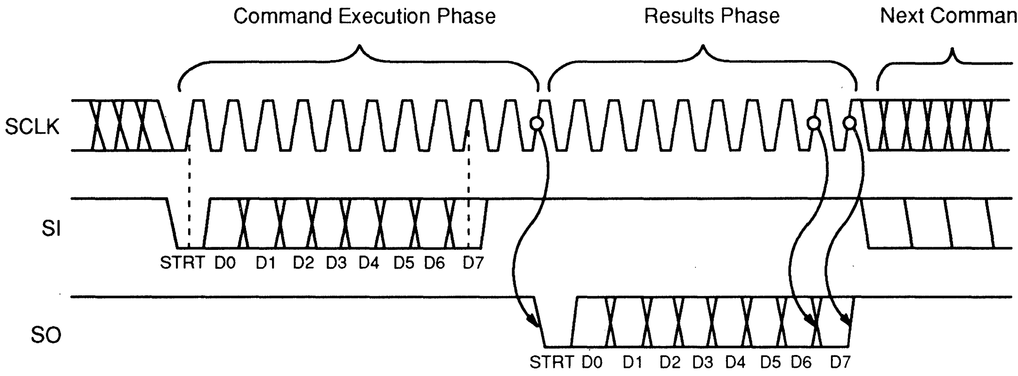
Note:

tRSTSET refers to synchronous Reset Timing.

*Externally generated (Figure 4) signal illustrates internal IMR clock phase relationships.

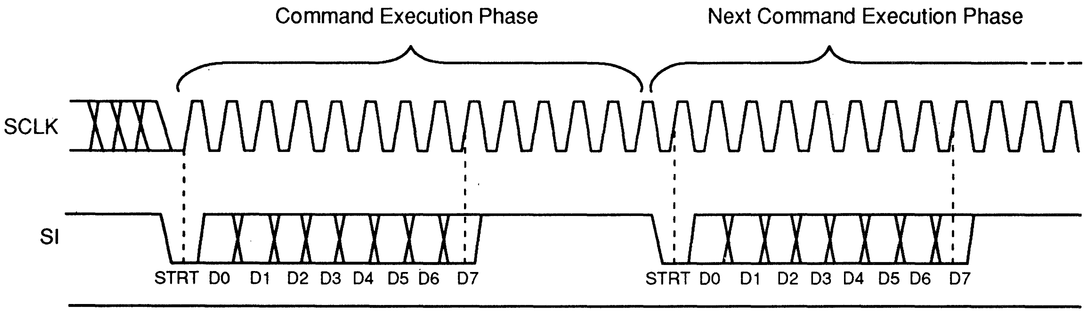
14396C-014B

Reset Timing



14396C-015B

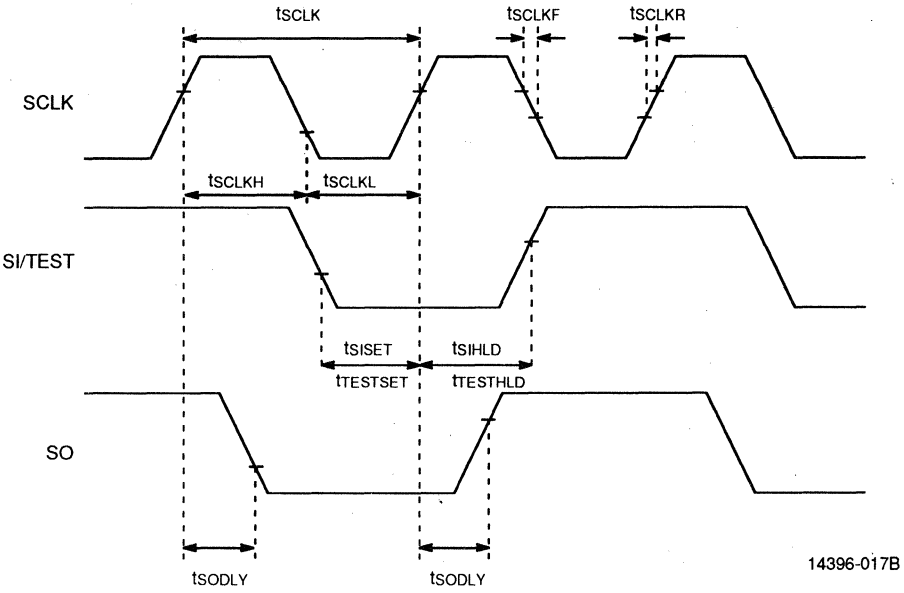
Management Command with Results Data


Command Execution Phase
Next Command Execution Phase
SCLK
SI
STRT D0 D1 D2 D3 D4 D5 D6 D7
STRT D0 D1 D2 D3 D4 D5 D6 D7
SO

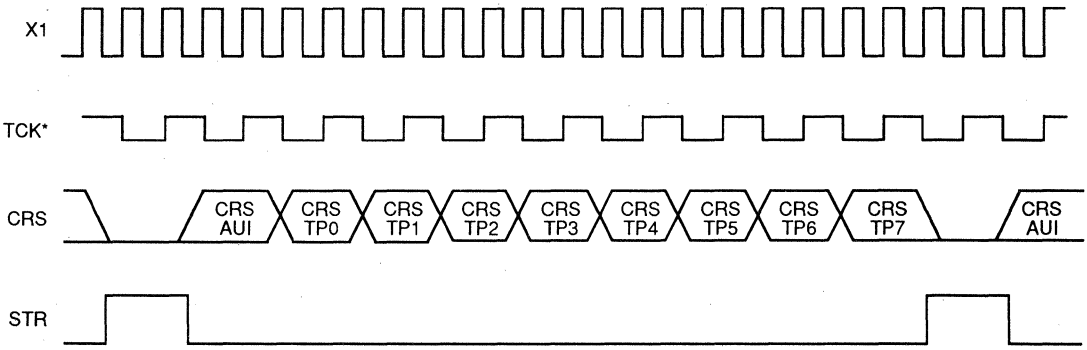
14396C-016B

No-results Management Command Timing

SWITCHING WAVEFORMS (Continued)



Management Port Clock Timing

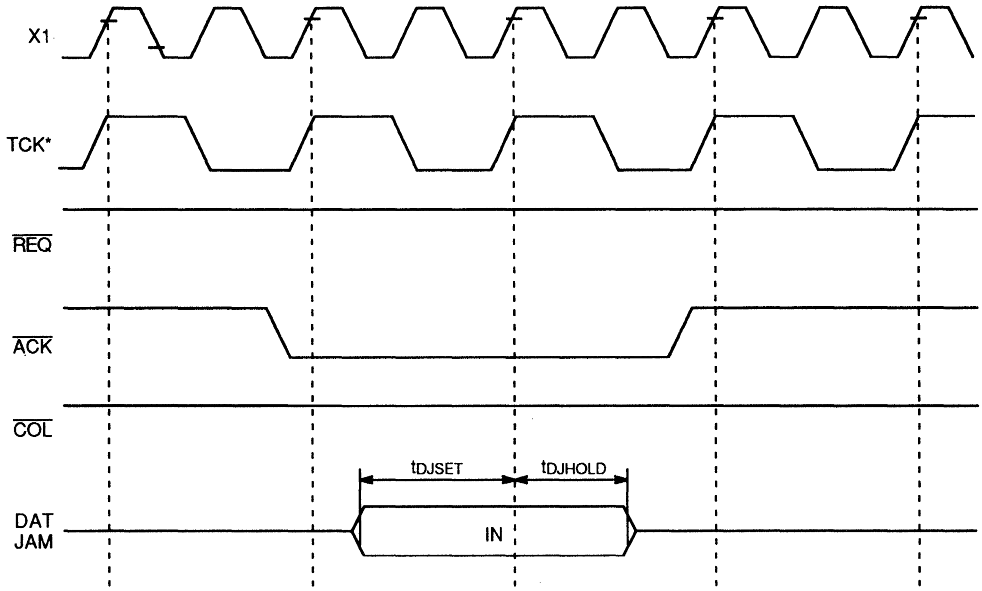


*Externally generated (Figure 4) signal illustrates internal IMR clock phase relationships.

14396C-018B

Carrier Sense Timing

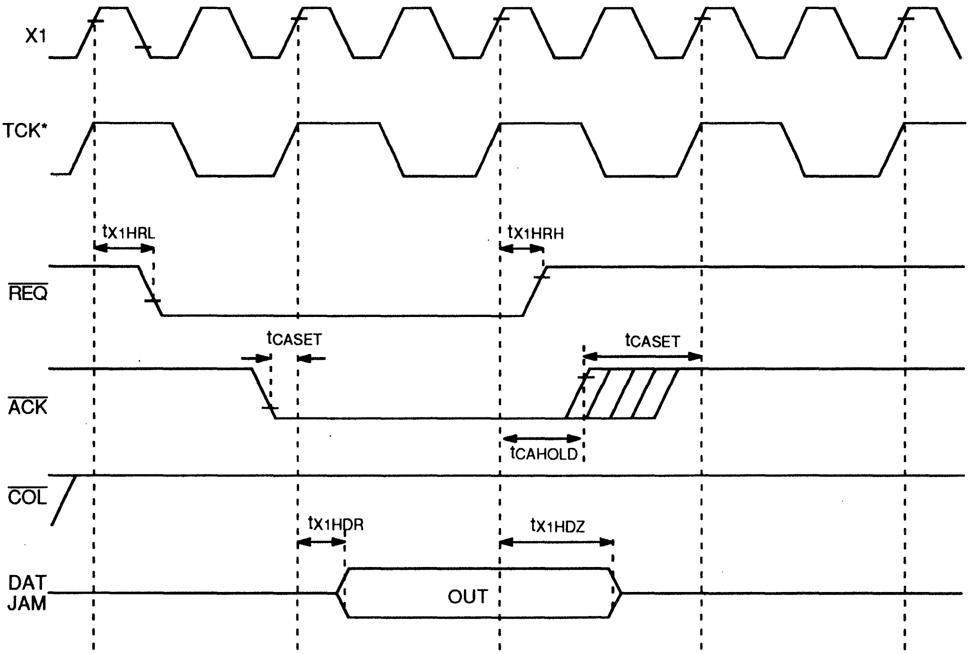
SWITCHING WAVEFORMS (Continued)



*Externally generated (Figure 4) signal illustrates internal IMR clock phase relationships.

14396C-019B

Expansion Port Input Timing

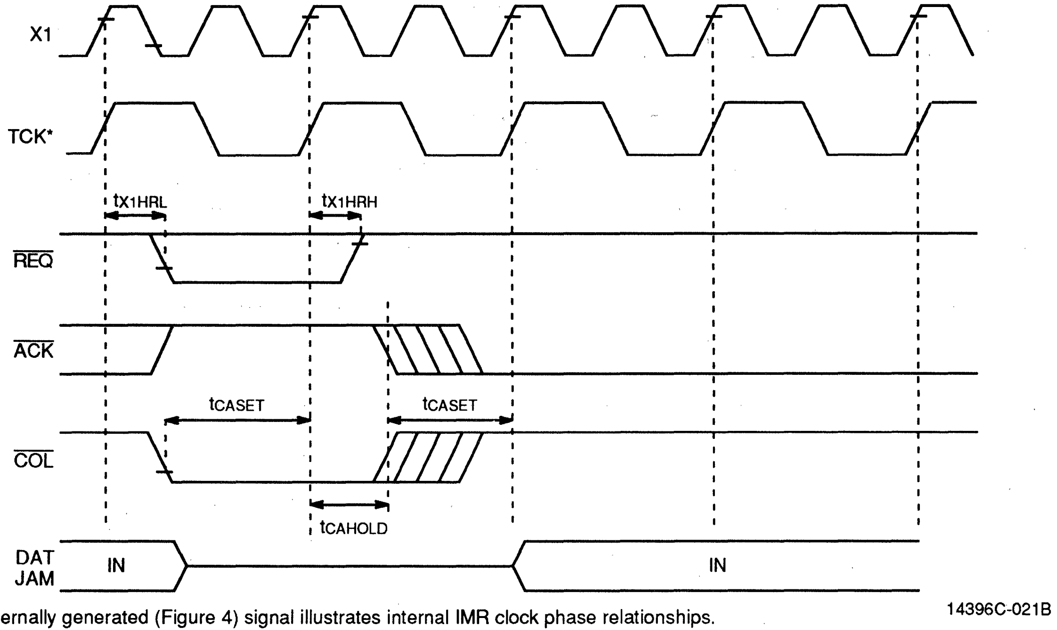


*Externally generated (Figure 4) signal illustrates internal IMR clock phase relationships.

14396C-020B

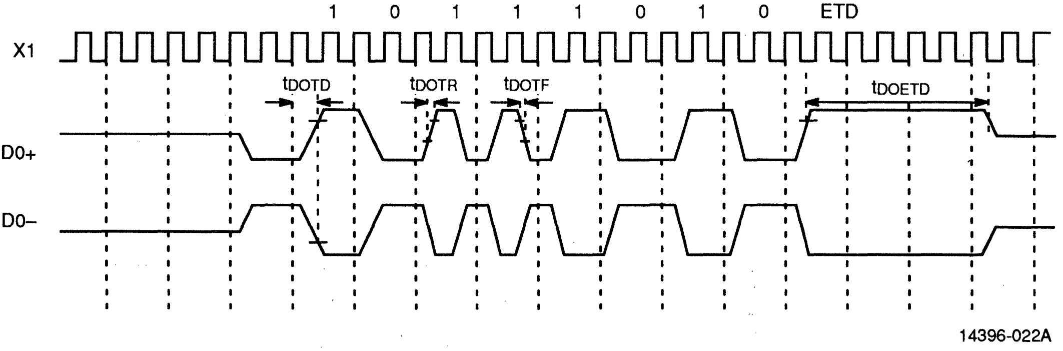
Expansion Port Output Timing

SWITCHING WAVEFORMS (Continued)

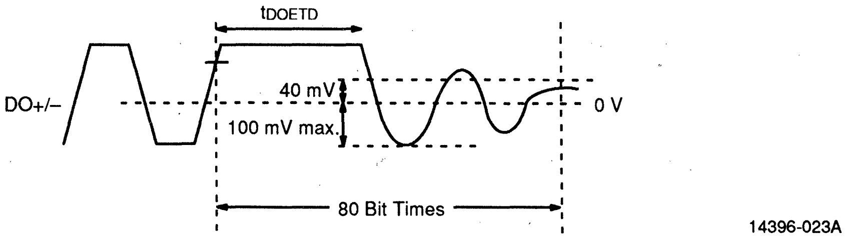


*Externally generated (Figure 4) signal illustrates internal IMR clock phase relationships.

Expansion Port Collision Timing

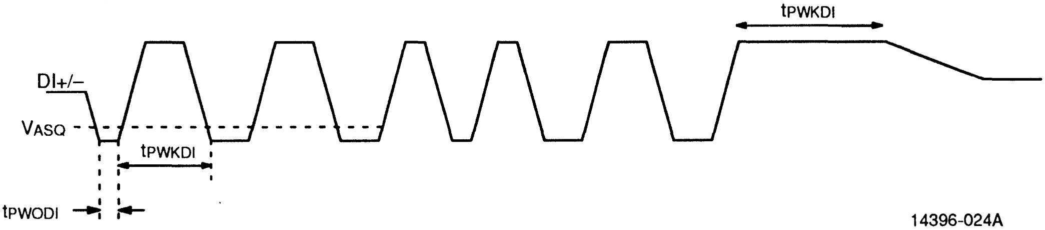


AUI DO Timing Diagram



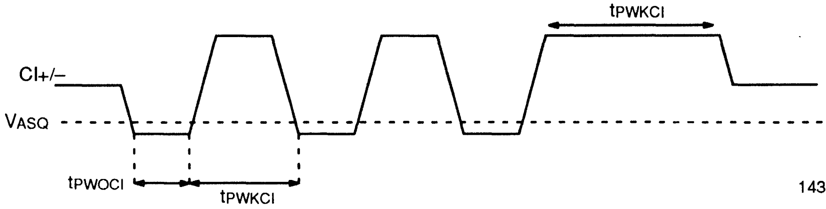
AUI Port DO ETD Waveform

SWITCHING WAVEFORMS (Continued)



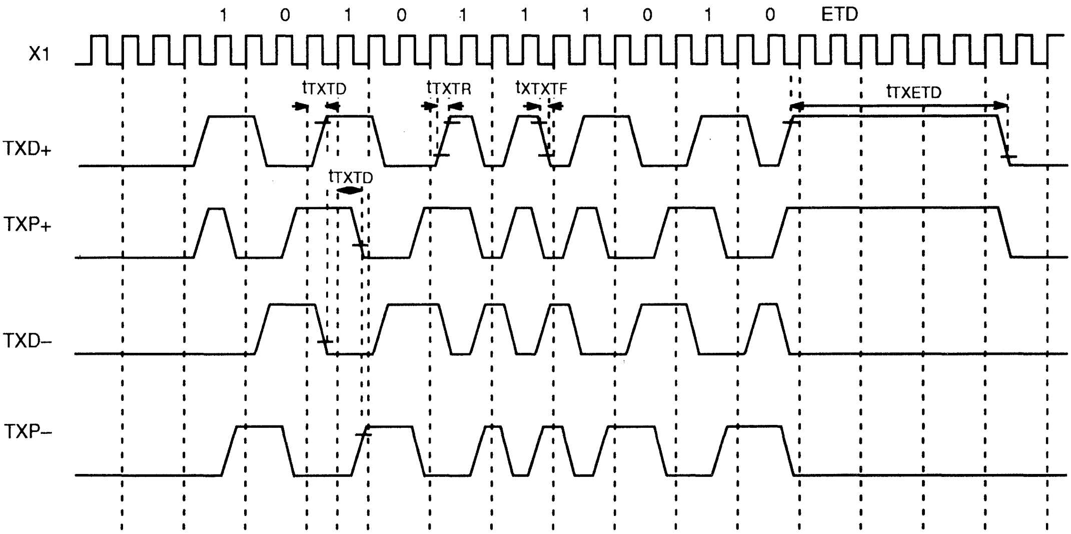
14396-024A

AUI Receive Timing Diagram



14396-025A

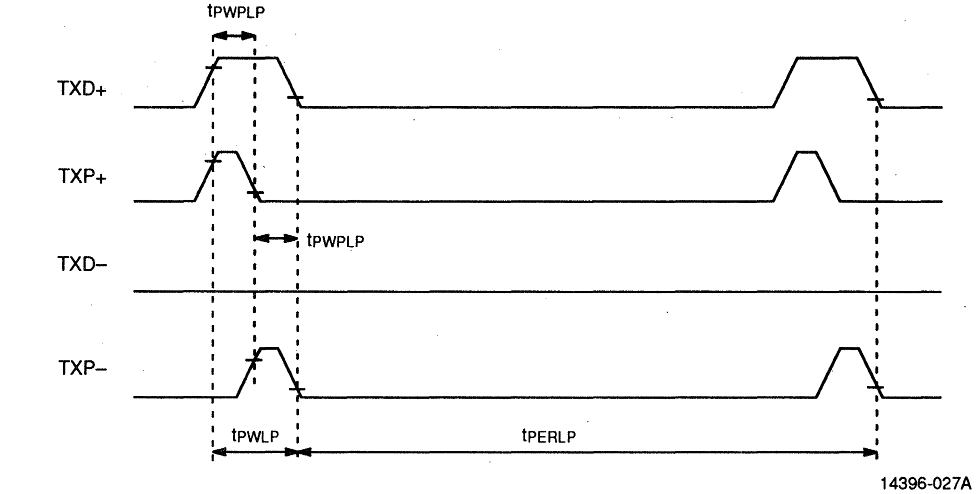
AUI Collision Timing Diagram



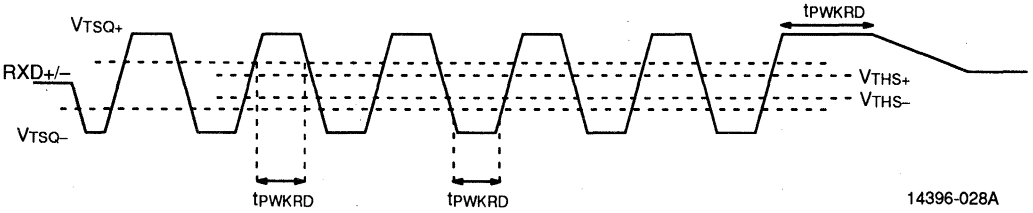
14396-026A

TP Ports Output Timing Diagram

SWITCHING WAVEFORMS (Continued)

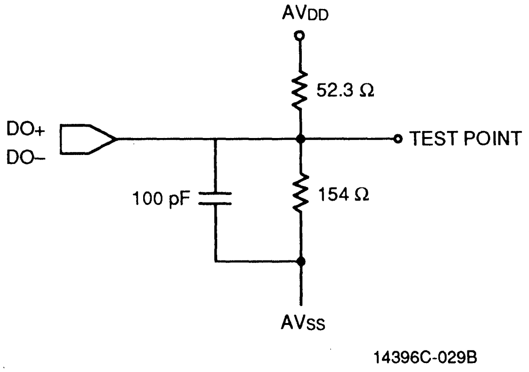


TP Idle Link Test Pulse

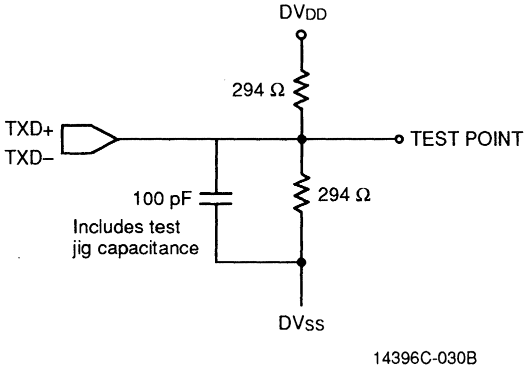


TP Receive Timing Diagram

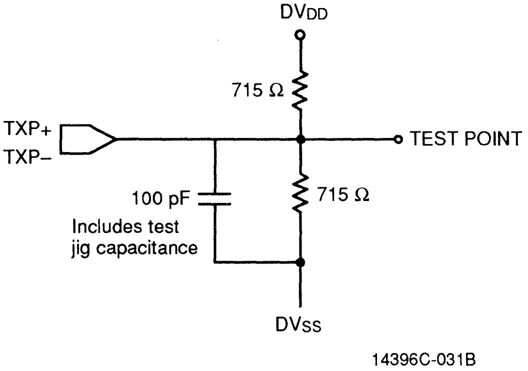
SWITCHING TEST CIRCUITS



AUI DO Switching Test Circuit



TXD Switching Test Circuit



TXP Outputs Test Circuit

APPENDIX

Glossary

Active Status

In a non-collision state, an IMR 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 is generating Jam Sequence), an IMR is considered active if any one or more network ports is receiving data. The IMR asserts the $\overline{REQ}$ line to indicate that it is active.

Collision

In a carrier sense multiple access/collision detect (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 IMRs, 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 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 when a Collision state is detected. This signal is transmitted by the IMR 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 (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.

SECTION 2
Am7990 Technical Manual

TABLE OF CONTENTS

APPENDIXES

CHAPTER 1
OVERVIEW

The LANCE (Local Area Network Controller for Ethernet) is a 48-pin VLSI device designed to greatly simplify interfacing a microcomputer or minicomputer to an IEEE 802.3/ Ethernet/Cheapernet Local Area Network. In addition to transferring data packets to and from the Ethernet transceiver, the LANCE has the ability to manage the data buffers in memory. Figure 1-1 is a block diagram of the LANCE.

This technical manual discusses the hardware and software considerations that may be useful for the user designing systems around the Am7990 LANCE. It examines the LANCE interfaces with typical microprocessors to show the flexibility under different environments. Bus bandwidth, bus latency, and system performance are discussed for different architectures.

The flowcharts of software drivers, along with some guidelines, are introduced for a better understanding of how the part may be programmed and used in typical applications. No attempt is made to explore software issues relating to the upper layer protocols of the ISO model.

As a communication controller, the LANCE manages descriptor rings and data buffers in memory for both transmitting and receiving data. It shares DMA access to this memory with other master bus devices such as CPUs.

In the transmitting mode, the LANCE transfers data from the current buffer to a 48-byte (FIFO) register in the LANCE called a Silo. The output of the Silo is serialized and then goes to a Serial Interface Adapter (SIA), the Am7992B. Design considerations for the SIA are not discussed in this manual. The output of the SIA is connected to the Ethernet lines through a transceiver. A pin-for-pin compatible IEEE 802.3/ Ethernet/Cheapernet transceiver, Am7996, is available from AMD.

In the receiving mode, the SIA receives data from the Ethernet cable. It transfers the data to the Silo in the LANCE. The LANCE transfers the data from the Silo to the current buffer in local memory. Figure 1-2 is a simplified diagram of the data flow.

To initialize the LANCE, it must be in the Bus Slave mode. When $\overline{CS}$ is active, the LANCE is in the Bus Slave mode.

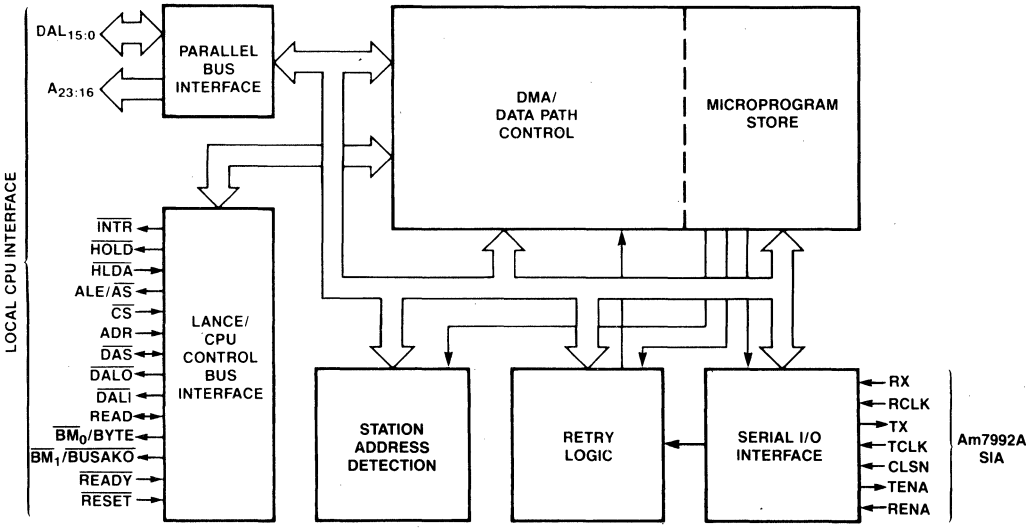


Figure 1-1 LANCE Block Diagram

Publication# 06363 Rev. C Amendment/0
Issue Date: December 1986

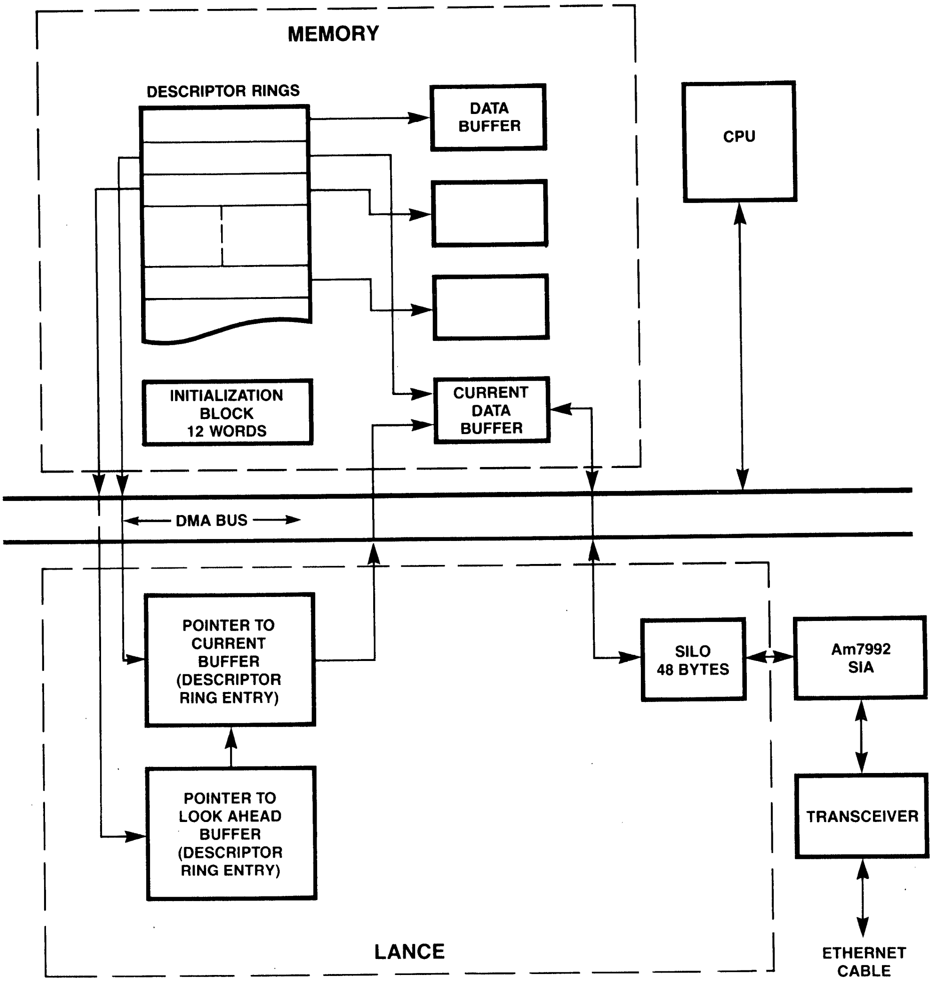

MEMORY
DESCRIPTOR RINGS
DATA BUFFER
CPU
INITIALIZATION BLOCK 12 WORDS
CURRENT DATA BUFFER
DMA BUS
POINTER TO CURRENT BUFFER (DESCRIPTOR RING ENTRY)
SILO 48 BYTES
Am7992 SIA
POINTER TO LOOK AHEAD BUFFER (DESCRIPTOR RING ENTRY)
TRANSCEIVER
LANCE
ETHERNET CABLE
06363A-1

Figure 1-2 Data Flow

The LANCE is initialized by entering configuration parameters into the four Control and Status Registers (CSR) and into an Initialization Block of 12 words. CSR_0 contains interrupt and error status flags and control bits. CSR_1 and CSR_2 contain the address of the Initialization Block. CSR_3 contains control bits to configure Byte Swap (BSWP), ALE Control (ACON), and Byte Control (BCON). The Initialization Block contains pointers to the descriptor rings and their length.

Buffer management is accomplished through message descriptors organized in ring structures in memory. Each message descriptor entry is four words long. There are separate descriptor rings for transmitting and receiving. Packets to be transmitted are loaded into buffers by the CPU and the addresses and lengths of the buffers are stored in consecutive entries in the Transmit Descriptor ring. Then, the ownership bit is set in each entry indicating that the LANCE owns the buffers. The LANCE polls the first descriptor entry and when it finds that it owns it, it proceeds to transmit the packet.

The LANCE maintains a copy of the descriptor entry that points to the current buffer. If the current buffer entry is not the final buffer for the packet being sent, the LANCE maintains a copy of the descriptor ring entry for the next buffer to be transmitted in the look-a-head buffer.

When a Receive Descriptor Ring entry is empty, the ownership bit for it is set meaning that the buffer it points to is available to the LANCE to store an incoming packet. After the LANCE has stored a packet in the buffer or filled the buffer, it resets the ownership bit thus giving control of the buffer to the host. It then polls the next descriptor entry to see if the next buffer is available for additional incoming packets.

The LANCE also enters status information into the descriptor entries.

It is recommended that the reader refer to the LANCE Data Sheet for further detail such as timing.

1.1 EARLY DESIGN STAGE TESTING

As with any other peripherals, a hierarchical approach should be taken to integrate the LANCE into a system for proper operation. The following outline shows the order in which the various LANCE functions should be developed and tested.

— RESET
— BUS SLAVE OPERATION
— INITIALIZATION
— BUS MASTER OPERATION
— DIAGNOSTICS
— NORMAL OPERATION

The initialization, Bus Master operation, and the diagnostic steps are closely tied together, such that each step requires the proper operation or arrangement of the other steps. A brief discussion of significant points for each step follows. A detailed description appears later in this chapter.

1.2 RESET

There are two ways to reset the LANCE: Hard Reset ($\overline{\text{RESET}}$ input to the LANCE), and Soft Reset (by setting the stop bit in CSR_0). Reset causes the LANCE to cease operation and clear its internal logic.

Control Status Registers (CSR_0 CSR_3) contents are affected as follows:

— CSR_0 is cleared. (Stop bit gets set.)
— CSR_3 is cleared.
— CSR_1, CSR_2 undefined by reset.

The LANCE operation may be resumed by setting the STRT bit in CSR_0 after it is stopped. However, the above condition implies that when the LANCE is stopped (Soft Reset), CSR_3 must be reprogrammed if the LANCE interface had been configured for non-default values. If CSR_3 has a non-zero value, it must be reloaded each time that the stop bit is set.

1.3 BUS SLAVE OPERATION

The LANCE enters the Bus Slave mode whenever $\overline{\text{CS}}$ becomes active. This mode is used for accessing the four Control Status Registers (CSR_0, CSR_1, CSR_2, CSR_3), and the Register Address Pointer (RAP). It takes the CPU normally two separate cycles to access CSR_0-CSR_3. The first cycle selects the CSR by writing to RAP, and the second cycle accesses the CSR pointed to by RAP. RAP is a latch and is not changed until it is rewritten; therefore, accessing the same CSR does not require the register select cycle.

During Bus Master Mode operation, the LANCE uses the parameters contained in CSR_1-CSR_3. Therefore, LANCE operation must be stopped by setting the Stop bit in CSR_0 whenever CSR_1-CSR_3 accesses are required. The LANCE and CPU interface can be tested by writing and then reading the Control Status Registers (CSR_0-CSR_3).

The CPU communicates with the LANCE in a handshake sequence by asserting the $\overline{\text{DAS}}$ signal as a request for Data Transfer, and receiving the READY output from the LANCE in return as an acknowledge, (refer to the LANCE Data Sheet for Bus Slave Timing and Read/Write operation). Note that there are two different types of delays involved in accessing the CSRs. The shorter delays refer to accesses to CSR_0, CSR_3, and RAP. The longer delays refer to accesses to CSR_1 and CSR_2. Proper operation of the LANCE in Bus Slave Mode is a prerequisite for the LANCE to operate in Bus Master Mode.

1.4 INITIALIZATION PROCEDURE

The LANCE is initialized (configured) for a desired operation by programming the CSR registers (CSR_1-CSR_3) and providing the parameters required by the LANCE in the Initialization Block. Proper initialization is a prerequisite for the LANCE operation in Bus Master mode. Refer to Figure 1-3 and Figure 1-4.

1.4.1 CSR PROGRAMMING

Depending upon which processor is to be interfaced to the LANCE, ACON, BCON, and BSWP in the Control Status Register 3 (CSR_3) must be programmed when initializing the LANCE.

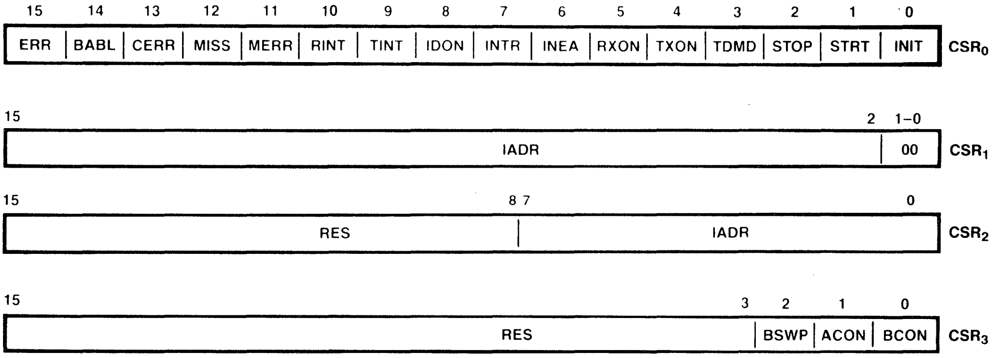
ACON (ALE control) defines the polarity of ALE signal when the LANCE is in Bus Master Mode. Multiplexed buses use this signal (ALE) to latch the least significant part of the Address Bus (A_{15}:A_0).

BCON (byte control) redefines the byte mask I/O pins (pins 15 and 16) for different ways to handle odd address boundaries and redefines $\overline{\text{HOLD}}$ I/O pin 17 to configure the LANCE as a daisy-chain system.

When BCON = 0, pins 15, 16, and 17 are defined as $\overline{\text{BM}}_0$, $\overline{\text{BM}}_1$, and HOLD. Pins 15 and 16 specify byte selection on the DAL lines as whole word, upper byte, lower byte, or none. Pin 17, $\overline{\text{HOLD}}$, is asserted by LANCE whenever it requires access to memory.

When BCON = 1, these pins are defined as BYTE, $\overline{\text{BUSAKO}}$, and $\overline{\text{BUSRQ}}$. The BYTE line (pin 15) is used in conjunction with the least significant bit of the DMA address to specify byte selection as whole word, lower output. If the LANCE is not requesting the bus and it receives $\overline{\text{HDLA}}$ $\overline{\text{BUSAKO}}$ is driven low. If the LANCE is requesting the bus when it receives $\overline{\text{HDLA}}$ $\overline{\text{BUSAKO}}$ will remain high. $\overline{\text{BUSRQ}}$ pin 17, will only be asserted if pin 17 is high prior to assertion when in the daisy chain configuration.

The BSWP bit in CSR_3 determines if the LANCE treats the high byte as the most significant or least significant byte when writing into the Silo (Bus Master Mode, read operation) or when reading it from the Silo (Bus Master Mode, write operation). This function facilitates operation with different 16-bit microprocessors. When BSWP = 1, the LANCE will swap the high and low bytes on DMA transfers between the Silo and and bus memory.



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ERR	BABL	CERR	MISS	MERR	RINT	TINT	IDON	INTR	INEA	RXON	TXON	TDMD	STOP	STRT	INIT	CSR_0

15 … 2	1-0	
IADR	00	CSR_1

15 … 8	7 … 0	
RES	IADR	CSR_2

15 … 3	2	1	0	
RES	BSWP	ACON	BCON	CSR_3

06363A-2

Figure 1-3 Control and Status Registers in LANCE

Transmit Descriptor Ring Pointer

31–29	28–24	23	3	2–0
TLEN	RES	TDRA		000

Receive Descriptor Ring Pointer

31–29	28–24	23	3	2–0
RLEN	RES	RDRA		000

Logical Address Filter

63	0
LADRF	

Physical Address

47	1	0
PADR		0

Mode Register

15	14	7	6	5	4	3	2	1	0
PROM	RES		INTL	DRTY	COLL	DTCR	LOOP	DTX	DRX

Initialization Block

15	0	
LADR		TMD_0

15	14	13	12	11	10	9	8	7	0	
OWN	ERR	RES	MORE	ONE	DEF	STP	ENP	HADR		TMD_1

15	12	11	0	
1111		BCNT		TMD_2

15	14	13	12	11	10	9	0	
BUFF	UFLO	RES	LCOL	LCAR	RTRY	TDR		TMD_3

Transmit Message Descriptor Ring Components

15	0	
LADR		RMD_0

15	14	13	12	11	10	9	8	7	0	
OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	HADR		RMD_1

15–12	11	0	
1111	BCNT		RMD_2

15–12	11	0	
RES	MCNT		RMD_3

Receive Message Descriptor Ring Components

06363A-3

Figure 1-4 Initialization Block and Descriptor Rings

Table 1-1. Processor Interface Requirements

	Z8000	**8086**	**68000**	**LSI-11**
Dedicated buses	**Z-bus (multiplexed)**	**Multibus (multiplexed)**	**Versabus (demultiplexed)**	**Q-bus (multiplexed)**
Address latch enable polarity	$\overline{\mathrm{AS}}$	ALE*	ALE	ALE
Byte addressing of memory	Byte/$\overline{\mathrm{Word}}$ signal	Upper Byte Strobe	Upper and lower strobes	Byte/$\overline{\mathrm{Word}}$ signal
Memory organization	Even address accesses upper byte; odd address accesses lower byte	Even address accesses lower byte; odd address accesses upper byte	Even address accesses upper byte; odd address accesses lower byte	Even address accesses lower byte; odd address accesses upper byte

* The Am7990 Tri-states ALE; the 8086 does not.

Table 1-1 shows how to program CSR$_3$ when interfaced to different processors. CSR$_1$ and CSR$_2$ are programmed to contain a pointer to the Initialization Block.

1.4.2 INITIALIZATION BLOCK

This block contains the operating parameters necessary for device operation. The Initialization Block is 12 words long in contiguous memory starting on a word boundary (See Figure 1-3). The Initialization Block contains:

- Mode of operation (Mode Register)
- Physical Address
- Logical Address Mask
- Location of Transmit and Receive Descriptor Rings
- Number of entries in Transmit and Receive Descriptor Rings.

1.4.3 DIAGNOSTICS

The Mode Register in the Initialization Block is used to set up the LANCE for diagnostic or normal operation. There are four user-programmable diagnostic modes. Each mode requires the user to change bits in the Mode Register and reinitialize the LANCE. The four modes are listed as follows:

1. Internal Loopback
2. CRC Logic Check
3. Collision Detection and Retry Logic
4. External Loopback.

The detailed description of the diagnostic functions and flowcharts appear in a later section of this technical manual.

1.4.4 NORMAL OPERATION

Once the LANCE interface with the system is checked out both in Bus Master and Bus Slave Mode, and the internal logic of the LANCE has been tested in diagnostic mode for both internal and external loopback test, the LANCE can be configured for normal operation via the Initialization Block. The user normally initializes the LANCE once for normal operation. Any change in the LANCE configuration requires the re-initialization of the LANCE. The user must make sure to process all the transmitted or received packets and rearrange the packets queued for transmission in the Transmit Ring before re-initializing the LANCE. This is necessary since, upon re-initialization, the LANCE sets its pointers to the beginning of the Transmit and Receive Rings.

1.5 BUS MASTER OPERATION

The LANCE must be in Bus Master mode to access the external memory. Local memory external to the LANCE is used for the Initialization Block, transmit/receive rings, and transmit/receive buffers. $\overline{\mathrm{HOLD}}$ and $\overline{\mathrm{HLDA}}$ of the LANCE both active indicate that the LANCE is a Bus Master. The $\overline{\mathrm{HLDA}}$ input to the LANCE may be directly connected to the HOLD output from the LANCE. This is desirable in some applications.

To guarantee that the LANCE is not switched into Slave Mode inadvertently, $\overline{\mathrm{CS}}$ must not be asserted when the LANCE in Bus Master Mode ($\overline{\mathrm{HLDA}}$ active). This is usually guaranteed by design when the LANCE and CPU share the same bus. However, in a dual port RAM design in which the LANCE is not on the same bus with the CPU, some external logic is required to monitor and control $\overline{\mathrm{HOLD}}$ and $\overline{\mathrm{HLDA}}$ before a $\overline{\mathrm{CS}}$ is asserted.

