Advanced
Micro
Devices

Am7990

Local Area Network Controller for Ethernet (LANCE)

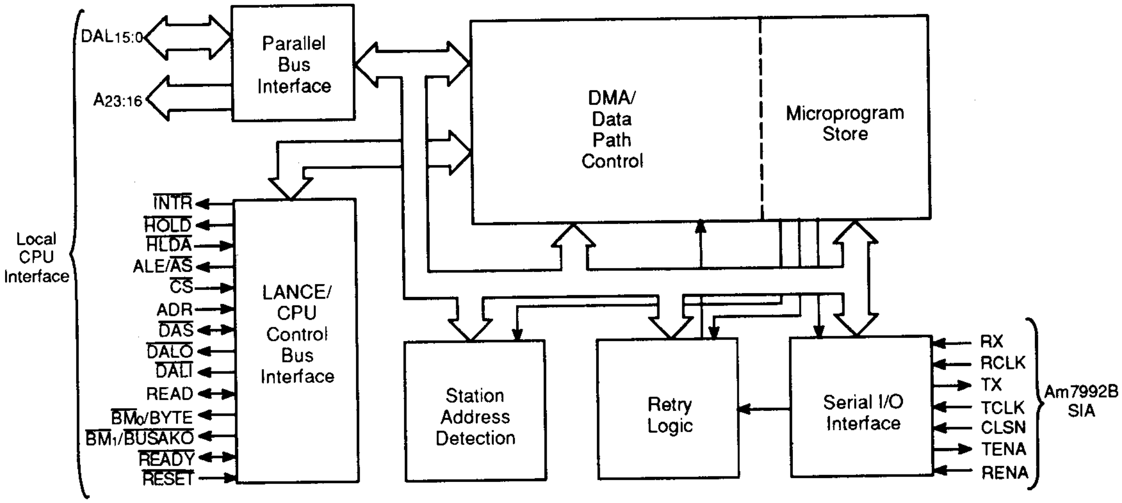
DISTINCTIVE CHARACTERISTICS

- **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


DAL15:0
A23:16
Parallel Bus Interface
DMA/ Data Path Control
Microprogram Store
Local CPU Interface
$\overline{INTR}$
$\overline{HOLD}$
$\overline{HLDA}$
ALE/$\overline{AS}$
$\overline{CS}$
ADR
$\overline{DAS}$
$\overline{DALO}$
$\overline{DALI}$
READ
$\overline{BM_0}$/BYTE
$\overline{BM_1}$/$\overline{BUSAKO}$
$\overline{READY}$
$\overline{RESET}$
LANCE/ CPU Control Bus Interface
Station Address Detection
Retry Logic
Serial I/O Interface
RX
RCLK
TX
TCLK
CLSN
TENA
RENA
Am7992B SIA

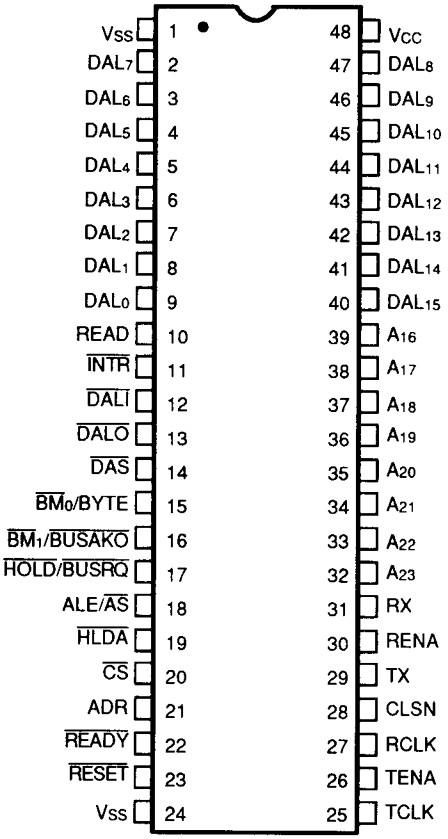
05698D-001A

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

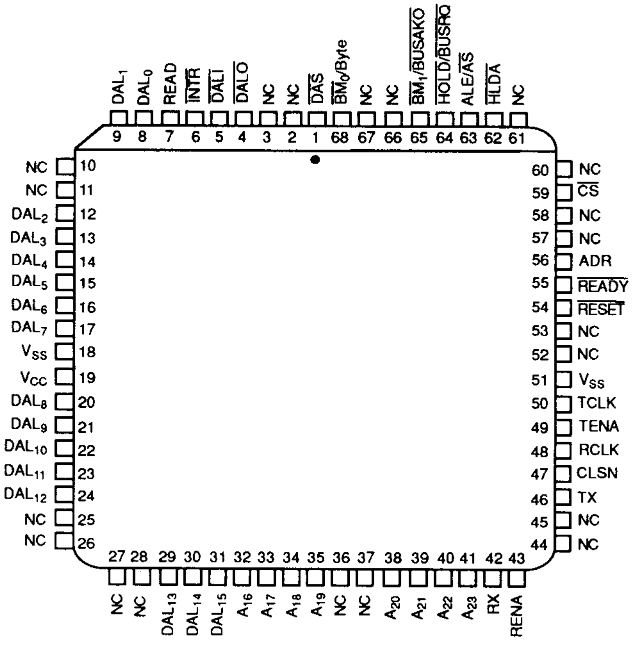
RELATED AMD PRODUCTS

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


Vss 1
DAL7 2
DAL6 3
DAL5 4
DAL4 5
DAL3 6
DAL2 7
DAL1 8
DAL0 9
READ 10
INTR 11
DALI 12
DALO 13
DAS 14
BM0/BYTE 15
BM1/BUSAKO 16
HOLD/BUSRQ 17
ALE/AS 18
HLDA 19
CS 20
ADR 21
READY 22
RESET 23
Vss 24
48 VCC
47 DAL8
46 DAL9
45 DAL10
44 DAL11
43 DAL12
42 DAL13
41 DAL14
40 DAL15
39 A16
38 A17
37 A18
36 A19
35 A20
34 A21
33 A22
32 A23
31 RX
30 RENA
29 TX
28 CLSN
27 RCLK
26 TENA
25 TCLK