Each DMA cycle initiated by the LANCE consists of 6 T-states (600 ns) plus any additional wait states (Tw) which is required for slower memories. Wait states may be added between T4 and T5 of the LANCE DMA cycle. The $\overline{\mathrm{READY}}$ input to the LANCE inserts wait states in 100 ns increments. To guarantee that no Wait State is added to the bus cycle, the $\overline{\mathrm{READY}}$ signal input to the LANCE must be asserted within 80 ns (max), following the falling edge of the ALE signal. Memory chips with 600 ns (max) cycles time and an access time of 280 ns from ALE falling edge, or 355 ns from Address Valid, can be used without adding any Wait States to the LANCE DMA cycles.

It is possible for one or more wait states to be added to the LANCE DMA cycle even with fast memory chips if the timing specification of the LANCE is not met. It is recommended that the memory cycle sequencer be synchronized with the falling edge of DAS and that $\overline{\text{READY}}$ be asserted within 80 ns following the falling edge of ALE in order to assure that no wait states are added.

In the Bus Master Mode, the LANCE has two types of DMA transfers: burst or single DMA cycle. Each burst transfer uses eight DMA cycles and transfers eight 16-bit words unless fewer than eight words remain to be transferred. Separate single word DMA cycles are used to access the transmit/receive rings and the Initialization Block.

The LANCE is equipped with some additional features, including two transceiver control signals, $\overline{\text{DALI}}$ (DAL In) and $\overline{\text{DALO}}$ (DAL Out). When in Bus Master Mode, the LANCE asserts $\overline{\text{DALI}}$ and $\overline{\text{DALO}}$ to enable the data transceivers for Read and Write operations. $\overline{\text{DALI}}$ is asserted only during the data portion of a read transfer. $\overline{\text{DALO}}$ is asserted during the write operation to enable the transceiver for both address and data (multiplexed Address/Data), and also during the read operation to transfer the address (refer to the LANCE Data Sheet). It is important to note that, during the read operation, there is a delay from deassertion of $\overline{\text{DALO}}$ to assertion of $\overline{\text{DALI}}$ or vice versa. This is to prevent bus contention which would cause electromagnetic interference (EMI).

1.6 GENERAL DESCRIPTION

The Am7990 (LANCE) is designed to operate in an environment that includes close coupling with a local memory and a microprocessor (HOST). The Am7990 LANCE is programmed by a combination of registers and data structures resident within the chip and in memory. There are four Control and Status Registers (CSRs) within the chip which are programmed by the HOST device. Once enabled, the chip has the ability to access memory locations to acquire additional operating parameters.

The Am7990 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:

1. 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 of Receive and Transmit Descriptor Rings
   - Number of Entries in Receive and Transmit Descriptor Rings
2. Receive and Transmit Descriptor Rings—Two ring structures, one each 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.
3. 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 chip may be summarized as:

1. Programming the chip's CSRs by a host device to locate an Initialization Block in memory. The byte control, byte addressing, and address latch enable modes are defined here also.
2. The chip loads itself with the information contained within the Initialization Block.
3. The chip accesses the descriptor rings for packet handling.

The parallel interface of the Local Area Network Controller for Ethernet (LANCE) has been designed to be "friendly" or easy to interface to a variety of popular 16-bit microprocessors. These microprocessors include the following: Z8000, 8086, 68000 and LSI-11. The LANCE has a 24-bit wide linear address space when it is in the Bus Master Mode allowing it to DMA directly into the entire address space of the above microprocessors.

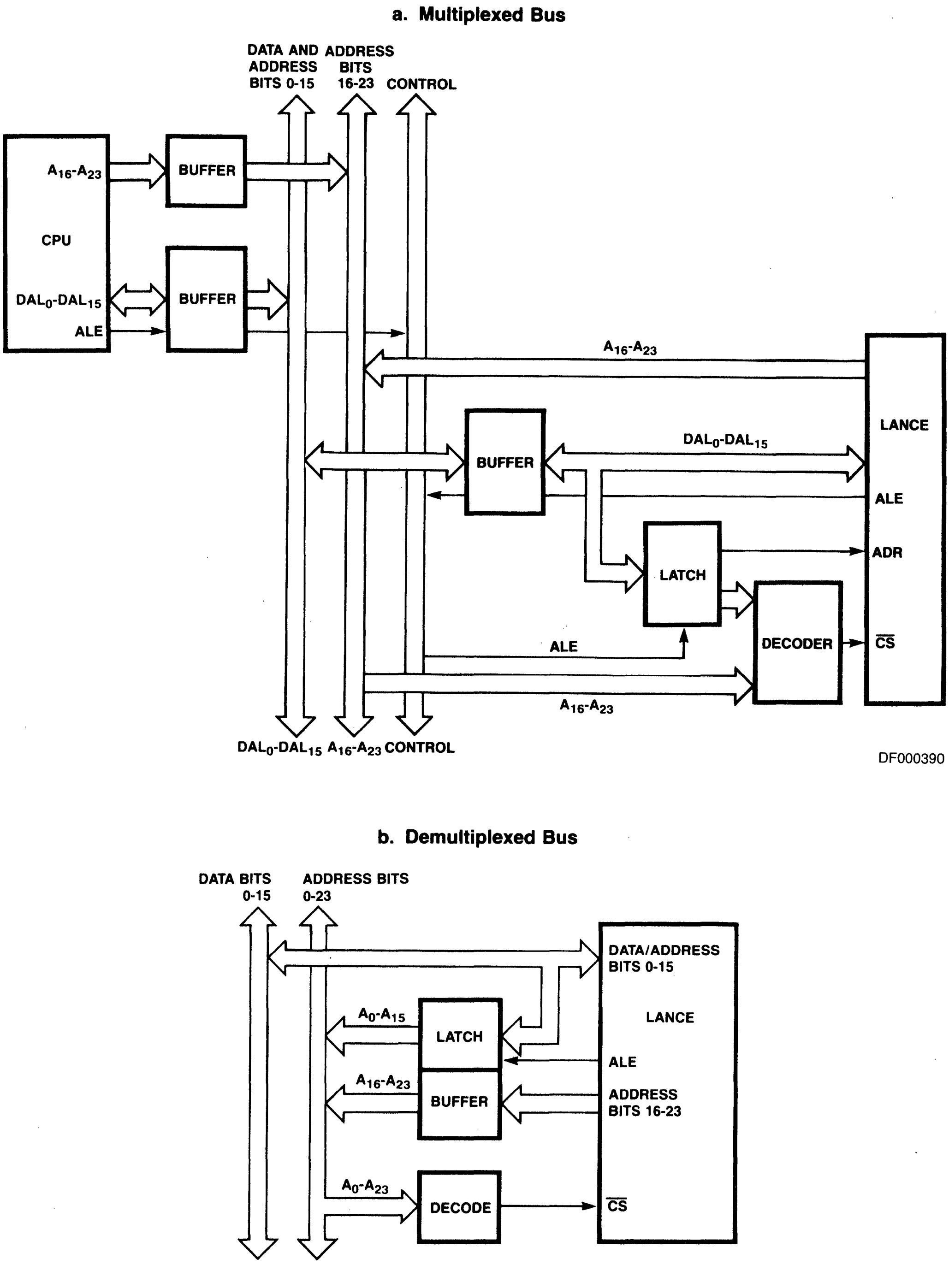
A programmable mode of operation allows byte addressing in one of two ways: A Byte/Word control signal compatible with the 8086 and Z8000, or an Upper Data Strobe and Lower Data Strobe signal compatible with microprocessors such as the 68000. A programmable polarity on the Address Strobe signal eliminates the need for external logic. The LANCE interfaces with both multiplexed and demultiplexed data buses and features control signals for address/data bus transceivers. See Figure 1-5.

During initialization, the CPU loads the starting address of the Initialization Block into two internal control registers. The LANCE has four internal control and status registers (CSR$_{0,\ 1,\ 2,\ 3}$) which are used for various functions such as the loading of the Initialization Block address, different programming modes and status conditions. The host processor communicates with the 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 handled as DMA under microword control.

Interrupts to the microprocessor are generated by the LANCE upon: 1) completion of its initialization routine, 2) the reception of a packet, 3) the transmission of a packet, 4) transmitter timeout error, 5) a missed packet and 6) memory error.

The cause of the interrupt is ascertained by reading CSR$_0$. Bit (06) of CSR$_0$, (INEA) enables or disables interrupts to the microprocessor. In systems where polling is used in place of interrupts, bit (07) of CSR$_0$ (INTR) indicates an interrupt condition.

The basic operation of the LANCE consists of two, distinct modes: transmit and receive. In the transmit mode, the LANCE chip directly accesses data (in a transmit buffer) in memory. It prefaces the data with a preamble, sync pattern, and calculates and appends a 32-bit CRC. This packet is then ready for transmission to the Am7992A SIA. On transmission, the first byte of data loads into the 48-byte FIFO. The LANCE then begins to transmit preamble while simultaneously loading the rest of the packet into FIFO for transmission.



a. Multiplexed Bus

b. Demultiplexed Bus

Figure 1-5 LANCE/CPU Interfacing

In the receive mode, packets are sent via the SIA to the LANCE. The packets are loaded into the 48-byte 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 checksum doesn't agree with the packet CRC, an error bit is set.

1.7 PIN DESCRIPTION

DAL_{00}-DAL_{15} **Data/Address Lines (Input/Output 3-State)**

The time multiplexed Address/Data bus. During the address portion of a memory transfer, DAL_{00}-DAL_{15} contains the lower 16 bits of the memory address. The upper 8 bits of address are contained in A_{16}-A_{23}.

During the data portion of a memory transfer, DAL_{00}-DAL_{15} contains the read or write data, depending on the type of transfer.

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

A_{16}-A_{23} **High Order Address Bus (Output 3-State)**

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

READ **(Input/Output 3-State)**

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

High — Data is taken off the DAL by the LANCE.

Low — Data is placed on the DAL by the LANCE.

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

High — Data is placed on the DAL by the LANCE.

Low — Data is taken off the DAL by the LANCE.

$\overline{BM_0}$/BYTE $\overline{BM_1}$/$\overline{BUSAKO}$ I/O pins 15 and 16 are programmable through bit (00) of CSR_3

$\overline{BM_0}$, $\overline{BM_1}$

If CSR_3 (00) BCON = 0,
I/O Pin 15 = $\overline{BM_0}$ (Output 3-state)
I/O Pin 16 = $\overline{BM_1}$ (Output 3-state)

$\overline{BM_0}$, $\overline{BM_1}$ (Byte Mask). This indicates the byte(s) on the DAL are to be read or written during this bus transaction. The 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.

$\overline{BM_1}$	$\overline{BM_0}$	
LOW	LOW	Whole Word
LOW	HIGH	Upper Byte
HIGH	LOW	Lower Byte
HIGH	HIGH	None

BYTE, $\overline{BUSAKO}$

If CSR_3 (00) BCON = 1,
I/O Pin 15 = BYTE (Output 3-state)
I/O Pin 16 = $\overline{BUSAKO}$ (Output)

Byte selection may also be done using the BYTE line and DAL_{00} line, latched during the address portion of the bus cycle. The LANCE drives BYTE only as a Bus Master and ignores it when a Bus Slave selection is done (similar to $\overline{BM_0}$, $\overline{BM_1}$.

Byte selection is done as outlined in the following table.

BYTE	DAL_{00}	
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 received $\overline{HLDA}$, $\overline{BUSAKO}$ will be driven low. If the LANCE is requesting the bus when it received $\overline{HLDA}$, $\overline{BUSAKO}$ will remain high.

Byte Swapping

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

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 the byte swap is activated, the only data that is swapped is data traveling to and from the Silo.

	Mode Bits	
Signal Line	**BSWP = 0 and BCON = 1**	**BSWP = 1 and BCON = 1**
BYTE = L and DAL_{00} = L	Word	Word
BYTE = L and DAL_{00} = H	Illegal	Illegal
BYTE = H and DAL_{00} = H	Upper Byte	Lower Byte
BYTE = H and DAL_{00} = L	Lower Byte	Upper Byte

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

Indicates, when asserted, that the 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.

ADR **Register Address Port Select (Input)**
When LANCE is 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 LANCE when CS is low.

ALE/$\overline{\text{AS}}$ **Address Latch Enable (Output 3-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 CSR$_3$.

As ALE (CSR$_3$ (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 $\overline{\text{AS}}$ (CSR$_3$ (01), ACON = 1), the signal pulses LOW during the address portion of the bus transaction. The low to high transition of AS can be used by a slave device to strobe the address into a register.

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

$\overline{\text{DAS}}$ **Data Strobe (Input/Output 3-State)**
Defines the data portion of the bus transaction. $\overline{\text{DAS}}$ 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. $\overline{\text{DAS}}$ is driven only as a Bus Master.

$\overline{\text{DALO}}$ **Data/Address Line Out (Output 3-State)**
An external bus transceiver control line. $\overline{\text{DALO}}$ is asserted when the LANCE drives the DAL lines. $\overline{\text{DALO}}$ is low only during the address portion if the transfer is a READ. It is low for the entire transfer if the transfer is a WRITE. $\overline{\text{DALO}}$ is driven only when LANCE is a Bus Master.

$\overline{\text{DALI}}$ **Data/Address Line In (Output 3-State)**
An external bus transceiver control line. $\overline{\text{DALI}}$ is asserted when the LANCE reads from the DAL lines. It is low during the data portion of a READ transfer, and remain high for the entire transfer if it is a WRITE. $\overline{\text{DALI}}$ is driven only when LANCE is a Bus Master.

$\overline{\text{HOLD}}$
$\overline{\text{BUSRQ}}$ **Bus Hold Request (Output Open Drain)**
Asserted by the LANCE when it requires access to memory. $\overline{\text{HOLD}}$ is held LOW for the entire ensuing bus transaction. The function of this pin is programmed through bit (00) of CSR$_3$ Bit (00) of CSR$_3$ is cleared when $\overline{\text{RESET}}$ is asserted.

When CSR$_3$ (00) BCON = 0

I/O pin 17 = $\overline{\text{HOLD}}$(Output Open Drain)

When CSR$_3$ (00) BCON = 1

I/O pin 17 = $\overline{\text{BUSRQ}}$(Output Open Drain)

$\overline{\text{BUSRQ}}$ will be asserted only if I/O pin 17 is high prior to assertion.

$\overline{\text{HLDA}}$ **Bus Hold Acknowledge (Input)**
A response to $\overline{\text{HOLD}}$. When $\overline{\text{HLDA}}$ is low in response to the chip's assertion of $\overline{\text{HOLD}}$, the chip is the Bus Master. $\overline{\text{HLDA}}$ deasserts upon the deassertion of $\overline{\text{HOLD}}$.

$\overline{\text{INTR}}$ **Interrupt (Output Open Drain)**
An attention signal that indicates, when active, that one or more of the following CSR$_0$ status flags is set: BABL, MERR, MISS, RINT, TINT or IDON. $\overline{\text{INTR}}$ is enables by bit 06 of CSR$_0$ (INEA = 1). $\overline{\text{INTR}}$ remains asserted until the source of Interrupt is removed.

RX **Receive (Input)**
Receive Input Bit Stream.

TX **Transmit Enable (Output)**
Transmit Output Bit Stream.

TENA **Transmit (Output)**
Transmit Output Bit Stream enable. When asserted, it enables the external transmit output.

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

CLSN **Collision (Input)**
A logical input that indicates that a collision is occurring on the channel.

RENA **Receive Enable (Input)**
A logical input that indicates the presence of carrier on the channel.

TCLK **Transmit Clock (Input)**
10MHz clock.

$\overline{\text{READY}}$ **(Input/Output Open Drain)**
When the LANCE is a Bus Master, $\overline{\text{READY}}$ is an asynchronous acknowledgement 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 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 LANCE is a Bus Master and an output when the LANCE is a Bus Slave.

$\overline{\text{RESET}}$ **(Input)**
Bus Reset Signal. Causes the LANCE to cease operation, clears its internal logic, and enter an idle state. The stop bit in CSR$_0$ is also set.

A 3.3K pull-up resistor should be used at the $\overline{\text{RESET}}$ pin of the LANCE.

V$_{CC}$ Power supply pin +5 volts ±5%.

A 0.1µf and a 10µf decoupling capacitors should be used for V$_{CC}$ to V$_{SS}$.

V$_{SS}$ Ground Pins 1 and 24 should be connected together externally, as close to the chip as possible.

1.8 FUNCTIONAL DESCRIPTION

1.8.1 ADDRESSING

Packets can be received using 3 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 programmed into the LANCE during an initialization cycle.

There are two types of logical address. One is group type mask where the 48-bit address in the packet is put through a hash filter in order 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 comparing the 48-bit incoming address with the prestored 48-bit logical address. The 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 the so-called "promiscuous mode" in which a node will accept all packets on the coax regardless of their destination address.

1.8.2 COLLISION DETECTION AND IMPLEMENTATION

The Ethernet CSMA/CD network access algorithm is implemented completely within the LANCE. In addition to listening for a clear coax before transmitting, Ethernet handles collisions in a predetermined way. Should two transmitters attempt to seize the coax at the same time, they will collide and the data on the coax 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 don't try to repeatedly access the network at the same time. Up to 16 attempts to access the network are made by the LANCE before reporting back an error due to excessive collisions.

COLLISION JAM

Collisions are detected by monitoring the CLSN I/O pin. If CLSN becomes asserted during a frame transmission, TENA will remain asserted for at least 32 additional bit times (including CLSN synchronization). This additional transmission after collision is called COLLISION JAM. If collision occurs during the transmission of the preamble, the LANCE continues to send the preamble followed by a JAM pattern. If collision occurs after the preamble, the LANCE sends the JAM pattern following the transmission of the current byte. The JAM pattern is any pattern except the CRC bytes.

COLLISION DURING RECEPTION

If CLSN becomes asserted during the reception of a packet, this reception is immediately terminated. The action taken depends on the timing of collision detection. The packet is rejected and the Silo pointer is reset if the collision occurs within six byte times (4.8us) because of an address mismatch. The packet is rejected as a runt packet if the collision occurs within 64 byte times (51.2us). If a collision occurs after 64 byte times (late collision), a truncated packet is written to the memory buffer with the CRC error bit usually set in the Status Word of the receive ring. Late collision error is not recognized in receive mode.

COLLISION DURING TRANSMISSION

When a transmission is terminated due to the assertion of CLSN (a collision within the first 64 byte times of the packet), the LANCE retries it 15 times. The LANCE does not re-read the descriptor entries from the TX ring upon each collision. The descriptor entries for the current buffer are internally saved. The scheduling of the retransmissions is determined by a controlled randomized process called "truncated binary exponential backoff". Upon the completion of the COLLISION JAM interval, the LANCE calculates a delay before retransmitting. The delay is an integral multiple of the slot time. The slot time is 64 byte times (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 of:

$0 \leqslant r < 2^k$

where: k = minimum (n, 10)

For example, if this is the third retry,

$n = 3$

$k = \min(3,10) = 3.$

$2^3 = 8$

Therefore, r is a random number from zero to eight.

If all 16 attempts fail, the LANCE sets the RTRY bit in the current Transmit Message Descriptor 3 (TMD_3) 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 a late collision occurs, the LANCE does not transmit the same packet again. It terminates the transmission, notes the LCOL error in TMD_3, and transmits 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 to start loading the Silo in anticipation of retransmission. To utilize the channel properly, it is important that the LANCE be ready to transmit when the backoff interval elapses.

TIME DOMAIN REFLECTOMETRY

The LANCE contains a time domain reflectometry counter (TDR). The TDR counter is ten bits wide. It counts at a 10MHz rate. It is cleared by the microprogram and counts upon the assertion of RENA during transmission. Counting stops if CLSN becomes true or if RENA goes inactive. The counter does not wrap around. When the counter reaches a value of all ones, that value is held until it is 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.

COLLISION PAIR TEST (HEARTBEAT, SQE)

An open in the cable causes a reflection which the coaxial cable transceiver detects as a collision. A short in the cable causes RENA not to become active or to drop out after becoming active. During the Interpacket Delay following the negation of TENA, the CLSN input is asserted (as a result of collision becoming active) by some transceivers as a self-test called Heartbeat or SQE (Signal Quality Error) test. If the CLSN input is not asserted within 2.0us following the completion of transmission (after TENA goes low), then the LANCE will set the CERR bit in CSR_0. CERR error does not cause an interrupt to occur (INTR = 0).

1.8.3 ERROR REPORTING AND DIAGNOSTICS

Extensive error reporting is provided by the LANCE. Error conditions reported relate to either the network as a whole or to 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-related errors include:

1. Babbling Transmitter — Transmitter attempting to transmit more than 1518 data bytes in one packet.
2. Collision — Collision detection circuitry non-functional.
3. Missed Packet — Insufficient buffer space.
4. Memory Timeout — Memory response failure

Packet-related errors include:

1. CRC — Invalid data
2. Framing — Packet did not end on byte boundary
3. Overflow/Underflow — Abnormal latency in servicing a DMA request
4. Buffer — Insufficient buffer space available

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

CYCLIC REDUNDANCY CHECK (CRC)

The LANCE utilizes the 32 bit CRC function used in the Autodin-II network. Refer to the Ethernet specification (Section 6.2.4 Frame Check Sequence Field and Appendix C, CRC Implementation) for more detail. The LANCE requirements for the CRC logic are:

1. TRANSMISSION — Mode bit 2 (LOOP) = 0, Mode bit 3 (DTCR) = 0. The LANCE calculates the CRC from the first bit following the Start bit to the last bit of the data field. The CRC value inverted is appended to the transmission in one unbroken bit stream.
2. RECEPTION — Mode bit 2 (LOOP) = 0. The LANCE performs a check on the input bit stream from the first bit following the Start bit to the last bit in the frame. The 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.
3. LOOPBACK — Mode bit 2 (LOOP) = 1, Mode bit 3 (DTRC) = 0. The 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.
4. LOOPBACK — Mode bit 2 (LOOP) = 1, Mode bit 3 (DTRC) = 1. The 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 LANCE normally operates as a half-duplex device. However, to provide an on-line operational test of the 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:

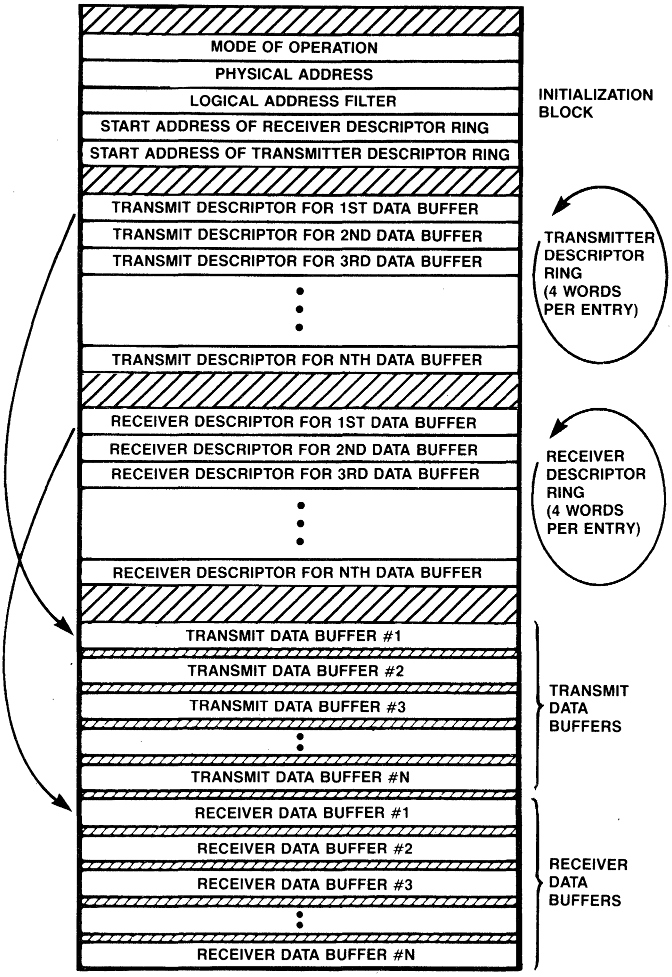
1. The packet length must be from eight to 32 bytes long, exclusive of the CRC bits.
2. Serial transmission does not begin until the Silo contains the entire output packet.
3. Moving the input packet from the Silo to the memory does not begin until the serial input bit stream terminates.
4. CRC may be either generated and appended to the output serial bit stream or may be checked on the input serial bit stream but not both in the same transaction.
5. In internal loopback, the packets should be addressed to the node itself. If not, the chip must be re-initialized with non-matching source and destination addresses after each test.
6. 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.
7. In external loopback, the Silo pointers sometimes get misaligned with heavy traffic on the network. It is recommended that the user repeats the test several times after the first time it fails. This can help isolate the failure to Silo pointer misalignment or other causes. For detail External Loopback Test Procedure, please refer to Appendix C.

Loopback is controlled by the INTL, DTCR, and LOOP (bits 2, 3,and 6) of the MODE register.

1.8.4 BUFFER MANAGEMENT

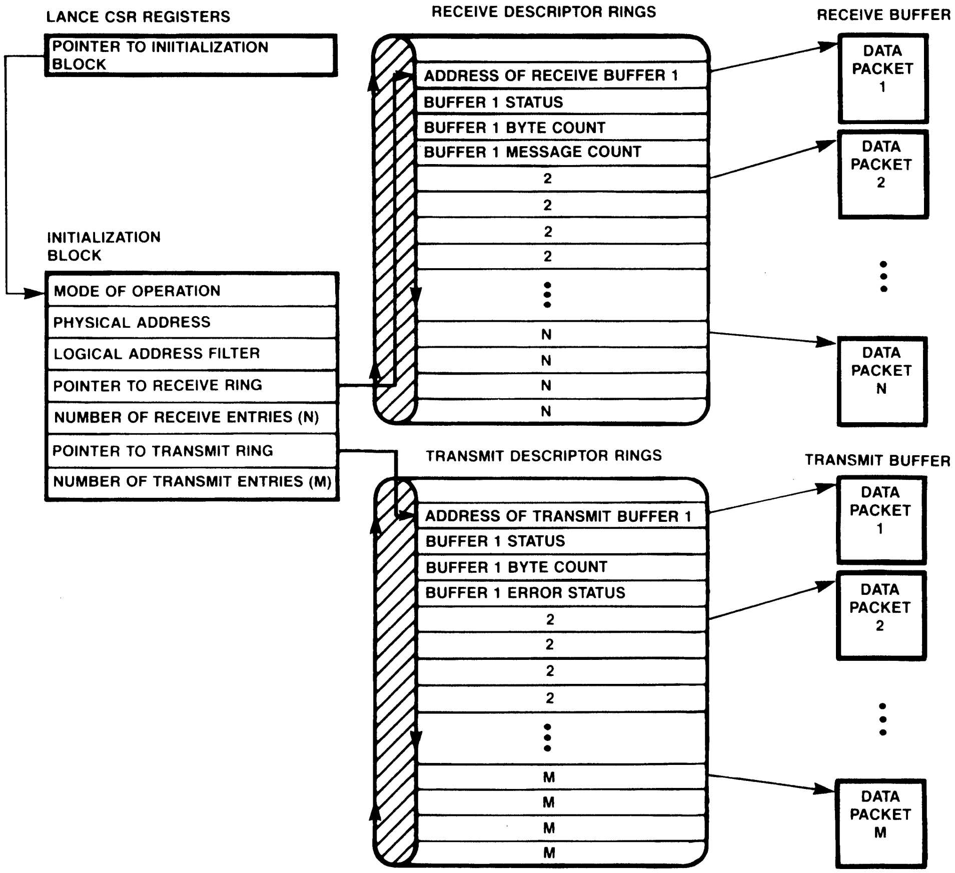
A key feature of the LANCE and its on-board DMA channel is the flexibility and speed of communication between the LANCE and the host microprocessor through common memory locations. The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings. Refer to Figure 1-6 and Figure 1-7.

The transmit and receive operations are described by separate descriptor rings. Up to 128 tasks may be queued on a descriptor ring awaiting execution by the LANCE. Each entry in a descriptor ring contains a pointer to a data buffer and an entry for the length of the buffer.


MODE OF OPERATION
PHYSICAL ADDRESS
LOGICAL ADDRESS FILTER
START ADDRESS OF RECEIVER DESCRIPTOR RING
START ADDRESS OF TRANSMITTER DESCRIPTOR RING
INITIALIZATION BLOCK
TRANSMIT DESCRIPTOR FOR 1ST DATA BUFFER
TRANSMIT DESCRIPTOR FOR 2ND DATA BUFFER
TRANSMIT DESCRIPTOR FOR 3RD DATA BUFFER
TRANSMIT DESCRIPTOR FOR NTH DATA BUFFER
TRANSMITTER DESCRIPTOR RING (4 WORDS PER ENTRY)
RECEIVER DESCRIPTOR FOR 1ST DATA BUFFER
RECEIVER DESCRIPTOR FOR 2ND DATA BUFFER
RECEIVER DESCRIPTOR FOR 3RD DATA BUFFER
RECEIVER DESCRIPTOR FOR NTH DATA BUFFER
RECEIVER DESCRIPTOR RING (4 WORDS PER ENTRY)
TRANSMIT DATA BUFFER #1
TRANSMIT DATA BUFFER #2
TRANSMIT DATA BUFFER #3
TRANSMIT DATA BUFFER #N
TRANSMIT DATA BUFFERS
RECEIVER DATA BUFFER #1
RECEIVER DATA BUFFER #2
RECEIVER DATA BUFFER #3
RECEIVER DATA BUFFER #N
RECEIVER DATA BUFFERS

DF000130

Figure 1-6 LANCE/Processor Memory Interface



05726A-15

Figure 1-7 LANCE Memory Management

Data buffers can be chained to handle a packet that is longer than the buffer. The LANCE searches the descriptor rings in a lookahead operation to determine the next empty buffer when chaining is needed or when back-to-back packets are being received. As each buffer is filled, an OWN bit in the descriptor ring entry is reset, allowing the host processor to process the data in the buffer.

RING ACCESS BY THE LANCE

After the LANCE is initialized and started, the CPU and the LANCE communicate via transmit and receive rings (in memory) for packet transmission and reception. The LANCE contains eight 16-bit registers. The first four registers store the four entries in the descriptor that points to the current buffer. The other four registers store the same information for the descriptor that points to the next buffer to be processed (the lookahead buffer). Refer to Figure 1-2.

TRANSMIT RING BUFFER MANAGEMENT

After the LANCE has been initialized, whenever the Ethernet is inactive, the LANCE automatically polls the first transmit ring entry in memory. The polling occurs every 1.6ms and consists of reading the status word (TMD_1) of the transmit ring entry until it finds the OWN bit and STP bit set to one. The OWN bit set to one in the transmit ring indicates a buffer ready to be transmitted. The STP bit indicates that this is the start of a packet.

When the LANCE finds that a buffer is to be transmitted, it transfers the low order bits of the buffer address from TMD_0 and byte count from TMD_2 into the current buffer pointer in LANCE. Each of these memory reads is done with a separate arbitration cycle for each transfer. The high order bits of the address are in TMD_1 which have already been transferred.

If the packet is larger than the buffer, the buffers are data-chained and the ENP (end of the packet) is zero in all of the buffers except the buffer containing the end of the packet. If ENP equals zero for the buffer being transmitted, the LANCE does one lookahead for the next buffer. The lookahead is done in between data transfers to or from the Silo. It consists of one single word DMA read (TMD_1) to find out if it owns the buffer and two more reads, (TMD_0 and TMD_2) if it does own it.

If LANCE does own the next buffer, it transfers the address and byte count of this buffer into the lookahead buffer in the LANCE. After the LANCE completes transferring the current buffer into the Silo, it clears the OWN bit for this buffer and immediately starts loading the Silo from the next buffer.

If the LANCE finds during the lookahead that it does not own the next transmit Descriptor Table Entry (DTE), it will transmit the current buffer and update the status of the current ring entry by setting the BUFF bit in TMD_3. BUFF error causes the transmit section of the LANCE to turn off (CSR_0, TXON = 0). The chip has to be re-initialized to turn transmitter on.

While transmitting, if a RTRY or LCOL (late collision) occurs, the LANCE stops transmitting the packet. It clears the OWN bit in TMD_1 and sets the TINT bit in CSR_0 causing an interrupt if INEA = 1 (interrupts enabled). The LANCE does not transmit the remainder of the buffers in this packet. Instead, it clears the own bit and sets the TINT bit in each Descriptor Table Entry it polls until it finds a buffer with both the STP bit and OWN bit set indicating the start of the next packet to send.

When the transmit buffers are not data-chained (current descriptor's ENP = 1), the LANCE does not perform any lookahead operation. It transmits the current buffer and updates the status TMD_1 (and TMD_3 if an error occurred) and then clears the OWN bit in TMD_1. The LANCE immediately checks the next descriptor in the ring to see if another packet is ready to be sent. If it is ready, the LANCE proceeds to read the buffer address and byte count and to transmit the packet. If no packet is ready, it resumes polling the ring every 1.6ms.

After each complete packet has been transmitted from the Silo to the cable, the LANCE updates the status in TMD_1 (and TMD_3 if an error occurred). Then it releases the last buffer of the packet to the CPU. It then immediately polls the transmit descriptor ring for another packet to send. This guarantees the back-to-back transmission of packets when packets are ready to transmit.

RECEIVE RING BUFFER MANAGEMENT

When the receiver is enabled, the LANCE polls the first receive ring entry once every 1.6 ms for buffer ownership so that it will be ready for the next incoming packet. After the LANCE owns a buffer, it reads the remainder of the buffer address (from RMD_0) and the byte count (from RMD_2) into the current buffer in the LANCE. After the LANCE has filled the first buffer with data from a received packet, it polls the next receive descriptor ring entry for the next buffer to fill.

When a packet arrives from the cable, and the LANCE does not own any buffer, it polls the receive ring entry once for the buffer. If it does not get the buffer, it sets the MISS error bit in CSR_0 and does not poll the receive ring entry again until the packet ends. (In the previous revision, Rev.B, the LANCE polled the entry every few microseconds until it either received the buffer or the Silo overflowed.)

Assuming that the LANCE owns a buffer when a packet arrives from the cable, the LANCE does one lookahead operation in between periods of data transfer from the Silo. The LANCE needs to have a buffer ready in case the current buffer needs data-chaining. The lookahead operation consists of one single word DMA read (RMD_1) to find out if it owns the buffer and two more reads, (RMD_0 and RMD_2) if it does own it.

Whether the LANCE owns the next buffer or not, it will transfer data from the Silo to the current buffer in burst mode (eight words for each DMA arbitration cycle). If the packet being received requires data-chaining and the LANCE does not own the next buffer, the LANCE updates the current buffer status by setting the BUFF error bit.

If the LANCE does own the next buffer and needs data-chaining, when the first buffer is full it releases it by clearing the OWN bit. It then starts filling the next buffer. In between data transfers from the Silo, it does one lookahead operation for ownership of the next buffer to be ready for more data-chaining if needed.

After each complete packet has been received and transferred from the Silo to the buffers, the LANCE updates the status in RMD_1 (and RMD_3 if an error occurred). Then it releases the last buffer of the packet to the CPU.

1.8.5 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/$\overline{\text{AS}}$.

BCON is the byte control bit that redefines the BYTE MASK I/O pins to handle odd address boundaries and redefines the $\overline{\text{HOLD}}$ I/O pin for daisy-chain operation when BCON = 1. In addressing word organized, byte addressable memories (BCON = 1), the BYTE signal is decoded along with the least significant address bit to determine odd or even (upper or lower byte) byte address.

When BCON = 1 in a daisy-chain operation, the $\overline{\text{BM}_1}$ pin is used as $\overline{\text{BUSAKO}}$ the $\overline{\text{HOLD}}$ pin is used as $\overline{\text{BUSRQ}}$ and the pin $\overline{\text{HLDA}}$ is used. When a DMA controller is used for arbitration, only the $\overline{\text{HOLD}}$ and $\overline{\text{HLDA}}$ pins are used.

When BCON = 0, the BYTE MASK pins 15 and 16 ($\overline{\text{BM}_0}$ and $\overline{\text{BM}_1}$) are used to specify byte selection on the DAL lines as whole word, upper byte, lower byte, or none.

All data transfers from the LANCE in the Bus Master mode are timed by ALE, $\overline{\text{DAS}}$, and $\overline{\text{READY}}$. The automatic adjustment of the LANCE cycle by the $\overline{\text{READY}}$ signal allows synchronization with variable cycle time memory due either to memory refresh or to dual port access. Bus cycles are a minimum of 600ns duration and can be increased in 100ns increments.

READ SEQUENCE (Master Mode)

The read memory cycle is started by placing valid addresses on DAL$_{00}$-DAL$_{15}$ and A$_{16}$-A$_{23}$. 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 $\overline{\text{AS}}$ are pulsed and the trailing edge of either can be used to latch addresses. DAL$_{00}$-DAL$_{15}$ go into a 3-state mode and $\overline{\text{DAS}}$ falls low to signal the beginning of the memory access. The memory responds by placing $\overline{\text{READY}}$ low to indicate that the DAL lines have valid data. The LANCE then latches memory data on the rising edge of $\overline{\text{DAS}}$ which in turn ends the memory cycle and $\overline{\text{READY}}$ returns to high.

Data transfers between the bus transceivers and the LANCE are controlled by the $\overline{\text{DALI}}$ and $\overline{\text{DALO}}$ lines. $\overline{\text{DALI}}$ diverts data toward the LANCE. $\overline{\text{DALO}}$ diverts data or addresses away from the LANCE. During a read cycle, $\overline{\text{DALO}}$ goes inactive before $\overline{\text{DALI}}$ becomes active to avoid "spiking" the bus transceivers.

WRITE SEQUENCE (Master Mode)

The write to memory cycle is similar to the read cycle except that the DAL lines change from containing addresses to containing data after ALE or $\overline{\text{AS}}$ goes inactive. After data is valid on the bus, $\overline{\text{DAS}}$ goes active. Data to memory is held valid for some time after $\overline{\text{DAS}}$ goes inactive.

1.8.6 LANCE DMA TRANSFER (Bus Master Mode)

The LANCE has two types of DMA transfers: Burst Mode and Single Word transfers.

BURST MODE

Burst mode is used for the transfer of packets between the Silo and processor memory. Each burst transfer is eight words long and is done in one DMA arbitration cycle. After the LANCE receives the bus acknowledge ($\overline{\text{HLDA}}$ = low) signal, it does eight word transfers using eight DMA cycles (600 ns per cycle minimum) before it releases the bus request signal ($\overline{\text{HOLD}}$ = low).

Burst transfers are always eight word transfers unless the buffer in memory starts on an odd byte boundary or there are fewer than eight words remaining to be transferred. If the buffer starts on an odd address, one byte and seven words are transferred on the first burst transfer. If fewer than eight words remain to be transferred at the end of a packet, the burst transfer transfers as many full words as remain plus a single byte (if any) as the last transfer in the burst.

SINGLE WORD DMA TRANSFERS

The LANCE initiates single word DMA transfers to access the Transmit Descriptor Ring, Receive Descriptor Ring, and Initialization Block. The LANCE does not initiate any burst transfers when it owns the descriptor and is accessing the descriptor entries (an average of 3 to 4 separate DMA cycles are required for a multi-buffer packet) or when it is reading the Initialization Block.

1.8.7 SILO OPERATION

The Silo provides temporary buffer storage for data being transferred between the parallel bus I/O pins and the serial bus I/O pins in the LANCE. The capacity of the Silo is 48 bytes. It is a first in first out (FIFO) buffer.

TRANSMIT

Data is loaded into the Silo from memory under internal microprogram control. The Silo must have more than eight words (16 bytes) empty before the LANCE requests the DMA bus (by asserting $\overline{\text{HOLD}}$). The LANCE starts sending the preamble to the cable (providing the line is idle) as soon as the first byte is loaded into the Silo from memory. If a collision is detected causing the transmitter to back off, loading the Silo continues until it contains 32 bytes ready for transmission. Reception has priority over transmission during the time that the transmitter is backing off.

RECEIVE

Data from the cable is loaded into the Silo through the serial input shift register during reception. Data is transferred from the Silo into memory under internal microprogram control. The LANCE waits until the Silo has at least eight words (16 bytes) of data before initiating a DMA burst transfer. Neither the preamble nor the sync is loaded into the Silo during reception.

MEMORY BYTE ALIGNMENT

Memory buffers begin and end on arbitrary byte boundaries. Parallel data is byte-aligned between the Silo and DAL lines (DAL$_0$-DAL$_{15}$). Byte alignment can be reversed by setting the Byte Swap (BSWP) BIT in CSR$_3$. Byte Swap (BSWP = 1) is done only on transfers between the Silo and memory. Refer to Table 1-2.

Table 1-2. Silo - Memory Byte Alignment

Word/byte in Memory	BSWP	Silo Byte	DAL Pins
Word from even memory address	0	n	0 - 7
	0	n + 1	8 - 15
	1	n	8 - 15
	1	n + 1	0 - 7
Byte from even memory address	0	n	0 - 7
	1	n	8 - 15
Byte from odd memory address	0	n	8 - 15
	1	n	0 - 7
Word to even memory address	0	n	0 - 7
	1	n + 1	8 - 15
Byte to even memory address	0	n	0 - 7
	0	x	8 - 15
	1	n	8 - 15
	1	x	0 - 7
Byte to odd memory address	0	x	0 - 7
	0	n	8 - 15
	1	x	8 - 15
	1	n	0 - 7

1.8.8 ETHERNET/IEEE 802.3 INTERFACE

The Ethernet (or IEEE 802.3 standard) is the data link and physical level in the communication system. The LANCE performs the encapsulation/decapsulation function of the data link layer and also interfaces the Ethernet through a Serial Interface Adapter (SIA) (Am7992) and a transceiver (Am7995).

A typical Ethernet/IEEE802.3 node is shown in Figure 1-8. This figure also shows a Cheapernet for comparison. Cheapernet costs less because it does not require a drop cable to the transceiver. The transceiver is a part of the DTE. The bulky Ethernet coaxial cable and tap connection is replaced by Rg58 cable and BNC connectors. Cheapernet runs at the same speed as Ethernet.

SERIAL TRANSMISSION

Serial transmission consists of sending an unbroken bit stream from the TX I/O pin of the LANCE. The bit stream has the following format: (Refer to Figure 1-9)

1. Preamble: A stream of alternating ones and zeros 62 bits long supplied by LANCE.
2. Sync.: Two successive ones following the preamble supplied by LANCE.
3. Destination Address: Six byte address supplied by user.
4. Source Address: Six byte address supplied by user.
5. Type: Two byte field for type or length information supplied by user.
6. Data: 46 to 1500 bytes of data serialized with the LSB first supplied by user.
7. CRC: Four byte cyclic redundancy check code supplied by LANCE. It is an inverted 32 bit polynomial calculated from the address, type field, and data field. It is not transmitted if the transmission of the data field is truncated for any reason, CLSN becomes asserted (collision detected), or MODE <3> DTCR = 1 in a normal or loopback transmission mode.

The format for the IEEE 802.3 MAC Frame is shown in Figure 1-10. It is almost identical to the Ethernet format except for the labeling and a pad preceding the CRC. The preamble is shorter and the sync is longer but the code is identical.

Transmission is indicated at the I/O pin by the assertion of TENA with the first bit of the preamble. TENA is negated after the last transmitted bit.

The LANCE starts transmitting the preamble when all of the following conditions are met:

1. There is at least one byte of data to be transmitted in the Silo.
2. No carrier detected. Both RENA and CLSN are inactive.
3. The interpacket delay has elapsed.
4. If a retransmission, the backoff interval has elapsed.

The Interpacket Gap Time (IPG) for back to back transmission is 9.6 to 10.6 microseconds including synchronization. The interpacket delay interval begins immediately after the negation of the RENA signal. During the first 4.1 us of the IPG, RENA is masked internally.

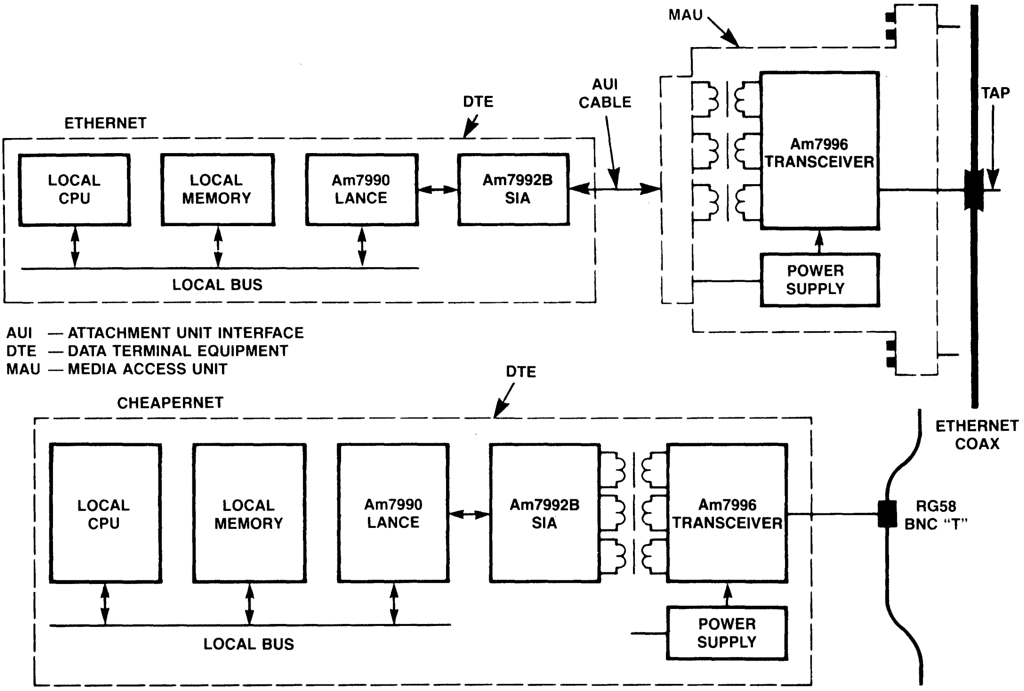
When the LANCE is waiting for the interpacket delay to elapse, it begins transmission immediately after the interpacket delay interval, independent of the state of RENA. However, RENA must be asserted during the time that TENA is high. Otherwise, the LCAR (loss of carrier) bit is set in TMD_3 after the packet has been transmitted.

SERIAL RECEPTION

Serial reception consists of receiving an unbroken bit stream on the RX I/O pin of the LANCE. The Destination Address and data are framed into bytes and enter the Silo. The Source Address and Type field are part of the data which are transparent to the LANCE. They pass through the Silo into memory along with the CRC. The LANCE strips off the preamble and sync bits. Refer to Figure 1-9 and Figure 1-10. The format is:

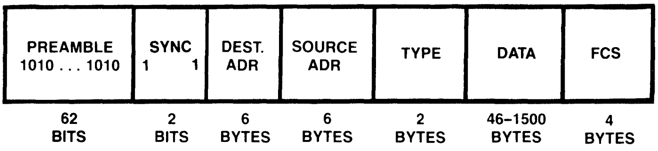
1. Preamble/Start bit: Two ones occurring a minimum of eight bit times after the assertion of RENA. The last one is the Start bit.
2. Destination Address: The 48 bits (six bytes) following the start bit.
3. Data: The serialized byte stream following the Destination Address. The last four complete bytes are the CRC.

Reception is indicated at the I/O pin by the assertion of RENA and the presence of clock on RCLK while TENA is inactive. After the eighth RCLK, (about 800ns), after RENA is asserted (goes high), the LANCE samples the incoming bits looking for the sync bits "11" following a zero. After it finds the sync bits, it starts calculating CRC on the incoming bits and loading the Silo on byte boundaries.



05998A-2

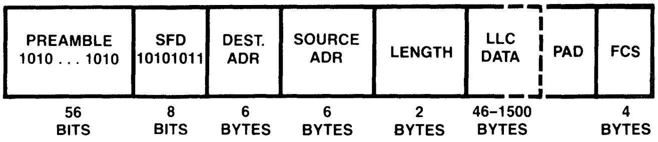
Figure 1-8 Typical Ethernet/IEEE 802.3 Node



PREAMBLE 1010 . . . 1010	SYNC 1 1	DEST. ADR	SOURCE ADR	TYPE	DATA	FCS
62 BITS	2 BITS	6 BYTES	6 BYTES	2 BYTES	46–1500 BYTES	4 BYTES

05726A-22

Figure 1-9 Ethernet Frame Format



PREAMBLE 1010 . . . 1010	SFD 10101011	DEST. ADR	SOURCE ADR	LENGTH	LLC DATA	PAD	FCS
56 BITS	8 BITS	6 BYTES	6 BYTES	2 BYTES	46–1500 BYTES		4 BYTES

05726A-23

Figure 1-10 IEEE 802.3 MAC Frame Format

The LANCE discards packets with less than 64 bytes (runt packet) and reuses the receive buffer for the next packet. This is the only case when the LANCE discards a packet. A runt packet is normally the result of a collision.

If RENA is asserted and remains asserted during the first 4.1us of IPG following a receive, the LANCE will not receive the packet. If this condition occurs following a transmit, the LANCE starts to look for the sync bits (011) about 800ns (8 bit times) after the 4.1us window has elapsed. Therefore, the packet may be received correctly if at least 8 bits of the preamble are received after the 4.1us window. Otherwise, the packet may contain a CRC error. (The LANCE may be locking to a sync pattern in the middle of data.) The packet may be discarded because the loss of data during the 4.1us window made it a runt packet.

If RENA is asserted after the 4.1us window, the LANCE treats this as the start of a new packet. It starts to look for the sync bits 8 bit times after RENA becomes active.

When a received packet terminates, if it does not end on a byte boundary, a framing error has occurred. The number of bits left over after the last full byte are called dribbling bits. The framing error is reported to the user as follows:

1. If there is no CRC error, then no framing error is reported (FRAM = 0) regardless of the number of dribbling bits.
2. If there is a CRC error, then the framing error is reported (FRAM = 1) if there are one or more dribbling bits.
3. If there are no dribbling bits, there is no framing error. There may or may not be a CRC error.

THE LANCE RECOVERY AND RE-INITIALIZATION

The transmitter and receiver section of the LANCE are turned on via the Initialization Block, Mode Register DRX and DTX bits. The state of the transmitter and the receiver are monitored through the CSR_0 RXON, TXON bits). The LANCE must be re-initialized to transmit/receive if the transmitter and/or receiver has not been turned on during the original initialization. The LANCE must also be re-initialized if either section shuts off because of an error (MERR, UFLO, TX BUFF).

The LANCE should be re-initialized with care. The user should rearrange the descriptors in the transmit and receive rings prior to re-initialization. This is necessary because the transmit and receive descriptor ring pointers are reset to the beginning of the rings during initialization.

To re-initialize the LANCE, the user must stop the LANCE by setting the stop bit in CSR_0 prior to setting the INIT bit (in CSR_0). Because setting the stop bit clears CSR_3, this register must be reprogrammed unless default values are used for BCON, ACON, and BSWP. CSR_1 and CSR_2 must be reloaded when the STOP bit is set.

The LANCE may be restarted without initialization by setting the STRT bit in CSR_0. The STRT bit starts the LANCE in accordance with the parameters setup in the Mode register. If DTX and DRX are set to zero in the Mode Register, the transmitter and the receiver will be turned on again when the STRT bit is set. However, this method of restart is not recommended if the LANCE stopped in the middle of a transmission or reception or if the buffers are data-chained.

ETHERNET/IEEE 802.3 DIFFERENCES

The only differences in the frame formats are:

1. The Ethernet labels the first 62 bits preamble and the next two bits sync. The IEEE 802.3 labels the first 56 bits preamble and the next 8 bits sync (SFD). The coding of the first 64 bits is identical.
2. The Ethernet has a two-byte type field to specify the type of packet. The length is not specified in the packet. The LANCE supplies the packet length during reception. The packet length consists of all the bits starting at the destination address and ending with the last byte of the packet. The Ethernet frame consists of the packet plus the preamble.
3. The IEEE 802.3 has no type field but in its place is a two-byte length field to specify the amount of data in the packet.
4. The IEEE 802.3 has a pad field at the end of the data portion and preceding the CRC. The use of a pad allows the user to send and receive packets that have less than 64 bytes of data. In contrast, Ethernet requires a minimum packet size of 64 bytes for packets that are not data-chained and 100 bytes for packets that are data-chained.
5. Ethernet, Version 2, specifies that the collision detect of the transceiver must be activated during the interpacket gap time.

1.9 LANCE REGISTERS AND BUFFERS

There are four Control and Status Registers (CSRs) resident within the chip. CSR_0 contains the current status information during operation and the Start, Stop, and Initialize control signals. CSR_1 and CSR_2 contain the address in memory of the Initialization Block. CSR_3 contains control bits for Byte Swap, ALE Control, and Byte Control. These registers are described following a description of the two ports used to access the CSRs. The CSRs are accessed through two bus-addressable ports, an address port (RAP) and a data port (RDP).

1.9.1 ADDRESS AND DATA PORTS

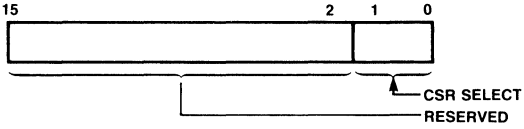
The CSRs are accessed in a two-step operation. The address of the CSR to be accessed is written into the address port (RAP) during a bus slave transaction. During a subsequent bus slave transaction, the data being read from (or written into) the data port (RDP) is read from (or written into) the CSR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

ADR I/O Pin	Port
L	Register Data Port (RDP)
H	Register Address Port (RAP)

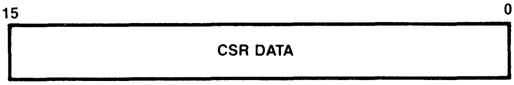
REGISTER ADDRESS PORT (RAP)



AF001490

Bit	Name	Description
15:02	RES	Reserved and read 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 CSR$_0$ 01 CSR$_1$ 10 CSR$_2$ 11 CSR$_3$

REGISTER DATA PORT (RDP)

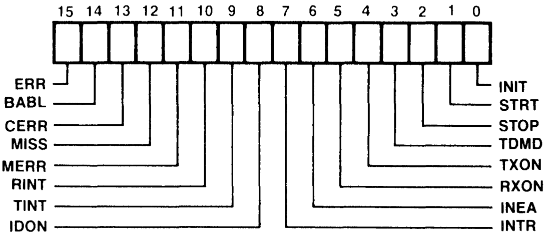


AF001450

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. CSR$_1$, CSR$_2$ and CSR$_3$ are accessible only when the STOP bit of CSR$_0$ is set. If the STOP bit is not set while attempting to access CSR$_1$, CSR$_2$ or CSR$_3$, the chip will return READY, but a READ operation will return undefined data. WRITE operation is ignored.

1.9.2 CONTROL AND STATUS REGISTERS

CONTROL AND STATUS REGISTER 0 (CSR$_0$)



AF000860

The LANCE updates CSR$_0$ by logical "OR"ing the previous and present value of CSR$_0$. CSR$_0$ is accessible when RAP = 00.

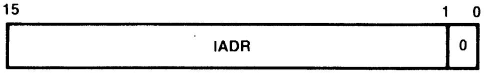
Bit	Name	Description
15	ERR	ERROR summary is set by the "OR" of BABL, CERR, MISS and MERR. ERR remains set as long as any of the error flags are true. ERR is read only; writing to it has no effect. It is cleared by Bus $\overline{\text{RESET}}$, by setting the STOP bit, or clearing the individual error 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 is set after 1519 data bytes have been transmitted; the chip continues to transmit until the whole packet is transmitted or there is a failure. When BABL error occurs, an interrupt is generated if INEA = 1. BABL is READ/CLEAR ONLY. It is set by the 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.
13	CERR	COLLISION ERROR is a collision after transmission transceiver test feature. It indicates that the collision input to the LANCE failed to activate within 2us after a LANCE-initiated transmission was completed. This function is also known as heartbeat. CERR is READ/CLEAR ONLY. It is set by the chip 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.

Bit	Name	Description
12	MISS	MISSED PACKET is set when the receiver loses a packet because it does not own a receive buffer and the silo has overflowed, indicating loss of data. Silo overflow is not reported because there is no receive ring entry in which to write status. When MISS is set, an interrupt is generated if INEA = 1. MISS is READ/CLEAR ONLY. It is set by the 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.
11	MERR	MEMORY ERROR is set when the LANCE is the Bus Master and has not received $\overline{\text{READY}}$ within 25.6us after asserting the address on the DAL lines. When a Memory Error is detected, the receiver and transmitter are turned off and an interrupt is generated if INEA = 1. MERR is READ/CLEAR ONLY. It is set by the 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 LANCE updates an entry in the Receive Descriptor Ring for the last buffer received before fall of carrier. When RINT is set an interrupt is generated if INEA = 1. RINT is READ/CLEAR ONLY. It is set by the 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 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 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 chip has completed the initialization procedure started by setting the INIT bit. The 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. It is set by the 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 "OR" of BABL, MISS, MERR, RINT, TINT and IDON. If INEA = 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 INTR I/O pin to be driven low when the Interrupt Flag is set. If INEA = 1 and INTR = 1, the INTR I/O 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 cannot be set while STOP bit is set. INEA can be set in parallel or after INIT and/or STRT bit is set.
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 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.
03	TDMD	TRANSMIT DEMAND - when set - causes the LANCE to access the Transmit Descriptor Ring without waiting for the polling time interval to elapse. TDMD need not be set to transmit a packet; it merely hastens the chip'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 $\overline{\text{RESET}}$ or by setting the STOP bit. Writing a "0" in this bit has no effect.

Bit	Name	Description
02	STOP	STOP disables the chip from all external activity when set and clears the internal logic. Setting STOP is the equivalent of asserting $\overline{\text{RESET}}$. The 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 overrides 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.
01	STRT	START enables the LANCE to send and receive packets, perform direct memory access, and do buffer management. STOP bit must be set prior to setting the STRT bit. Setting STRT clears the STOP bit. INIT and STRT must not be set in the same cycle. Set INIT first and wait for IDON (IDON=1) before setting STRT bit. INIT is READ/WRITE WITH ONE ONLY. Writing a "0" into this bit has no effect. STRT is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
00	INIT	INITIALIZE, when set, causes the chip to begin the initialization procedure and access the Initialization Block. STOP bit must be set prior to setting the INIT bit. Setting INIT clears the STOP bit. INIT and STRT must not be set in the same cycle. Set INIT first and wait for IDON (IDON=1) before setting STRT bit. INIT is READ/WRITE WITH "1" ONLY. Writing a "0" into this bit has no effect. INIT is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.

CONTROL AND STATUS REGISTER 1 (CSR_1)

READ/WRITE: Accessible only when the STOP bit of CSR_0 is a ONE and RAP = 01.

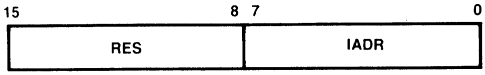


AF000970

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 (CSR_2)

READ/WRITE: Accessible only when the STOP bit of CSR_0 is a ONE and RAP = 10.



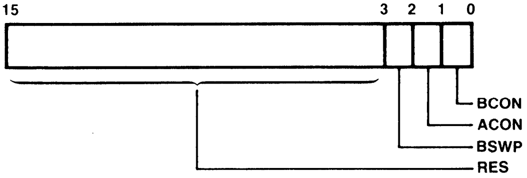
AF000920

Bit	Name	Description
15:08	RES	Reserved.
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 (CSR_3)

CSR_3 allows redefinition of the Bus Master interface.

READ/WRITE: Accessible only when the STOP bit of CSR_0 is ONE and RAP = 11. CSR_3 is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit in CSR_0.



AF000900

Bit	Name	Description
15:03	RES	Reserved and read as "0".
02	BSWP	BYTE SWAP allows the LANCE to operate in systems that consider bits (15:08) of data to be at an even address and bits (07:00) to be at an odd address. When BSWP = 1, the LANCE will swap the high and low bytes on DMA data transfers between the silo and bus memory. Only data from silo transfers is swapped; the Initialization Block data and Descriptor Ring entries are NOT swapped. BSWP is READ/WRITE and cleared by $\overline{\text{RESET}}$ or by setting the STOP bit in CSR_0.

Bit	Name	Description
01	ACON	ALE CONTROL defines the assertive state of ALE when the LANCE is a Bus Master. ACON is READ/WRITE and cleared by $\overline{RESET}$ and by setting the STOP bit in CSR$_0$. 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 $\overline{RESET}$ or by setting the STOP bit in CSR$_0$.

BCON	Pin 15	Pin 16	Pin 17
0	$\overline{BM_0}$	$\overline{BM_1}$	$\overline{HOLD}$
1	BYTE	$\overline{BUSAKO}$	$\overline{BUSRQ}$

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

1.9.3 INITIALIZATION BLOCK

The Initialization Block consists of five registers; Mode Register, Physical Address Register, Logical Address Filter, Receive Descriptor Ring Pointer, and Transmit Descriptor Ring Pointer. Chip initialization includes the reading of the Initialization Block in memory to obtain the operating parameters.

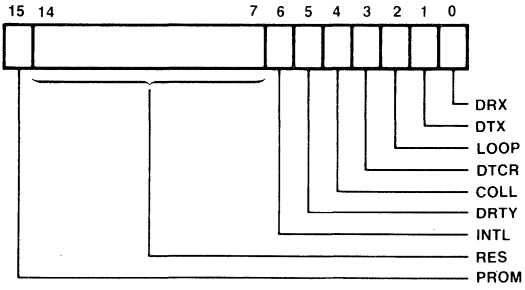
The Initialization Block is read by the chip when the INIT bit in CSR$_0$ is set. The INIT bit should be set before or concurrent with the STRT bit to insure proper parameter initialization and chip operation. After the chip has read the Initialization Block, IDON is set in CSR$_0$ and an interrupt is generated if INEA = 1.

The following is a definition of the Initialization Block.

Highest Address	TLEN-TDRA (31:16)	IADR + 22
	TDRA (15:00)	IADR + 20
	RLEN-RDRA (31:16)	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	MODE	IADR + 00

MODE REGISTER

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



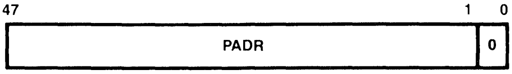
AF000510

Bit	Name	Description
15	PROM	PROMISCUOUS mode. When PROM = 1, all incoming packets are accepted.
14:07	RES	RESERVED.
06	INTL	INTERNAL LOOPBACK is enabled when the LOOP bit is set. Internal loopback lets the LANCE receive its own transmitted packet. Since this represents full duplex operation, the packet size is 8–32 bytes. INTERNAL LOOPBACK operates in promiscuous mode and when the packets are addressed to the node itself. If the source and destination are different, the LANCE must be re-initialized after each test. The LANCE will not receive external packets in INTERNAL LOOPBACK mode. EXTERNAL LOOPBACK allows the LANCE to transmit a packet through the SIA transceiver cable out to the Ethernet coax. It is used to determine the operability of all circuitry and connections between the LANCE and the coaxial cable. Multicast addressing in EXTERNAL LOOPBACK is valid only when DTCR=1 (user needs to append the 4-byte CRC). In EXTERNAL LOOPBACK, the Silo READ/WRITE pointers can get misaligned under heavy traffic. The packet may get corrupted, or not be received. Therefore, EXTERNAL LOOPBACK should be executed several times in order to isolate this type of error in receiving the lost packet. INTL is only valid if LOOP = 1; otherwise it is ignored.

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

Bit	Name	Description
05	DRTY	DISABLE RETRY. When DRTY = 1, the 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 (TMD$_3$).
04	COLL	FORCE COLLISION. This bit allows the collision logic to be tested. The LANCE must be in internal loopback mode of 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 TMD$_3$.
03	DTCR	DISABLE TRANSMIT CRC. When DTCR = 0, the transmitter generates and appends a CRC to the transmitted packet. When DTCR = 1, the CRC logic is allocated to the receiver and no CRC is generated or sent with the transmitted packet. During loopback, DTCR = 0 will cause a CRC to be generated on the transmitted packet, but no CRC check is done by the receiver since the CRC logic is shared and cannot generate and check CRC at the same time. The generated CRC is 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 checks the CRC on the received data and reports any errors. Since the CRC generator is used to generate the hash filter, the multicast addressing cannot be used when DTCR = 1.
02	LOOP	LOOPBACK allows the LANCE to operate in full duplex mode for test purposes. The packet size is 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 packet is smaller than the minimum size Ethernet packet (64 bytes). LOOP = 1 allows simultaneous transmission and reception for a message constrained to fit within the Silo. The chip waits until the entire message is in the Silo before serial transmission begins. The incoming data stream fills the Silo from behind as it is being emptied. Moving the received message out of the Silo to memory does not begin until reception has ceased. 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 LANCE to not access the Transmitter Descriptor Ring and therefore no transmissions are attempted. DTX = 1 clears the TXON bit in CSR$_0$ when initialization is complete.
00	DRX	DISABLE THE RECEIVER causes the chip to reject all incoming packets and not access the Receive Descriptor Ring. DRX = 1 clears the RXON bit in the CSR$_0$ when initialization is complete.

PHYSICAL ADDRESS REGISTER (PADR)



AF000520

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

LOGICAL ADDRESS FILTER (LADRF)

AF000500

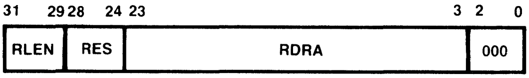
Bit	Name	Description
63:00	LADRF	The 64-bit mask used by the LANCE to accept logical addresses.

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 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 are strobed into a register. This register is used to select one of the 64-bit positins in the Logical Address Filter. If the selected filter bit is a "1" the address is accepted and the packet is 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 physical address that was loaded through the Initialization Block.

The Broadcast address, which is all Ones, 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.

RECEIVE DESCRIPTOR RING POINTER



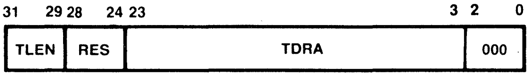
AF000490

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

Bit	Name	Description
28:24	RES	RESERVED
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 Rings are aligned on quadword boundaries.

TRANSMIT DESCRIPTOR RING POINTER



AF000480

Bit	Name	Description
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

Bit	Name	Description
28:24	RES	RESERVED
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 Rings are aligned on quadword boundaries.

1.9.4 DESCRIPTOR RINGS

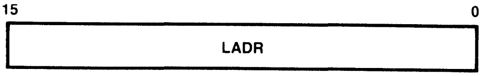
Buffer Management is accomplished through message descriptors organized in ring structures in memory. Each message descriptor entry is four words long. There are two rings allocated for the device: a receive ring and a transmit ring. The device is capable of polling each ring for buffers to either empty or fill with packets to or from the channel. The device is also capable of entering status information in the descriptor entry. Chip polling is limited to looking at the descriptor entry following the current entry.

The location of the descriptor rings and their length are found in the Initialization Block, accessed during the initialization procedure by the chip. Writing a "ONE" into the STRT bit of CSR_0 will cause the chip to start accessing the descriptor rings and enable it to send and receive packets.

The LANCE communicates with a HOST device through the ring structures in memory. Each entry in the ring is either "owned" by the chip 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.

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

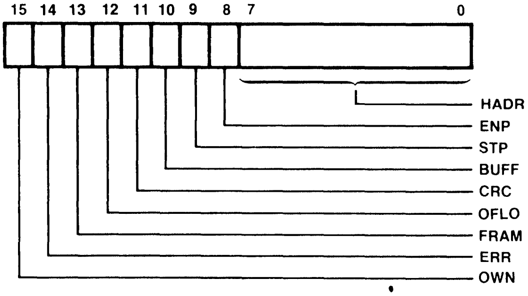
RECEIVE MESSAGE DESCRIPTOR 0 (RMD_0)



AF000940

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 unchanged by the LANCE.

RECEIVE MESSAGE DESCRIPTOR 1 (RMD$_1$)

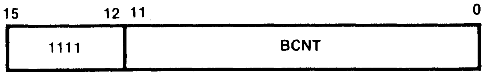


AF000870

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = 0) or by the chip (OWN = 1). The 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 chip 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 contains a non-integer multiple of eight bits and there is a CRC error. If the incoming packet has no CRC error, then FRAM will not be set even if the packet has a non-integer multiple of eight bits. FRAM is not valid in internal loopback mode. See Note 1.
12	OFLO	OVERFLOW error indicates that the Silo overflowed and all or part of the incoming packet was lost because no memory buffer was available. See Note 1.
11	CRC	CRC indicates that the receiver has detected a CRC error on the incoming packet. See Note 1.
10	BUFF	BUFFER ERROR is set any time the chip does not wn 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) silo overflow occurred before the chip received the next STATUS. If a Buffer Error occurs, an Overflow Error may also occur internally in the Silo, but is not 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 of a packet. It is used for data chaining buffers.
08	ENP	END OF PACKET indicates that this is the last buffer of 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. See Note 1.
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 LANCE.

Note 1. Disregard the error status in RMD$_1$ when both OVFL and ENP are set (no overflow error).
ONFL is valid only when the ENP is "0".
CRC and FRAM are valid only when ENP is "1".

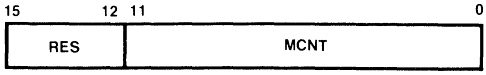
RECEIVE MESSAGE DESCRIPTOR 2 (RMD$_2$)



AF000930

Bit	Name	Description
15:12		MUST BE ONES. This field (all Ones) is written by the Host and unchanged by the 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 unchanged by the LANCE. Minimum buffer size is 64 bytes for the first buffer packet.

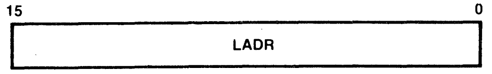
RECEIVE MESSAGE DESCRIPTOR 3 (RMD$_3$)



AF000950

Bit	Name	Description
15:12	RES	RESERVED and read 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.

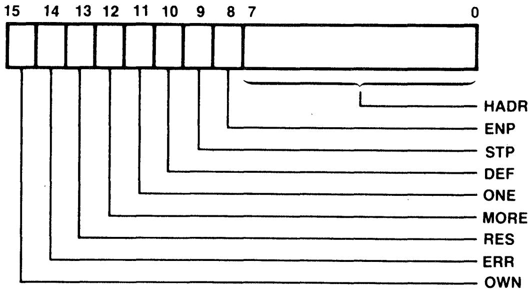
TRANSMIT MESSAGE DESCRIPTOR 0 (TMD$_0$)



AF000940

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 unchanged by the LANCE.

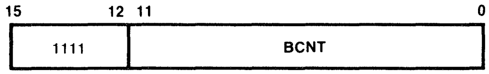
TRANSMIT MESSAGE DESCRIPTOR 1 (TMD$_1$)



AF000880

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = 0) or by the LANCE (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by this descriptor. The chip clears the OWN bit after transmitting the contents of the buffer. Neither the host nor the chip must alter a descriptor entry after it has relinquished ownership.
14	ERR	ERROR summary is the "OR" of LCOL, LCAR, UFLO or RTRY.
13	RES	RESERVED bit. The LANCE will write this bit with a "0".
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. One flag is not valid when LCOL is set.
10	DEF	DEFERRED indicates that the chip had to defer while trying to transmit a packet. This condition occurs if the channel is busy when the chip is ready to transmit.
09	STP	START OF PACKET indicates that this is the first buffer of this packet. It is used for data chaining buffers. STP is set by the host and unchanged by the chip. The STP bit must be set in the first buffer of the packet or the LANCE will skip over this descriptor, poll the next descriptor(s) entry 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 chip 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 unchanged by the 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 unchanged by the LANCE.

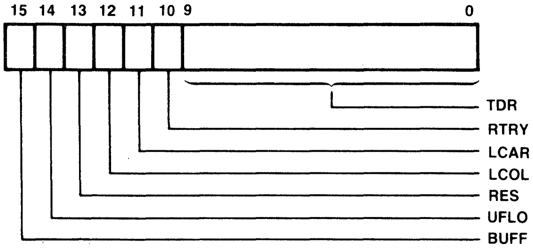
TRANSMIT MESSAGE DESCRIPTOR 2 (TMD$_2$)



AF000980

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

TRANSMIT MESSAGE DESCRIPTOR 3 (TMD$_3$)



AF000890

Bit	Name	Description
15	BUFF	BUFFER ERROR is set by the LANCE during transmission when the chip 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 Silo underflow occurred before the LANCE received the next STATUS signal. BUFF is set by the chip. BUFF error disables the transmitter (CSR_0 = TXON = 0) If a Buffer Error occurs, an Underflow Error will also occur. BUFF error is not valid when LCOL, or RTRY errors are set.
14	UFLO	UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that the Silo has emptied before the end of the packet was reached.
13	RES	RESERVED bit. The chip 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 LANCE does not retry on late collisions.
11	LCAR	LOSS OF CARRIER is set when the carrier input (RENA) to the LANCE goes false during a chip-initiated transmission. The chip does not retry upon loss of carrier. It will continue to transmit the whole packet LCAR until packet is finished. 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 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 chip and is valid only if RTRY is set.

CHAPTER 2
BUS INTERFACE CONSIDERATIONS

This chapter discusses the requirements of the LANCE in various configurations relating to the bus master or masters, bus bandwidth, and bus latency.

The LANCE may be required to fit into an already designed architecture or a new architecture may be designed around the LANCE. It is obvious that the latter case results in a cleaner design and more reliable system. The following analysis is being developed from a simple architecture where the LANCE and CPU are the only master devices in the system, to a more complex system with multiple master devices.

A system may be configured with:

The LANCE and one CPU as the only bus masters
The LANCE in systems with multiple other masters
The LANCE in a Daisy Chain operation
The LANCE with 8-bit microprocessors

Following a brief definition of bus bandwidth, bus latency, and LANCE DMA cycles, the LANCE bus latency and DMA operation are discussed in each of the above configurations. The bus bandwidth and latency are different for each of these configurations.

2.1 BUS BANDWIDTH DEFINITION

Bus bandwidth is defined as the percentage of the time that the bus is held and controlled by a device. Assuming that no wait states are added to the DMA cycles in a simplified system environment, the LANCE acquires the bus within a few cycles after requesting it. In order to handle the Ethernet transmission data rate of 10 Megabits per second, the LANCE must request the bus approximately every 12.8 us. The LANCE uses the bus for 4.8 us for eight words of DMA transfer. Therefore, 37.5% (4.8/12.8) of bus bandwidth is taken by the LANCE and the remainder is left to the CPU to perform other tasks. Any wait state added to the LANCE 8-word DMA cycle increases the LANCE DMA cycle by 800 ns and increases the bus bandwidth taken by the LANCE by 6.25 percentage points (see Figure 2-1).

2.2 BUS LATENCY DEFINITION

The term "bus latency" is defined as follows:

Bus latency is the amount of time from the time that the DMA device requests the bus (LANCE, HOLD=L) until the time it acquires the bus (LANCE, HLDA=L). Allowable Bus Latency is the amount of time that a DMA device can wait for the bus after requesting it without the loss of data or other functions.

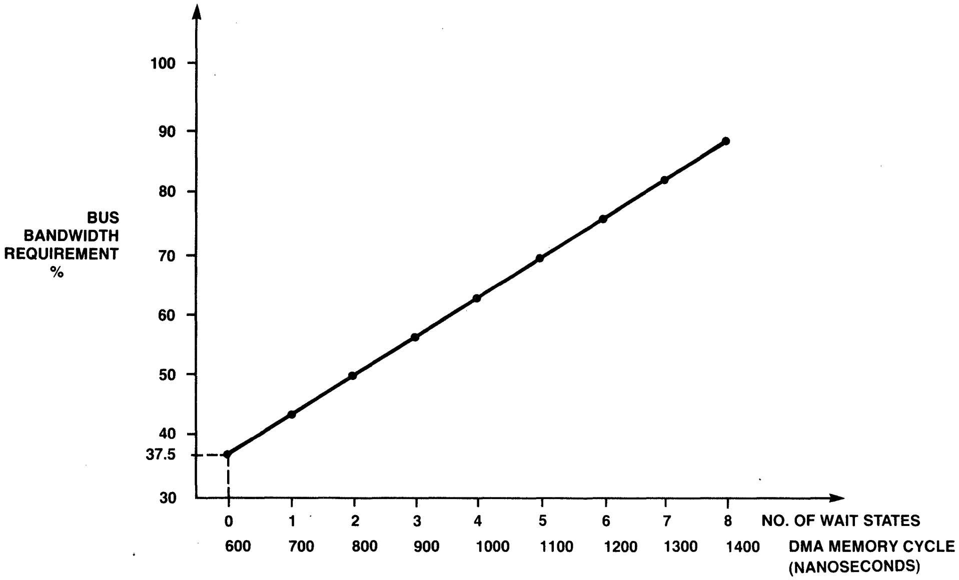


Figure 2-1 Bus Bandwidth Requirement

06363A-4

Bus latency is a limiting factor in assigning peripheral devices to a bus. Devices sensitive to the value of bus latency are: communication controllers, display memory scanners, and dynamic memories when a DMA approach is used.

Communication devices require that data be transferred to/from memory before an overflow/underflow condition occurs. The CRT controllers must scan the display memory to update and/or refresh the screen periodically. Each DMA device on a bus increases the bus latency when sharing the bus with other DMA devices. Therefore, the bus latency of each DMA device should be taken into account in designing a system.

2.3 LANCE DMA CYCLE (BUS MASTER MODE)

The LANCE is capable of two different types of DMA Transfers:

— Burst mode DMA
— Single word DMA

Burst mode DMA is used for transmission of packets from memory to the Silo (read) or for reception of packets from the Silo to memory (write).

The burst transfers are eight consecutive word reads (transmit) or writes (receive) that are done on a single bus arbitration cycle. In other words, after the LANCE receives the bus acknowledge, (HLDA = Low), it will do eight word transfers, or one byte and seven word transfers, (8 DMA cycles, at a minimum of 600 ns per cycle) before releasing the bus request signal by HOLD going high. If there are more than 8 words of data in the Silo in Receive Mode when the LANCE releases the bus (HOLD deasserted), the LANCE will request the bus again within 700 ns (HOLD dwell time). Burst DMAs are always eight cycle transfers unless there are less than eight words left to be transferred to/from the Silo.

The LANCE initiates single word DMA transfers to access the Transmit Rings, Receive Rings, or Initialization Block. The LANCE will not initiate any burst DMA transfer from the time that it gets to own the descriptor to the time it accesses the descriptor entries in the ring (an average of three to four separate DMA cycles for a multibuffer packet) or to the time it reads the Initialization Block.

2.4 LANCE AND ONE CPU ON THE BUS MASTER

In this configuration, the LANCE shares the bus with one master device. A dedicated node processor may be assigned to service the LANCE and the host CPU, or a single processor may be responsible for both the LANCE and the network tasks management. The bus bandwidth requirement should not be critical with only two master devices on the bus. The LANCE will request the bus when there are at least 16 bytes of data available in the Silo (receive mode) or when there are more than 16 locations available in the Silo (transmit mode). The CPU normally acknowledges the bus request within two to four cycles.

Once the LANCE acquires the bus, it will transfer eight words to/from the Silo before it relinquishes the bus. The transfer time is equivalent to about 4.8 us without any wait states. With 8.0 us Dwell time, the LANCE requests the bus every 12.8 us (8.0 + 4.8) until the whole packet is transmitted or received. The ring access is done with a single DMA arbitration cycle, either between the time that the LANCE is transferring data to/from the Silo (data chaining) or when the last buffer of a packet (receive or transmit) has been relinquished to the LANCE. When the LANCE owns the buffer, it does three to four separate DMA cycles, with 1.0 to 2.0 us dwell time for every buffer. The LANCE does not perform any DMA burst between ring access time.

2.4.1 BUS LATENCY WITH ONE OTHER MASTER

The above discussion assumes that the LANCE receives the bus acknowledge (HLDA) fast enough (within a few cycles). The large Silo size (48 bytes) allows the LANCE to tolerate a longer bus latency. When the LANCE is in receive mode and there are eight words in the Silo, the time required for the LANCE to receive the bus acknowledge can be stretched to nearly 25.6 us (16 words) before the Silo overflows. However, the very next bus request (HOLD) from the LANCE should be acknowledged within 8.0 us because there are already about 19 words in the Silo. The LANCE always releases the bus after eight word transfers, even if there are already eight or more words residing in the Silo. The second bus request will be asserted within 700 ns, when there are eight or more words in the Silo (dwell time).

2.5 LANCE WITH MULTIPLE MASTER DEVICES ON THE BUS

The LANCE is flexible when it is integrated with other master devices on the bus. A large FIFO (also referred to as Silo) 48 bytes deep, and a unique programmable interface make LANCE versatile in a multimaster environment.

The impact of integrating a master device onto an existing bus system requires careful investigation at the start of system development. The following discussion summarizes the areas of concern to the user when designing the LANCE into a multimaster bus.

In designing a bus with multiple master devices, some restrictions are imposed on the master devices. These requirements allow the bus to be utilized efficiently, and each device gets a fair share of the bus without degrading the bus bandwidth.

Some typical system bus requirements are:

1. Each device on the bus is allowed to hold the bus for only a limited time, once it has acquired it. There are many system requirements which must be considered to determine the maximum time any bus master is allowed to remain on the bus. If this time is exceeded, some type of monitoring device generates a bus error and forces the device off the bus.

The LANCE holds the bus for a minimum of 4.8 us during the DMA burst and a minimum of 600 ns when accessing the rings or the Initialization Block. If the memory access time is longer than the LANCE DMA cycle, the LANCE DMA cycle period can be stretched (wait states being added to the LANCE DMA cycle) by the use of some external logic and by controlling the READY input to the LANCE. Eventually, the READY should be granted to the LANCE before the occurrence of either an overflow or underflow error condition, or a 25.6 usec memory timeout error (MERR).

2. Any device on the bus may be kicked off the bus (preemptive DMA) at any time. This is usually enforced by a bus arbiter when a higher priority device needs to acquire the bus. Refer to Figure 2-2.

The $\overline{\text{HLDA}}$ to the LANCE cannot be deasserted while the $\overline{\text{HOLD}}$ from the LANCE is active. Once the LANCE has acquired the bus, it will not release the bus until it has finished its 8 DMA cycles (or less, if it happens to be the last few bytes of the packet) while transferring data to/from the Silo. However, the external logic pointed out previously may be used to add wait states to the LANCE DMA cycle and to isolate its address/data bus, through buffers or transceivers, from the system bus. Thus preemptive DMA can be supported by the use of external logic.

3. Within an existing bus, the master devices already on the bus may not have enough bus bandwidth left to allow for any new device to be added. The new device may be required to stay off the bus and provide its own dedicated dual port memory.

The LANCE supports a dual port RAM architecture with some external logic. The external logic is needed to allow access to the LANCE internal register, CSR_0, during the normal operation. This register contains the interrupt/control status register bits. The external logic must monitor and control $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$ of the LANCE so that the CPU can access the CSR_0 when the LANCE is not in the middle of a series of DMA accesses.

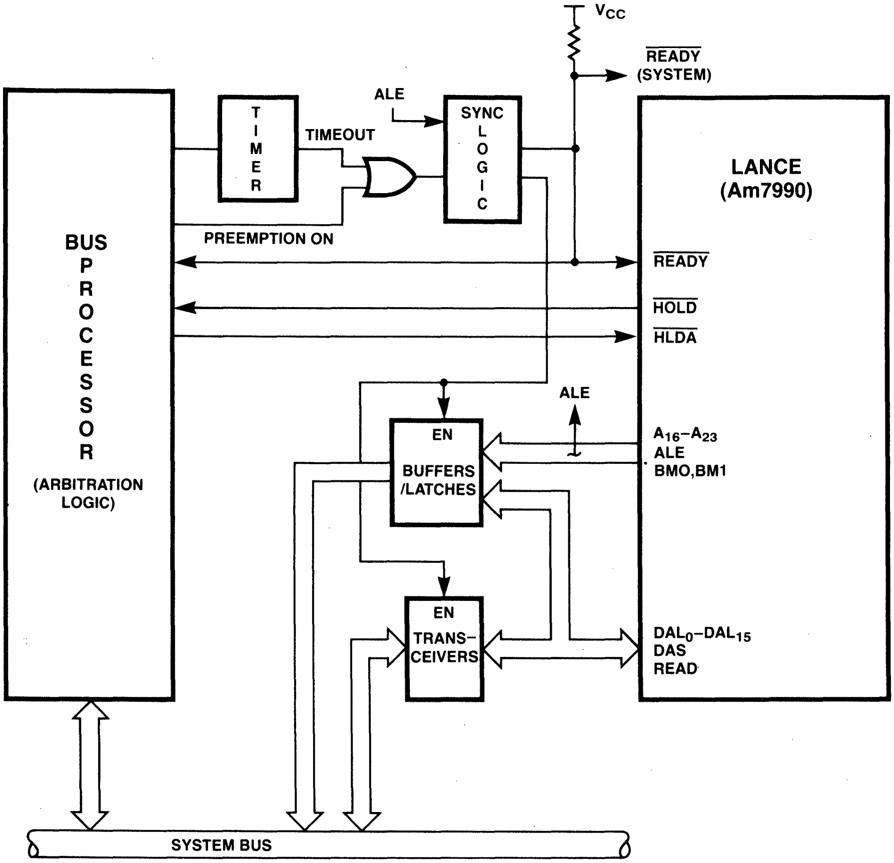


Figure 2-2 Block Diagram of Pre-emption in the LANCE

06363A-5

2.5.1 BUS LATENCY FOR MULTIMASTER ENVIRONMENT

The amount of allowable bus latency for a master device must be evaluated prior to adding a new device into a multimaster environment. We will study the allowable bus latency for the LANCE under two different cases: a single buffer per packet (no data-chaining) and multiple buffer segments per packet (data-chaining). The main difference is that, in data-chaining, the LANCE does a separate lookahead operation (requiring three DMA accesses for each buffer segment) to access the ring between the periods it is doing DMA burst. Therefore, this additional time overhead reduces the allowable bus latency. This is especially significant when short buffer sizes are assigned and long packets are in process.

NO DATA-CHAINING

When the buffers are not data-chained, the single DMA access to the ring per packet may be disregarded especially for large packets. In the receive mode, words enter the Silo from the Ethernet at the rate of eight words every 12.8 microseconds. The LANCE transfers words from the Silo is 8-word bursts requiring 4.8 microseconds per burst.

Therefore the allowable bus latency for LANCE is 12.8 minus 4.8 or 8 microseconds disregarding the single DMA access to the descriptor ring per packet.