05698D-002A

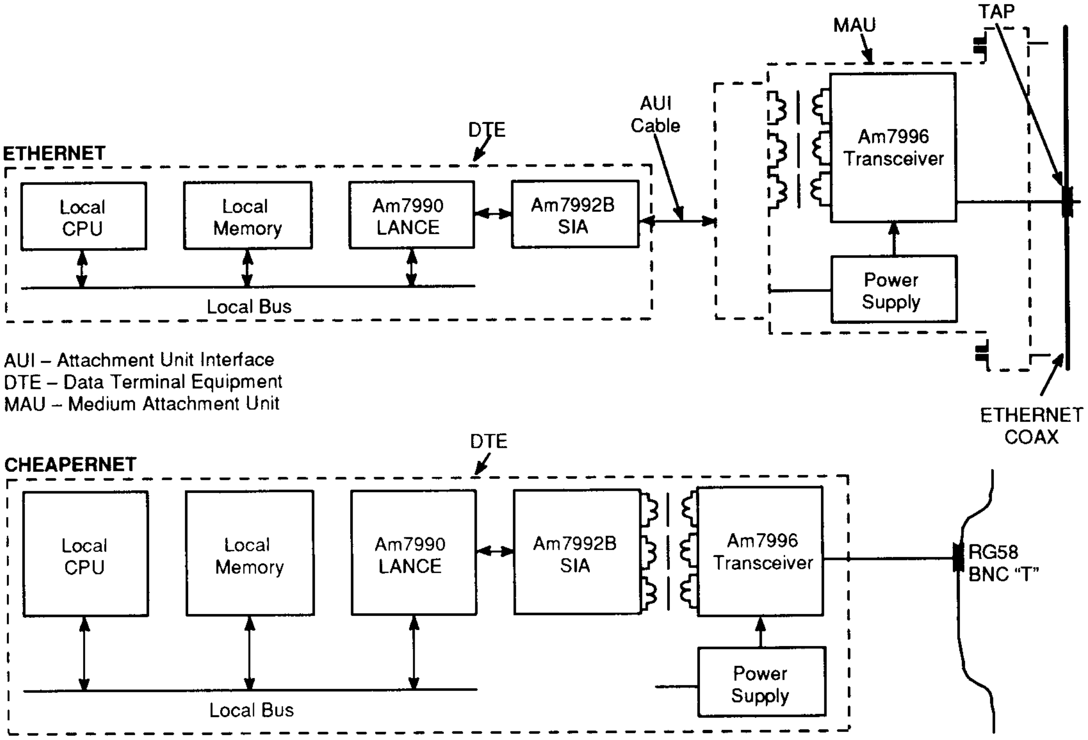

DAL1 9
DAL0 8
READ 7
INTR 6
DALI 5
DALO 4
NC 3
NC 2
DAS 1
BM0/Byte 68
NC 67
NC 66
BM1/BUSAKO 65
HOLD/BUSRQ 64
ALE/AS 63
HLDA 62
NC 61
NC 10
NC 11
DAL2 12
DAL3 13
DAL4 14
DAL5 15
DAL6 16
DAL7 17
VSS 18
VCC 19
DAL8 20
DAL9 21
DAL10 22
DAL11 23
DAL12 24
NC 25
NC 26
60 NC
59 CS
58 NC
57 NC
56 ADR
55 READY
54 RESET
53 NC
52 NC
51 VSS
50 TCLK
49 TENA
48 RCLK
47 CLSN
46 TX
45 NC
44 NC
27 NC
28 NC
29 DAL13
30 DAL14
31 DAL15
32 A16
33 A17
34 A18
35 A19
36 NC
37 NC
38 A20
39 A21
40 A22
41 A23
42 RX
43 RENA

05698D-003A

Note:

Pin 1 is marked for orientation.

TYPICAL ETHERNET/CHEAPERNET NODE


ETHERNET
DTE
AUI Cable
MAU
TAP
Local CPU
Local Memory
Am7990 LANCE
Am7992B SIA
Am7996 Transceiver
Power Supply
Local Bus
ETHERNET COAX
AUI – Attachment Unit Interface
DTE – Data Terminal Equipment
MAU – Medium Attachment Unit
CHEAPERNET
DTE
Local CPU
Local Memory
Am7990 LANCE
Am7992B SIA
Am7996 Transceiver
Power Supply
Local Bus
RG58 BNC "T"

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 $\overline{\text{CS}}$ is LOW.

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

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 $\overline{\text{AS}}$ (CSR$_3$ (01), ACON = 1), the signal pulses LOW during the address portion of the bus transaction. The LOW-to-HIGH transition of $\overline{\text{AS}}$ 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.

$\overline{\text{BM}}_0$/BYTE, $\overline{\text{BM}}_1$/$\overline{\text{BUSAKO}}$

(Output, Three-state)

The two pins are programmable through bit (00) of CSR$_3$

$\overline{\text{BM}}_0$, $\overline{\text{BM}}_1$ — If CSR$_3$ (00) BCON = 0
PIN 15 = $\overline{\text{BM}}_0$ (Output Three-state) (48-Pin DIPs)
PIN 16 = $\overline{\text{BM}}_1$ (Output Three-state) (48-Pin DIPs)

$\overline{\text{BM}}_0$, $\overline{\text{BM}}_1$ (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.

$\overline{\text{BM}}_1$	$\overline{\text{BM}}_0$	
LOW	LOW	Whole Word
LOW	HIGH	Upper Byte
HIGH	LOW	Lower Byte
HIGH	HIGH	None

BYTE, $\overline{\text{BUSAKO}}$ — If CSR$_3$ (00) BCON = 1
PIN 15 = BYTE (Output Three-state) (48-Pin DIPs)
PIN 16 = $\overline{\text{BUSAKO}}$ (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 $\overline{\text{BM}}_0$, $\overline{\text{BM}}_1$).

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

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

$\overline{\text{CS}}$

Chip Select (Input)

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

$\overline{\text{DALI}}$

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

An external bus transceiver control line. $\overline{\text{DALI}}$ 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. $\overline{\text{DALI}}$ is driven only when LANCE is a Bus Master.

$\overline{\text{DALO}}$

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

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

$\overline{\text{DAS}}$

Data Strobe (Input/Output, Three-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.

$\overline{\text{HLDA}}$

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.

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

$\overline{\text{HOLD}}$/$\overline{\text{BUSRQ}}$

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

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

When CSR$_3$ (00) BCON = 1

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

If the LANCE wants to use the bus, it looks at $\overline{\text{HOLD}}$/$\overline{\text{BUSRQ}}$; 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.

$\overline{\text{INTR}}$

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 enabled by bit 06 of CSR0 (INEA = 1). $\overline{\text{INTR}}$ 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.