The size of the Silo provides some protection against a one time unusual delay in granting the bus after a LANCE bus request. The LANCE automatically requests the DMA bus as soon as eight words are in the Silo. The Silo may accumulate an additional sixteen words (32 bytes) on a one time basis before the LANCE becomes the bus master. This means that the bus latency is 25.6 us (32 x .8) for the first time that the LANCE requests the bus. However, subsequent bus requests must be granted within 8 us without fail to prevent a Silo overflow. The 25.6 us bus latency can apply for one bus access at any time that the Silo has eight or fewer words in it.

The eight microsecond bus latency corresponds to an accumulation of five words in the Silo. The bus latency may be stretched by some amount when the packets are shorter.

WITH DATA-CHAINING

When the buffers are data-chained, the overhead caused by the lookahead operation in accessing the ring must be taken into consideration in calculating the bus latency. This lookahead is performed between the periods in which the LANCE is doing DMA bursts. It consists of three separate single DMA reads when the LANCE owns the buffer. The LANCE does not initiate any DMA burst (8 DMA cycles) between the three single DMA reads.

The lookahead operation may take as long as 5.8 microseconds plus three times the bus latency of a single DMA read. This is calculated as follows: The dwell time (the time interval between the LANCE's request to the bus to access the ring and the acquisition of the bus) is approximately 1.0 to 2.0 us for each access. Two ring accesses are required. Three ring access transfers at .6 microseconds each are required.

The following calculation leads to the worst case allowable bus latency for the LANCE, for large packet (1518 bytes) and small buffer sizes (64 bytes), when the buffers are data-chained.

Assume:

— No wait state is added to the LANCE DMA cycle.

— $\overline{\text{HLDA}}$ assertion to bus driver enable delay is not included in the calculation.

64-byte transfer (1 buffer) time =
4 x 4.8 = 19.2 us (4 bus requests)

Ring access transfer (3 single DMA) =
3 x 0.6 = 1.8 us (3 bus requests)

Ring access dwell time =
2 x 2.0 = 4.0 us

64-byte transfer time from/to Ethernet cable to/from Silo = 64 x 0.8 = 51.2 us

$$\begin{array}{c}\text{ALLOWABLE}\\ \text{BUS LATENCY}\end{array} = \frac{51.2 - (19.2 + 1.8 + 4.0)}{7} = 3.74 \text{ us}$$

Because of the lookahead overhead, the data-chained buffer mode has less allowable bus latency than the single buffer mode which does not require data-chaining.

2.6 LANCE IN A DAISY-CHAIN CONFIGURATION

It may be desirable to integrate the LANCE into a daisy-chained environment (for example, Z-BUS). The BCON bit in CSR$_3$ can be programmed for daisy-chain operation. As a result, Pins 15, 16, and 17 ($\overline{\text{BM}_0}$, $\overline{\text{BM}_1}$, $\overline{\text{HOLD}}$) will function as BYTE, $\overline{\text{BUSAKO}}$, and $\overline{\text{BUSRQ}}$ respectively. In a daisy chain environment with pre-emption, the LANCE should be assigned to a high priority slot. This is necessary to meet the bus latency requirement under the worst case conditions.

In a daisy-chain environment without pre-emption (Zilog daisy-chain), it is possible for a device, independent of slot position, to wait for the remainder of the devices in the chain to finish their bus accesses before it can request the bus again. (This can occur when the other devices request the bus at the same time.) The impact to the LANCE is an underflow or overflow error condition if the LANCE bus latency requirement is not met. The addition of hardware, to provide preemptive DMA capability and/or to enforce a bus time limit to the devices on the bus, may not be worthwhile if the probability of this happening in a normal system environment is reasonably low.

2.7 LANCE WITH 8-BIT MICROPROCESSORS

Although the LANCE is a 16-bit device that interfaces easily to 16-bit microprocessors, it can also be interfaced to 8-bit microprocessors with some external logic.

In integrating the LANCE with 8-bit microprocessors, there are two possible configurations: 16-bit memory and 8-bit memory. Each solution has trade-offs in cost, performance, and complexity of implementation. Each approach is considered below.

2.7.1 8-BIT MICROPROCESSORS WITH 8-BIT MEMORY

At a glance, this seems to be a simple and not too costly implementation because it requires fewer memory chips. However, as we will see later, the implementation can be quite complex.

For this interface to work properly, it takes twice as long to do the data transfers as with 16-bit memories and a 16-bit microprocessor. The data transfers are between the LANCE and CPU (LANCE in Slave Mode), or the LANCE and 8-bit memories (LANCE in Master Mode). The interface to the LANCE should look like a 16-bit interface. The DMA word transfer cycle between the LANCE and the CPU or memory must be converted to two separate byte transfers. This type of architecture degrades the system performance by one-half.

For some lower cost applications, this may not be a bottleneck. However, it is important to note that the LANCE DMA transfer cycles take twice as long. Therefore, the potential of the LANCE getting into an overflow or underflow error condition is increased, especially if the LANCE bus requests are not acknowledged promptly.

Bus latency for a 16-bit microprocessor interface is based on a 4.8 us DMA cycle. The 8-bit microprocessors must perform twice as many DMA cycles as the 16-bit microprocessor, therefore, the bus latency for the 8-bit microprocessors is based on 9.6 us DMA cycles (assuming that an 8-bit microprocessor runs as fast as a 16-bit which is not a safe assumption). The allowable bus latency for an 8-bit microprocessor is therefore less than for a 16-bit microprocessor and the bus bandwidth is almost doubled. The LANCE must request the bus once every 12.8 us, and since it takes 9.6 us to do an 8-word transfer, the LANCE thus takes up 75% of the bus bandwidth and the allowable bus latency is 3.2 usec.

The bus latency and bandwidth limitations can be overcome by designing a dual-port RAM shared by the 8-bit processor and the LANCE. Some additional logic may be required to provide an 8-bit channel for use by the 8-bit processor, and a 16-bit channel to be used by the LANCE and any other 16-bit controllers.

2.7.2 8-BIT MICROPROCESSORS WITH 16-BIT MEMORY

Using a 16-bit memory is equivalent to having an 8-bit processor on a 16-bit bus. The memory is normally organized into lower and upper bytes. Although this approach is costly in terms of using twice as many memory chips, it is a better solution in terms of performance and ease of implementation.

This architecture is nearly equivalent to the LANCE being interfaced to a 16-bit processor. The LANCE DMA cycles, bus latency, and the bus bandwidth analysis are the same as those discussed involving 16-bit microprocessors. There is still a drawback in the area where the CPU has to access the LANCE. The CPU must go through two separate consecutive cycles for every word transferred to the LANCE; however, this is not significant since the LANCE to CPU interface does not require a fast response time. With enough receive buffers allocated to the LANCE, the packets are received independently of the CPU's interrupt response time.

CHAPTER 3
LANCE-CPU INTERFACE CONSIDERATIONS

3.1 GENERAL LANCE INTERFACE REQUIREMENTS

The following is a list of key points to be considered in designs using the LANCE:

- There are three open drain outputs on the LANCE. They are $\overline{\text{INTR}}$ (Pin 11), $\overline{\text{HOLD}}$ (Pin 17), and $\overline{\text{READY}}$ (Pin 22). These signals must be pulled up external to the LANCE chip by 3.3k or larger resistors.
- $\overline{\text{DAS}}$ and ALE are usually used as strobe signals. It would therefore be a good idea to pull them up to insure that a floating level will not affect other circuits.
- Decoupling capacitors with values of 0.1 and 5.0 uf are connected, in parallel, from the V_{CC} Pin of the LANCE to Ground.
- Voltage level on the $\overline{\text{RESET}}$ Pin of the LANCE should be maintained at above 3.4 volts with a pull-up resistor.

3.1.1 LANCE INTERFACE TO POPULAR MICROPROCESSORS

Several problems are associated with interfacing a peripheral device to a CPU. One major problem involves the various control signals that each chip uses. Unless the two families are designed to be pin-for-pin compatible, there are always going to be minor variations between them. Part of the pin incompatibility involves genuine signal differences while other pins only require name changes. Another problem with interfacing a general purpose peripheral to a CPU is the timing differences.

Most of these timing differences can be resolved by adding external logic. The Am7990 LANCE has been designed to be interfaced easily with the popular 16-bit microprocessors (8086/80186, 68000, Z8000*, LSI-II**). Most of the interface logic is embedded in the chip and is program selectable.

Although the LANCE itself has a multiplexed bus, it can easily be interfaced to demultiplexed buses with a minimal amount of effort. This chapter discusses the interfaces to four popular 16-bit microprocessors (68000, Z8001, 80186, and 8086). These designs assume that the processor and the LANCE reside on the same board. Address buffers and data transceivers are set up to be shared by the processor and the LANCE. All of these designs use PAL*** devices to reduce the parts count.

3.2 68000 TO LANCE INTERFACE

The versatility of the LANCE makes it easier for the user to interface with demultiplexed buses. Figure 3-1 is an example of how to interface the 68000 to the LANCE. The goal of this interface is compatibility with 8-MHz and faster 68000s while minimizing parts count. The two flip-flops are needed to adapt the LANCE bus request handshake to the 68000.

Auto-vectoring is used since the Am7990 does not return a vector during Interrupt Acknowledge cycles. The BYTE and $\overline{\text{DAS}}$ signals of LANCE are used to generate $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$ when LANCE is in Bus Master Mode. It takes two latches to demultiplex the LANCE Address/Data Lines (DAL_0-DAL_{15}) to adapt it to the 68000 Address Bus. The user should program the BSWP, BCON to "1", and ACON to "0" in CSR_3 when initializing the LANCE. The AmPAL22V10 device could also be used to eliminate the two flip-flops shown in Figure 3-1.

3.2.1 68000 TO LANCE INTERFACE PAL

```
PAL16L8                  JOE BRCICH
PAT002                   2 FEB 84
File: 68K90.PAL

68000 TO LANCE INTERFACE
ADVANCED MICRO DEVICES

/AS  RW  BYTE  /HOLD  NC  /BG  A0  NC  /BGACK  GND
/CS  /TB  /UDS  /DAS  /CLR1  BR  /CLR2  /LDS  /RB  VCC

IF (/BGACK) RB = CS*RW*UDS + CS*RW*LDS; 68000 Master

IF (/BGACK) TB = CS*/RW; 68000 Master

IF (BGACK) UDS = DAS*/A0*BYTE + /BYTE*DAS

IF (BGACK) LDS = DAS*A0*BYTE + /BYTE*DAS

IF (/BGACK) DAS = UDS*LDS

CLR1 = /AS*BG

CLR2 = BGACK

/BR = /HOLD
```

* Z8000 is a trademark of Zilog, Inc.
** LSI-II is a registered trademark of Digital Equipment Corp.
*** PAL is a registered trademark of Monolithic Memories, Inc.

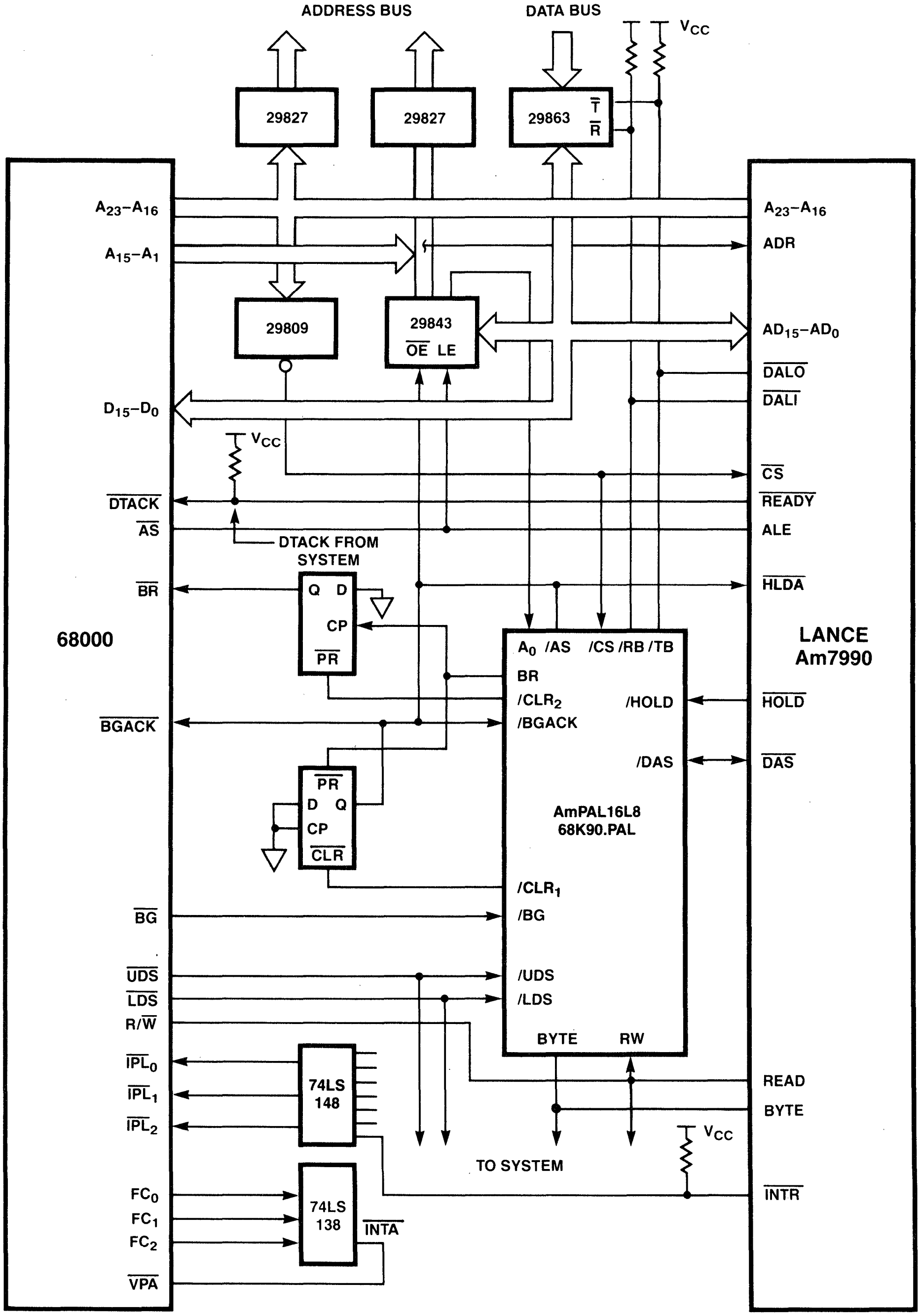


Figure 3-1 68000 to LANCE Interface

06363A-6

3.3 Z8001 TO LANCE INTERFACE

The Z8001 interface to the LANCE is easily accomplished since both have a multiplexed bus and most of the control signals can be directly connected (see Figure 3-2). This design also uses the PAL device, AmPAL16L8, to reduce the parts count. The $\overline{\text{INTR}}$ pin of the LANCE is connected to the $\overline{\text{NVI}}$ pin of Z8001, since LANCE does not return a vector during the Interrupt Acknowledge cycle. The PAL device uses the status lines ST$_3$-ST$_0$ (when Z8001 is the Bus Master) or $\overline{\text{HLDA}}$ from the LANCE (when LANCE is the Bus Master) to generate M, the memory request signal. The user should program the ACON, BCON, and BSWP to "1" prior to initializing the LANCE. When the LANCE is Bus Master, $\overline{\text{DALI}}$ and $\overline{\text{DALO}}$ control the transceiver. When the CPU is Bus Master, T and R are generated from R/$\overline{\text{W}}$ and $\overline{\text{DS}}$ to control the transceiver.

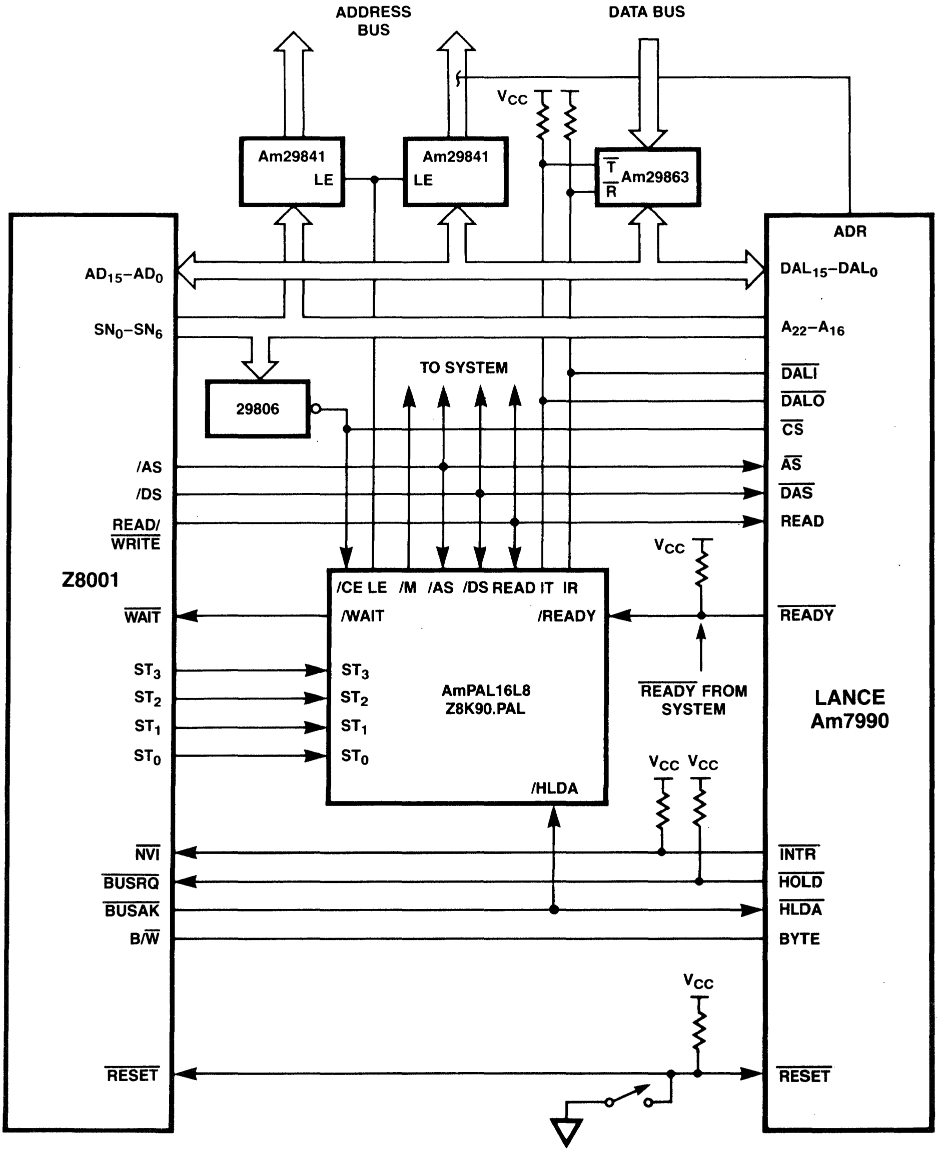


Figure 3-2 Z8001 to LANCE Interface

06363A-7

3.3.1 Z8001 TO LANCE INTERFACE PAL

```
AMPAL16L8                 RASOUL M. OSKOUY
PAT001                    MARCH 13, 1984
File: Z8K90.PAL

Z8001 TO LANCE INTERFACE
ADVANCED MICRO DEVICES

/AS  /HLDA READ /CS   /DS /READY ST3 ST2 ST1 GND

ST0 /T   /M    /WAIT LE /R     NC  NC  NC  VCC

IF /(HLDA)    T = /READ*/CS

IF (/HLDA)    R = READ*DS*/CS

              M = /HLDA*ST3*/ST2*/ST1*/ST0 +

                  /HLDA*/ST2*/ST1*ST0 +

                  /HLDA*ST3*ST2*/ST1 +

                  HLDA

           WAIT = /READY

            /LE = AS
```

3.3.2 DESCRIPTION

Most of the Z8001 and Am7990 control signals can directly be connected. The Z8001 status lines outputs, ST0-ST3, and LANCE (Am7990) HLDA input are used to indicate whether the cycle is a memory or non-memory cycle. The INTR pin of LANCE is connected to NVI pin of Z8001, since the LANCE does not return a vector during the interrupt acknowledge cycle.

Note: Program ACON, BCON, BSWP to "1" in CSR3 Reg.

3.4 80186 TO LANCE INTERFACE

This interface also uses a PAL design to reduce the parts count. The 80186/LANCE address and data buses can be connected directly together since they both have multiplexed buses. It seems natural to program the LANCE for ALE output. However, the PAL equations or a discrete design are easier if AS (CSR$_3$, ACON=1) is used. This is because the LANCE three-states ALE, the 186 does not. The INTR, READY, and HOLD signals from the LANCE are open-drain and should be pulled up. The BM$_1$ signal from the LANCE or BHE from the 186 along with A$_0$ can be used to decode the data transfer type (Word/Byte). The external address buffers and data transceivers are enabled by the LANCE and the 186. The buffers and transceivers are enabled by whichever device is the master. The user should program the BCON, BSWP to "0," and ACON to "1" in CSR$_3$. Figure 3-3 shows the 80186 to LANCE interface connections.

3.4.1 LANCE IN BUS SLAVE MODE

The LANCE enters the Bus Slave Mode whenever the CS is active. In this example, the 186 becomes the master. RD or WR of the 186 can be used to generate DAS which can be deasserted after the LANCE asserts its READY output. The DTR and DEN of the 186 are used to control the data transceivers. When the LANCE is not selected, the DTR of 186 is used to generate the READ signal to the LANCE since DTR stays valid (High for Read, Low for Write operation) during the whole cycle.

3.4.2 LANCE IN BUS MASTER MODE

The LANCE enters the Bus Master Mode whenever the HOLD and HLDA signals are both active. The READ and DAS outputs are used to generate separate RDand WR signals to meet the 186 system requirements. The DALI and the DALO outputs of the LANCE are used to control the data transceivers. The DAS signal asserted by the LANCE remains asserted until either the READY input is received or 25.6 us has elapsed from the time DAS was asserted. The time-out error is noted in CSR$_0$ (MERR=1) and the 186 is notified via interrupt.

3.4.3 186 TO LANCE INTERFACE PAL

```
PAL16L8                   JOE BRCICH
PAT001                    1 OCT 83
File: 186-90.PAL

80186 TO LANCE INTERFACE
ADVANCED MICRO DEVICES

ALE  /AS  DTR  NC  NC  DEN  NC  /READY  HLDA  GND
/CS  ARDY  READ  /R  /T  /DAS  /WR  /RD  LE  VCC

IF (/HLDA)     DAS = RD + WR

IF (/HLDA)   /READ = DTR

IF (/HLDA)       T = DTR * /CS

IF (/HLDA)       R = /DTR*DEN*/CS

IF (HLDA)       RD = READ*DAS

IF (HLDA)       WR = /READ*DAS

               /LE = /ALE + /AS

             /ARDY = /READY
```

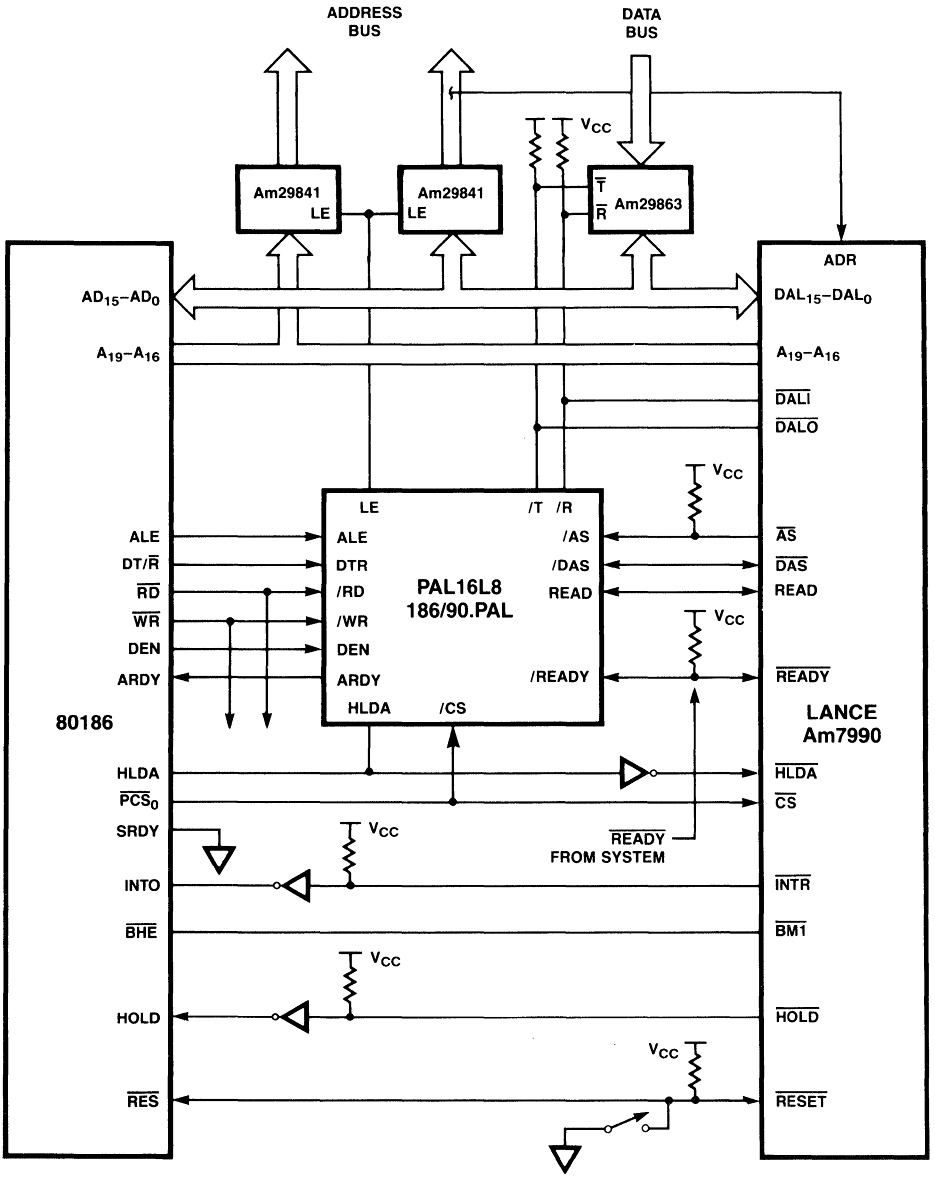
3.4.4 DESCRIPTION

This PAL design assumes that the 186 and LANCE are on the same board. The data bus buffer is only enabled if the LANCE is not selected. It seems natural to program the LANCE for ALE output. However, the PAL equations or indeed a discrete design is easier if AS is used. This is because the LANCE three-states ALE while the 186 does not. Note that data is valid on the falling edge of WR in min mode meeting the apparent requirement of the LANCE in early data sheets. Data set up time is specified with respect to the rising edge of DAS in later data sheets giving the designer more flexibility. All transfers to or from the LANCE must be word transfers.

Note: Program ACON to "1," and BCON, BSWP to "0" in CSR3 REG.

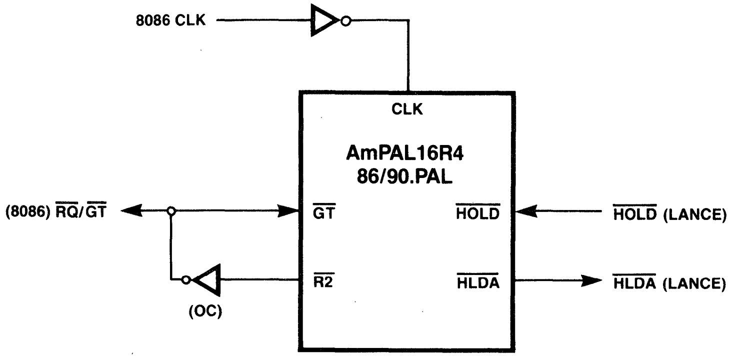
3.5 8086 TO LANCE INTERFACE

The LANCE interface to the 8086 and to the 80186 is quite similar, except that the bus request handshake is different when 8086 is configured in maximum mode. The 8086 has a bidirectional signal for both bus request and bus grant (RQ/GT). Both the bus request (RQ), and bus grant (GT) to/from 8086 are one CPU-clock wide and are synchronous to the CPU clock. Figure 3-4 shows a PAL design to do this conversion. Figure 3-5 shows the interface timing for this PAL. This PAL design could also be utilized to include other external logic requirements for interfacing the LANCE to 8086.



06363A-8

Figure 3-3 80186 to LANCE Interface



06363A-9

Figure 3-4 8086 $\overline{RQ}/\overline{GT}$, Am7990 HOLD/HLDA Conversion PAL

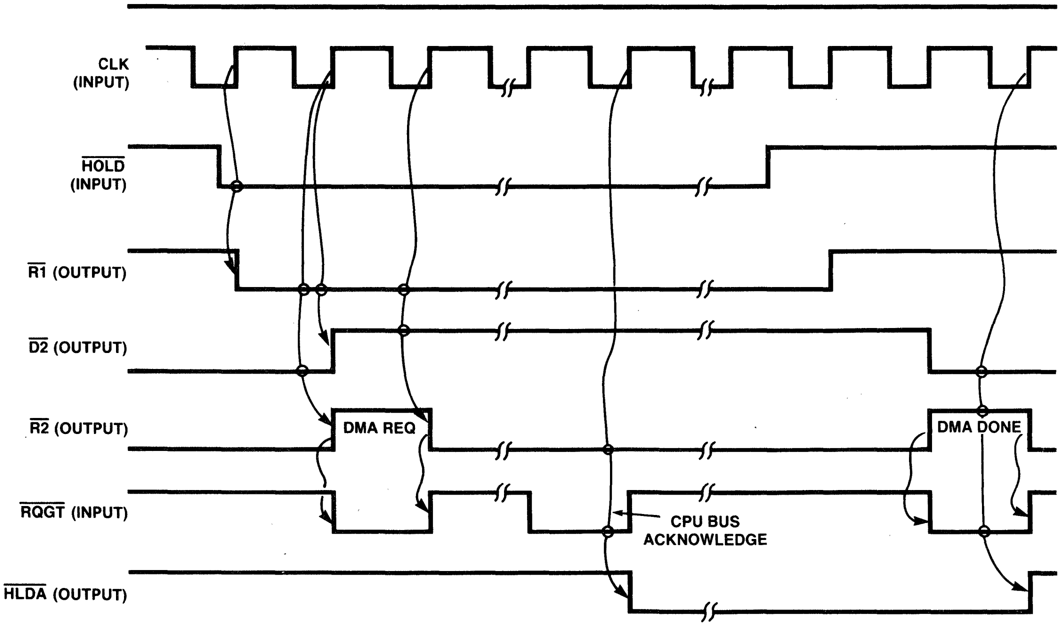


Figure 3-5 AmPAL16R4, 8086 (max. mode)/LANCE Interface Timing Diagram

06363A-10

3.5.1 8086 $\overline{\text{RQ}}/\overline{\text{GT}}$ CONVERSION TO LANCE HOLD/HLDA PAL

```
AMPAL16R4                  RASOUL OSKOUY
PAT001                     MARCH 20, 1984

8086 RQ/GT CONVERSION TO LANCE HOLD/HLDA
ADVANCED MICRO DEVICES

CLK  NC  H̅O̅L̅D̅  G̅T̅     NC  NC  NC  NC  NC  GND
NC   NC  NC    H̅L̅D̅A̅  D̅2̅  R̅2̅  R̅1̅  NC  NC  VCC

       R1 := HOLD
       D2 := R̅1̅
       R2 := R1*D̅2̅ + R̅1̅*D2
     HLDA := GT*R̅2̅ + HLDA*D̅2̅
```

3.5.2 DESCRIPTION

This PAL converts the request and grant ($\overline{\text{RQ}}/\overline{\text{GT}}$) of 8086, when configured in maximum mode, to the LANCE Am7990 $\overline{\text{HOLD}}/\overline{\text{HLDA}}$. Both request (input to 8086) and grant (output from 8086) are one clock wide and are synchronous to the CPU clock.

CHAPTER 4
SOFTWARE CONSIDERATIONS

4.1 SOFTWARE TASKS

First the LANCE interface with the system is checked out both in Bus Master and Bus Slave Mode, and the internal logic of the LANCE is tested in diagnostic mode for both internal and external loopback test. Then, the LANCE is configured for normal operation via the Initialization Block. The user normally initializes the LANCE once for normal operation. Any changes in LANCE configuration require re-initialization.

The data handling tasks include assigning and chaining data buffers and checking the format rings for format errors. Transmit and receive drivers are required to set up the transmit and receive descriptors, buffer pointers, and related tasks. Status monitoring routines are also required. If re-initialization is necessary, the user must first process all the transmitted or received packets and to rearrange the packets queued for transmission in the Transmit Ring. This is necessary since the LANCE sets its pointers to the beginning of the Transmit and Receive Rings upon initialization.

The user must also process the interrupts and report the errors.

These tasks and the software support required are discussed in the following order:

Initialization
Diagnostics
Main Program
Descriptors and Buffers
Interrupts
Transmit Drivers
Receive Drivers
Status Monitoring and Error Processing

The flow charts and discussion outlined below examine one approach to typical driver routines which may be used to operate the LANCE. They are designed to guide the user in programming the LANCE within a typical application. However, the flowcharts are not intended as a direct implementation, which is system dependent.

4.2 INITIALIZATION

The LANCE is initialized (configured) for a desired operation by programming the CSR registers (CSR_0-3) and providing the parameters required by the LANCE in the Initialization Block. Refer to Chapter 1 for an overview of the system including the Initialization Block. For detailed information, refer to the LANCE data sheet. Initialization is a prerequisite for the LANCE operation in Bus Master mode.

Upon power-up or hardware reset, the internal logic of the LANCE is cleared, and the Stop bit in CSR_0 gets set. Once the Stop bit is set, either by a hardware reset or by software, all the activity in the LANCE stops. The Stop bit in CSR_0 must be set prior to any access to CSR_1-CSR_3. ACON, BCON and BSWP bits in the CSR_3 are programmed for the desired hardware interface. CSR_1 and CSR_2 get updated to contain the pointer to Initialization Block. (Refer to LANCE data sheet for more description on CSR registers.)

CSR_1 and CSR_2 are normally not affected when the Stop bit is set in CSR_0. (When the Stop bit is set, CSR_3 is cleared and CSR_0 is set to 0004 Hex, Stop bit set.) However, it is recommended to reload CSR_1 and CSR_2 whenever the Stop bit is set.

The Initialization Block contains the parameters (Mode Register, Node Address, Logical Address Filter, Receive, and Transmit Descriptor Ring Pointers) needed for normal or diagnostic operation.

The flow chart shown in Figure 4-1 shows a typical initialization routine which starts by invoking the diagnostic operation with error loop capabilities and if the diagnostic passes are successful, it initializes the LANCE for normal operation.

4.3 DIAGNOSTICS

The Mode Register Initialization Block can be used to set up the LANCE for diagnostic or normal operation. There are four user-programmable diagnostic modes. Each mode requires the user to change bits in the Mode Register and re-initialize the LANCE. The four modes are discussed in the following order:

1. Internal Loopback
2. External Loopback.
3. CRC Logic Check.
4. Collision Detection and Retry Logic.

4.3.1 INTERNAL LOOPBACK

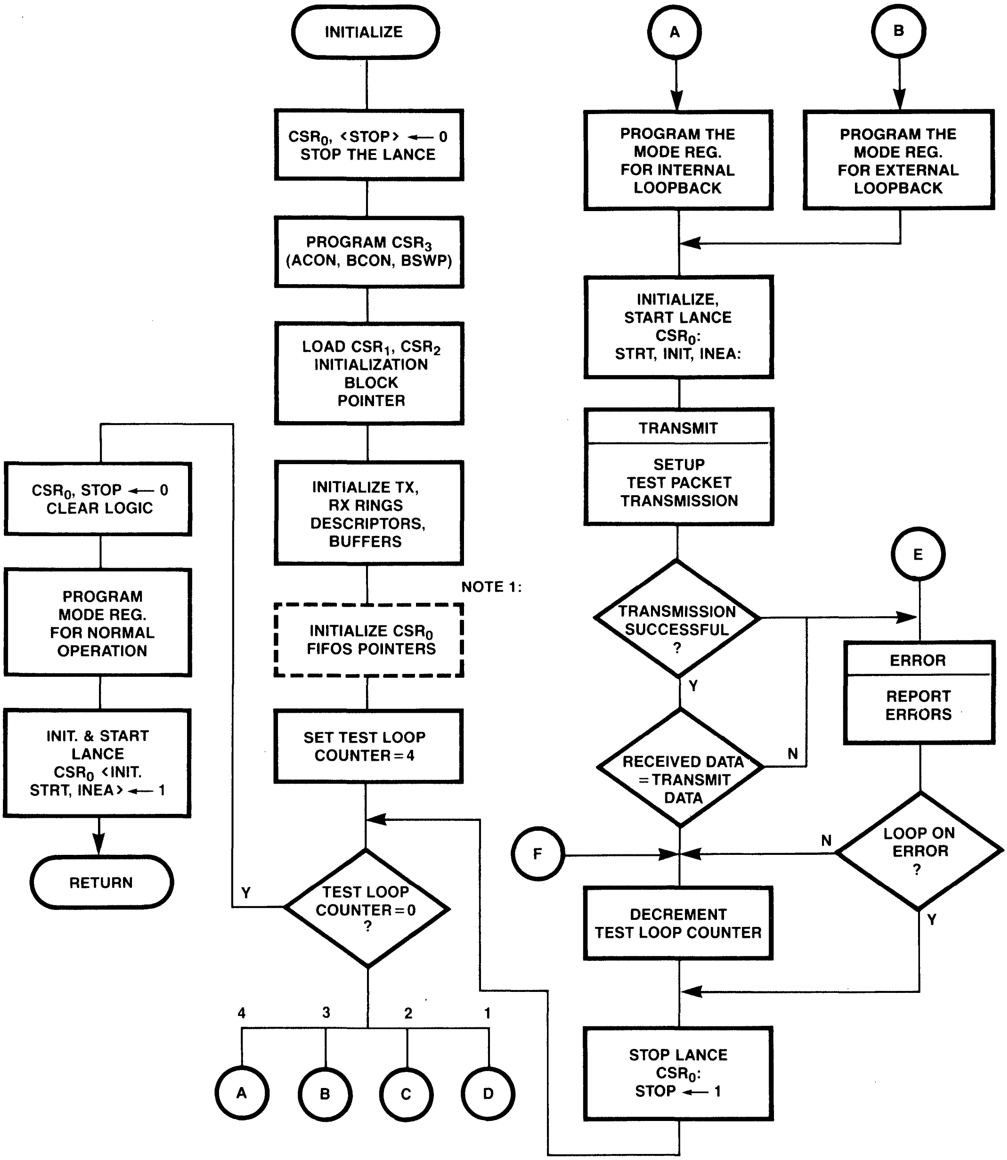
Although Ethernet is half-duplex, when in Loopback Mode the LANCE can operate in full-duplex by limiting the packet size to 8-32 bytes. The LANCE will not start the transmission of the loopback packet unless the whole packet (up to 32 bytes) is in the SILO. All the logic and data paths are tested in internal loopback except the transmit/receive drivers and receivers for the transmit data and receive data.

The LCAR bit should not be set in internal loopback mode since the LANCE does not monitor the RENA signal input to the LANCE. The LCAR bit is set when the sequence described below happens:

a) The LANCE is initialized in internal loopback mode. The loopback test can transmit and receive normally, as long as the transmit address matches the node address.

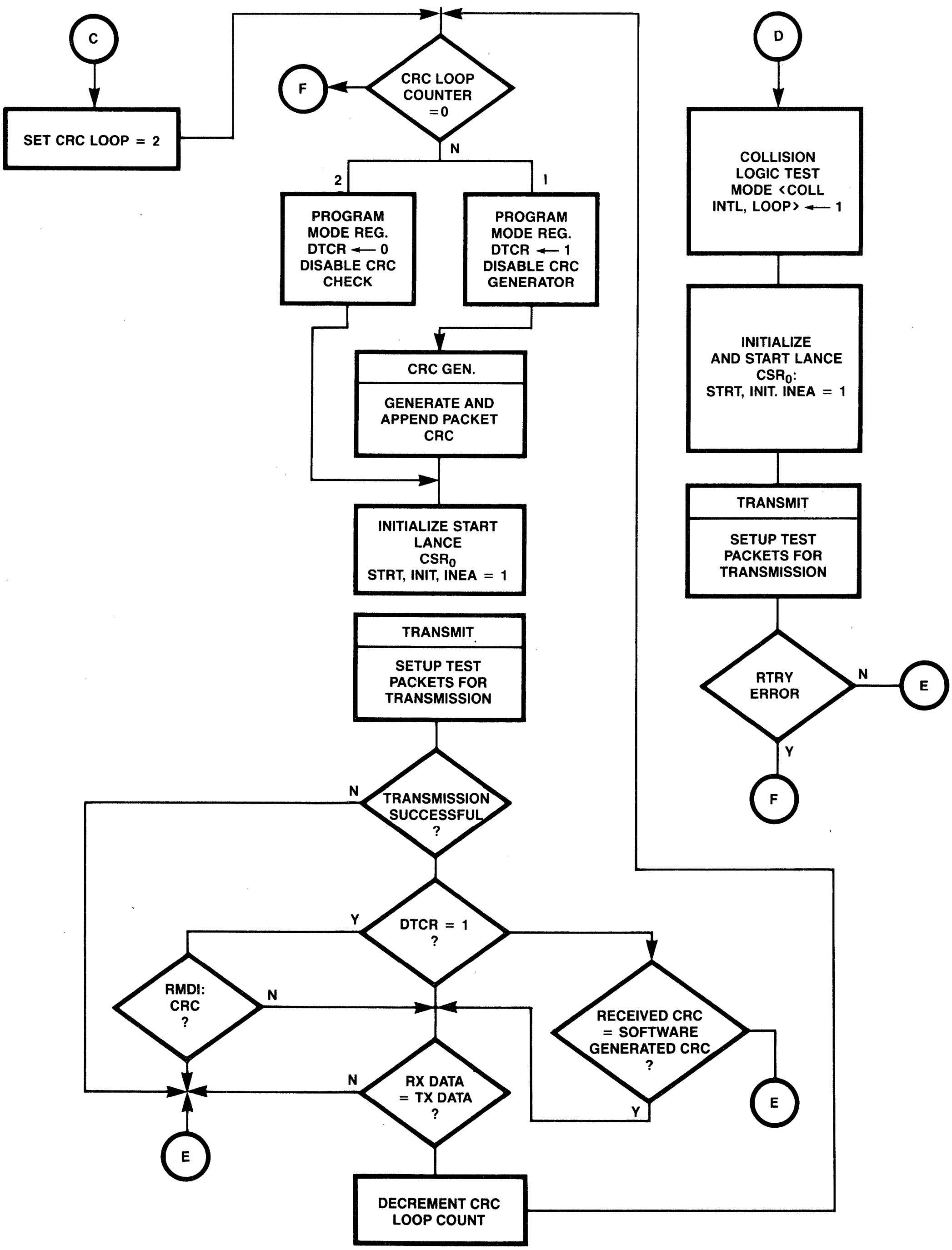
b) If at some point the LANCE transmits a non-matching packet, the packet will be sent but it will not be received. LCAR (lost carrier) will be set.

c) After transmitting a non-matching packet, the LANCE will no longer transmit in internal loopback mode, even if the matching address is put back. If the LANCE is then set-up to send a matching or a non-matching packet, the LANCE reads the descriptors from the Transmit Ring and relinquishes the descriptor to the CPU with LCAR bit set. It never DMAs data into the SILO. It does not actually transmit the packet out. Therefore, it does not receive the packet.