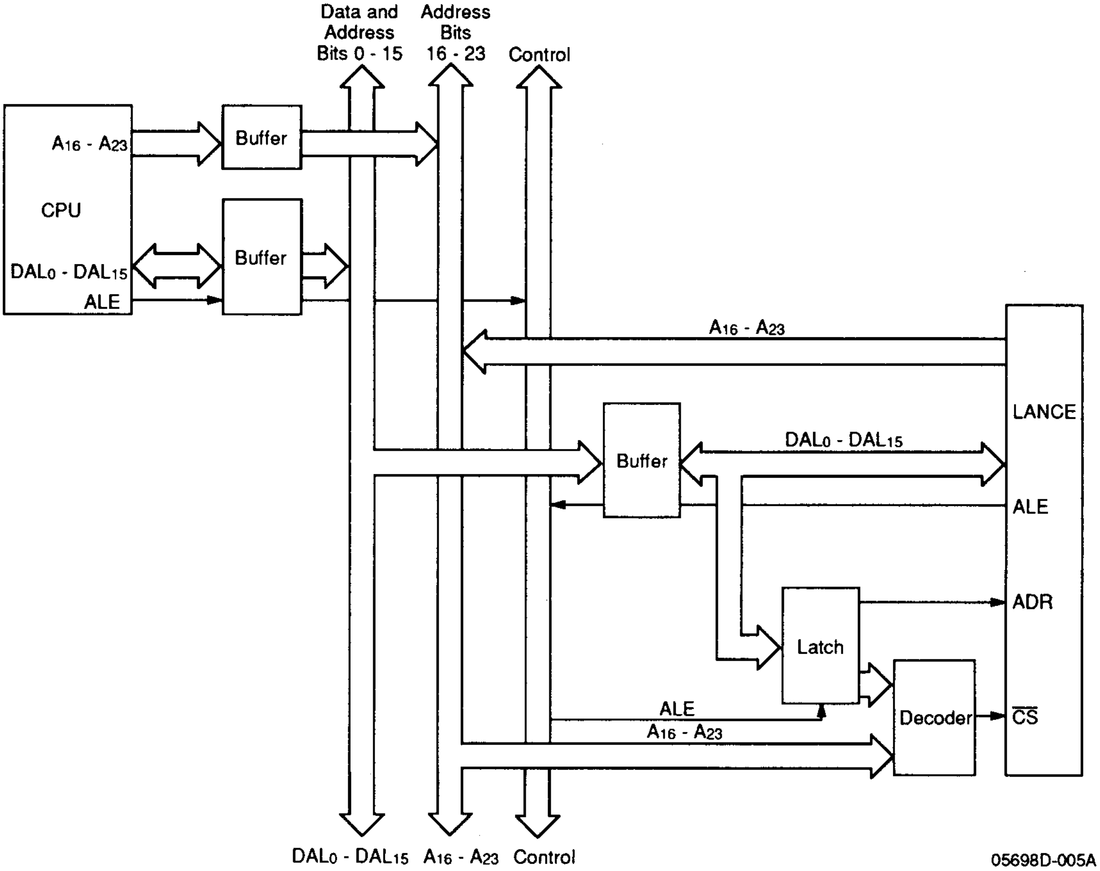

Data and Address Bits 0 - 15
Address Bits 16 - 23
Control
A_{16} - A_{23}
Buffer
CPU
Buffer
DAL_0 - DAL_{15}
ALE
A_{16} - A_{23}
LANCE
Buffer
DAL_0 - DAL_{15}
ALE
ADR
Latch
ALE
A_{16} - A_{23}
Decoder
$\overline{CS}$
DAL_0 - DAL_{15}
A_{16} - A_{23}
Control
05698D-005A

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

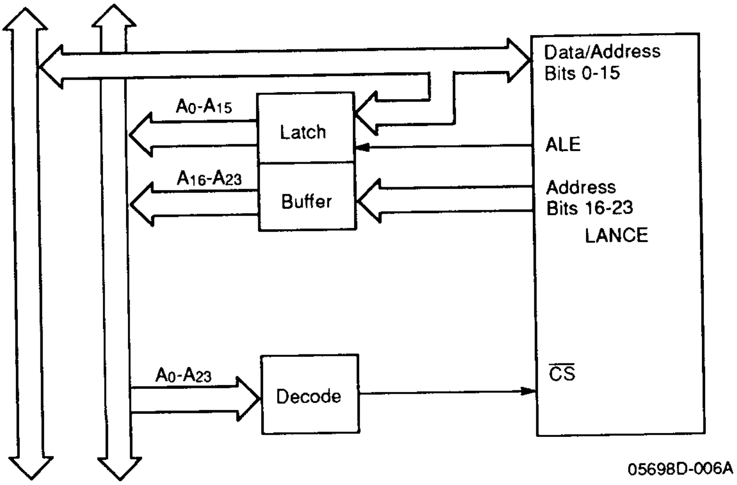


Figure 1-2. LANCE/CPU Interfacing—Demultiplexed Bus

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.

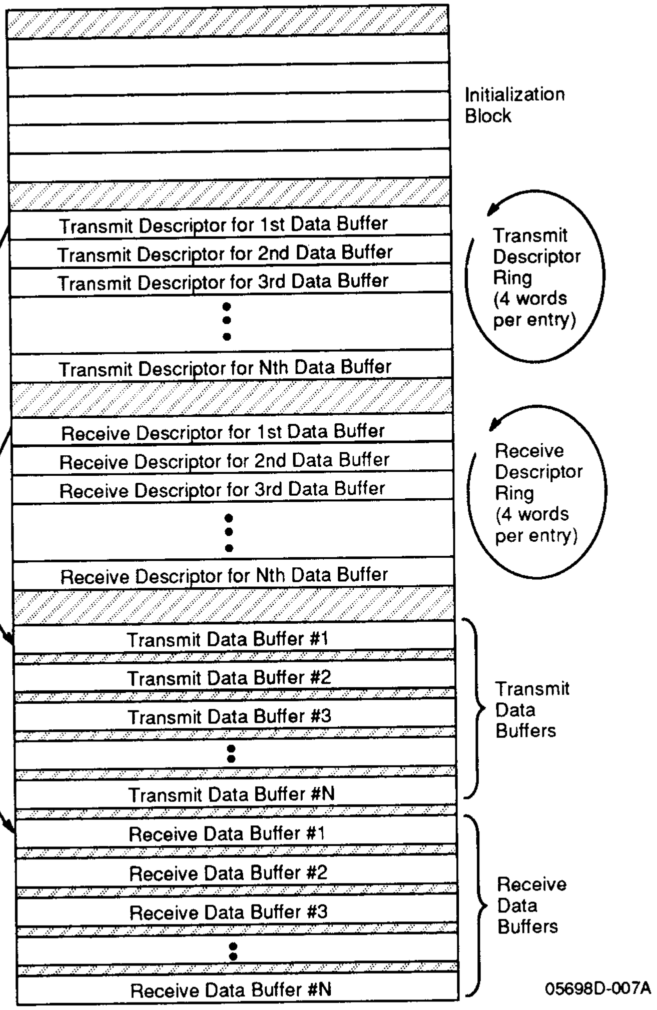

Initialization Block
Transmit Descriptor for 1st Data Buffer
Transmit Descriptor for 2nd Data Buffer
Transmit Descriptor for 3rd Data Buffer
Transmit Descriptor for Nth Data Buffer
Transmit Descriptor Ring (4 words per entry)
Receive Descriptor for 1st Data Buffer
Receive Descriptor for 2nd Data Buffer
Receive Descriptor for 3rd Data Buffer
Receive Descriptor for Nth Data Buffer
Receive 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
Receive Data Buffer #1
Receive Data Buffer #2
Receive Data Buffer #3
Receive Data Buffer #N
Receive Data Buffers
05698D-007A