NOTE 1: Optional step if CSR stacking is used. Not required for real time Interrupt Processing

Figure 4-1 Initialization/Diagnostic Flowchart (Sheet 1 of 2)


C
SET CRC LOOP = 2
CRC LOOP COUNTER =0
F
N
2
1
PROGRAM MODE REG. DTCR ◄— 0 DISABLE CRC CHECK
PROGRAM MODE REG. DTCR ◄— 1 DISABLE CRC GENERATOR
CRC GEN.
GENERATE AND APPEND PACKET CRC
INITIALIZE START LANCE CSR0 STRT, INIT, INEA = 1
TRANSMIT
SETUP TEST PACKETS FOR TRANSMISSION
N
TRANSMISSION SUCCESSFUL ?
Y
DTCR = 1 ?
RMDI: CRC ?
N
RECEIVED CRC = SOFTWARE GENERATED CRC ?
Y
E
N
RX DATA = TX DATA ?
E
DECREMENT CRC LOOP COUNT
D
COLLISION LOGIC TEST MODE <COLL INTL, LOOP> ◄— 1
INITIALIZE AND START LANCE CSR0: STRT, INIT. INEA = 1
TRANSMIT
SETUP TEST PACKETS FOR TRANSMISSION
RTRY ERROR
N
E
Y
F

06363A-11

Figure 4-1 Initialization/Diagnostic Flowchart (Sheet 2 of 2)

In the internal loopback mode, when a packet is transmitted that is not addressed to the node itself, LCAR is set and indicates this condition. If the address detection logic of the LANCE is being tested for a non-matching test in internal loopback mode, the first non-matching transmit packet always gets transmitted. Only the packets following the transmission of the non-matching packet fall into this condition and never get transmitted. Re-initializing the LANCE is the only option when this problem occurs. This problem will not occur as long as the LANCE is transmitting packets to itself (matching addresses) in internal loopback mode. The external loopback can be used for testing the packets with non-matching addresses.

4.3.2 EXTERNAL LOOPBACK

External Loopback operates in much the same fashion as Internal Loopback, except that the test packet is actually transmitted out, starting from the LANCE to the SIA, to the transceiver, and finally to the coaxial cable. The transmitted data wraps around, in the same fashion, from the coaxial cable to the LANCE. External Loopback test will not interfere with any other node attached to the Ethernet cable, since the packet is normally addressed to the node itself or gets discarded as a runt packet (loopback packet size max = 32 bytes).

The multicast addressing is not operational in External Loopback Mode. The LANCE transmits the packet but it will not receive it. However, DTCR (Disable Transmit CRC) in the Mode Register can be set to "1" when multicast addressing in External Loopback Mode is desired. In this case, the user has to append the correct four byte CRC when transmitting a packet or there will be a CRC error when the packet is received. The multicast addressing can be tested in Internal Loopback Mode, or Receive Mode when another node is transmitting.

4.3.3 CRC GENERATOR/CHECKER LOGIC TEST

During the loopback test (internal or external), the CRC logic may be tested. Since the LANCE operates in half-duplex mode, the CRC generator or checker must be tested separately as follows. The Transmit CRC Generator is tested by programming DTCR=0 in the Mode Register. This enables the transmit CRC generation and disables the Receive CRC Checker. Assuming the test packet is 32 bytes, the LANCE will transmit 32 bytes and store 36 bytes of data in the receive buffer. The extra four bytes are the CRC generated by Transmit CRC Generator. The received CRC bytes are compared against the software generated CRC.

The Receive CRC Checker is tested by setting the DTCR bit in the Mode Register. The user then calculates the CRC for the test packet and appends it to the end of the test packet for transmission (test packet max size = 32 + 4 bytes CRC). The Receive CRC Checker checks the last four bytes of the packet for CRC error and updates the receive descriptors accordingly. The CRC logic may be tested in either External or Internal Loopback Mode.

4.3.4 COLLISION DETECTION/RETRY LOGIC TEST

The collision detection can be tested by forcing collision, setting the COLL, LOOP, and INTL bits of the Mode Register) to "1." The LANCE is then initialized and started (CSR_0: INIT,STRT,INEA=1). The collision bit, COLL, being set will cause the LANCE to detect a collision on each loopback attempt. The result is 16 total transmission attempts with a retry error reported in TMD_3.

The SIA/transceiver collision interface to the LANCE is checked by placing the LANCE in External Loopback. The LANCE will look for a collision detection within 2 us following the completion of a transmission. This is a transceiver self test feature which is known as SQE (Signal Quality Error) test or "Heartbeat." The CERR bit gets set if the Heartbeat is missing within the specified time. The failure can be caused by the external collision path (transceiver-to-SIA-to-LANCE) and/or the transceiver itself.

4.4 MAIN PROGRAM

The LANCE is configured for normal operation via the Initialization Block. LANCE is initilized once for normal operation. However, any configuration change requires re-initialization. Figure 4-2 shows a flow diagram of the main program for normal operation.

4.5 DESCRIPTOR AND BUFFER PITFALLS

4.5.1 TRANSMIT MESSAGE DESCRIPTORS, TMD1 AND TMD3

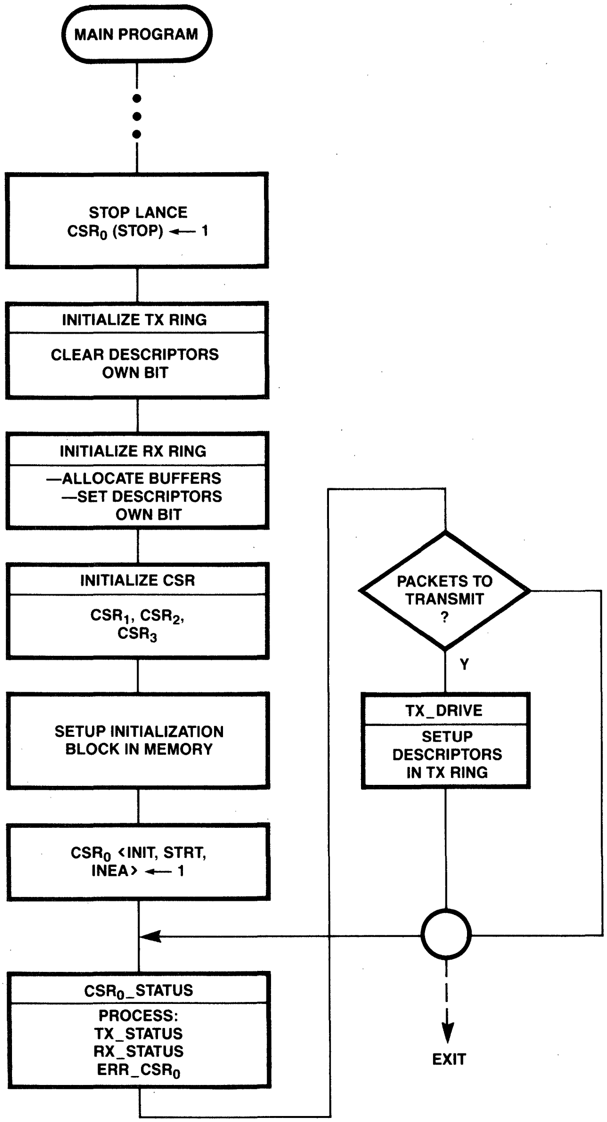
"One" flag in TMD_1 is not valid when "LCOL" is set in TMD_3.

4.5.2 TRANSMIT DESCRIPTOR RING FORMAT

When the LANCE is polling the Transmit Ring, it will skip over Transmit Ring entries having a bad format. For the first buffer in a packet, when the LANCE owns the buffer and STP (Start of Packet) is not set, it will simply turn the ownership back over to the host. The LANCE will then generate a TINT interrupt and go on to the next buffer. The LANCE continues to skip over transmit buffers (by clearing the OWN bit and setting TINT bit) until it finds a buffer with a good format (both STP and OWN bit set) to transmit. When the LANCE finds a transmit buffer owned by the host (OWN = 0), it means that there are no more buffers to transmit.

4.5.3 DATA CHAINING

During either a receive or a transmit with data chaining, the LANCE generates a done flag (RINT or TINT) and interrupt only at the end of the chain. It will not generate any interrupt flags whenever a buffer of the chain is filled or transmitted, unless there is an error. The ownership bits, however, are turned over to the host as each buffer is completed. This can cause a problem if the first transmit buffer is short as explained below.



06363A-12

Figure 4-2 Main Program for the LANCE in Normal Operation

4.5.4 TRANSMIT BUFFERS

When in Transmit Mode, the LANCE can transfer up to 32 bytes of data into the Silo before transmission is started if the cable is busy at the time. A collision could occur at any time during the transmission of the first 64 bytes (first slot time). During this transmission, 64 additional bytes could be read from the buffer. Therefore, the first transmit buffer must be at least 96 bytes long to ensure that the LANCE does not relinquish the first buffer before it has sent the first 64 bytes of the packet over the Ethernet cable. Including a small extra margin, 100 bytes was chosen as the minimum size required for the first buffer when data chaining to ensure proper handling of collisions.

At the beginning of a transmission, the LANCE internally stores the address and byte count of the first transmit ring entry, in case a retransmission is required. In data-chaining, after each buffer is read into the Silo, the buffer is returned to the host by resetting the "own" bit. As explained above, if the first transmit buffer is too short, a collision could occur after the first buffer was completed and the ownership bit turned over to the host.

Since a collision causes an automatic retry and since the LANCE has the address of the first buffer, the LANCE will attempt to retransmit the buffer that it had just given back to the host. This violates the mutual exclusion rule of the ring ownership by accessing a ring entry that it does not own. This can also cause incorrect and unexpected transmit ring status changes as follows: if the host is polling the ownership bit, it would see ownership of the first transmit buffer and start to service it. At the same time, the LANCE may be retransmitting it and even updating the status for that buffer.

When data-chaining is not used, the LANCE does not return the buffer to the host until the entire packet has been sent without collision or other error. Therefore, for buffers that are not data-chained, the minimum buffer size for the first buffer is the minimum packet size of 64 bytes required by Ethenet to ensure proper handling of collisions. However, the LANCE is capable of transmitting a packet as small as one byte.

To summarize, when the buffers are data chained in Transmit Mode, the minimum size for the first transmit buffer is 100 bytes. When the buffers are not data-chained, the minimum size for the transmit buffer is 64 bytes to conform to Ethernet standards. These minimums prevent the LANCE from violating the mutual exclusion principle of the ring ownership bits as a result of a collision. The mutual exclusion rule states that the LANCE must not access a ring entry it does not own. The slot time is the time it takes to transmit 64 bytes over Ethernet. A collision can occur at any time during the first slot time.

4.5.5 RECEIVE BUFFERS

The minimum size for the first receive buffer is specified as 64 bytes. If this spec is not adhered to, the LANCE could cause the violation of the mutual exclusion principle of the ring ownership bits.

This happens if the receive buffers are less than 64 bytes each and a packet that is larger than the first buffer but less than 64 bytes (a runt packet) arrives. The LANCE fills the first buffer, updates that buffer's status and turns over the ring ownership to the host. When the packet terminates as a runt, the LANCE discards it. The LANCE backs up its internal receive ring address pointer and byte count registers to their values at the beginning of the reception so that it can use this buffer for the next packet. However, this buffer has already been turned over to the host.

When the next packet arrives, the LANCE will reuse the buffer that it filled during the previous runt packet although it is now owned by the host. This violates the mutual exclusion rule of the ring ownership bits.

If the host had been polling the ownership bits, it could have already used the data in that receive buffer although it was not valid data. The host expects the next receive packet to start at the next receive buffer but the LANCE starts at the previous buffer. If the next receive packet is a legal packet, the LANCE will data chain at least two buffers because the first buffer is too small. The host then starts looking at the second buffer of the chain because it has already serviced the previous buffer when the ownership bit was turned over during the previous runt packet. Under these conditions, the host will not see the expected packet format (the destination address, source address, etc.) in the receive buffer.

4.6 INTERRUPTS

The interrupt pin is simply an OR of the interrupt causing conditions. The occurrence of another interrupt after the interrupt pin is asserted does not generate another transition on the interrupt pin. Interfaces that require a transition per interrupt may use the following software routine to force a transition on the interrupt pin.

READ CSR_0 AND STORE IN MEMORY OR A REGISTER

CLEAR THE INTERRUPT ENABLE (BIT 06) IN THE STORED CSR_0

WRITE CSR_0 WITH THE STORED VALUE (BIT 06 IS NOW CLEAR)

SET THE INTERRUPT ENABLE BIT IN CSR_0

At the time the interrupt is first serviced, this routine first reads CSR_0 to capture the register and store its value in memory or a register. Then the stored value is written to CSR_0 with Bit 06 clear, clearing the interrupt-causing conditions present at the time the register was read. They are cleared because they are all "write-one-to-clear" bits. This clears out the old interrupt flags and preserves any new ones set after the register was read. It causes the LANCE to deassert the interrupt pin because interrupt enable is now clear. The last step is to set interrupt enable again. This must be done with a Write Only instruction, such as MOVE, so that any new interrupt flags are not cleared by a Read-Modify-Write operation. At this time, if there were any new interrupt flags set after CSR_0 was first read, the interrupt pin will be reasserted generating a new transition for the new interrupt flags.

It is possible to have more than one interrupt-causing condition for each interrupt generated. For example, if an interrupt is received and the RINT bit is set, there could have been more than one receive buffer turned over to the host, because other receive buffers could have been filled before the host services the interrupt. This is also true for transmit interrupts. In fact, a receive buffer could have been turned over to the host after a transmit interrupt has been generated. This happens if a transmit interrupt was generated, if the host software read CSR_0, and if a receive buffer was filled before the host software finishes the transmit routine. Therefore, it is wise to check both the Receive and Transmit Rings for buffers owned by the host whenever an interrupt occurs.

LANCE asserts the interrupt signal ($\overline{INTR}$) every time it updates the internal register, CSR_0, provided the INEA bit is set. Interrupt remains asserted until one of the following conditions is met:

1. Source of interrupt is removed.
2. $\overline{RESET}$ is asserted.
3. Stop bit gets set in CSR_0.

There are four types of interrupts noted in CSR_0:

1. Initialization done.
2. Fatal and non-fatal errors.
3. Receive Descriptor update.
4. Transmit Descriptor update.

It is important and necessary to acknowledge the interrupts in real time so that the segment of the LANCE status being updated is not lost. This also enables the processing of transmit and receive buffers promptly.

There is only one interrupt condition, MISS error, which requires a response. When MISS error occurs, it must be cleared by the user, or no more packets will be received, even if some receive buffers have been relinquished to the LANCE. (The MISS error requirement applies only to the LANCE Rev. B. The LANCE Rev. C does not require the MISS bit to be cleared to receive the following packets.) However, there is enough time (67.2 us) between the MISS error bit getting set and the arrival of another packet. The 67.2 us is measured based on back-to-back short packets (64 bytes) addressed to this node, and the size of the SILO (48 bytes).

No other interrupt bits in CSR_0 will result in the loss of any packet. If interrupt is not serviced before the next interrupt occurs, then the new status bits get ORed with the old ones before CSR_0 is updated. It is possible for the user to lose the sequence of status being updated.

The discussion below describes one way of handling a long interrupt latency. Since some software overhead may be involved in responding to the interrupts, a buffering scheme (software FIFO) may be used to ensure that all the events are recognized and serviced in the order of occurrence.

A software FIFO as large as Transmit/Receive Ring size is allocated to store the CSR_0s in the order of interrupt occurrence. The interrupt service routine merely reads the CSR_0 and stores it for later processing by status processing routines thus making it as short as possible. (See Figure 4-3.) This approach will assure that all the events are captured in the order of occurrence.

4.7 TRANSMIT DRIVER CONSIDERATIONS

The host communicates with the LANCE, via the Transmit Ring, for packet transmission. There can be up to 128 descriptor entries, four words per entry, in the Transmit Ring.

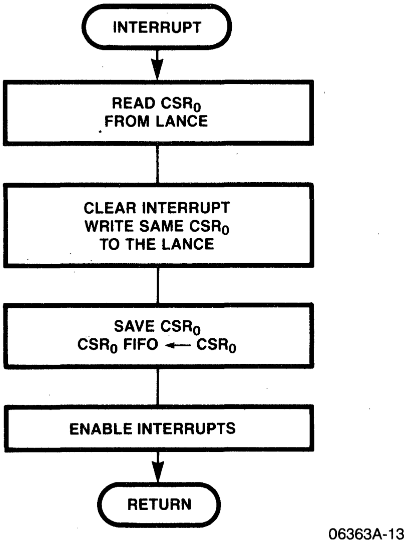

INTERRUPT
READ CSR0 FROM LANCE
CLEAR INTERRUPT WRITE SAME CSR0 TO THE LANCE
SAVE CSR0 CSR0 FIFO ← CSR0
ENABLE INTERRUPTS
RETURN
06363A-13

Figure 4-3 Interrupt Service Routine

The user should have two pointers associated with the Transmit Ring:

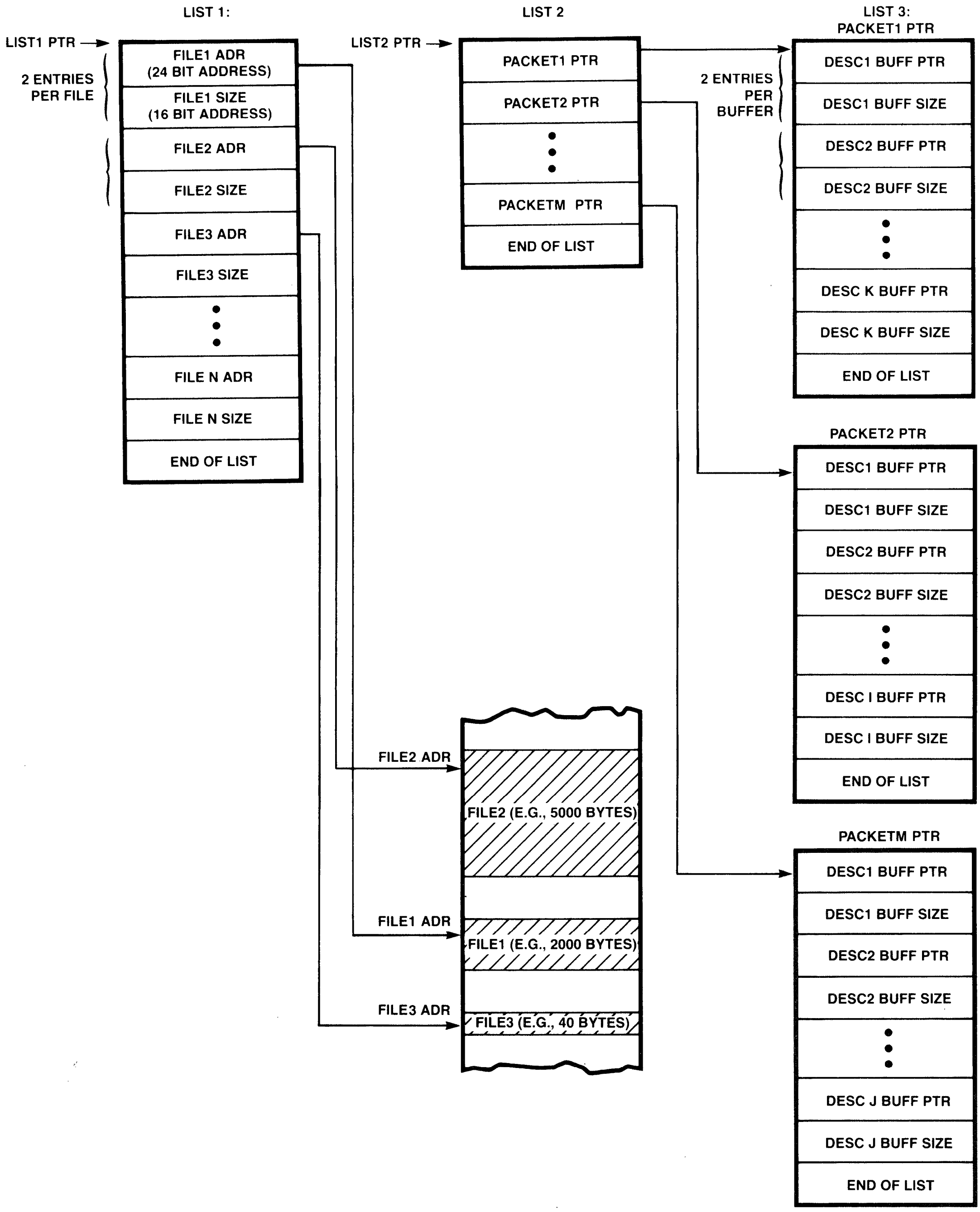
1. TXDESW — Transmit Descriptor Write Pointer
2. TXDESR — Transmit Descriptor Read Pointer

TXDESW points to the first descriptor of the packet queued for transmission by the host (OWN=1, STP=1). TXDESR points to the first descriptor of the packet just relinquished by the chip (OWN=0, TINT=1).

TXDESR and TXDESW originally point to the same location, to the beginning of the Transmit Ring. Once the packet transmission starts, the TXDESW pointer normally leads the TXDESR pointer. The user sets up the descriptors by flowing/wrapping around the ring, and LANCE chases the user sequentially to send the buffers out.

Transmit data buffers are usually scattered around in the memory. Before initiating a transmission, the user needs to invoke the lower level transmit driver routine to change the size of large and small buffers scheduled for transmission. One approach, as illustrated in Figure 4-4, is to shrink down the large buffers to 1500 bytes each and chain the small buffers together by taking advantage of LANCE's buffer management data chaining scheme. Three types of lists can be associated with transmit routines to transmit a large block of data (for example, 10K bytes).

List 1:
This list contains pointers to buffers (any location in memory), with different buffer sizes (no limitation in buffer sizes), to be transmitted as one block in the order defined sequentially in the list. LIST1PTR is the pointer to this list. Each entry on the list designates the buffer pointer and buffer size. End of list is recognized by an end of list flag (for example, FFFF...).



06363A-14

Figure 4-4 Transmit Block Buffer Packing

List 2:
One of the transmit routines uses List 1 to generate List 2. Each entry on List 2 is a pointer to a packet (List 3). List 2 is a translation of a block of data (List 1) with large and/or small buffers to one or more packets. End of list is recognized by the end of list flag (for example, FFFF...). LIST2PTR is a pointer as an input to the lower level transmit driver routine to transmit the whole block (all the packets in List 2).

List 3:
Each entry on this list designates the buffer pointer and the buffer size (<=1514 bytes; see Note 1) for each buffer that is chained together to be transmitted as one packet. At the end of the packet, a list flag is used to designate the end of the list. The lower level transmit driver routine transfers both the buffer pointers and buffer sizes directly to the transmit ring descriptors. The buffer pointed to by the first entry in list 3 must contain the header information (source address, destination address, and type (or length) field.

NOTE 1

> The user should be cautious about address and type field insertion for transmission of each packet. Since LANCE does not insert destination or source addresses or type field, the user must append the address and type field (14 bytes) to the beginning of each packet.

The transmit driver routine, TXDRIVE, shown in Figure 4-5 uses LIST2PTR as the pointer to the packet list to set-up the transmit descriptors for transmission. Transmit driver routine (Figure 4-5) and TX_Status routine (Figure 4-6) keep track of the two pointers, TXDESW and TXDESR by using a software FIFO. The transmit driver routine writes the pointer, TXDESW, to the FIFO and the TX_Status routine reads the pointer, TXDESR, from the FIFO. The FIFO size should be as large as the transmit ring size. This approach allows the transmission process of the packet to be performed in the correct order and minimizes the software complexity of managing the Transmit Ring.

As pointed out earlier, the TXDESW pointer normally leads the TXDESR. If these two pointers point to the same location (TXDESR = TXDESW), it is an indication of having either a small ring (much less than 128) or the LANCE transmitter has been turned off because of some particular error(s). Should the transmitter shut off, the user is notified via interrupt. CSR_0 Register, TXON Bit, may then be checked to see if the transmitter is on.

The LANCE normally polls the Transmit Ring descriptor once every 1.6 ms until it finds a packet (defined/addressed in the descriptor) to be sent out. However, the user may demand faster polling time (every 1-2 us) by continuously setting the TDMD bit in CSR_0, after it gets cleared by the LANCE. The user should normally set the TDMD bit every time it relinquishes (sets the OWN bit) the first descriptor of a packet to the LANCE.

NOTE

> As shown in the transmit routine flowchart, the user should not relinquish the descriptors (buffers) until all the descriptors for the packet have been set up. (ENP bit is set in the last descriptor.) After the descriptors are set up, the user should clear the OWN bit of each descriptor in the reverse order (last descriptor first, first descriptor last). This technique will prevent transmit buffer error (TMD_3, BUFF error).

As shown in Figure 4-6, there are two variables, TXDESR and TINTCNT, used for processing the packets that LANCE has transmitted. TINTCNT corresponds to the number of packets that have been transmitted (times that the TINT bit has been set in CSR_0) but not processed by the CPU to update their status. TXDESR is the pointer to the first descriptor of the packet relinquished by the LANCE and not processed by the CPU. This mechanism is used to assure servicing the packets in the order of transmission and to prevent TINT being set twice without the CPU awareness that two packets have been transmitted by the LANCE.

When checking for errors, the user needs to check both the ERR and BUFF bits, since the ERR does not include the BUFF error. When processing errors, BUFF or UFLO errors should be treated as fatal errors since these errors are due to a malfunction in hardware and/or software designed around the LANCE and it is under user's control. LCOL error should also be treated as a fatal error. In a maximum Ethernet configuration, a collision might occur within the first 64 bytes of the packet but not beyond. LCOL error can also be due to a fault (open or short) in the coaxial cable.

LCAR error can be due to a malfunction in the receive section at the transceiver or fault between transceiver and SIA (Receive ± pair) or SIA and LANCE (RENA input to the LANCE). The LANCE requires RENA to become and remain active before TENA is deasserted; otherwise an LCAR error will be set. If RENA drops out while transmitting (TENA High), an LCAR error will get set and the LANCE will continue to transmit the packet. A fault (short) in the coaxial cable can also cause an LCAR error.

The RTRY error is set when the LANCE attempts 16 times to transmit the packet and does not succeed. This error is less likely to occur. However, this error can occur in a very heavily loaded network, a failure of a node in the network, or an open in the coaxial cable.

As shown in the flowchart in Figure 4-6, after the packet has been transmitted, in particular when the buffers are data chained, the user should check the status of all descriptors for errors (ERR bit). This is because RTRY error is updated in the first descriptor, and LCOL and UFLO can occur anywhere from the first descriptor to the last descriptor of the packet. When late collision occurs (LCOL=1), the LANCE continues to transmit the packet.

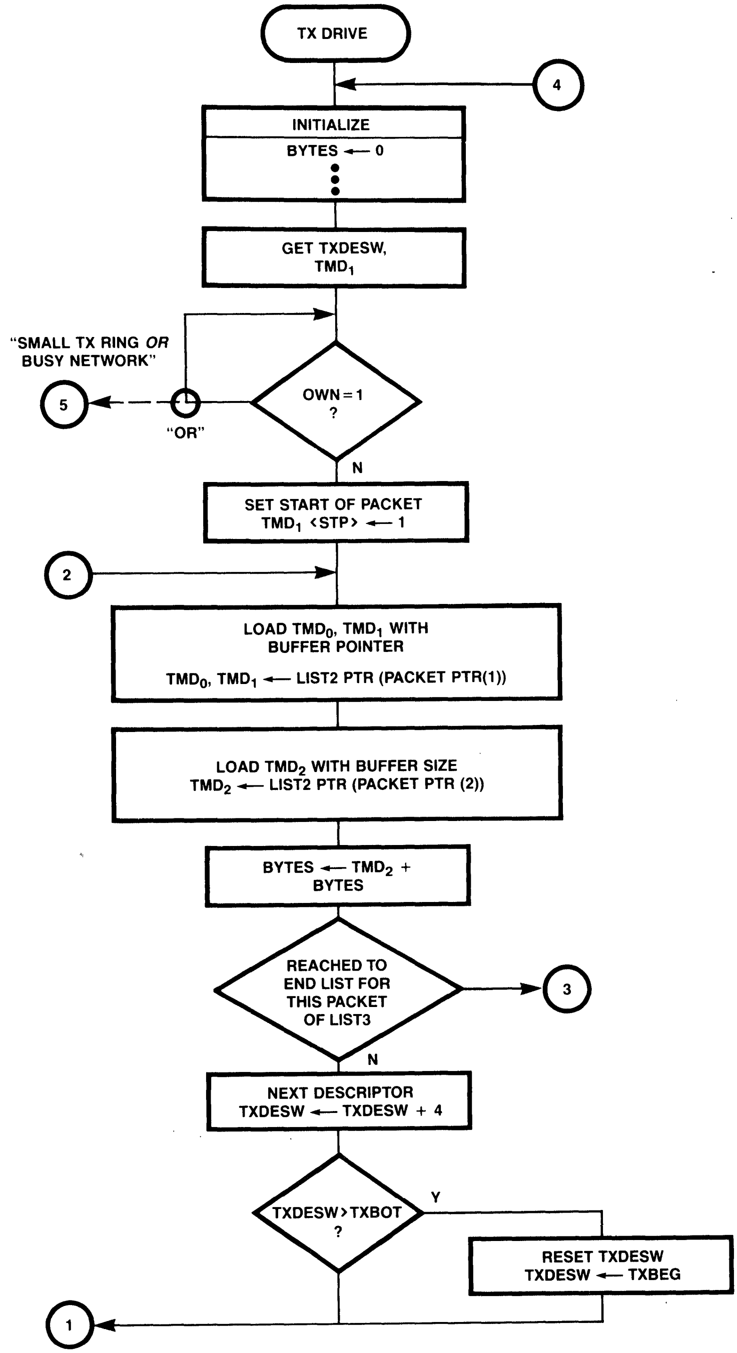
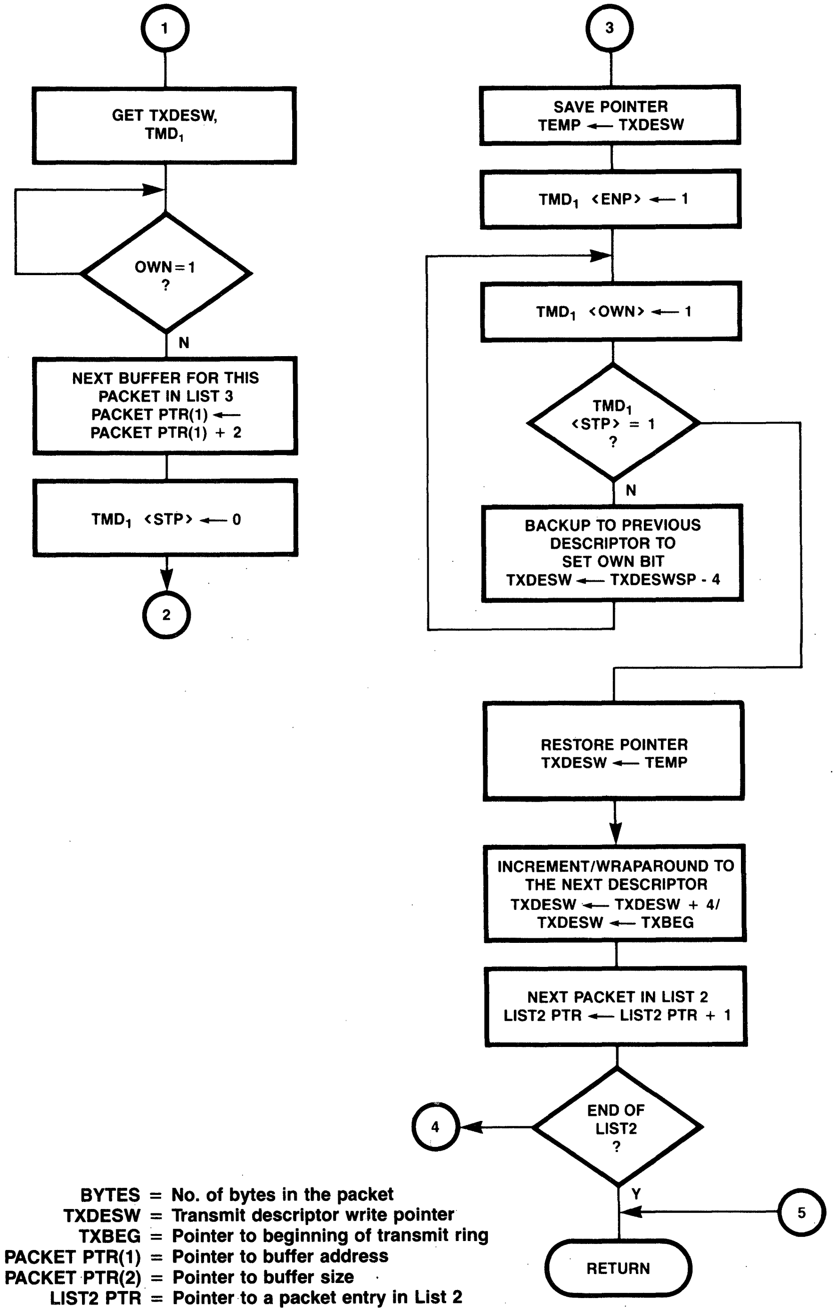
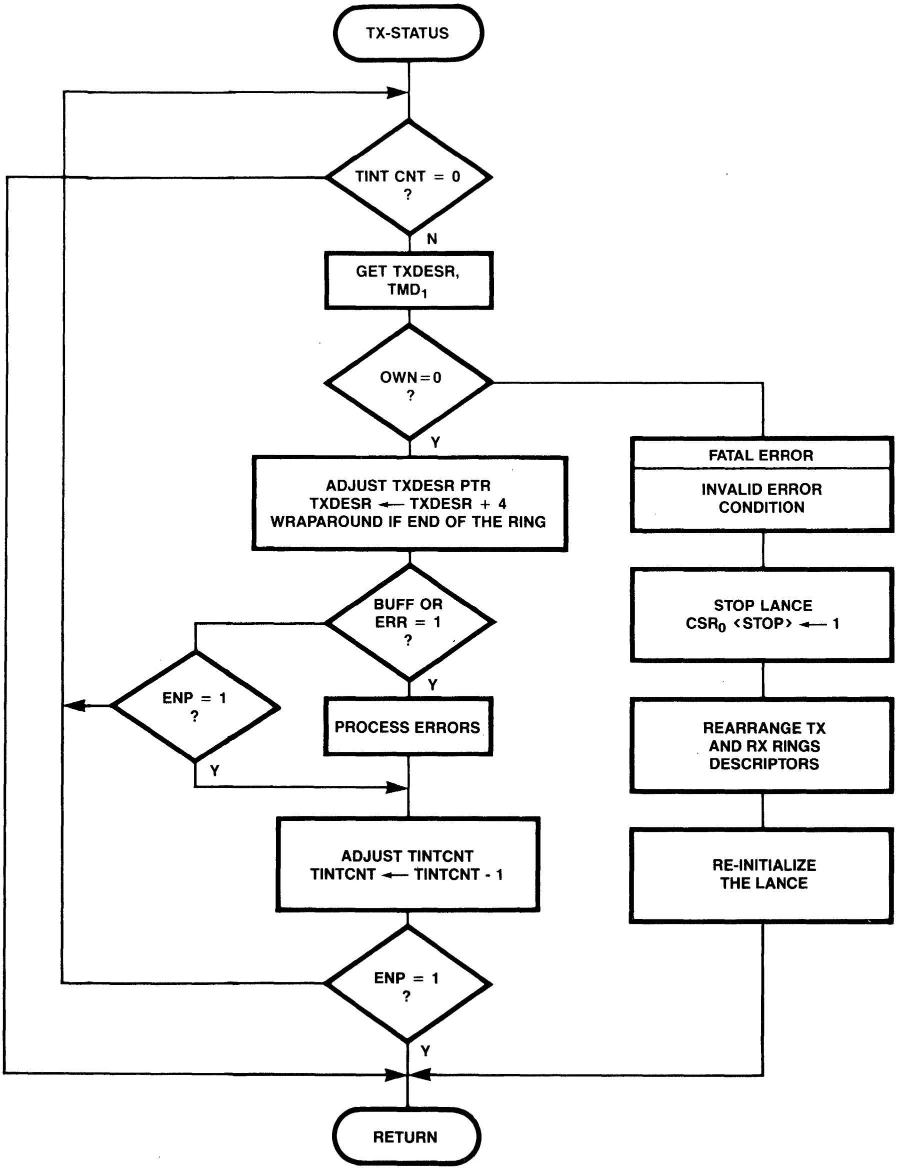

TX DRIVE
4
INITIALIZE
BYTES ← 0
GET TXDESW, TMD1
"SMALL TX RING OR BUSY NETWORK"
5
"OR"
OWN=1 ?
N
SET START OF PACKET TMD1 <STP> ← 1
2
LOAD TMD0, TMD1 WITH BUFFER POINTER
TMD0, TMD1 ← LIST2 PTR (PACKET PTR(1))
LOAD TMD2 WITH BUFFER SIZE
TMD2 ← LIST2 PTR (PACKET PTR (2))
BYTES ← TMD2 + BYTES
REACHED TO END LIST FOR THIS PACKET OF LIST3
3
N
NEXT DESCRIPTOR TXDESW ← TXDESW + 4
TXDESW > TXBOT ?
Y
RESET TXDESW TXDESW ← TXBEG
1

Figure 4-5 Transmit Routine (TXDRIVE) to Setup Packets for Transmission (Sheet 1 of 2)


1
GET TXDESW, TMD1
OWN=1 ?
N
NEXT BUFFER FOR THIS PACKET IN LIST 3
PACKET PTR(1) ←— PACKET PTR(1) + 2
TMD1 <STP> ←— 0
2
3
SAVE POINTER
TEMP ←— TXDESW
TMD1 <ENP> ←— 1
TMD1 <OWN> ←— 1
TMD1 <STP> = 1 ?
N
BACKUP TO PREVIOUS DESCRIPTOR TO SET OWN BIT
TXDESW ←— TXDESWSP - 4
RESTORE POINTER
TXDESW ←— TEMP
INCREMENT/WRAPAROUND TO THE NEXT DESCRIPTOR
TXDESW ←— TXDESW + 4/
TXDESW ←— TXBEG
NEXT PACKET IN LIST 2
LIST2 PTR ←— LIST2 PTR + 1
END OF LIST2 ?
4
Y
5
RETURN
BYTES = No. of bytes in the packet
TXDESW = Transmit descriptor write pointer
TXBEG = Pointer to beginning of transmit ring
PACKET PTR(1) = Pointer to buffer address
PACKET PTR(2) = Pointer to buffer size
LIST2 PTR = Pointer to a packet entry in List 2

06363A-15

Figure 4-5 Transmit Routine (TXDRIVE) to Setup Packets for Transmission (Sheet 2 of 2)



TINTCNT = No. of TINT interrupts
TXDESR = Pointer to the first descriptor of the packet relinquished by the LANCE and not processed by the CPU
TXBOT = Pointer to the bottom of the ring
TXBEG = Pointer to the beginning of the ring.

06363A-16

Figure 4-6 Transmit Packet Status Processing (TX-STATUS)

4.8 RECEIVE DRIVER CONSIDERATIONS

The host communicates with the LANCE, via the Receive Ring, for packet reception. There can be up to 128 descriptor entries, four words per entry, in the Receive Ring.

The user needs to allocate some buffers through the descriptor entries in the Receive Ring for the arrival packets, if the receiver is enabled (turned on via Initialization Block). Multiple buffers with different sizes scattered around in the memory may be allocated to the LANCE for reception. Receive driver routines and buffer management tasks can be relatively complex when the user needs to keep up with incoming back-to-back packets.

The user has 3 major tasks to handle at all times:

1. Respond fast enough to LANCE's DMA requests
2. Always have enough available receive buffers for arriving packets
3. Provide fast interrupt response time and short interrupt service routine to capture/distinguish the events in the order of occurrence for each packet. However, the packets will not be lost if interrupt processing is not in real time; CSR_0 update is an OR operation of the new CSR_0 and the previous CSR_0.

The user must keep track of the incoming receive packets, process the packets (check status and empty buffers), and allocate enough available receive buffers to store all incoming packets. A global pointer associated with the Receive Ring is needed to process the received packets sequentially in the order of arrival. This pointer, Received Descriptor Pointer (RXDPTR), is used in RX-Status routine (Figure 4-7) to locate the buffers for the received packet, check the integrity of the packet, and update the value of RXDPTR (pointer to the first descriptor of the next packet).

The node processor and the host CPU software may use a semaphore approach to communicate with each other. This is accomplished by allocating a software FIFO to be a Write Only by the node processor, and a Read Only by the host CPU. This FIFO has multiple entries each of which contains two locations: a pointer to the first descriptor for a packet (RXDPTR) and the number of descriptors (buffers) for this packet (RXDPSNO). The host CPU is interrupted by the node processor every time an entry is added to the list (FIFO). The host CPU, in return, reads the entry from FIFO to empty the buffer(s) and relinquish the buffers to the LANCE using the RXDPTR and RXDESNO. This technique speeds up the received packet processing time and diminishes the probability of missing packets as a result of no receive buffers being available.

The data chaining scheme in the LANCE makes efficient use of the memory during the packet reception. It is practical to assign small buffers (for example, 256 bytes) for receive buffers allocated to the LANCE. With small buffers, there is less unused space in each buffer when storing packets that are short. For packets larger than the buffer, data-chaining is used. Storing small packets in large buffers results in large unused and unusable areas of memory and results in the data being in fragmented areas scattered around in the memory.

RINTCNT variable shown in Figure 4-7 contains the number of times that the RINT bit has been set in CSR_0 without being processed by the user. This value corresponds to the number of packets which have been received with or without the error and are awaiting processing. Figure 4-7 uses RINTCNT and RXDPTR to process back-to-back packets in the order of arrivals.

As it implies in Figure 4-7, the user should start checking the RMD_1 of each descriptor until it finds either ENP or ERR bit set. IF ERR bit is set and ENP bit is not, it indicates that an error (such as BUFF or OVFL) occurred before the packet ended. If ENP bit is set, it indicates that the packet has ended (cable has gone idle) and the current descriptor is the last descriptor for this packet. When ENP and ERR are both set, it is an indication that there is a CRC or both CRC and framing error.

It is important to note that the LANCE interrupts the user when it updates the last descriptor (clears the OWN bit) for current received packet. It does not interrupt (does not set RINT) for each descriptor update when the packet consists of multiple descriptors.