Figure 2-1. LANCE/Processor Memory Interface

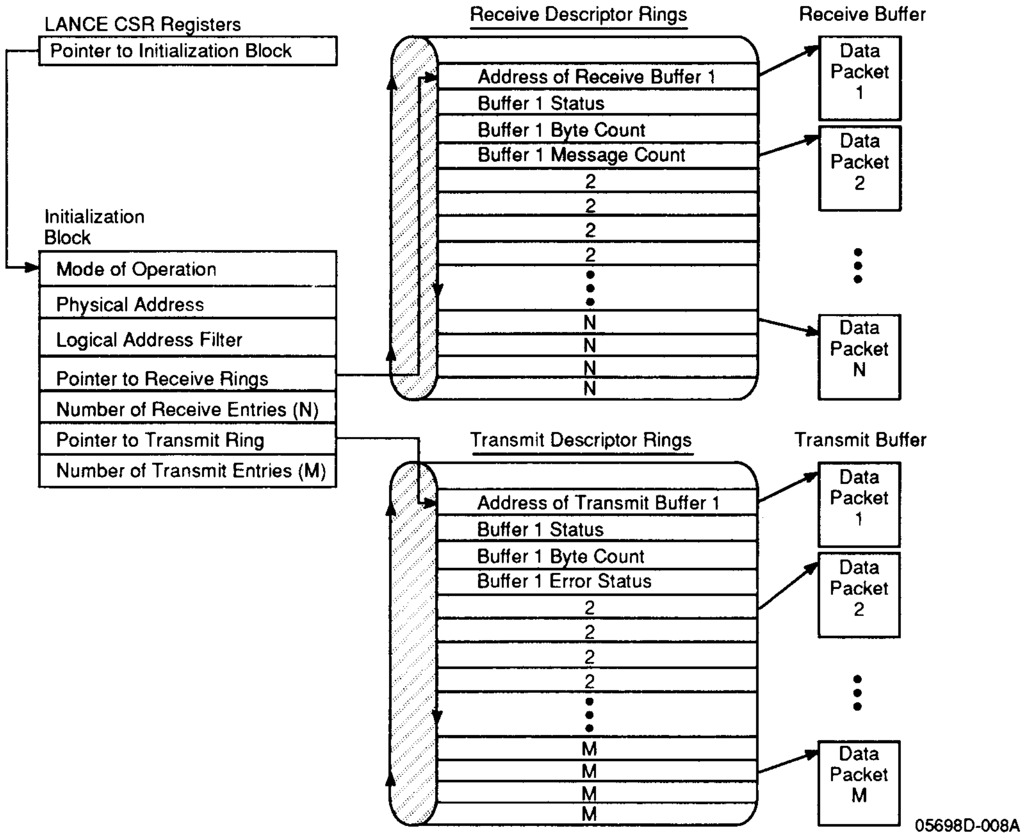

LANCE CSR Registers
Pointer to Initialization Block
Initialization Block
Mode of Operation
Physical Address
Logical Address Filter
Pointer to Receive Rings
Number of Receive Entries (N)
Pointer to Transmit Ring
Number of Transmit Entries (M)
Receive Descriptor Rings
Address of Receive Buffer 1
Buffer 1 Status
Buffer 1 Byte Count
Buffer 1 Message Count
2
2
2
2
N
N
N
N
Receive Buffer
Data Packet 1
Data Packet 2
Data Packet N
Transmit Descriptor Rings
Address of Transmit Buffer 1
Buffer 1 Status
Buffer 1 Byte Count
Buffer 1 Error Status
2
2
2
2
M
M
M
M
Transmit Buffer
Data Packet 1
Data Packet 2
Data Packet M
05698D-008A

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/$\overline{\text{AS}}$).

BCON is used for programming the pins, for handling either the BYTE/$\overline{\text{WORD}}$ 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 ($\overline{\text{BM}}_0$ and $\overline{\text{BM}}_1$) indicate which byte is addressed. When the BYTE scheme is chosen, the $\overline{\text{BM}}_1$ pin can be used for performing the function $\overline{\text{BUSAKO}}$.

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

LANCE in Bus Slave Mode

The LANCE enters the Bus Slave Mode whenever $\overline{\text{CS}}$ 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, $\overline{\text{CS}}$, READ, and $\overline{\text{DAS}}$ 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 $\overline{\text{READY}}$. $\overline{\text{CS}}$, READ, $\overline{\text{DAS}}$, and ADR must remain stable throughout the cycle. Refer to Figure 3.

Write Sequence (Slave Mode)

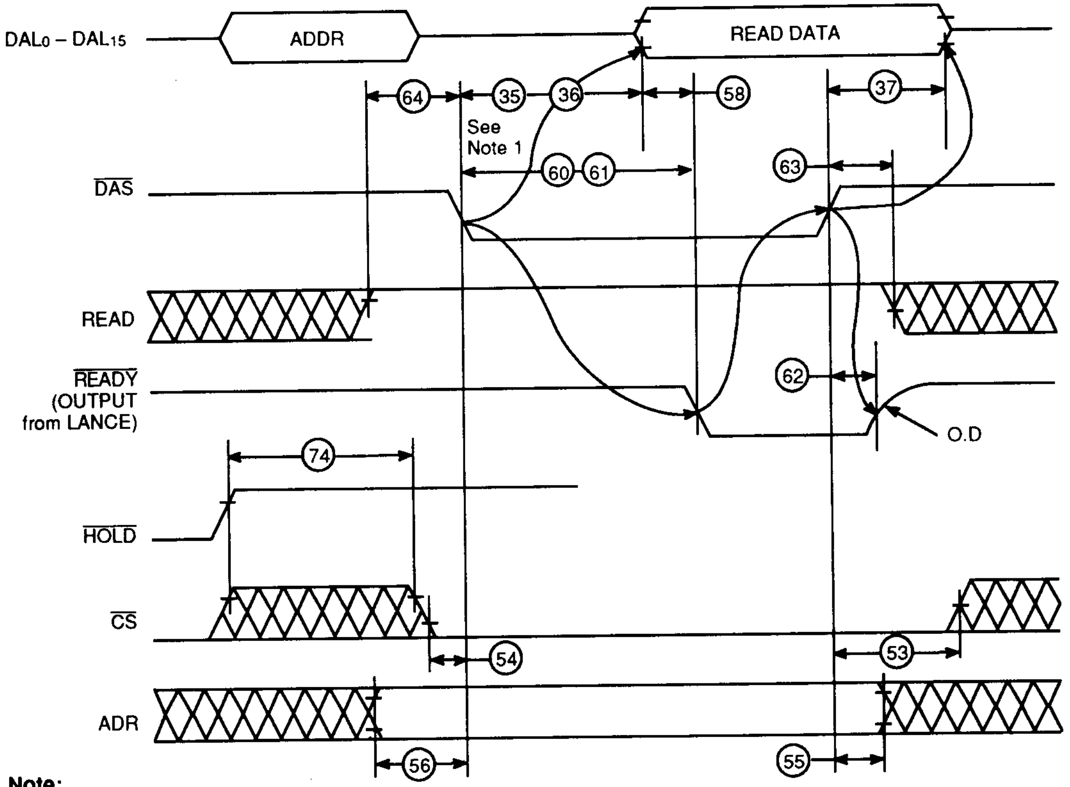
This cycle is similar to the read cycle, except that during this cycle, READ is not asserted (READ is LOW). The DAL buffers are tristated which configures these lines as inputs. The assertion of $\overline{\text{READY}}$ by LANCE indicates to the memory device that the data on the DAL lines have been stored by LANCE in its appropriate CSR register. $\overline{\text{CS}}$, READ, $\overline{\text{DAS}}$, 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 $\overline{\text{READY}}$) to start floating about 100 nsec. after $\overline{\text{READY}}$ goes active. If $\overline{\text{DAS}}$ and $\overline{\text{CS}}$ are held active for more than 400 nsec. after $\overline{\text{READY}}$ start to float, the LANCE can start a second slave cycle. $\overline{\text{DAS}}$ and $\overline{\text{CS}}$ should be deasserted within 400 nsec. after $\overline{\text{READY}}$ 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, $\overline{\text{DAS}}$, 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 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.

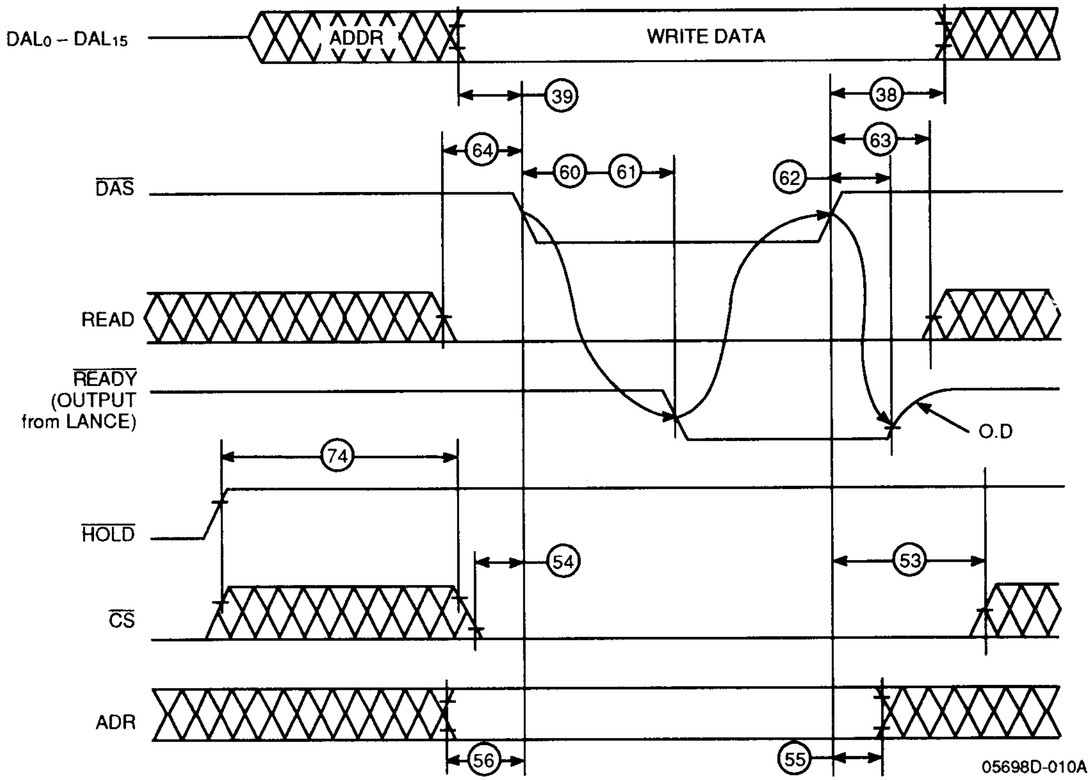


Figure 4. Bus Slave Write Timing

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.