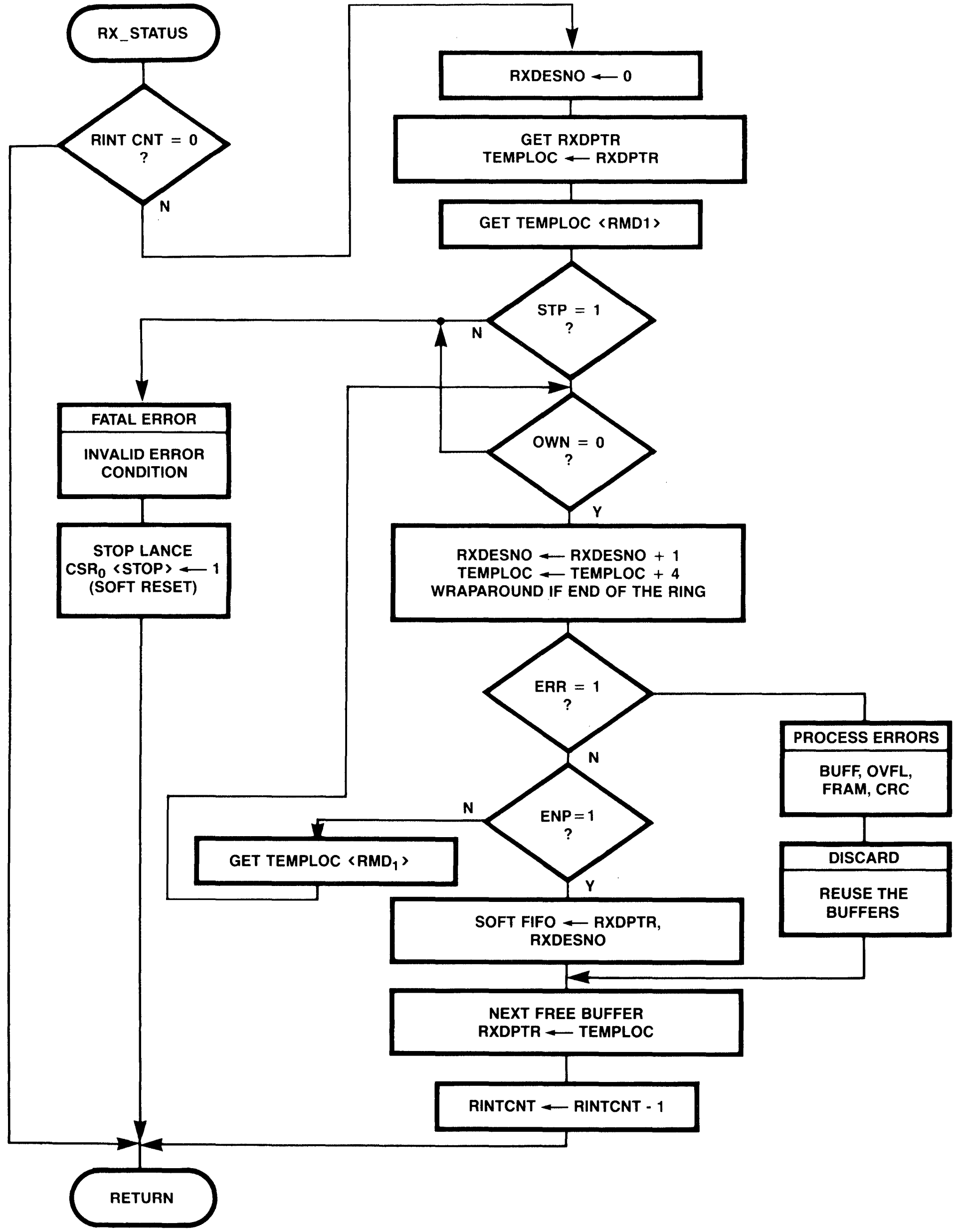
When the LANCE interrupts the CPU due to RINT, packet reception, the user should clear the source of interrupt (here RINT bit) so that it can distinguish the arrival of the next packet from the current one when RINT bit is set again.

The minimum time (worst case) for the RINT bit to be set again is as follows: At the end of a packet reception from the cable, the Silo has 48 bytes for the LANCE to transfer into a buffer before issuing a RINT. This transfer requires three DMA sequences of 4.8 us each plus two bus dwell times of .7 us. each (bus request response time, $\overline{HOLD}$ to $\overline{HLDA}$, is ignored) for a total of 15.8 us. However, the transfer does not start until after the interpacket gap time of 9.6 us followed by 6.4 us of preamble time. The transfer starts just as the data from the next packet starts to enter the Silo. In this case, the RINT bit is set 15.8 us after the next packet starts to fill the Silo. If the worst case happens and the Silo overflows, another RINT interrupt is issued for the error condition. The time required for the second RINT to occur is 22.6 us ([48 × 0.8] - 15.8). This is the interrupt latency of the LANCE for RINT bit to get set again.

This worst case interrupt latency only applies if the LANCE is not assured of prompt response to DMA bus requests. If the LANCE DMA bus latency is eight us, and the LANCE requests the bus as soon as eight words are in the Silo, there will be a maximum of thirteen words in the Silo when the end of the packet arrives. The LANCE takes 4.8 us to transfer the first eight words and 6.5 us (5 × (.6 + .7)) to transfer the remaining five words, one word at a time for a total of 11.3 us before the first RINT is set.

The packet intergap time is 9.6 us and the preamble is 6.4 us for a total of 16 us before data from the next packet starts to fill the empty Silo. If the LANCE now fails to acquire the DMA bus, the Silo will overflow causing a second RINT 43.1 us after the first RINT (16 - 11.3 + 48 × .8). This is the interrupt latency under these more reasonable conditions.

The errors associated with packet reception are BUFF, OVFL, CRC, and FRAM errors. The packets are normally discarded when any of these errors occur. Note that there may be an OVFL error when there is a BUFF error.



RINTCNT	=	No. of RINT interrupts
RXDPTR	=	Pointer to the first descriptor of the packet received by the LANCE and not processed by the CPU
RXDESNO	=	No. of descriptors for the received packet
Soft FIFO	=	A software FIFO shared by the host and the node processor. It contains two entries per packet, RXDPTR, RXDESNO has write only by the node processor and read only by the host CPU.

Figure 4-7 Receive Packet Status Processing

06363A-17

4.9 STATUS MONITORING AND ERROR PROCESSING

CSR_0_Status routine shown in Figure 4-8 monitors a software FIFO used by the interrupt handler to save the contents of CSR_0 after each interrupt (Figure 4-3). Each CSR_0 record is checked for IDON, RINT, TINT, and ERR bits. If the ERR bit is set, it calls another routine, ERR_CSR_0 (Figure 4-9), to process the fatal and non-fatal errors.

The IDON bit, when set, indicates that the initialization of LANCE has been done.

The counter variable, RINTCNT is incremented whenever RINT is set. RINTCNT is monitored by the routine, RX-Status (Figure 4-7). RX-Status uses RINTCNT to determine if any receive packets need to be processed. The value of RINTCNT corresponds to the number of packets which have been received by the LANCE but have not been processed by the CPU. Similarly, TX-Status routine uses TINTCNT to be able to process the results of the transmitted packets in the order of transmission.

TINTCNT corresponds to the number of packets to be processed by the CPU which have been transmitted by the LANCE. The counter variable, TINTCNT is incremented whenever the TINT bit is set. TINTCNT is monitored by the routine, TX-Status (Figure 4-6).

ERR_CSR_0 routine shown in Figure 4-9 illustrates the typical actions which users would normally take when processing the errors. The errors, MERR and BABL should be recognized as fatal errors. CERR and MISS are considered non-fatal errors.

MERR error indicates that the LANCE did not receive any acknowledgement from the system ($\overline{\text{READY}}$ High) 25.6 us after the start of the memory cycle. The problem can be the result of the LANCE accessing an invalid address (wrong buffer pointers might have been given to the LANCE in the descriptors) or it may be a memory interface problem between the LANCE, CPU, and memory.

BABL error occurs after the LANCE starts loading the 1519th byte to the SILO (maximum Ethernet packet is 1518 bytes). The LANCE continues to send the remainder of the packet following 1518 bytes. When this error occurs, the user should stop the transmission by setting the Stop bit in CSR_0. It is the user's responsibility not to have the packet more than 1514 bytes (1514 if CRC is generated by the LANCE, DTCR=0; 1518 if CRC is to be generated by the user, DTCR=1). It is worthwhile to mention that the LANCE can send up to 4K bytes (BCNT=12 bits in TMD_2) for each packet where it sets the BABL error following 1518 bytes.

CERR (collision, heartbeat, or SQE Test error) is a transceiver test feature. During this test, CERR is set if the LANCE does not see the collision within 2.0 us after a chip initiated transmission. Most transceivers (IEEE 802.3/Ethernet compatible) have the SQE Test feature built in. Within some delay (less than 2.0 us) following a transmission, they send a 10.0 MHz signal on collision pair. In the LANCE/SIA application, the SIA translates the 10.0 MHz differential signal to a TTL signal on CLSN line to the LANCE. CERR error does not cause an interrupt. The user can normally check CERR when the TINT bit gets set following transmission of the next packet. When the LANCE/SIA is interfaced with a transceiver which does not have the SQE Test function, the user simply ignores the existence of the CERR bit. Typical application where SQE Test may not be desirable is in repeater designs.

4.9.1 MISS ERROR

In the LANCE Rev. B, when a MISS error is set in CSR_0, it must be cleared immediately, or the next packets addressed to this node will not be received, even if some receive buffers have been relinquished to the LANCE. The LANCE Rev. C does not require the MISS bit to be cleared to receive the following packets.

4.9.2 RECEIVER and/or TRANSMITTER TURN OFF DUE to ERROR

There are three types of error which cause the transmitter to be turned off (CSR_0, TXON=0):

(1) Memory error (MERR)
(2) Underflow (UFLO)
(3) Buffer error (BUFF)

The only type of error which causes the receiver to be turned off (CSR_0, RXON=0) is a memory error (MERR). The chip has to be restarted in order to turn the transceiver and/or receiver back on again. The easiest way to restart the LANCE is by setting the Stop bit and then setting the Start bit in CSR_0. This turns the receiver and transmitter on again because DRX and DTX are still cleared in the Mode Register. However, setting the Stop bit may cause some confusion for the user if the LANCE is in the middle of a transmission or reception or if the buffers are data chained.

It is recommended that the LANCE be re-initialized, rather than setting the Start bit when it stops. Care must be applied when stopping the LANCE because setting the STOP bit clears CSR_3. CSR_3 must be reprogrammed if it contained a non-zero value before the LANCE was stopped. The LANCE can also be re-initialized by setting the INIT bit, but this is not necessary unless a change of operating parameters is desired.

4.10 PROGRAMMER ORGANIZATION OF THE LANCE REGISTER INFORMATION

The 8086 System register organization is different than the organization of 68000 registers. Therefore the programmer needs to know the organization of the system to be used with the LANCE so that the Initialization Block information and the Descriptor Ring entries are stored correctly.

In a 16-bit word in the 68000 system, byte 0 consists of bits 8 thru 15 (the most significant bits) and byte 1 consists of bits 0 thru 7. The word, AD 80, in 68000 assembly code is stored in memory as AD 80. This is the correct order of bytes for the LANCE control registers.

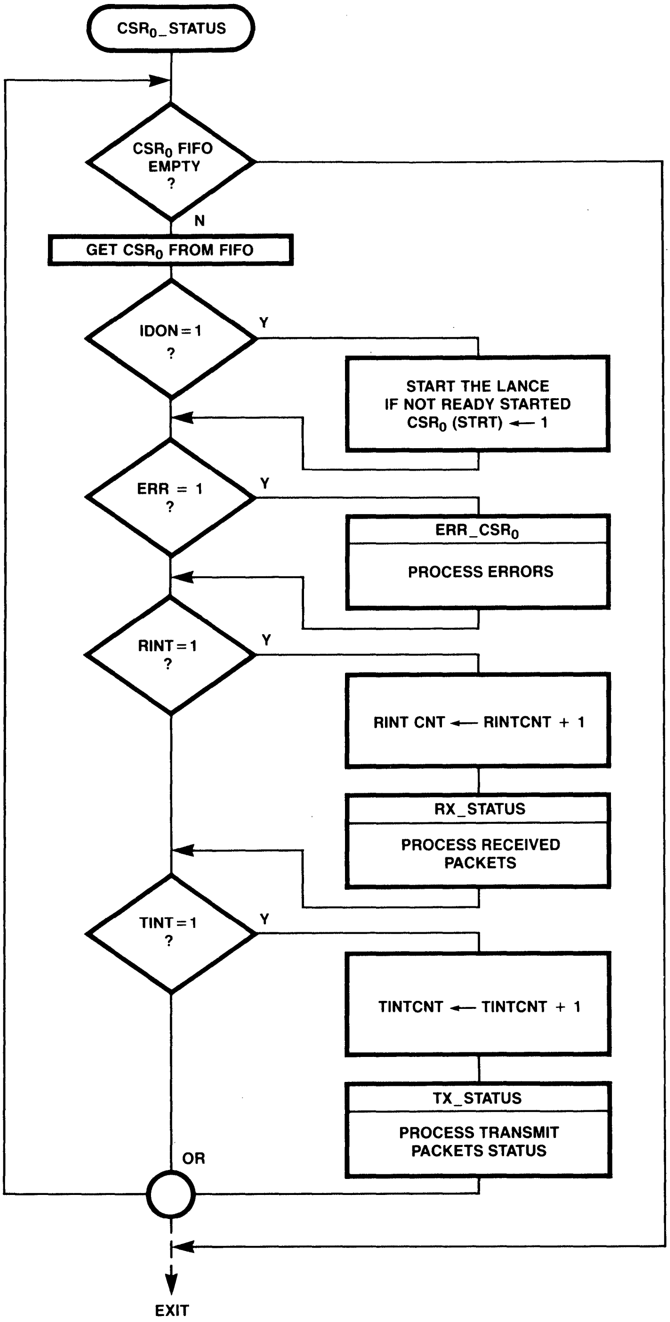
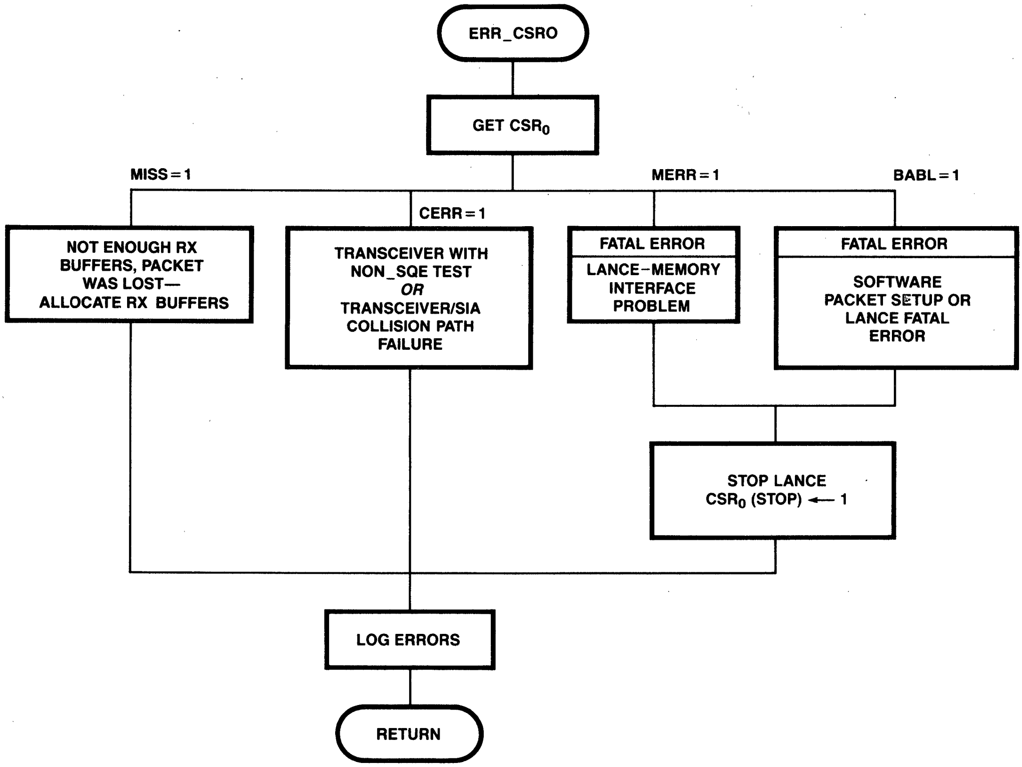


Figure 4-8 CSR$_0$ Status Processing


ERR_CSRO
GET CSR_0
MISS=1
CERR=1
MERR=1
BABL=1
NOT ENOUGH RX BUFFERS, PACKET WAS LOST— ALLOCATE RX BUFFERS
TRANSCEIVER WITH NON_SQE TEST OR TRANSCEIVER/SIA COLLISION PATH FAILURE
FATAL ERROR
LANCE-MEMORY INTERFACE PROBLEM
FATAL ERROR
SOFTWARE PACKET SETUP OR LANCE FATAL ERROR
STOP LANCE
CSR_0 (STOP) ← 1
LOG ERRORS
RETURN

06363A-19

Figure 4-9 CSR_0 Error Bit Processor, ERR-CSR0

However, data must be serialized for transmission and serialized data is handled as bytes rather than words. Serial byte data in ascending order is stored in memory as byte 1 (bytes 0–7) followed by byte 0 (bits 8–15). In serializing the data for transmission, the byte in the low-order bit positions is sent first. Therefore, the 68000 data being transmitted or received must be byte-swapped. The LANCE has a byte-swapping mechanism to perform this function. It is a control bit called BSWP in the Control and Status Register 3. When it is set to 1, the LANCE swaps the high and low bytes on DMA data transfers between the Silo and memory.

Figures 4-10, 4-11, and 4-12 show the Initialization Block and Descriptor Ring organization of control information required in a 68000 program for proper functioning in the LANCE. Figure 4-16 shows the relationship between control data in a 68000 register, in memory, and in a LANCE register. Figure 4-17 shows the relationship of serialized data and memory with and without byte swapping.

The 8086 stores sixteen-bit words with the low order byte (byte 0, bits 0–7) on an even byte boundary and byte 1 (the most significant eight bits) on the next higher byte address. For example, a data word stored in an 8086 register or in a program as AD 80, is stored in memory as 80 AD.

The LANCE stores word data in its control registers in the same order as it is stored in memory. It is the programmer's responsibility to store control information in the order needed by the LANCE. Figures 4-13, 4-14, and 4-15 give an 8086 programmer's view of the LANCE Initialization Block and Descriptor Ring information. Figure 4-16 shows the relationship between control data in an 8086 register, in memory, and in a LANCE register.

Because of the order that the 8086 stores bytes in memory during word transfers, no byte swapping of transmitted or received data is required when an 8086 system is used with the LANCE. The BSWP control bit is set to zero.

Word	Byte 0								Byte 1								
Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bits
IADR + 0	PROM	Reserved								INTL	DRTY	COLL	DTCR	LOOP	DTX	DRX	Mode Word
IADR + 2	PADR (bits 15:8)								PADR (bits 7:0)								Physical Address Register
IADR + 4	PADR (bits 31:24)								PADR (bits 23:16)								
IADR + 6	PADR (bits 47:40)								PADR (bits 39:32)								
IADR + 8	LADRF (bits 15:8)								LADRF (bits 7:0)								Logical Address Filter
IADR + 10	LADRF (bits 31:24)								LADRF (bits 23:16)								
IADR + 12	LADRF (bits 47:40)								LADRF (bits 39:32)								
IADR + 14	LADRF (bits 63:56)								LADRF (bits 55:48)								
IADR + 16	RDRA (bits 15:8)								RDRA (bits 7:0)								Receive Descriptor Ring Pointer
IADR + 18	RLEN (31:29)			Reserved					RDRA (bits 23:16)								
IADR + 20	TDRA (bits 15:8)								TDRA (bits 7:0)								Transmit Descriptor Ring Pointer
IADR + 22	TLEN (31:29)			Reserved					TDRA (bits 23:16)								

06363A-20

Figure 4-10. Initialization Block Organization in 68000 Assembly Code

Word	Byte 0								Byte 1								
Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bits
RDRA + 0	RBADR (bits 15:8)								RBADR (bits 7:0)								
RDRA + 2	OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	RBADR (bits 23:16)								
RDRA + 4	1	1	1	1	BCNT (bits 11:8)				BCNT (bits 7:0)								
RDRA + 6	Reserved				MCNT (bits 11:8)				MCNT (bits 7:0)								

06363A-21

Figure 4-11. Receive Descriptor Ring Entry Organization in 68000 Assembly Code

Word	Byte 0								Byte 1								
Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bits
TDRA + 0	TBADR (bits 15:8)								TBADR (bits 7:0)								
TDRA + 2	OWN	ERR	Res	MORE	ONE	DEF	STP	ENP	TBADR (bits 23:16)								
TDRA + 4	1	1	1	1	BCNT (bits 11:8)				BCNT (bits 7:0)								
TDRA + 6	BUFF	UFLO	RES	LCOL	LCAR	RTRY	TDR	(9:8)	TDR (bits 7:0)								

06363A-22

Figure 4-12. Transmit Descriptor Ring Entry Organization in 68000 Assembly Code

Word Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bits
	Byte 1								Byte 0								
IADR + 0		INTL	DRTY	COLL	DTCR	LOOP	DTX	DRX	PROM	Reserved							Mode Word
IADR + 2	PADR (bits 7:0)								PADR (bits 15:8)								Physical Address Register
IADR + 4	PADR (bits 23:16)								PADR (bits 31:24)								
IADR + 6	PADR (bits 39:32)								PADR (bits 47:40)								
IADR + 8	LADRF (bits 7:0)								LADRF (bits 15:8)								Logical Address Filter
IADR + 10	LADRF (bits 23:16)								LADRF (bits 31:24)								
IADR + 12	LADRF (bits 39:32)								LADRF (bits 47:40)								
IADR + 14	LADRF (bits 55:48)								LADRF (bits 63:56)								
IADR + 16	RDRA (bits 7:0)								RDRA (bits 15:8)								Receive Descriptor Ring Pointer
IADR + 18	RDRA (bits 23:16)								RLEN (31:29)				Reserved				
IADR + 20	TDRA (bits 7:0)								TDRA (bits 15:8)								Transmit Descriptor Ring Pointer
IADR + 22	TDRA (bits 23:16)								TLEN (31:29)				Reserved				

06363A-23

Figure 4-13. Initialization Block Organization in 8086 Assembly Code

Word Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bits
	Byte 1								Byte 0								
RDRA + 0	RBADR (bits 7:0)								RBADR (bits 15:8)								
RDRA + 2	RBADR (bits 23:16)								OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	
RDRA + 4	BCNT (bits 7:0)								1	1	1	1	BCNT (bits 11:8)				
RDRA + 6	MCNT (bits 7:0)								Reserved				MCNT (bits 11:8)				

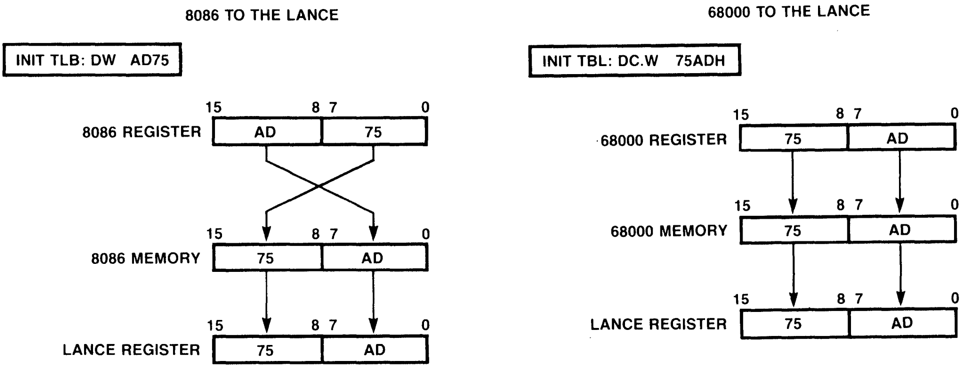
06363A-24

Figure 4-14. Receive Descriptor Ring Entry Organization in 8086 Assembly Code

Word Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bits
	Byte 1								Byte 0								
TDRA + 0	TBADR (bits 7:0)								TBADR (bits 15:8)								
TDRA + 2	TBADR (bits 23:16)								OWN	ERR	RES	MORE	ONE	DEF	STP	ENP	
TDRA + 4	BCNT (bits 7:0)								1	1	1	1	BCNT (bits 11:8)				
TDRA + 6	TDR (bits 7:0)								BUFF	UFLO	RES	LCOL	LCAR	RTRY	TDR (9:8)		

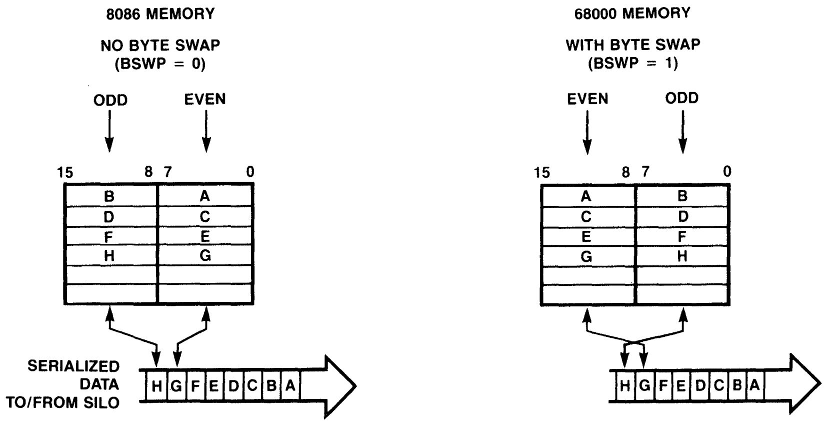
06363A-25

Figure 4-15. Transmit Descriptor Ring Entry Organization in 8086 Assembly Code



06363A-26

Figure 4-16 Control Data Transfers Between the LANCE and CPU



06363A-27

Figure 4-17 Serial Data Storage in Memory

APPENDIX A
GLOSSARY

ACON	ALE Control bit in CSR$_3$
ADR	Register Data/Address Port Select
ALE	Address Latch Enable output pin on the LANCE
AS	Address Strobe output for use by slave device
BABL	Error flag in CSR$_0$ (more than 1518 bytes in one packet)
BCNT	Buffer Byte Count entries in RMD$_2$ and TMD$_2$
BCON	Byte Control bit in CSR$_3$ for Byte Mask and Hold I/O pins
BMo/BYTE	Input/output select
BSWP	Byte Swap bit in CSR$_3$
BUFF	Buffer Error flag in RMD$_1$ and TMD$_3$
BUSAKO	Bus request output in daisy chain configuration of the LANCE. If the LANCE is requesting the bus when it receives $\overline{\text{HLDA}}$, $\overline{\text{BUSAKO}}$ remains high. Otherwise, $\overline{\text{BUSAKO}}$ is set low when $\overline{\text{HLDA}}$ arrives.
BUSRQ	Bus Request output pin on the LANCE used in daisy chain configuration (CSR$_3$ BCON = 1).
Bus Bandwidth	Percentage of time a device holds and controls the DMA bus.
Bus Latency	Allowable bus latency is the time a device can wait for the DMA bus after requesting it without the loss of data or other functions.
Bus Master	When a device has access to a DMA bus, it is a bus master.
BM	Byte Mask bits on the LANCE indicating which byte(s) on the DAL lines are to be read or written
CERR	Collision Error flag in CSR$_0$
CLSN	Collision logical input to the LANCE indicating the presence of a 10.0 MHz differential signal in the collision pairs (collision ±) at the transceiver interface cable
COLL	Force Collision bit in the Mode Register in the LANCE
CRC	Cyclic Redundancy Check flag in RMD$_1$
CS	Chip Select input when asserted puts the LANCE into bus slave mode
CSR	Control and Status Registers in the LANCE
DAL	Data/Address Line pins on the LANCE
DALI	Data/Address Lines input control
DALO	Data/Address Lines output control
DAS	Data Strobe is an input/output pin on the LANCE used to distinguish the data portion of a bus transfer from the address portion.
DEF	Deferred flag in TMD$_1$. Transmission deferred
DMA	Direct Memory Access
DMA Cycle	The time required for the LANCE to transfer one word over the DMA bus (600 ns plus any wait states). In the burst mode, the LANCE transfers eight words in 4.8 us each time it acquires the DMA bus.
DRTY	Disable Retry flag in Moter (allows only one transmission attempt)
DRX	Disable the Receiver bit in Mode Register
DS	Data Strobe on the Z8001
DTCR	Disable Transmit CRC bit in Mode Register
DTX	Disable Transmitter bit in Mode Register
Daisy-chain	A method of connecting bus master devices to control interrupt priorities.
Dwell time	The time interval between the LANCE's release of the bus to the next bus request. It is the time interval that the $\overline{\text{HOLD}}$ line remains inactive.
ENP	End of Packet flag in RMD$_1$ and TMD$_1$
ERR	Error flag in descriptor rings, RMD$_1$ and TMD$_1$ and in Am7990 Control and Status Register CSR$_0$ ("or" of BABL, CERR, MISS, and MERR)
FIFO	First In First Out data buffer (Silo)
FRAM	Framing Error flag in RMD$_1$
HADR	High Order 8 Address Bits of buffer described in this entry of RMD$_1$ or TMD$_1$
HLDA	Hold Acknowledge input to the LANCE makes it the bus master.
HOLD	Bus Hold Request output asserted by the LANCE when it requires the bus.
IADR	Low order 16 bits of address of Initialization Block in CSR$_1$, high order 8 bits in CSR$_2$.
IDON	Status bit in CSR$_0$ indicates initialization has been done
INEA	Interrupt Enable bit in CSR$_0$ enables the LANCE INTR output.
INIT	Initialize flag in CSR$_0$ in the LANCE
INT	Interrupt
INTL	Internal Loopback flag in the Mode Register
INTR	Interrupt flag in CSR$_0$. Active when one or more of the CSR$_0$ status flags, BABL, IDON, MERR, MISS, RINT, or TINT, are set.
ISO	International Standards Organization
LADR	Low Order 16 Address Bits of buffer of this descriptor in RMD$_0$ and TMD$_0$
LADRF	Logical Address Filter in Initialization Block. 64-bit mask for logical addresses
LANCE	Local Area Network Controller for Ethernet (Am7990)

LCAR	Loss of Carrier flag in TMD$_3$
LCOL	Late Collision flag in TMD$_3$
Latency	See Bus Latency
LOOP	Mode Register bit to control loopback for test
MCNT	Message Byte Count entry in RMD$_3$
MERR	Memory error flag in CSR$_0$ set when the LANCE as a bus master has not received READY within 25.6 us after asserting the address on the DAL lines.
MISS	Missed Packet flag in CSR$_0$ set when the receiver loses a packet because no buffers are available.
MORE	More than one entry needed (to transmit packet) flag in TMD$_1$
OFLO	Overflow Error flag in RMD$_1$. Received data lost.
ONE	Exactly One Entry Needed (to transmit packet) flag in TMD$_1$
OWN	Descriptor entry owner (host/chip) flag in RMD$_1$ and TMD$_1$
PADR	48-bit Physical Address assigned to chip (entry in Initialization Block)
PAL	Programmable Array Logic
PROM	Promiscuous mode (accepts all incoming packets) flag in Mode Register
RAM	Random Access Memory
RAP	Register Address Port (bus addressable register) in LANCE
RAS	Register Address Select
RDP	Data Port (bus addressable register) in the LANCE
RDRA	Receive Descriptor Ring base address entry in Initialization Block
READ	Input/output pin on the LANCE. High in Bus Master mode is input to read DAL lines, low means the LANCE has placed data on the DAL lines. The meaning is reversed in the Bus Slave mode.
RENA	Receive Enable input to the LANCE. Indicates presence of carrier on the channel.
RES	Reserved
RESET	Reset (Am7990 input) stops the LANCE operation, clears its internal logic (CSR$_0$ and CSR$_3$), and sets it into idle state with STOP bit set in CSR$_0$.
RINT	Receiver Interrupt flag in CSR$_0$ set after the LANCE has received a packet and updated the Receive Descriptor Ring. Also set if the reception is stopped due to a failure.
RLEN	Receive Ring Length
RQ/GT	Request/Grant bus exchange handshake
RTRY	Retry Error flag in TMD$_3$ (transmitter failed in 16 attempts)
RW	Read/Write Select
RX	Receive Input
RXON	Receiver On flag in CSR$_0$
Runt	A packet that is less than 64 bytes long
RX	Receive input bit stream to the LANCE
SIA	Serial Interface Adapter
STP	Start of Packet flag in TMD$_1$ and RMD$_1$
STRT	Start bit in CSR$_0$
Silo	A 48-byte FIFO memory in the LANCE used to transfer data between the local memory and the serial interface adapter
Slave Mode	The slave mode of the LANCE (CS asserted) is used to initialize the LANCE.
STOP	Control bit in CSR$_0$ when set stops all external activity of the LANCE
T-state	Timing cycle (100 ns) in the LANCE. One LANCE DMA cycle consists of six T-states plus any required wait states.
TDMD	Transmit Demand for DMA bus. Control bit in CSR$_0$ to bypass DMA wait time
TDR	Time Domain Reflectometer flag in TMD$_3$ in the LANCE
TDRA	Transmit Descriptor Ring base address in Initialization Block
TENA	Transmit Enable output on the LANCE used to enable external transmit logic
TINT	Transmitter Interrupt flag in CSR$_0$ set after the LANCE has completed sending a packet and has updated the Transmit Descriptor Ring. Also set if the transmission is stopped due to a failure.
TLEN	Transmit Ring Length
TMD	Transmit Message Descriptor entry in Transmit Descriptor Ring
TX	Transmit (output)
TXON	Transmitter On flag in CSR$_0$
TX	Transmit output bit stream on the LANCE
UFLO	Underflow Error flag in TMD$_3$

APPENDIX B

8086 program for multicast addressing and hash filter.

```
                     ;         SUBROUTINE TO SET A BIT IN THE HASH FILTER FROM A
                     ;         GIVEN ETHERNET LOGICAL ADDRESS
                     ;         ON ENTRY SI POINTS TO THE LOGICAL ADDRESS WITH LSB FIRST
                     ;                  DI POINTS TO THE HASH FILTER WITH LSB FIRST
                     ;         ON RETURN SI POINTS TO THE BYTE AFTER THE LOGICAL ADDRESS
                     ;                   ALL OTHER REGISTERS ARE UNMODIFIED
                     ;
                               PUBLIC  SETHASH
                               ASSUME  CS:CSE61
                     ;
 = 1DB6              POLYL     EQU     1DB6H       ;CRC POLYNOMINAL TERMS
 = 04C1              POLYH     EQU     04C1H
                     ;
0000                 CSE61     SEGMENT PUBLIC 'CODE'
                     ;
0000                 SETHASH   PROC    NEAR
0000 50                        PUSH    AX          ;SAVE ALL REGISTERS
0001 53                        PUSH    BX
0002 51                        PUSH    CX
0003 52                        PUSH    DX
0004 55                        PUSH    BP
                     ;
0005 B8 FFFF                   MOV     AX,0FFFFH   ;AX,DX = CRC ACCUMULATOR
0008 BA FFFF                   MOV     DX,0FFFFH   ;PRESET CRC ACCUMULATOR TO ALL 1'S
000B B5 03                     MOV     CH,3        ;CH = WORD COUNTER
                     ;
000D 8B 2C           SETH10:   MOV     BP,[S1]     ;GET A WORD OF ADDRESS
000F 83 C6 02                  ADD     SI,2        ;POINT TO NEXT ADDRESS
0012 B1 10                     MOV     CL,16       ;CL = BIT COUNTER
                     ;
0014 8B DA           SETH20:   MOV     BX,DX       ;GET HIGH WORD OF CRC
0016 D1 C3                     ROL     BX,1        ;PUT CRC31 TO LSB
0018 33 DD                     XOR     BX,BP       ;COMBINE CRC31 WITH INCOMING BIT
001A D1 E0                     SAL     AX,1        ;LEFT SHIFT CRC ACCUMULATOR
001C D1 D2                     RCL     DX,1
001E 81 E3 0001                AND     BX,0001H    ;BX = CONTROL BIT
0022 74 07                     JZ      SETH30      ;DO NOT XOR IF CONTROL BIT = 0
                     ;
                     ;         PERFORM XOR OPERATION WHEN CONTROL BIT = 1
                     ;
0024 35 1D86                   XOR     AX,POLYL
0027 81 F2 04C1                XOR     DX,POLYH
                     ;
002B 0B C3           SETH30:   OR      AX,BX       ;PUT CONTROL BIT IN CRC0
002D D1 CD                     ROR     BP,1        ;ROTATE ADDRESS WORD
002F FE C9                     DEC     CL          ;DECREMENT BIT COUNTER
0031 75 E1                     JNZ     SETH20
0033 FE CD                     DEC     CH          ;DECREMENT WORD COUNTER
0035 75 D6                     JNZ     SETH10
                     ;
                     ;         FORMATION OF CRC COMPLETE, AL CONTAINS THE REVERSED HASH
                               CODE
                     ;
0037 B9 000A                   MOV     CX,10
003A D0 E0           SETH40:   SAL     AL,1        ;REVERSE THE ORDER OF BITS IN AL
003C D0 DC                     RCR     AH,1        ;AND PUT IT IN AH
003E E2 FA                     LOOP    SETH40
```

```
                          ;
                          ;         AH NOW CONTAINS THE HASH CODE
                          ;
      0040 8A DC                    MOV     BL,AH        ;BL = HASH CODE, BH IS ALREADY ZERO
      0042 B1 03                    MOV     CL,3         ;DIVIDE HASH CODE BY 8
      0044 D2 EB                    SHR     BL,CL        ;TO GET TO THE CORRECT BYTE
      0046 B0 01                    MOV     AL,01H       ;PRESET FILTER BIT
      0048 80 E45 07                AND     AH,7H        ;EXTRACT BIT COUNT
      004B 8A CC                    MOV     CL,AH
      004D D2 E0                    SHL     AL,CL        ;SHIFT BIT TO CORRECT POSITION
      004F 08 01                    OR      [DI + BX],AL ;SET IN HASH FILTER
      0051 5D                       POP     BP
      0052 5A                       POP     DX
      0053 59                       POP     CX
      0054 5B                       POP     BX
      0055 58                       POP     AX
      0056 C3                       RET
                          ;
      0057                SETHASH   ENDP
                          ;
      0057                CSEG1     ENDS
                          ;
                                    END
```