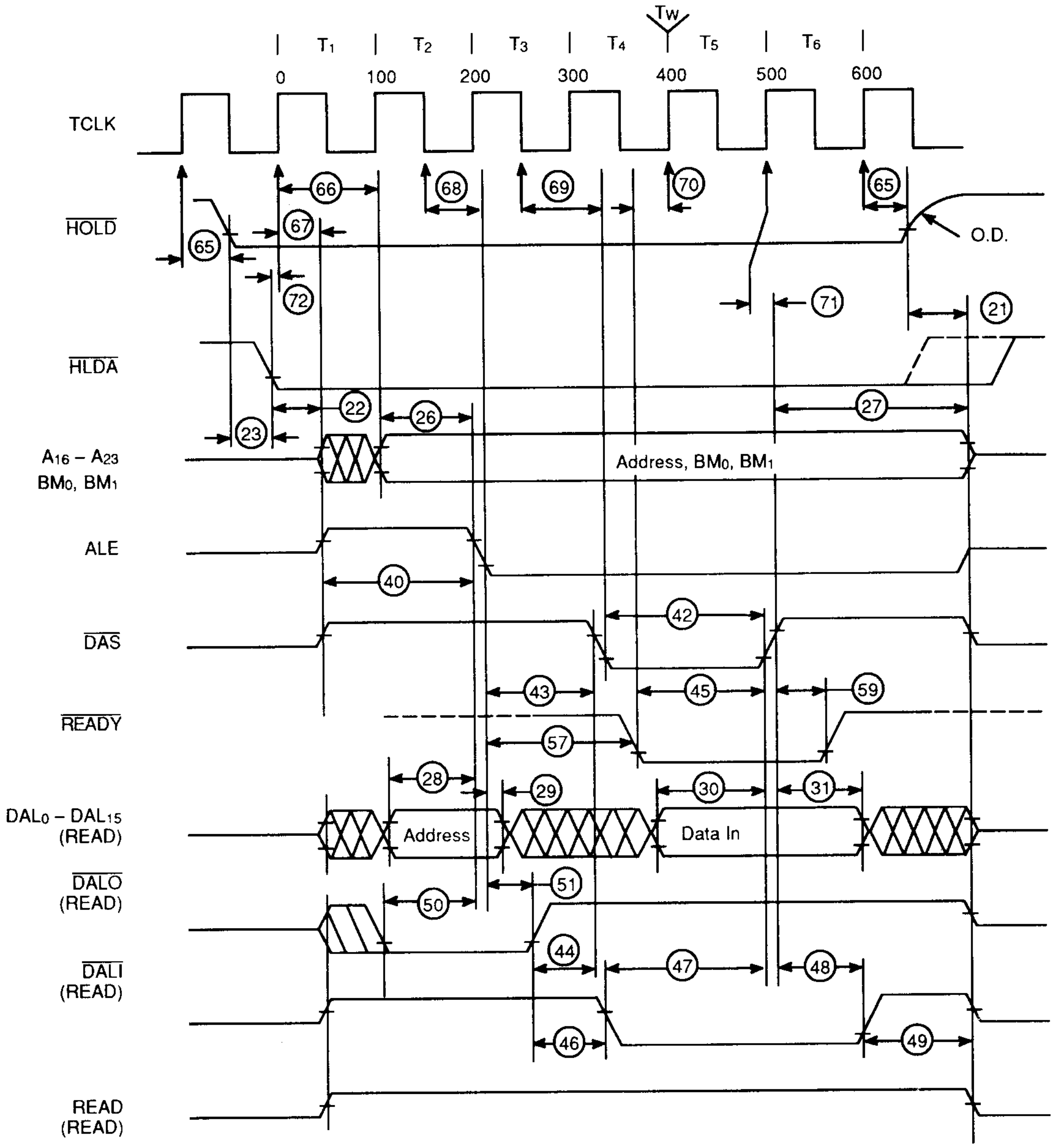
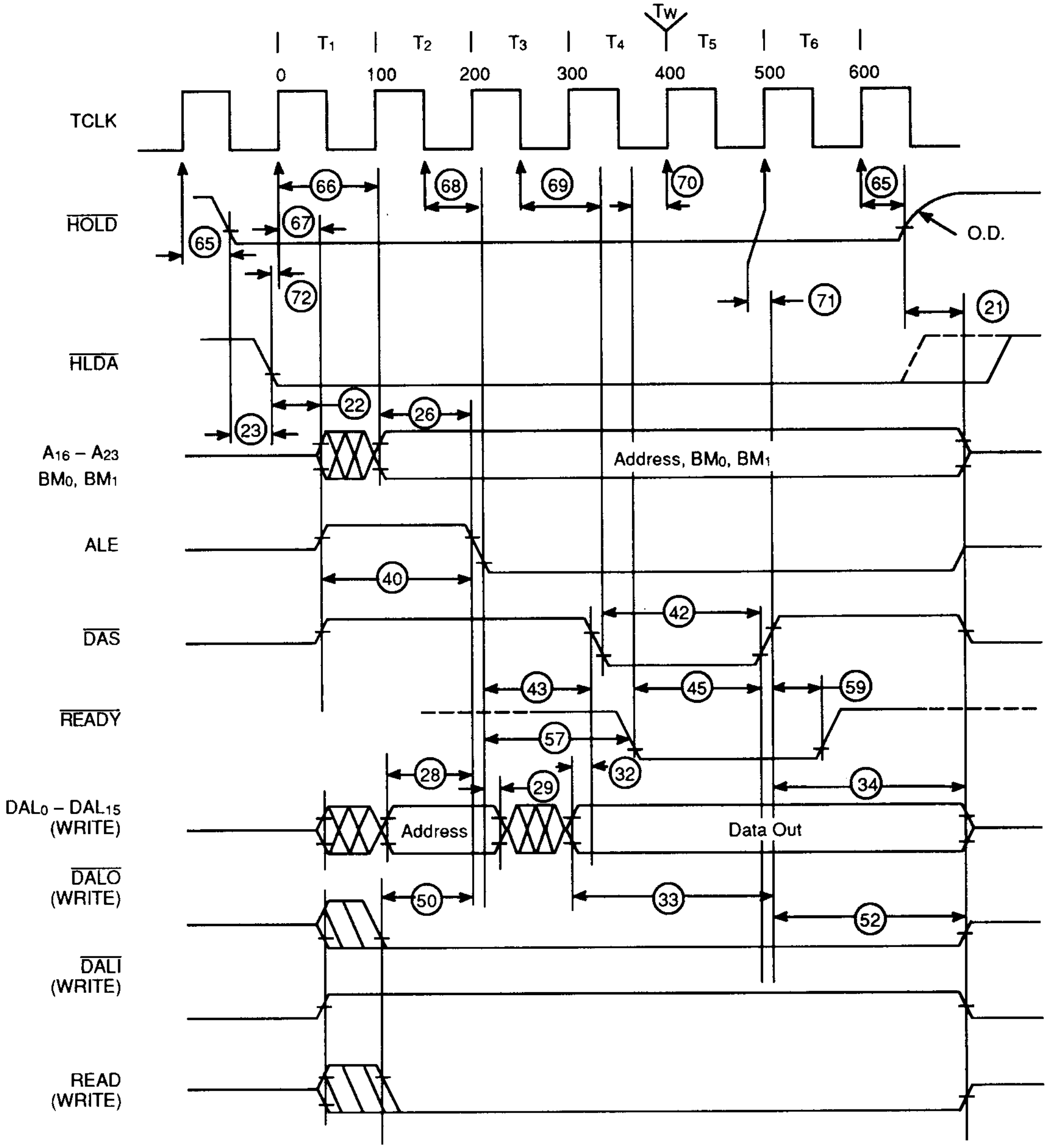
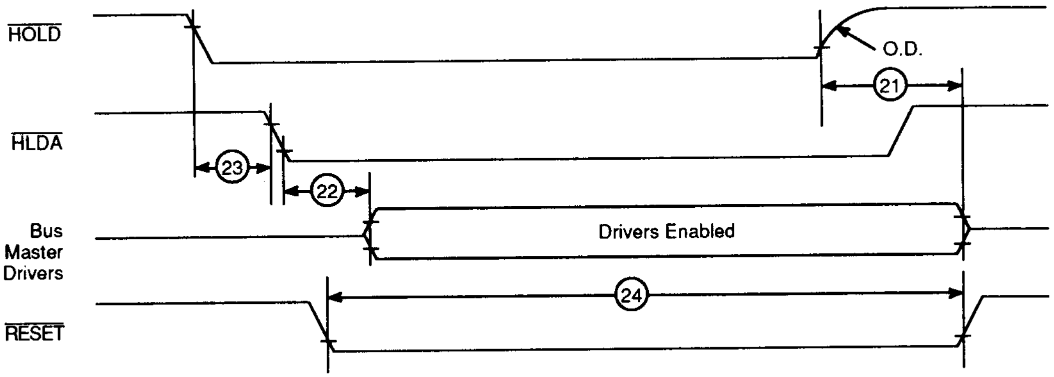

Tw
T1
T2
T3
T4
T5
T6
0
100
200
300
400
500
600
TCLK
HOLD
HLDA
A16 – A23
BM0, BM1
ALE
DAS
READY
DAL0 – DAL15
(READ)
DALO
(READ)
DALI
(READ)
READ
(READ)
O.D.
Address, BM0, BM1
Address
Data In
⑥⑤
⑥⑥
⑥⑦
⑥⑧
⑥⑨
⑦⓪
⑦①
⑦②
㉑
㉒
㉓
㉖
㉗
㉘
㉙
㉚
㉛
㊵
㊷
㊸
㊹
㊺
㊻
㊼
㊽
㊾
(50)
(51)
(57)
(59)
05698D-011A

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


Tw
T1
T2
T3
T4
T5
T6
0
100
200
300
400
500
600
TCLK
HOLD
HLDA
A16 – A23
BM0, BM1
ALE
DAS
READY
DAL0 – DAL15
(WRITE)
DALO
(WRITE)
DALI
(WRITE)
READ
(WRITE)
Address, BM0, BM1
Address
Data Out
O.D.
65
66
67
68
69
70
65
72
71
21
22
23
26
40
42
43
45
59
57
28
29
32
34
50
33
52

05698D-012A

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



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)

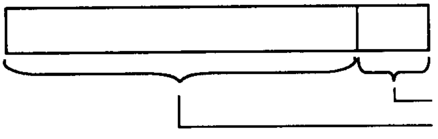
Register Data Port (RDP)

15 … 0
CSR DATA

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)



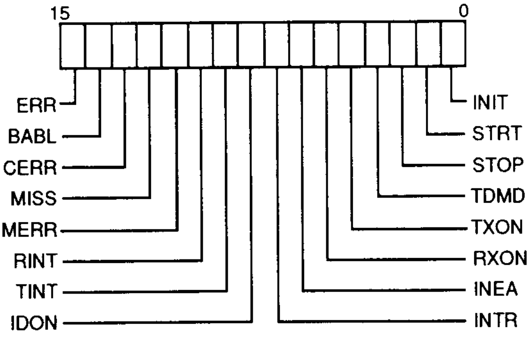
05698D-015A

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 $\overline{RESET}$.

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 $\overline{RESET}$, 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 $\overline{RESET}$ 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 $\overline{RESET}$ 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 $\overline{RESET}$ or by setting the STOP bit.

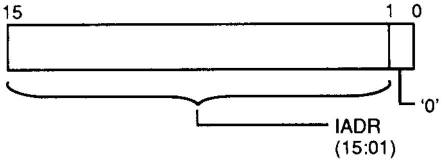
Bit	Name	Description
11	MERR	MEMORY ERROR is set when the LANCE is the Bus Master and has not received $\overline{\text{READY}}$ within 25.6 µs after asserting the address on the DAL lines. When a Memory Error is detected, the receiver and transmitter are turned off (CSR$_0$, TXON = 0, RXON = 0) and 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 $\overline{\text{RESET}}$ 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 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 $\overline{\text{RESET}}$ 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 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 $\overline{\text{RESET}}$ or by setting the STOP bit.
07	INTR	INTERRUPT FLAG is set by the "ORing" of BABL, MISS, MERR, RINT, TINT and IDON. If INEA = 1 and INTR = 1, the $\overline{\text{INTR}}$ pin will be LOW. INTR is READ ONLY; writing this bit has no effect. INTR is cleared by RESET, by setting the STOP bit, or by clearing the condition causing the interrupt.
06	INEA	INTERRUPT ENABLE allows the $\overline{\text{INTR}}$ pin to be driven LOW when the Interrupt Flag is set. If INEA = 1 and INTR = 1, the $\overline{\text{INTR}}$ pin will be Low. If INEA = 0, the $\overline{\text{INTR}}$ pin will be HIGH, regardless of the state of the Interrupt Flag. INEA is READ/WRITE and cleared by RESET or by setting the STOP bit. INEA 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 $\overline{\text{RESET}}$ 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.

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.

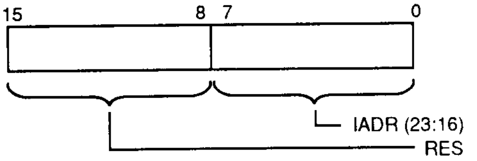


05698D-017A

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.



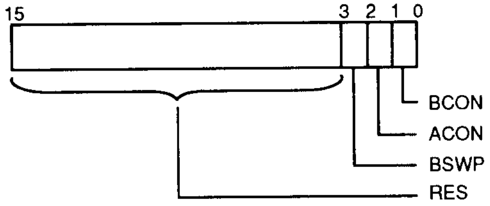
05698D-018A

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 $\overline{RESET}$ 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 $\overline{RESET}$ 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 $\overline{RESET}$ 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 $\overline{HOLD}$ 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 $\overline{RESET}$ or by setting the STOP bit in CSR_0.

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

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.

Initialization

Initialization Block

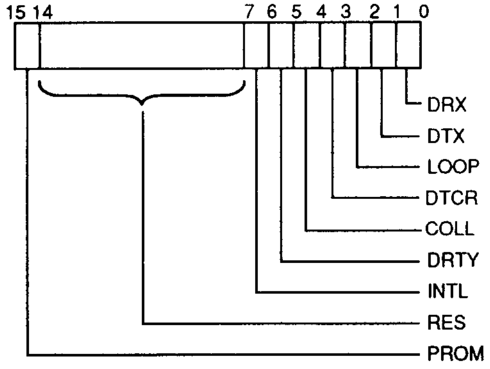
Chip initialization includes the reading of the initialization block in memory to obtain the operating parameters. The following is a definition of the Initialization Block.

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

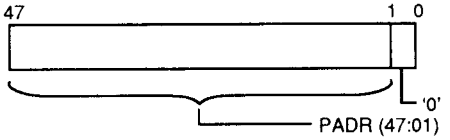
The Mode Register allows alteration of the LANCE's operating parameters. Normal operation is with the Mode Register clear.



05698D-020A

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>15</td><td>PROM</td><td>PROMISCUOUS mode. When PROM = 1, all incoming packets are accepted.</td></tr>
<tr><td>14:07</td><td>RES</td><td>RESERVED. Read as zeroes. Write as zeroes.</td></tr>
<tr><td>06</td><td>INTL</td><td>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.
<table>
<tr><th>LOOP</th><th>INTL</th><th>LOOPBACK</th></tr>
<tr><td>0</td><td>X</td><td>No loopback, normal</td></tr>
<tr><td>1</td><td>0</td><td>External</td></tr>
<tr><td>1</td><td>1</td><td>Internal</td></tr>
</table></td></tr>
<tr><td>05</td><td>DRTY</td><td>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).</td></tr>
</table>

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.