Basic computer program example to generate the hash filter, for multicast addressing, in the 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 C(32) : REM CRC REGISTER = 32 BITS
160  PRINT "ENTER STARTING ADDRESS'; : INPUT A$
170  IF LEN (A$) < > 12 THEN 160 : REM THE INPUT ADDRESS STARTING MUST BE 12 CHARS
180  REM
190  REM   UNPACK STARTING ADDRESS INTO ADDRESS ARRAY
200  REM
210  M = 0
220  FOR I = 0 TO 47 : A(I) = 0 : NEXT I
230  FOR N = 12 TO 1 STEP -1
240  Y$ = MID$ (A$,N,1)
250  IF Y$ = "0" THEN 420
260  IF Y$ = "1" THEN A(M) = 1 : GOTO 420
270  IF Y$ = "2" THEN A(M + 1) = 1 : GOTO 420
280  IF Y$ = "3" THEN A(M + 1) = 1 : A(M) = 1 : GOTO 420
290  IF Y$ = "4" THEN A(M + 2) = 1 : GOTO 420
300  IF Y$ = "5" THEN A(M + 2) = 1 : A(M) = 1 : GOTO 420
310  IF Y$ = "6" THEN A(M + 2) = 1 : A(M + 1) = 1 : GOTO 420
320  IF Y$ = "7" THEN A(M + 2) = 1 : A(M + 1) = 1 : A(M) = 1 : GOTO 420
330  A(M + 3) = 1
340  IF Y$ = "8" THEN 420
350  IF Y$ = "9" THEN A(M) = 1 : GOTO 420
360  IF Y$ = "A" THEN A(M + 1) = 1 : GOTO 420
370  IF Y$ = "B" THEN A(M + 1) = 1 : A(M) = 1 : GOTO 420
380  IF Y$ = "C" THEN A(M + 2) = 1 : GOTO 420
390  IF Y$ = "D" THEN A(M + 2) = 1 : A(M) = 1 : GOTO 420
400  IF Y$ = "E" THEN A(M + 2) = 1 : A(M + 1) = 1 : GOTO 420
410  IF Y$ = "F" THEN A(M + 2) = 1 : A(M + 1) = 1 : A(M) = 1
420  M = M + 4
430  NEXT N
```

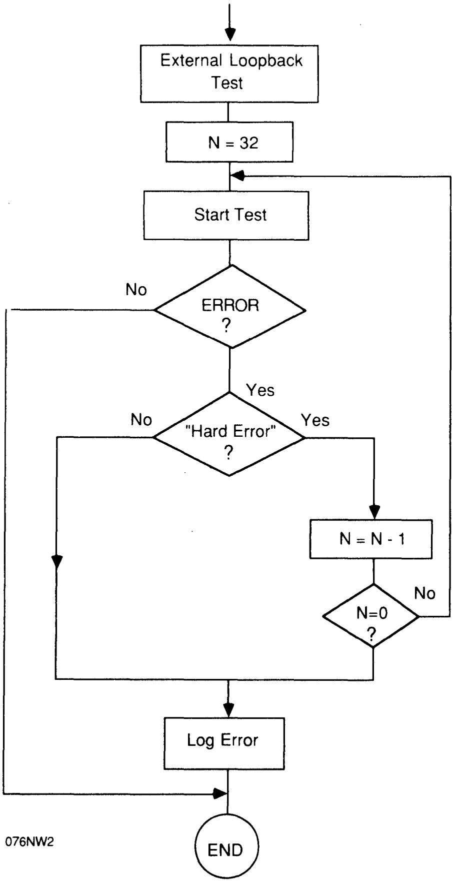
```
440  REM
450  REM   PERFORM CRC ALGORITHM ON ARRAY A(0-47)
460  REM
470  FOR I=0 TO 31 : C(I)=1 : NEXT I
480  FOR N=0 TO 47
490  REM LEFT CRC REGISTER BY 1
500  FOR I=32 TO 1 STEP -1 : C(I)=C(I-1) : NEXT I
510  C(0)=0
520  T=C(32) XOR A(N) : REM T=CONTROL BIT
530  IF T < > THEN 600 : REM JUMP IF CONTROL BIT=0
540  C(1)=C(1) XOR 1 : C(2)=C(2) XOR 1 : C(4)=C(4) XOR 1
550  C(5)=C(5) XOR 1 : C(7)=C(7) XOR 1 : C(8)=C(8) XOR 1
560  C(10)=C(10) XOR 1 : C(11)=C(11) XOR 1 : C(12)=C(12) XOR 1
570  C(16)=C(16) XOR 1 : C(22)=C(22) XOR 1 : C(23)=C(23) XOR 1
580  C(26)=C(26) XOR 1
590  C(0)=1
600  NEXT N
610  REM
620  REM   CRC COMPUTATION COMPLETE, EXTRACT HASH NUMBER FROM C(0) TO C(5)
630  REM
640  HH=32*C(0) + 16*C(1) + 8*C(2) + 4*C(3) + 2*C(4) + C(5)
650  PRINT "THE HASH NUMBER FOR ";A$;" IS ";HH
660  GOTO 160
```

MAPPING OF LOGICAL ADDRESS TO FILTER MASK

LAF Reg Bits Set	LAF Loc	Destination Address Accepted	LAF Reg Bits Set	LAF Loc	Destination Address Accepted
	Dec	(Hex)		Dec	(Hex)
0	0	0000 0000 0085	0	32	0000 0000 0021
	1	0000 0000 00A5		33	0000 0000 0001
L	2	0000 0000 00E5	L	34	0000 0000 0041
A	3	0000 0000 00C5	A	35	0000 0000 0071
F	4	0000 0000 0045	F	36	0000 0000 00E1
	5	0000 0000 0065		37	0000 0000 00C1
0	6	0000 0000 0025	2	38	0000 0000 0081
	7	0000 0000 0005		39	0000 0000 00A1
	8	0000 0000 002B		40	0000 0000 008F
	9	0000 0000 000B		41	0000 0000 00BF
	10	0000 0000 004B		42	0000 0000 00EF
	11	0000 0000 006B		43	0000 0000 00CF
	12	0000 0000 00EB		44	0000 0000 004F
	13	0000 0000 00CB		45	0000 0000 006F
	14	0000 0000 008B		46	0000 0000 002F
15	15	0000 0000 00BB	15	47	0000 0000 000F
0	16	0000 0000 00C7	0	48	0000 0000 0063
	17	0000 0000 00E7		49	0000 0000 0043
	18	0000 0000 00A7		50	0000 0000 0003
	19	0000 0000 0087		51	0000 0000 0023
L	20	0000 0000 0007	L	52	0000 0000 00A3
A	21	0000 0000 0027	A	53	0000 0000 0083
F	22	0000 0000 0067	F	54	0000 0000 00C3
	23	0000 0000 0047		55	0000 0000 00E3
1	24	0000 0000 0069	3	56	0000 0000 00CD
	25	0000 0000 0049		57	0000 0000 00ED
	26	0000 0000 0009		58	0000 0000 00AD
	27	0000 0000 0029		59	0000 0000 008D
	28	0000 0000 00A9		60	0000 0000 000D
	29	0000 0000 0089		61	0000 0000 002D
	30	0000 0000 00C9		62	0000 0000 006D
15	31	0000 0000 00E9	15	63	0000 0000 004D

Appendix C

External Loopback Test Flow Chart



N = Max. number of times to repeat the test.

External Loopback Test Procedure

Due to the problem of Silo Pointer Mis-alignment in the LANCE's External Loopback, the following gives the terminology used and procedures recommended for performing the External Loopback Test.

Terminology:

LANCE Hard Errors:

These can be caused by External Loopback Silo Pointer Mis-alignment (false Hard Error), or they can be real Hard Errors in the network that the software can correct. Examples of real Hard Errors are: LCAR, RTRY, CRC, FRAM, BABL, MISS,OFLO, BUFF.

LANCE Soft Error:

This is a real error. It is not a result of External Loopback Silo Pointer Mis-alignment, and the software must correct. The Soft Error is: CERR.

System Hard Errors:

These errors signal a hardware failure in the system. Examples for this type of error are: MERR, UFLO. These are not caused by External Loopback Silo Pointer Mis-alignment.

Testing Procedure:

When a LANCE Hard Error occurs and the source cannot be determined, repeat the External Loopback Test until it passes; or until a real Hard Error, a Soft Error, or a System Hard Error is found; or until it has continuously failed for a predetermined number of times (N). The error in the last attempt is then logged. If a Soft error, or System Hard Error occurs, an error handling routine will take the proper action and the error is logged.

SECTION 3
Applications

Modifications Needed to Port LANCE Software to the ILACC 32-Bit Environment

This is a list of the modifications necessary to port a previously written Am7990 (LANCE) driver to the Am79C900 (ILACC) environment. The major task in porting the software is restructuring the data objects that the 32-bit oriented ILACC will access. These changes constitute the minimum modification for driver functionality and do not make use of the extended features of the ILACC (extended register set, additional interrupt conditions, etc.):

RAP. No modification is necessary if the reserved bit field (bit 2 - bit 15) was filled with zeros.

CSR0. No modification necessary since the 16 status bits are mapped identically as they were in the LANCE.

CSR1. The Initialization Block must start on an even word boundary. Thus, IADR should have bits 1 and 0 equal to zero.

CSR2. The bits 15 through 8 are no longer reserved. The upper 16 bits of the 32 bit IADR should be placed here.

CSR3. No modification necessary if the reserved bit field was filled with zeros (these zeros will enable the new interrupt signals.)

CSR4 Interrupt Masks. In order for the ILACC to be compatible with LANCE interrupt conditions, LBDM and TXSTRTM (Loopback Done Mask and Transmit Start Mask) must be set. This is done after ILACC RESET, by writing 0x0004 to RAP and 0x0005 (or 0x0045, see below) to RDP.

CSR4 BACON Bits. The ILACC ignores the Byte Control bits BCON and BSWP in CSR3. If the designed ILACC bus interface is for a 32 Bit 80X86 processor, the BACON bits must be programmed to 00 after ILACC RESET, by writing 0x0004 to RAP and 0x0005 to RDP. For a 32 Bit 680X0 processor interface, the BACON bits in CSR4 must be programmed to 01 (0x0045 RDP.) Note that CSR4 is not cleared by CSR1 STOP, and needs to be written only once after RESET.

Initialization Block. As mentioned earlier, the 28-byte ILACC initialization block must begin on an even word boundary (address bits 1 and 0 must equal zero). Contrast this to the 24-byte LANCE initialization block structure which starts on an even byte boundary. In addition, the components of the ILACC initialization block are reordered as follows:

IADR+0x0h: Mode Register (same position as before.)

IADR+0x2h: RLEN byte (lower 4 bits reserved, upper 4 indicate RLEN. See below.)

IADR+0x3h: TLEN byte (lower 4 bits reserved, upper 4 indicate TLEN. See below.)

IADR+0x4h: 48-bit Physical Address (low to high bit order.)

IADR+0xAh: 16 bits reserved.

IADR+0xCh: 64-bit Logical Address Filter (low to high-bit order.)

IADR+0x14h: 32-bit RDRA (see below.)

IADR+0x18h: 32-bit TDRA (see below.)

RLEN and TLEN of the Initialization Block must be modified to be 4-bits wide instead of 3. The previous 3-bit LANCE values must be shifted one bit to the right into the reserved bit fields (the reserved bit fields in the RLEN/TLEN bytes for the LANCE are shortened by one bit for use with the ILACC.)

RDRA and TDRA are now 32-bit addresses. Note that each descriptor ring now must start at an 8 word boundary. Thus, RDRA and TDRA must have bits 3 through 0 equal to zero.

Mode Register. No modification is necessary if the reserved bit field was filled with zeros. This configuration selects the internal SIA.

Descriptors. Descriptors (both RX and TX) must start at and be placed at 8 word (16-byte) intervals, instead of 4 word intervals. Even though the ILACC descriptors are 6 words wide, memory space must be allocated for 8 words per descriptor. The descriptors are restructured as follows (assume RMD0 and TMD0 are base addresses for two descriptors):

RMD0+0h: 32-bit buffer address. Special address must be word bound

RMD0+4h: Identical to RMD2 in LANCE

RMD0+6h: Identical to RMD1 in LANCE, except that low byte is reserved

RMD0+8h: Identical to RMD3 in LANCE

RMD0+0Ah: Word containing two new 8-bit status counters, RPC and RCC

TMD0+0h: 32-bit buffer address. Specified address must be word bound

TMD0+4h: Identical to TMD2 in LANCE

TMD0+6h: Identical to TMD1 in LANCE, except that low byte is reserved

TMD0+8h: Word containing a new 4-bit status counter, TCC

TMD0+0Ah: Identical to TMD3 in LANCE

The new Transmit Collision Count (TCC), Runt Packet Count (RPC), and Receive Collision Count (RCC) indicators can be ignored for a minimum software modification.

Publication #	Rev.	Amendment	Issue Date
14532	A	/0	5/90

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-

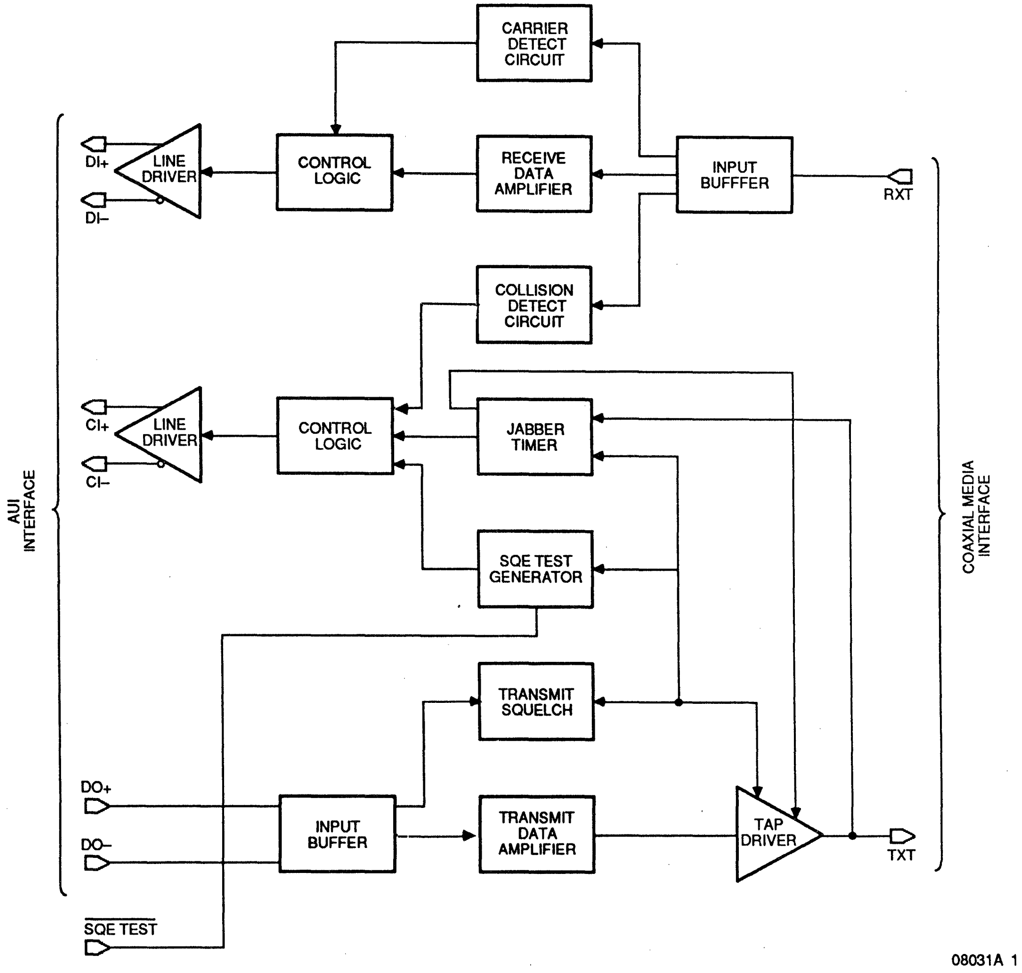


Figure 1. Am7996 Block Diagram

Publication #	*Rev.*	*Amendment*	*Issue Date*
08031	B	/0	3/90

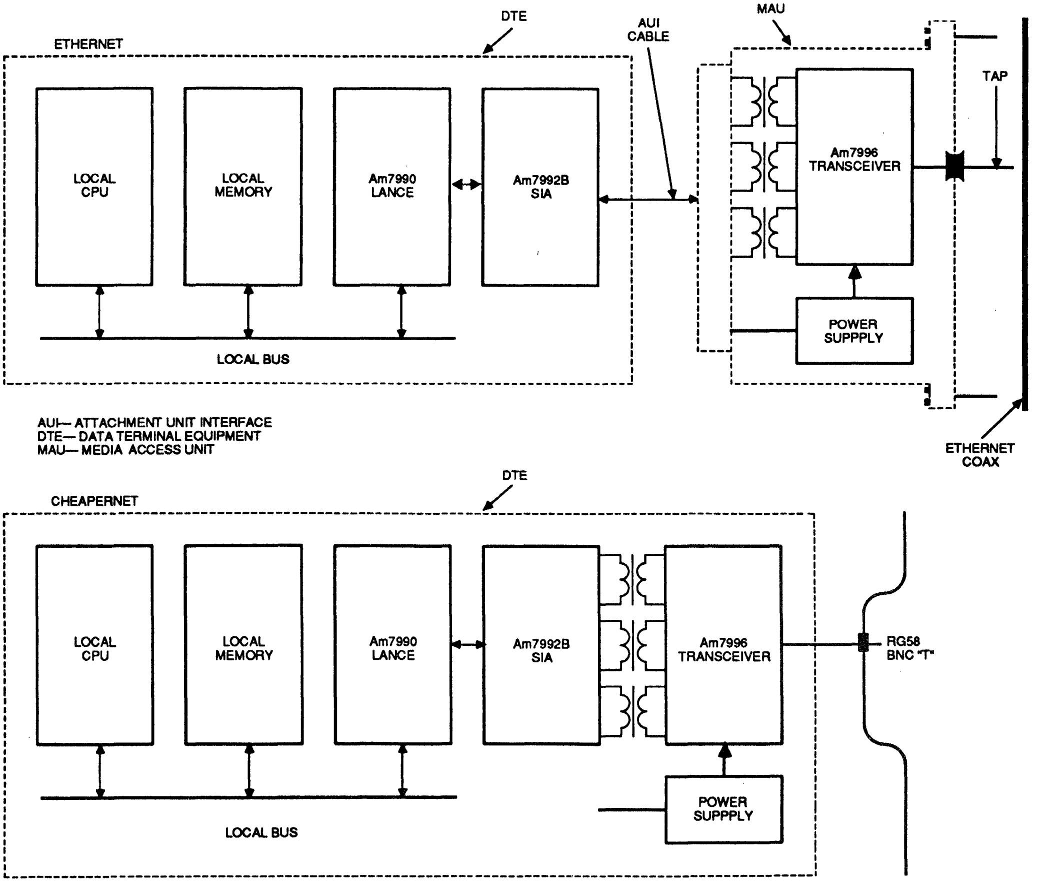
erates independently. In the transmit section, data is received differently from the Data Terminal Equipment (DTE) and transmitted out, single ended, to the medium (coax cable). The Jabber function guards the medium from node transmissions that are excessive in length. The receive section listens to 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.



08031A 2

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.

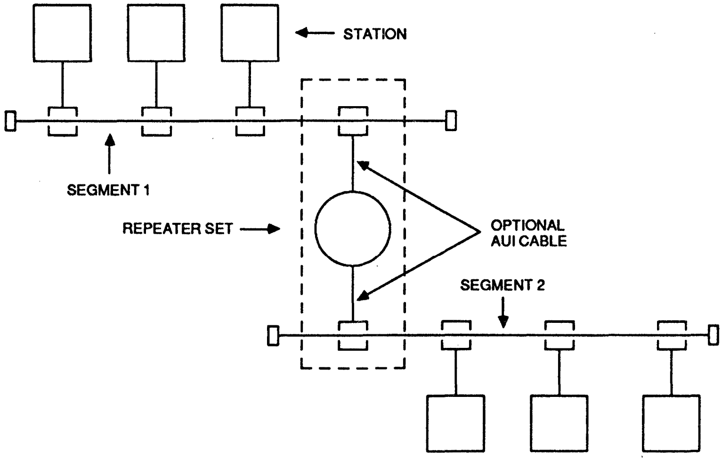
In an Ethernet installation, up to 100 Media Access Units (MAU) may be connected to one cable segment of 500 meters. In a Cheapernet installation, up to 30 MAUs may be connected to one cable segment of 185 meters. In either Ethernet or Cheapernet, repeaters may be used to connect up to five segments 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.

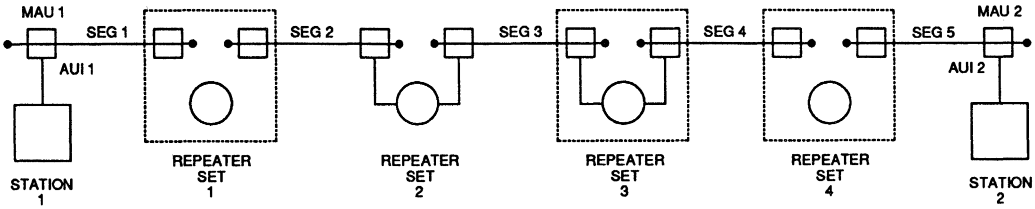
The salient features of Ethernet and Cheapernet are shown in Table 1.

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 REQURING 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± to the DTE when $\overline{\text{SQE TEST}}$ 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±, is intact and operational.

Pin-strappable SQE Test Option: The SQE test is selectable via the $\overline{\text{SQE TEST}}$ 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).

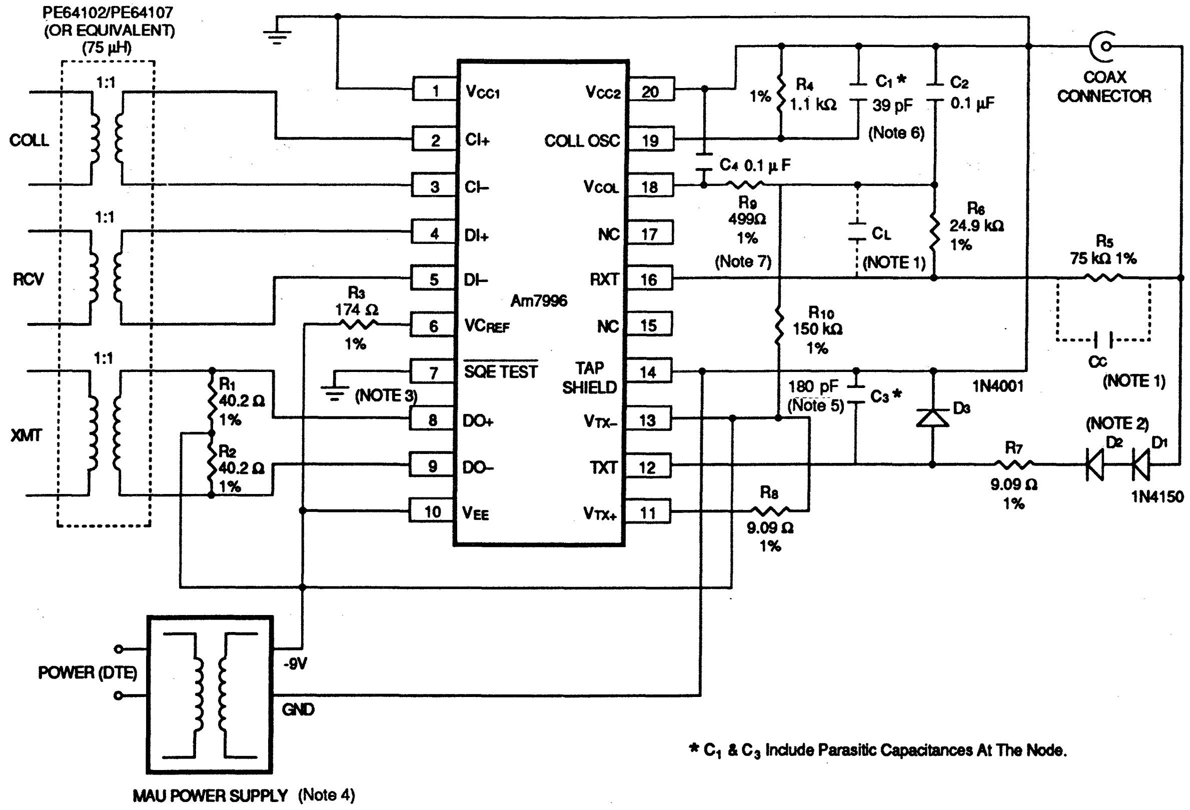
Collision Detection in Non-Repeater Applications: Receive Mode collision detection is normally not necessary when designing a MAU in a non-repeater application. This is because the received packet, the packet participating in collision, usually ends up as a runt packet (a packet less than 64 bytes long) 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-



* 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 = 1/3 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.

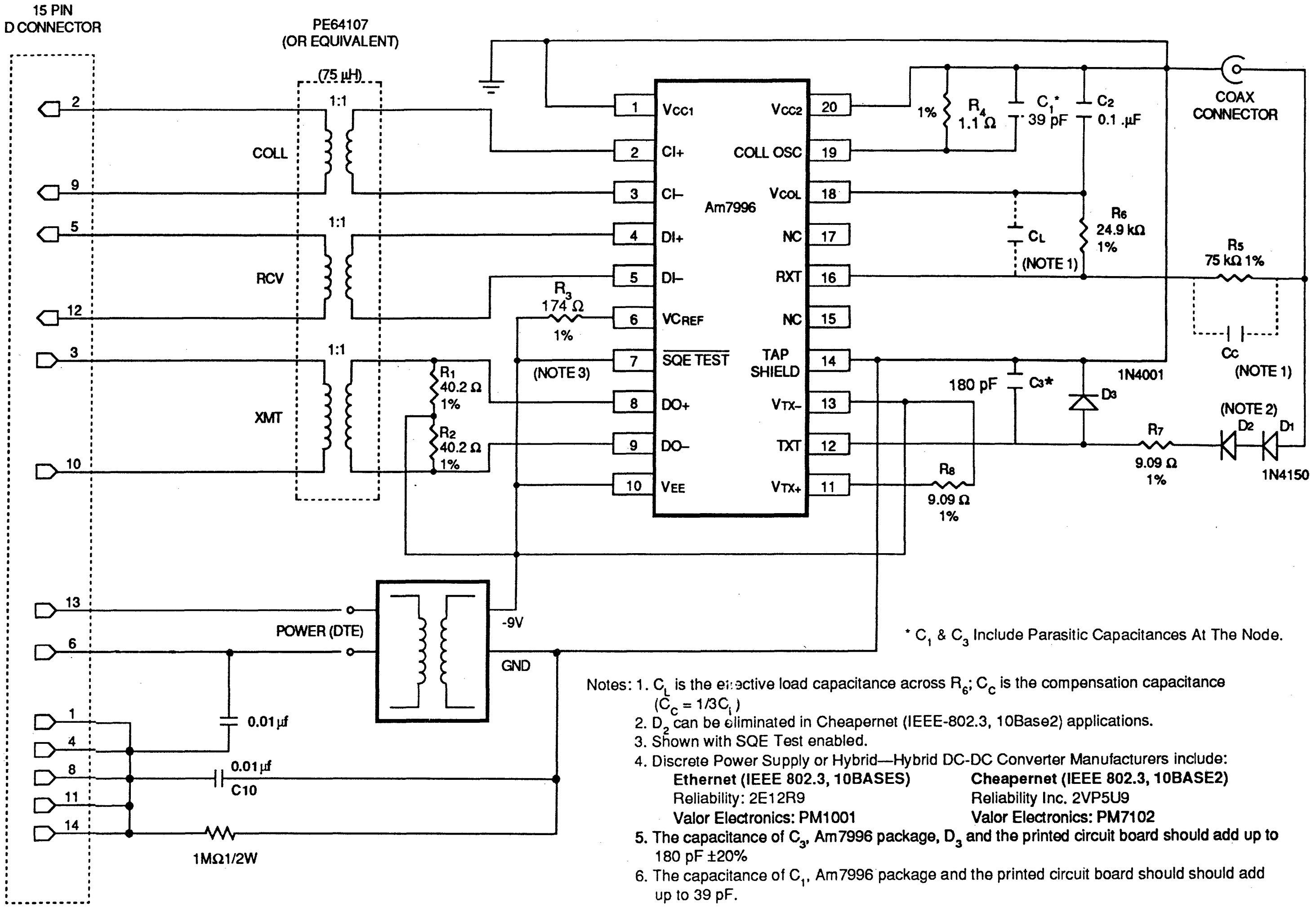
When the chip is not transmitting, the transmit path

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 ($C_C = 1/3C_1$)

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 $2.331 R_4 C_1$.

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).

$$V_{RXT} - V_{COL} = \frac{(V_{COAX} - V_{COL})(1/C_L)}{(1/C_C) + (1/C_L)}$$

$$(V_{RXT} - V_{COL})(1 + C_L/C_C) = V_{COAX} - V_{COL}$$

$$\text{Let: } y = C_L/C_C$$

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:

$0 = 4V_{RXT} - 3V_{COL}$

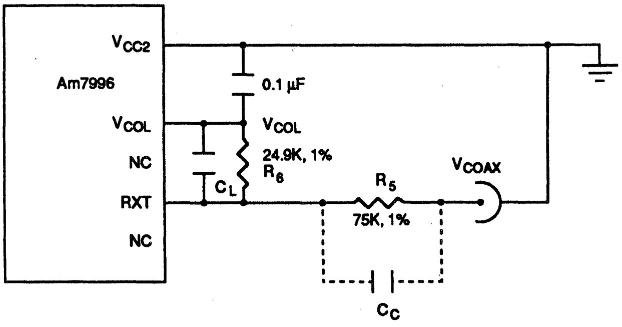
$V_{RXT} = -1200$ mVDC

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.

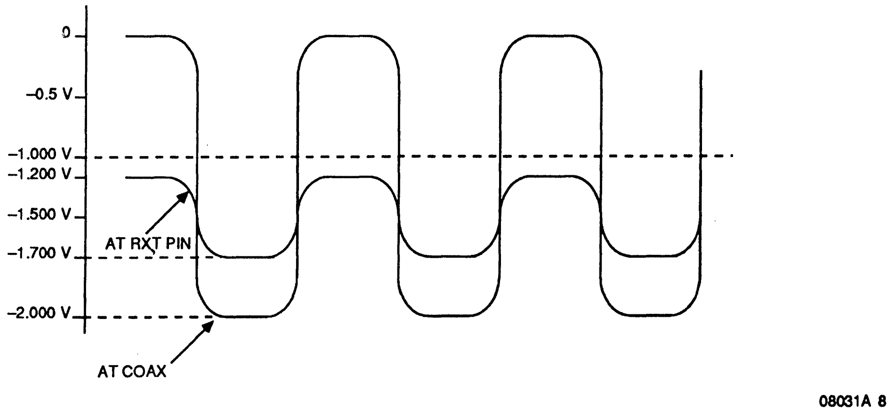


Figure 7. Attenuator Compensation Technique

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:

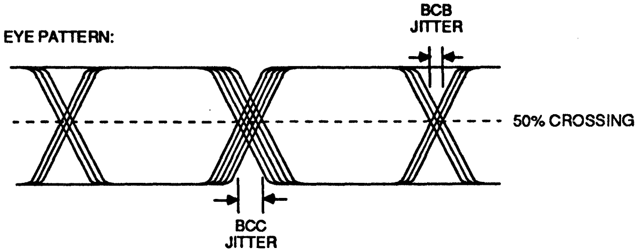
Bit Cell Center (BCC)
Bit Cell Boundary (BCB)

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.



08031A 9

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.

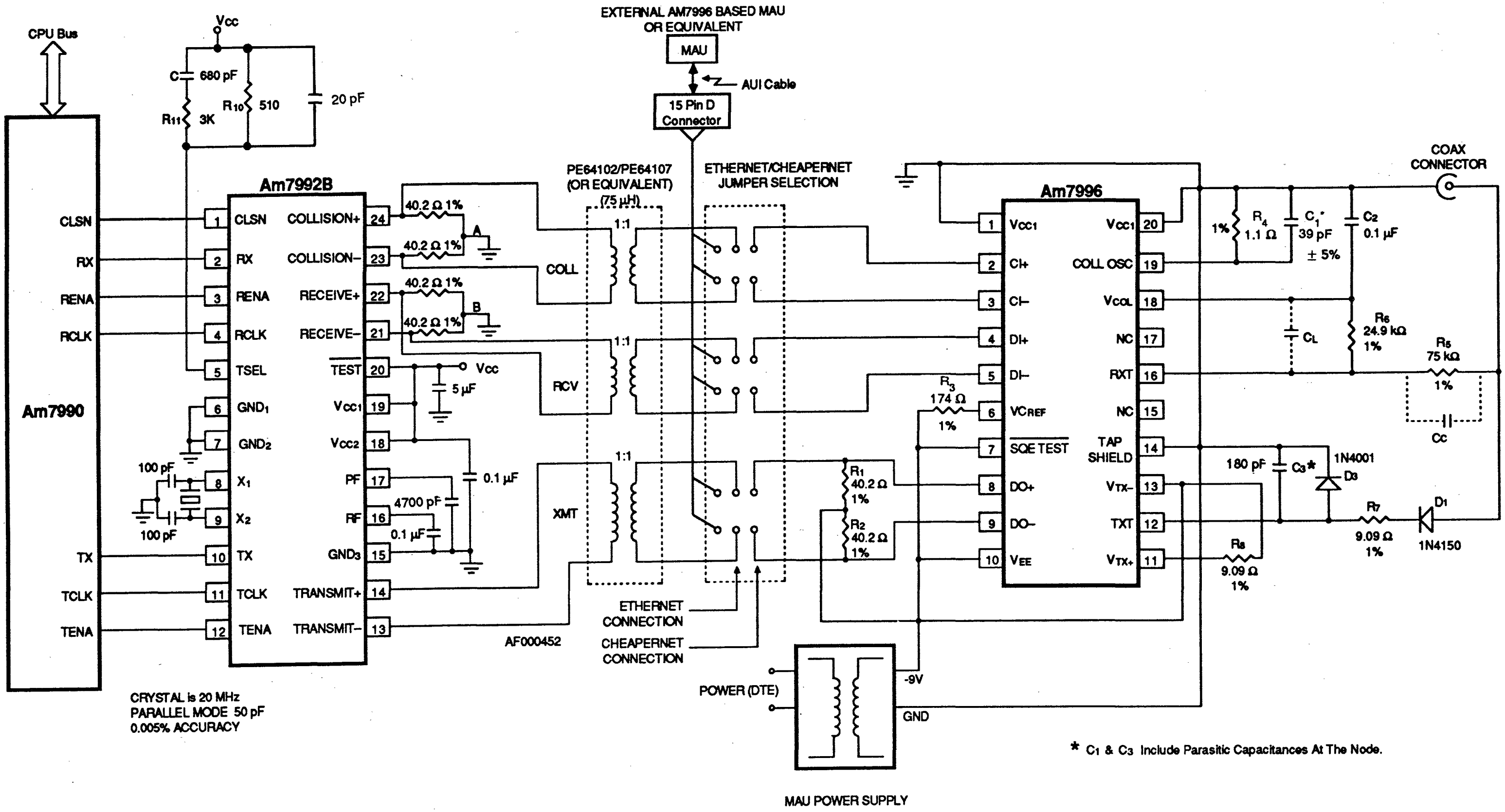
Figure 9 shows a typical Cheapernet implementation using Am7990, Am7992B, and Am7996 (also refer to AMD stand alone Ethernet/Cheapernet evaluation board). For a detailed discussion of the external Ethernet/Cheapernet components to the Am7996, refer to the External Component Design Guidelines presented 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 $\overline{\text{TEST}}$ 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±,



CRYSTAL is 20 MHz
PARALLEL MODE 50 pF
0.005% ACCURACY

* C1 & C3 Include Parasitic Capacitances At The Node.

08031A 10

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.

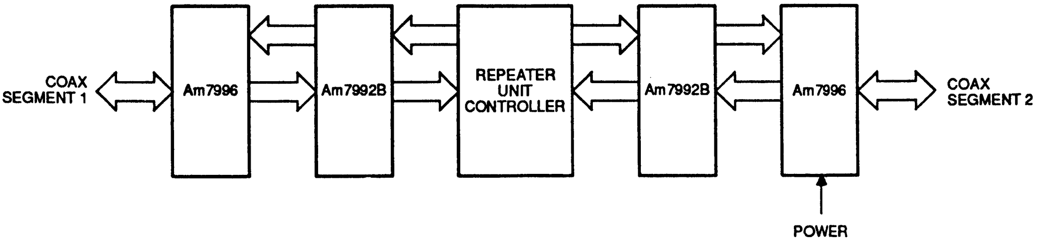


Figure 10. Typical Repeater Unit Block Diagram

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.

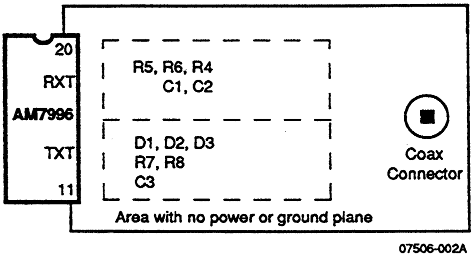
The only layout restriction for the transmitter circuit is that the longest current path from the TXT pin (pin 12) to the coaxial cable's center conductor must be no longer than 4 inches.

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.

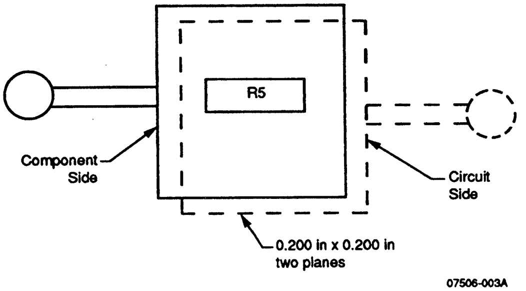
If the above layout rules are followed, the parasitic capacitance in parallel with R6 will be about 6 pF. This parasitic capacitance is shown in the schematics as C_L. (Note that C_L is a parasitic capacitance. Do not add a discrete capacitor in parallel with R6.) The capacitor labelled C_C in the schematics 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.



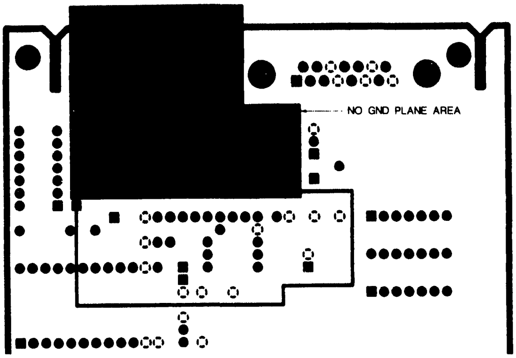
BD008560

Figure 11. PC Board Outline



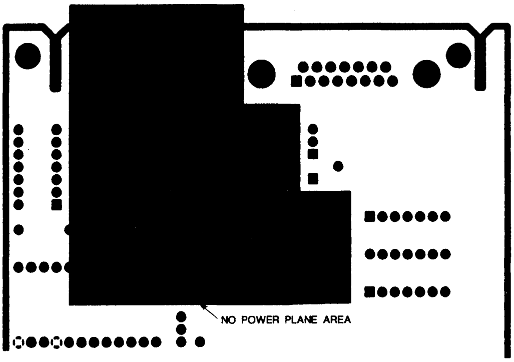
BD008550

Figure 12 PC Board Trace Capacitor



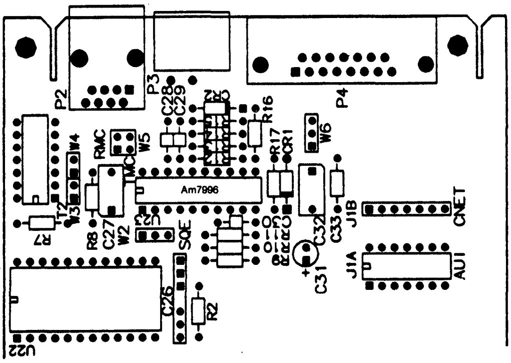
CD012021

GND PLANE

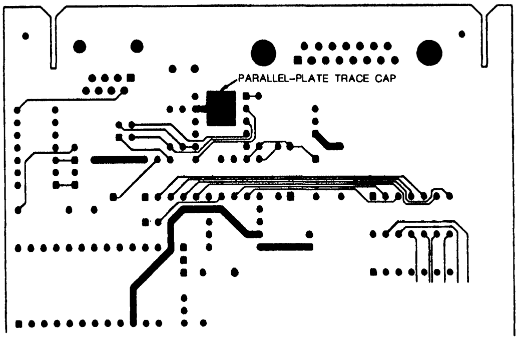


CD012011

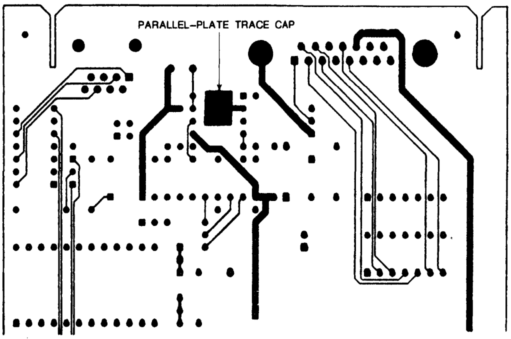
POWER PLANE



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 4
Development Tools

LANCE-AT-KT

Advanced Micro Devices

Am7990 Based Ethernet/Cheapernet/Twisted Pair Half Card Evaluation Kit for PC/AT™

DISTINCTIVE CHARACTERISTICS

- **LANCE-AT-KT/x Integrated with 10BASE2 transceiver and AUI port**
- **LANCE-AT-KT/xT 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 Local Area Network Controller for Ethernet (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 (LANCE-AT-KT/x only)
  - On board transceiver for Twisted Pair connection (LANCE-AT-KT/xT 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 Novell NetWare™ driver object code, allowing immediate connection to a working LAN environment**

GENERAL DESCRIPTION

The Ethernet/Cheapernet/IEEE 802.3 evaluation kit, called the LANCE-AT-KT, 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 LANCE-AT-KT/x card supports the 10BASE2 (Cheapernet) connection, while the LANCE-AT-KT/xT 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 multi-master 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.

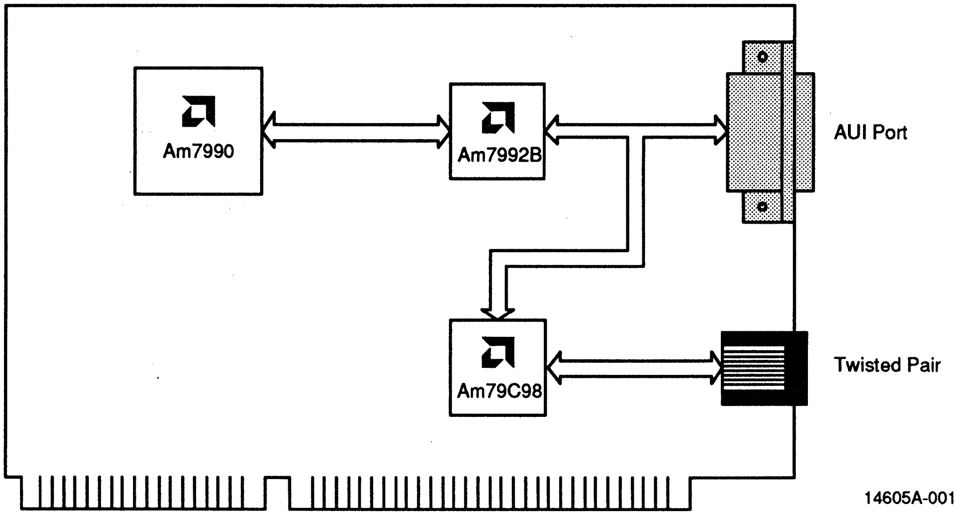
The LANCE-AT-KT comes with a floppy diskette of software programs. The software includes a high-level demonstration program, a low-level driver/monitor, Packet Driver (version 1.07), and Novell NetWare™ driver object code. The demonstration program contains an ISO data link layer with a menu driven interface 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 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 LANCE driver in 80286 assembly language. The Novell NetWare object code transforms the 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. The LANCE-AT-KT and its NetWare driver is certified by Novell.

In addition to the board and the software diskette, the LANCE-AT-KT kit comes with a user's manual including device specifications, device application notes, a local area network controller technical manual, the AMD Ethernet/IEEE 802.3 Family data book, and cable hook-up hardware.

Publication# 14605 Rev. B Amendment/0
Issue Date: January 1992

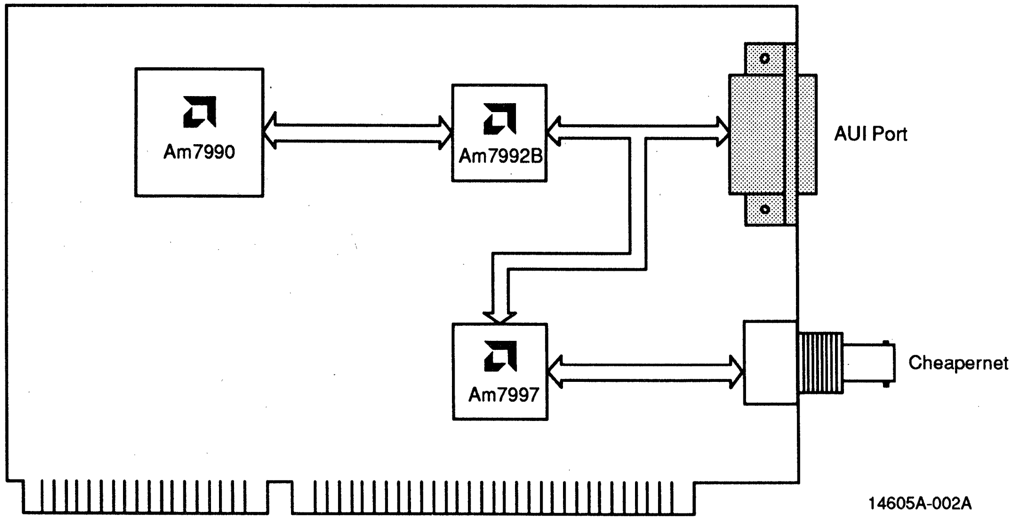
BLOCK DIAGRAM

LANCE-AT-KT/xT



14605A-001A

LANCE-AT-KT/x



14605A-002A

ILACC-MAC-KT

Advanced
Micro
Devices

Am79C900 Based 32-Bit NuBus™ Card Ethernet Evaluation Kit for Macintosh II®

DISTINCTIVE CHARACTERISTICS

- **Implements a working Ethernet evaluation node using a Mac II as host**
- **Complete solution utilizes the Am79C900 Integrated Local Area Communications Controller (ILACC) high performance 32-Bit Ethernet controller**
- **Development software allows the system designer access to register level functions of the ILACC, thereby facilitating quick development of customized software drivers**
- **On-board transceiver supports 802.3 10BASE2 (Cheapernet) connection. Standard AUI port can be connected to10BASE5 (Ethernet) or 10BASE-T (Twisted Pair) media**

GENERAL DESCRIPTION

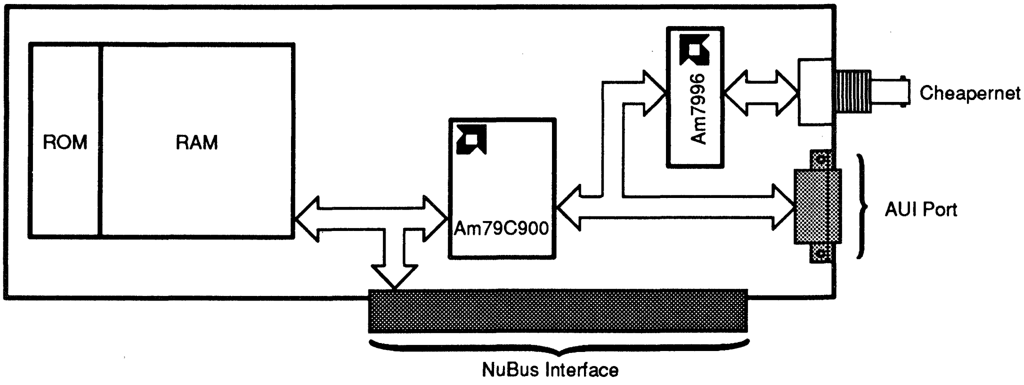
The Ethernet/Cheapernet/IEEE 802.3 evaluation kit (ILACC-MAC-KT) is a design evaluation vehicle for AMD's 32-Bit ILACC Ethernet chip. It is intended for use in the Apple Macintosh II family of computers, and represents a highly integrated Ethernet network adapter implementation. The ILACC-MAC-KT utilizes a shared memory architecture, and contains the Declaration ROM and user ROM capability. The card supports Ethernet and 10BASE-T via the AUI port and Cheapernet (10BASE2) with an onboard transceiver. The choice of network media operation is jumper selectable.

When the ILACC-MAC-KT is installed in a host machine, the system becomes a platform upon which to evaluate network hardware and to develop software for the 32-bit ILACC Ethernet controller. The system designer can therefore minimize the time required for familiarization with and prototyping of ILACC based Ethernet applications.

The ILACC-MAC-KT comes with ILACC Monitor (IMON) software. This software monitor allows the user to assign physical and logical addresses, establish connections, and send and receive data packets. The monitor lets the user view and change the contents of the ILACC's registers, the memory resident Initialization Block, and the data buffer Descriptor Rings. The program also allows the designer to establish operational loops for hardware data probing. In the IMON environment, commands can be entered via the command line for single step debugging or through batch files for automatic or repetitious operations.

In addition to the the board and the software diskette, the ILACC-MAC-KT kit comes with a user's manual including schematic diagrams, device specifications, and coaxial cable hook-up hardware.

BLOCK DIAGRAM



Publication# 14781 Rev. B Amendment/0
Issue Date: January 1992

Advanced
Micro
Devices

Am7996EVAL-HW

Ethernet/Cheapernet Transceiver Evaluation Kit

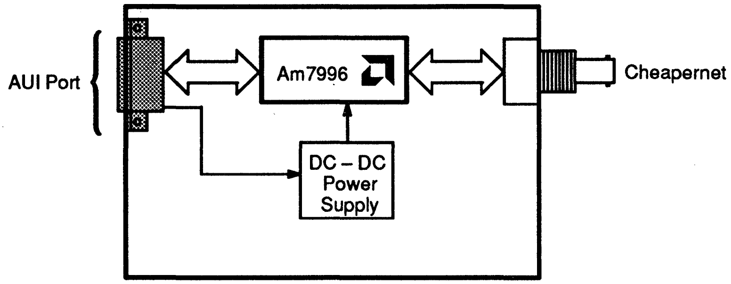
DISTINCTIVE CHARACTERISTICS

- **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

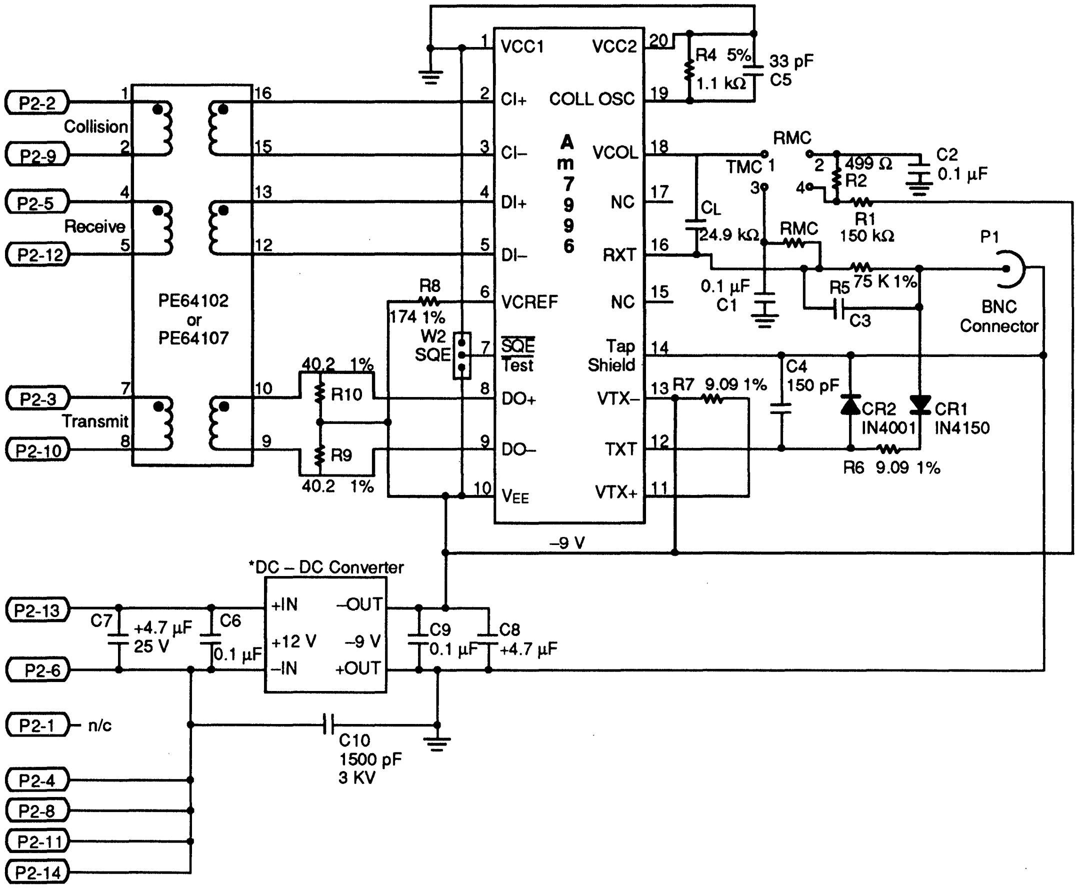
The jumper settings specify the type of collision detect that is used.

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

Am7997EVAL-HW

Advanced Micro Devices

Stand-alone Ethernet/Cheapernet IEEE 802.3 Compliant Tap Transceiver Evaluation Board

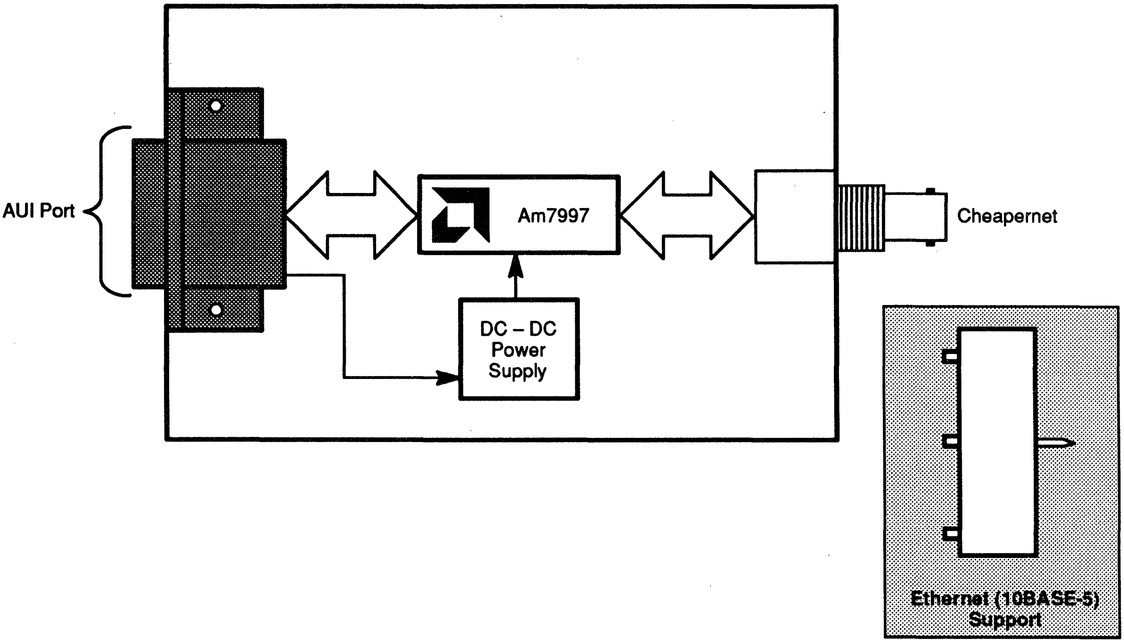
DISTINCTIVE CHARACTERISTICS

- **Fully compliant to the IEEE 802.3 Ethernet specification for both 10BASE-2 and 10BASE-5**
- **Standard version connects the standard AUI port interface to 10BASE-2 (Cheapernet) BNC connector. PC board can also support the 10BASE-5 'vampire' connector**
- **Jumper selectable Signal Quality Error (SQE) test allows evaluation in both DTE and repeater applications**
- **Self contained board facilitates rapid evaluation of the AUI-to-coax system block, and demonstrates optimal analog design criteria necessary for IEEE 802.3 compliance**

GENERAL DESCRIPTION

The Am7997EVAL-HW board is a self contained module that implements all the functions of a fully compliant Medium Attachment Unit (MAU) for IEEE 802.3 10BASE-5 (Ethernet) or 10BASE-2 (Cheapernet). The evaluation board is based on the 802.3 compliant Am7997 coaxial tap transceiver chip. This 3"x3" board can be used to connect any AUI port interface to either the 'thin' coaxial cable used in Cheapernet networks or to the 'thick' cable used in 10BASE-5 applications. 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 (DTE) or network repeater applications. In addition, optimal PC board layout is employed. The Am7997EVAL-HW kit comes complete with the AMD 802.3 Handbook, schematics, application note, and PC board layout artwork.

BLOCK DIAGRAM



15090-001A

Publication# 15090 Rev. B Amendment/0
Issue Date: January 1992

LAYOUT CONSIDERATIONS

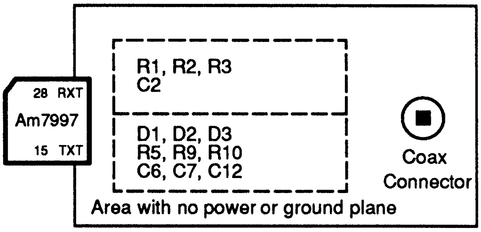
To protect the transceiver from the environment and to achieve optimum performance, the Am7997 is designed to be used with two sets of external components: the transmitter circuit consisting of components D1, D2, D3, R5, R9, R10, C6, C7, and C12, and the receiver circuit consisting of components R1, R2, R3, C2, C_L, and C_C (C_L is a parasitic capacitance rather than a discrete component). These combined circuits are shown in the schematic. The resistor tolerances for these circuits are specified as 1%.

The primary layout restriction for the transmitter circuit is that the longest current path from the TXT pin (PLCC pin 17) to the coaxial cable's center conductor must be as short as possible. Also, the transmitter circuitry should be physically isolated from power and ground planes, i.e. no power or ground planes under the transmitter circuit components.

The layout of the receiver circuit is even more critical. To minimize parasitic capacitance that can degrade the received signal, the external receiver circuit should also be physically isolated from power and ground planes. There must be no power or ground plane under the area of the P.C. board that includes PLCC pins 22 through 1, R2, R3, 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 R2–R3 attenuator. Also, the RXT pin should be as close as possible to the coaxial cable connector.

The layout of the receiver and transmitter circuits can be simplified by omitting power and ground planes from the whole area on the right side of the Am7997 as shown in Figure 1. Also, the TAP SHIELD pin (PLCC pin 28) should be connected directly to the shield of the coax connector without any other P.C. board ground trace hanging off of it.

If the above layout rules are followed, the parasitic capacitance in parallel with R2 will be about 6pF. 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 R2). The capacitor labeled C_C in the schematics is the total capacitance in parallel with R3 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 R3 to R2, which is 3:1. This means that an additional 1 pF of capacitance must be added in parallel with R3.



15090-002A

Figure 1.

This extra 1pF of capacitance can easily be added by building a capacitor out of P.C. board traces right under R3. One example of this is a 0.100 x 0.100 inch parallel plate capacitor on two layers of the P.C. board as shown in Figure 2 and Figure 3 (These dimensions assume that the P.C. board is made from 0.060 inch thick G-10 material. The top plate of the capacitor should be connected to one lead of R3, with the bottom plate connected to the other lead of R3.

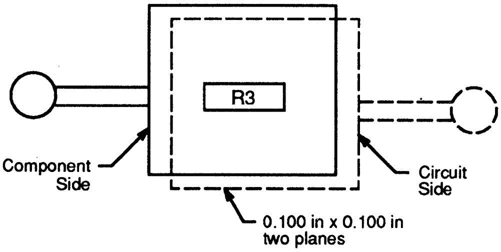
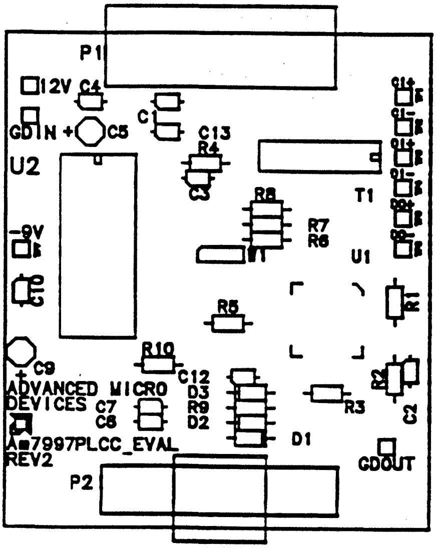
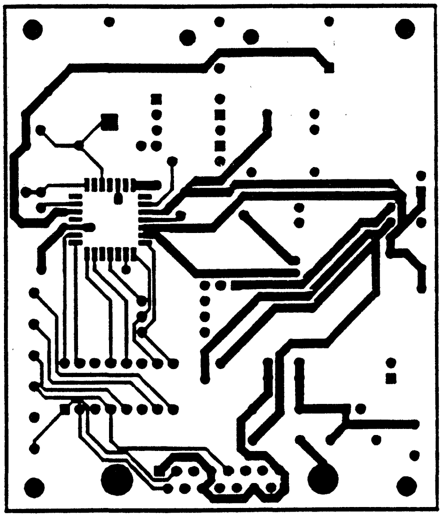


Figure 2. 15090-003A



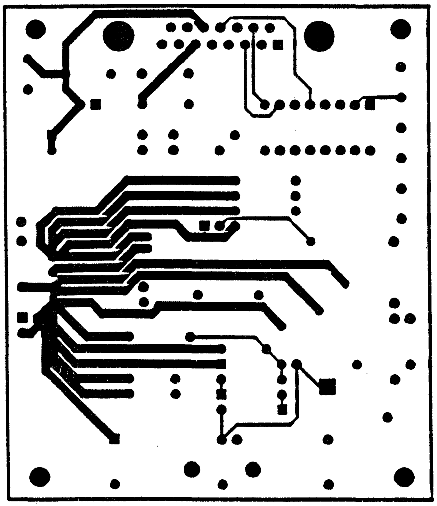
15090-004A

Component Side Silkscreen



15090-005A

Component Side



15090-006A

Solder Side

C1	.001µF GAPKAP
C2	0.1µF
C3	1000pF 3KV
C4	0.1µF
C5	4.7µF
C6	1µF
C7	1000pF
C9	4.7µF
C10	0.1µF
C11	NOT USED
C12	180pF
C13	0.01µF 50V
D1	1N4153
D2	1N4153
D3	MUR120
P1	15-Pin D Connector Male
P2	BNC Connector
R1	1.96KΩ 1%
R2	24.9KΩ 1%
R3	75.0KΩ 1%
R4	1MΩ 5% 0.5W
R5	2.74KΩ 1%
R6	237Ω 1%
R7	40.2Ω 1%
R8	40.2Ω 1%
R9	9.09Ω 1%
R10	27.0KΩ 1%
T1	LT6031 Pulse Transformer
U1	Am7997JC
U2	PM7005 DC-DC Converter
W1	3-Pin Jumper Block

(All Resistors are 1% except where noted)

Figure 3. Suggested Printed Circuit Board Layout for a Double-Sided PCB Application

Am79C98EVAL-HW

10BASE-T (Twisted Pair Ethernet) Transceiver Evaluation Kit

Advanced Micro Devices

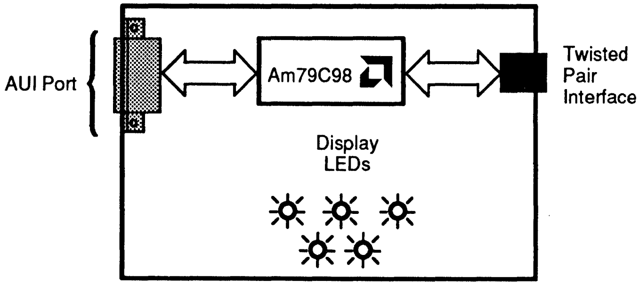
DISTINCTIVE CHARACTERISTICS

- **Connects 10BASE-T network (using an RJ45 connector) to Ethernet controller via the standard Attachment Unit Interface (AUI).**
- **Jumper selectable enable/disables for the following functions: Signal Quality Error (SQE), Idle Link Test, and Link Polarity Reversal Detection/Correction.**
- **Self contained board permits immediate connection to "Twisted Pair Ethernet" network.**

GENERAL DESCRIPTION

The Am79C98EVAL-HW board is a self contained module that implements all the functions of a Medium Attachment Unit (MAU) for the IEEE 802.3 Draft 10 for 10BASE-T (Twisted Pair Ethernet). This compact board can be used to connect the Data Terminal Equipment (DTE) (via an AUI port interface) to the twisted pair wiring used in 10BASE-T "Twisted Pair Ethernet" networks. The evaluation system contains onboard LED indicators to help the designer monitor the status of the network interface, including transmit/receive, collision, Idle Link status, and Polarity Reversal Detection and Correction. A jumper selectable SQE test (Heartbeat) enable/disable lets the board function in both DTE (SQE enabled) or network repeater (SQE disabled) applications. The Idle Link Test and Polarity Reversal Detection/Correction functions can also be separately disabled based on user configurable jumpers. In addition, optimal application configuration and PC board layout are employed. The Am79C98EVAL-HW kit comes complete with schematics, PC board artwork, and twisted pair wiring connection hardware.

BLOCK DIAGRAM



IMR-VELCRO-HW

Am79C980 Integrated Multiport Repeater Evaluation Kit

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **Stand alone 8-port 10BASE-T hub**
- **8 10BASE-T ports and 1 AUI port for added versatility**
- **Serial Management Port interface for monitoring and programming the board with the IBM PC demo software**
- **Only a single 12 V DC power supply required**
- **On-board transformer/filter modules for filtering and network isolation**
- **Contains status LEDs for either carrier sense or link status**
- **Expandable to a 24-port hub (plus 3 AUI ports)**

GENERAL DESCRIPTION

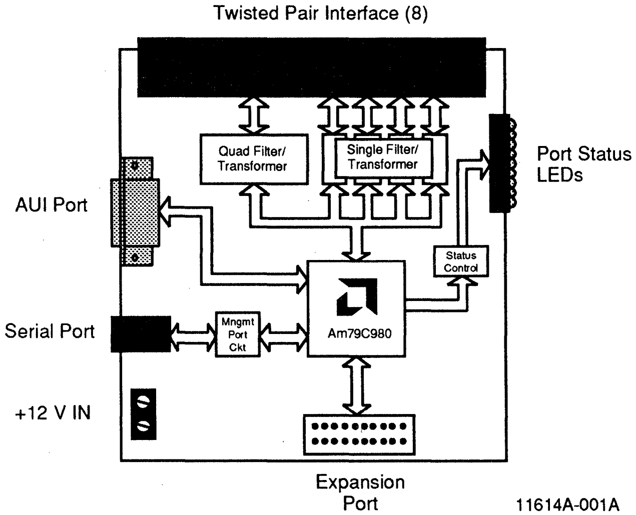
The Twisted Pair Transceiver evaluation kit (IMR-VELCRO-HW) is a design evaluation vehicle for AMD's Am79C980 Integrated Multiport Repeater (IMR™). The board is a self contained module that fully implements the Repeater functions specified by section 9 of the IEEE 802.3 standard and the Twisted Pair Transceiver functions conforming to the 10BASE-T standard.

The 5½″ x 5¾″ board contains eight integral Twisted Pair Medium Attachment Units (MAUs) and an IEEE 802.3 (section 7) compliant Attachment Unit Interface (AUI) to provide a simple platform to allow evaluation of the baseline feature set of the IMR. This system permits the network designer to monitor activities on the card through the serial Management Port. The board has an RS-232 interface that allows the designer to program the IMR and to read back the chip status with the provided software. The board also contains status LEDs for monitoring carrier sense or link status. With the on-board expansion bus, the hub is expandable to a maximum of 24 Twisted Pair MAUs and 3 AUIs.

The kit comes with an interactive IMR Management Port Monitor program that runs on an IBM PC/AT or compatible. This program exercises both the management port and the Twisted Pair ports (when used in conjunction with the LANCE-AT-KT card) of the repeater and monitors the status of the transmission. The monitor allows the user to assign physical and logical addresses, establish connections, and send and receive data packets.

In addition to the evaluation board and the software diskette, the IMR-VELCRO-HW kit comes complete with a user's manual including schematic diagrams, device specifications, device application notes, and the AMD Ethernet/IEEE-802.3 Family data book.

BLOCK DIAGRAM



11614A-001A

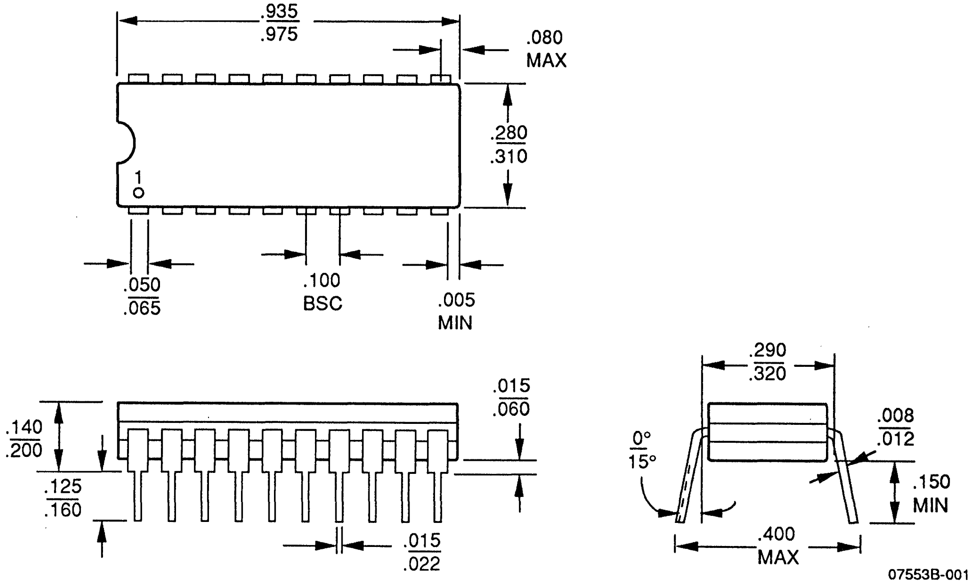
Publication# 16614 Rev. A Amendment/0
Issue Date: January 1992

SECTION 5
Physical Dimensions*

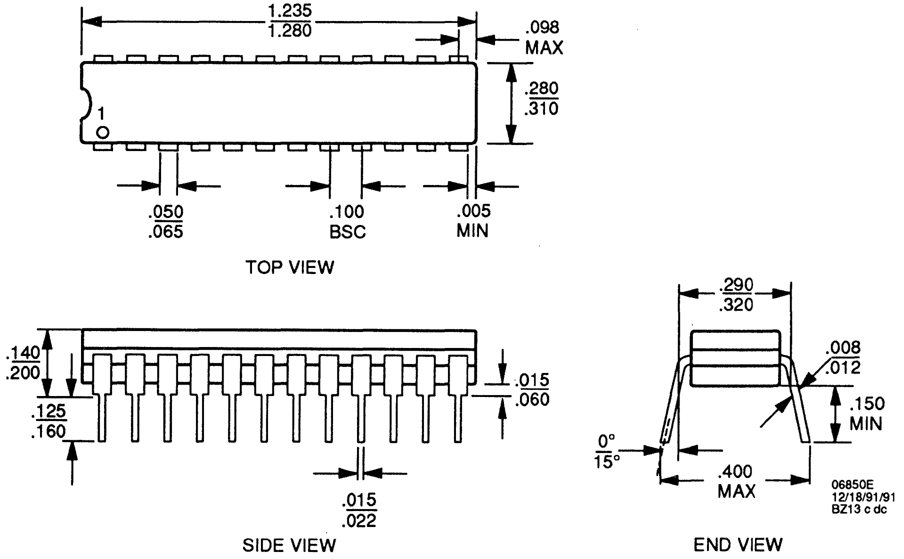
* For reference only. All dimensions are measured inches unless otherwise specified. BSC is an ANSI standard for Basic Space Centering.

Physical Dimensions

CD 020
20-Pin Ceramic DIP



CD3024
24-Pin (Slim) Ceramic DIP

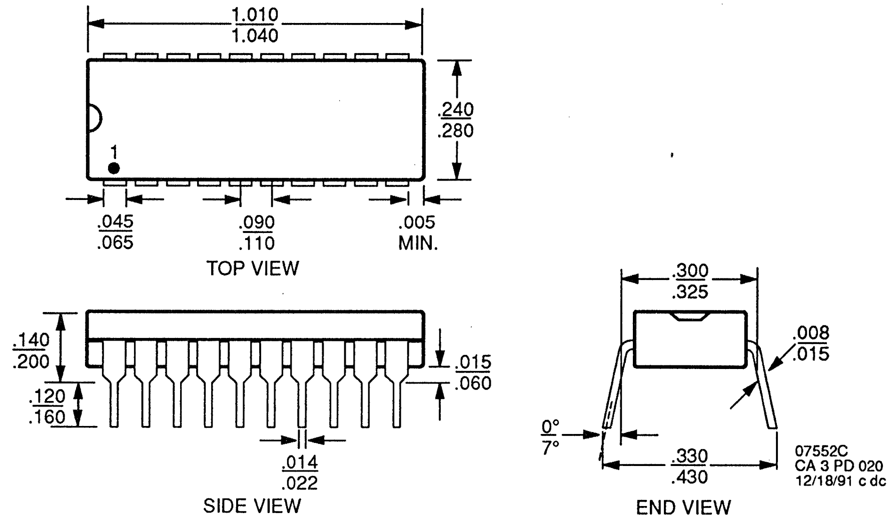


TOP VIEW

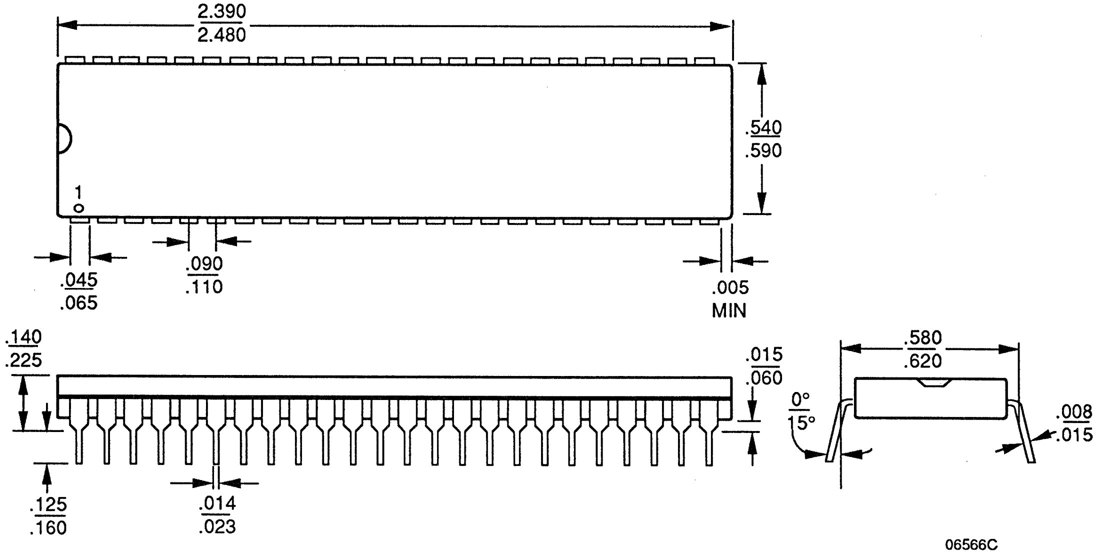
SIDE VIEW

END VIEW

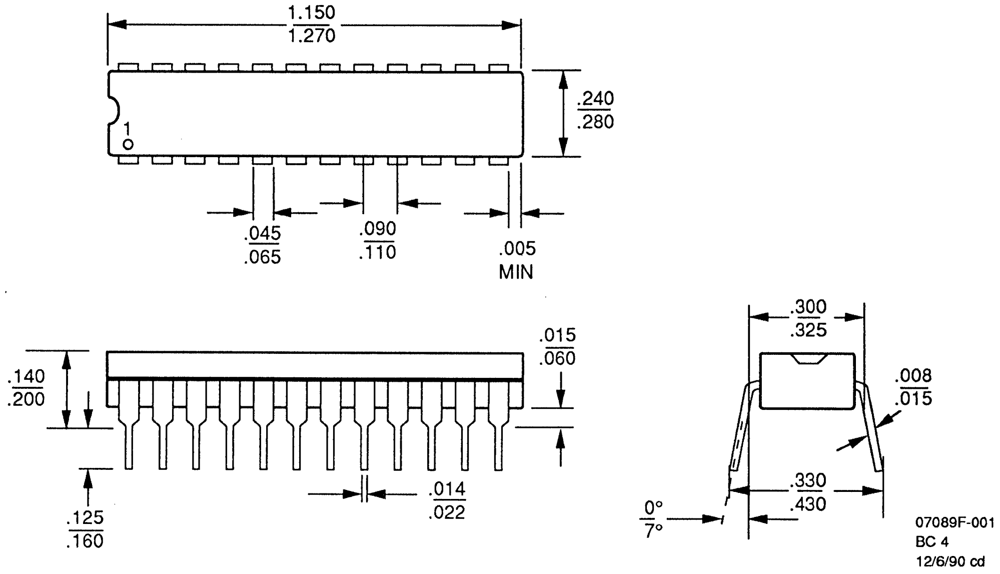
PD 020
20-Pin Plastic DIP



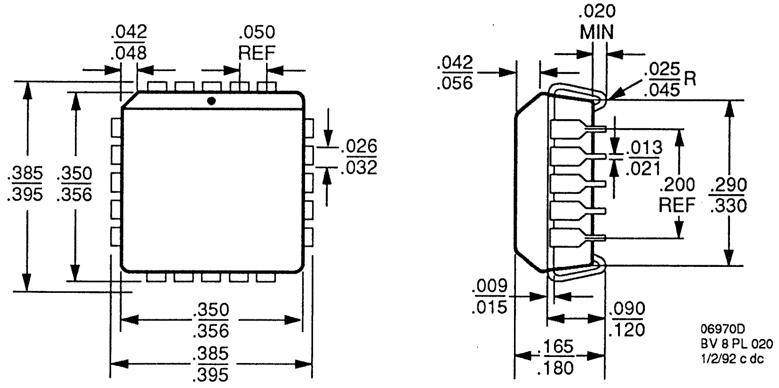
PD 048
48-Pin Plastic DIP



PD3024
24-Pin (Slim) Plastic DIP



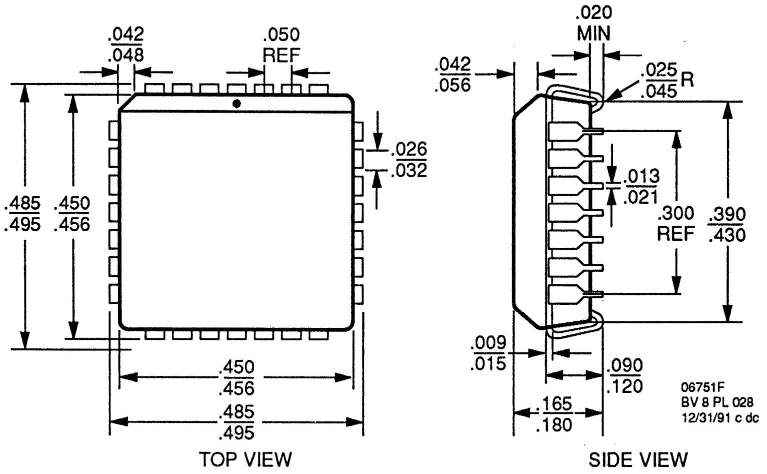
PL 020
20-Pin Plastic Leaded Chip Carrier



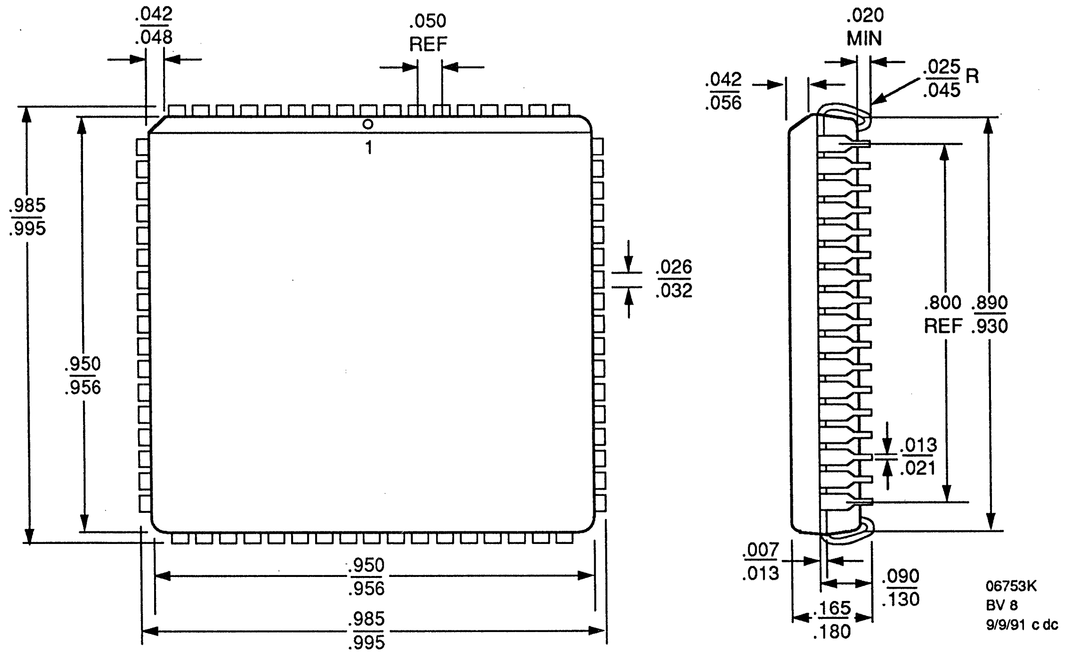
TOP VIEW

SIDE VIEW

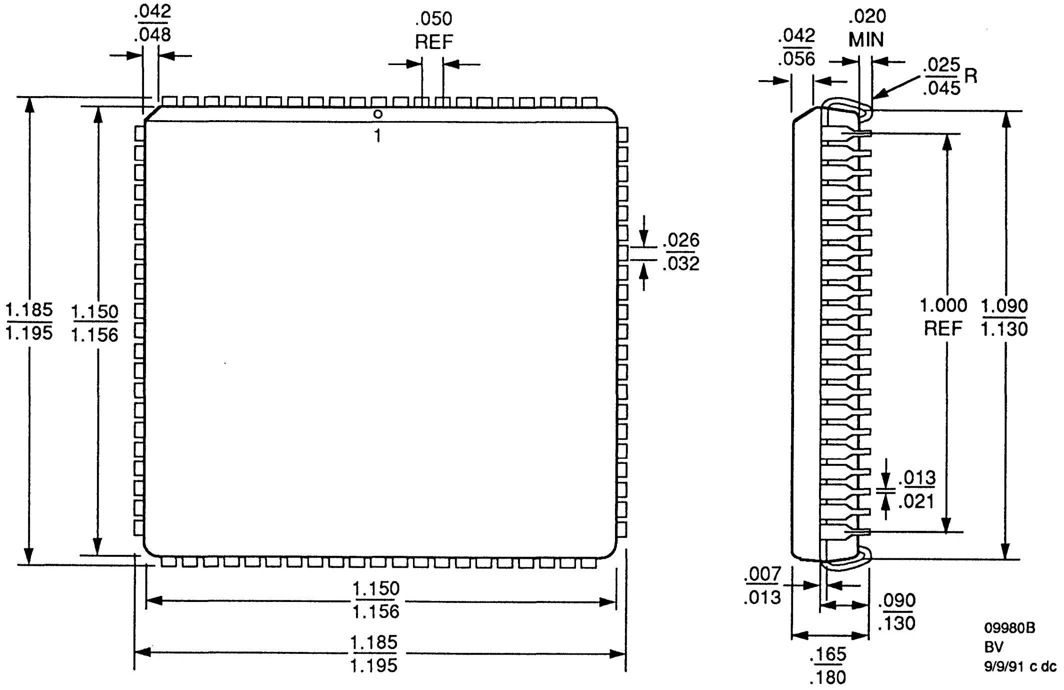
PL 028
28-Pin Plastic Leaded Chip Carrier



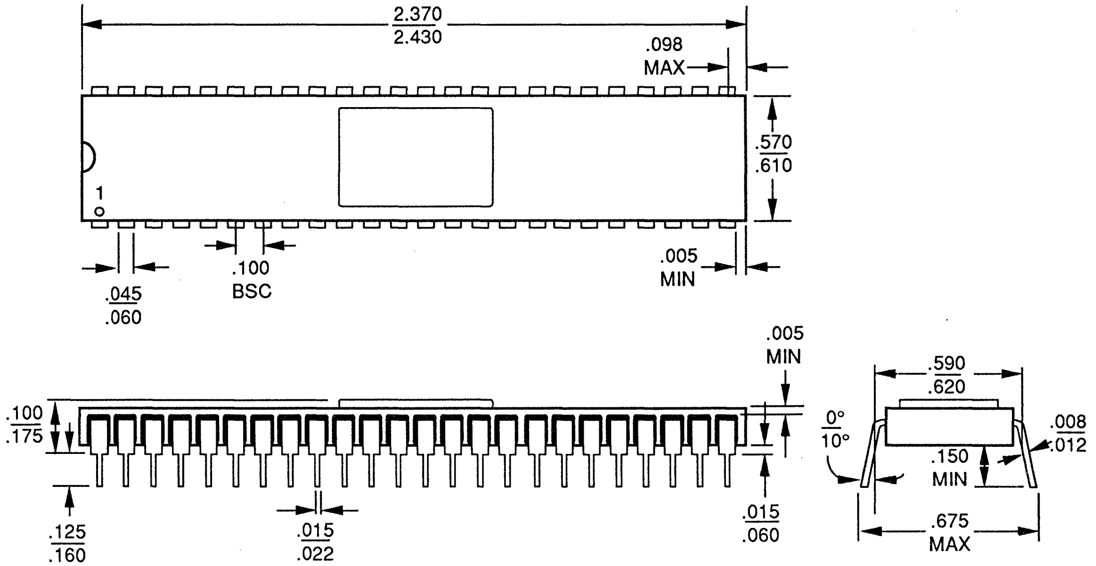
PL 068
68-Pin Plastic Leaded Chip Carrier



PL 084
84-Pin Plastic Leaded Chip Carrier



SD 048
48-Pin Sidebrazed Ceramic DIP



07644C
AW 25
8/14/91 c dc

Sales Offices

North American

ALABAMA	(205) 882-9122
ARIZONA	(602) 242-4400
CALIFORNIA,	
Culver City	(213) 645-1524
Newport Beach	(714) 752-6262
Sacramento(Roseville)	(916) 786-6700
San Diego	(619) 560-7030
San Jose	(408) 452-0500
Woodland Hills	(818) 992-4155
CANADA, Ontario,	
Kanata	(613) 592-0060
Willowdale	(416) 224-5193
COLORADO	(303) 862-9292
CONNECTICUT	(203) 264-7800
FLORIDA,	
Clearwater	(813) 530-9971
Ft. Lauderdale	(305) 776-2001
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,	
Liverpool	(315) 457-5400
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
UTAH	(801) 264-2900

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	(49) 6172-24061
	FAX	(49) 6172-23195
München	TEL	(089) 4114-0
	FAX	(089) 406490
HONG KONG,	TEL	(852) 865-4525
Wanchai	FAX	(852) 865-1147
ITALY, Milano	TEL	(02) 3390541
	FAX	(02) 3498000
JAPAN,		
Atsugi	TEL	(0462) 29-8460
	FAX	(0462) 29-8458
Kanagawa	TEL	(0462) 47-2911
	FAX	(0462) 47-1729
Tokyo	TEL	(03) 3346-7550
	FAX	(03) 3342-5196

International (Continued)

Osaka	TEL	(06) 243-3250
	FAX	(06) 243-3253
KOREA, Seoul	TEL	(82) 2-784-7598
	FAX	(82) 2-784-8014
LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
NORWAY, Oslo area	TEL	(02) 53 13 24
(Hövik)	FAX	(02) 58 22 62
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) 828008
(Warrington)	FAX	(0925) 827693
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	(416) 676-9720
Lachine, Quebec – VITEL ELECTRONICS	(514) 636-5951
ILLINOIS	
Skokie – INDUSTRIAL REPRESENTATIVES,INC	(708) 967-8430
INDIANA	
Huntington – ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3450
Indianapolis – ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3450
IOWA	
LORENZ SALES	(319) 377-4666
KANSAS	
Merriam – LORENZ SALES	(913) 469-1312
Wichita – LORENZ SALES	(316) 721-0500
KENTUCKY	
ELECTRONIC MARKETING CONSULTANTS, INC	(317) 921-3452
MICHIGAN	
Birmingham – MIKE RAICK ASSOCIATES	(313) 644-5040
Holland – COM-TEK SALES, INC	(616) 392-7100
Novi – 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
Columbus – DOLFUSS ROOT & CO	(614) 885-4844
Westlake – DOLFUSS ROOT & CO	(216) 899-9370
OREGON	
ELECTRA TECHNICAL SALES, INC	(503) 643-5074
PENNSYLVANIA	
RUSSELL F. CLARK CO.,INC.	(412) 242-9500
PUERTO RICO	
COMP REP ASSOC, INC	(809) 746-6550
WASHINGTON	
ELECTRA TECHNICAL SALES	(206) 821-7442
WISCONSIN	
Brookfield – INDUSTRIAL REPRESENTATIVES,INC	(414) 789-9393

Advanced Micro Devices reserves the right to make changes in its product without notice in order to improve design or performance characteristics. The performance characteristics listed in this document are guaranteed by specific tests, guard banding, design and other practices common to the industry. For specific testing details, contact your local AMD sales representative. The company assumes no responsibility for the use of any circuits described herein.

RECYCLED & RECYCLABLE

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

1/3/92
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
TOLL-FREE
(800) 538-8450

APPLICATIONS
HOTLINE & LITERATURE
ORDERING
(800) 222-9323
(408) 749-5703

RECYCLED &
RECYCLABLE

Printed in USA
WCP-25M-1/92-0
14287B