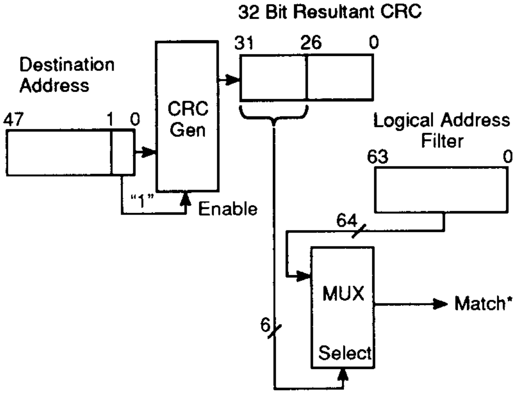
The task of mapping a logical address to one of 64-bit positions requires a simple computer program (see Appendix A) which uses the same CRC algorithm (used in LANCE and defined per Ethernet) to calculate the HASH (see Figure 7).

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.

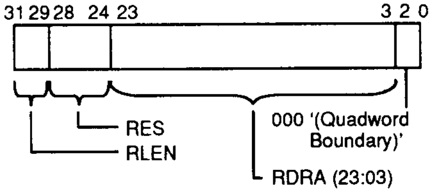


*Match - 1, the packet is accepted
Match - 0, the packet is rejected

05698D-023A

Figure 7. Logical Address Filter Operation

Receive Descriptor Ring Pointer



05698D-024A

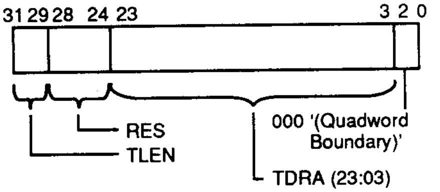
Bit	Name	Description
31:29	RLEN	RECEIVE RING LENGTH is the number of entries in the receive ring expressed as a power of two.

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

Bit	Name	Description
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

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.

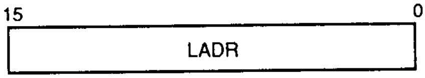
The LANCE communicates with a HOST device through the ring structures in memory. Each entry in the ring is either owned by the LANCE or the HOST. There is an ownership bit (OWN) in the message descriptor entry. Mutual exclusion is accomplished by a protocol which states that each device can only relinquish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

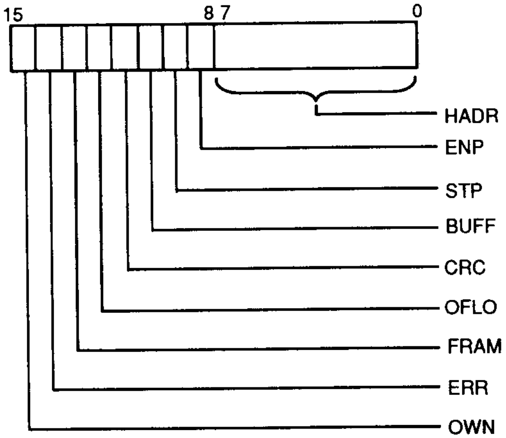
Receive Message Descriptor 0 (RMD_0)



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.

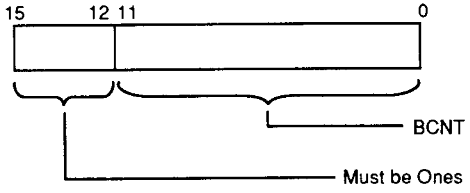
Receive Message Descriptor 1 (RMD_1)



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.

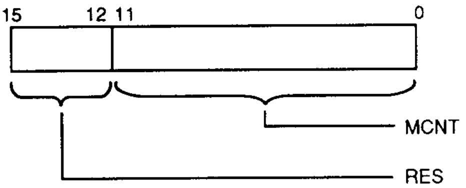
Receive Message Descriptor 2 (RMD_2)



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 (RMD3)

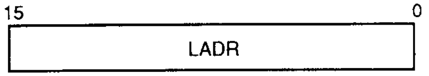


05698D-029A

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

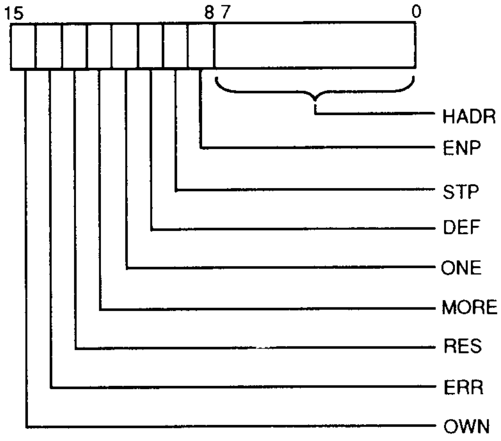
Transmit Message Descriptor 0 (TMD0)



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 (TMD1)

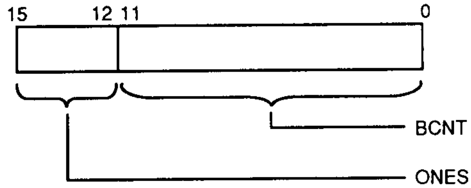


05698D-031A

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.

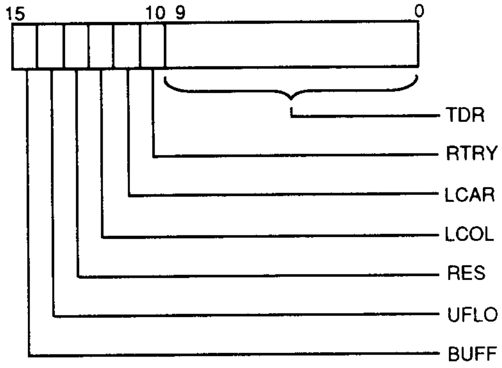
Transmit Message Descriptor 2 (TMD_2)



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).

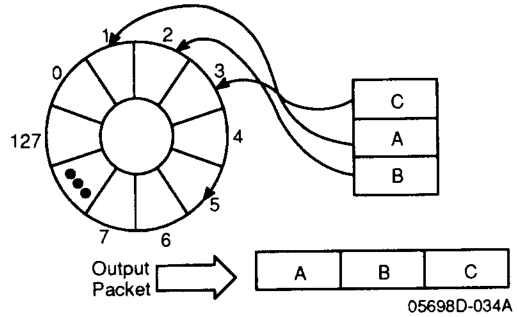
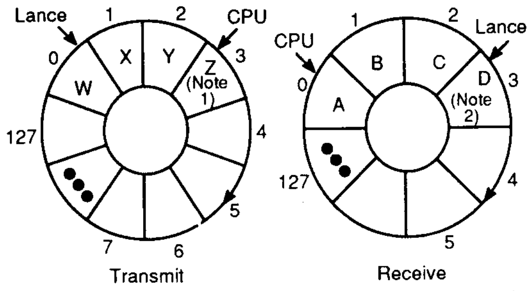


Figure 8-1. Data Chaining (Transmit)



Notes:

1. W, X, Y, Z are the packets queued for transmission.
2. A, B, C, D are the packets received by the 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 ($\overline{HOLD}$ 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 TMD$_3$, 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

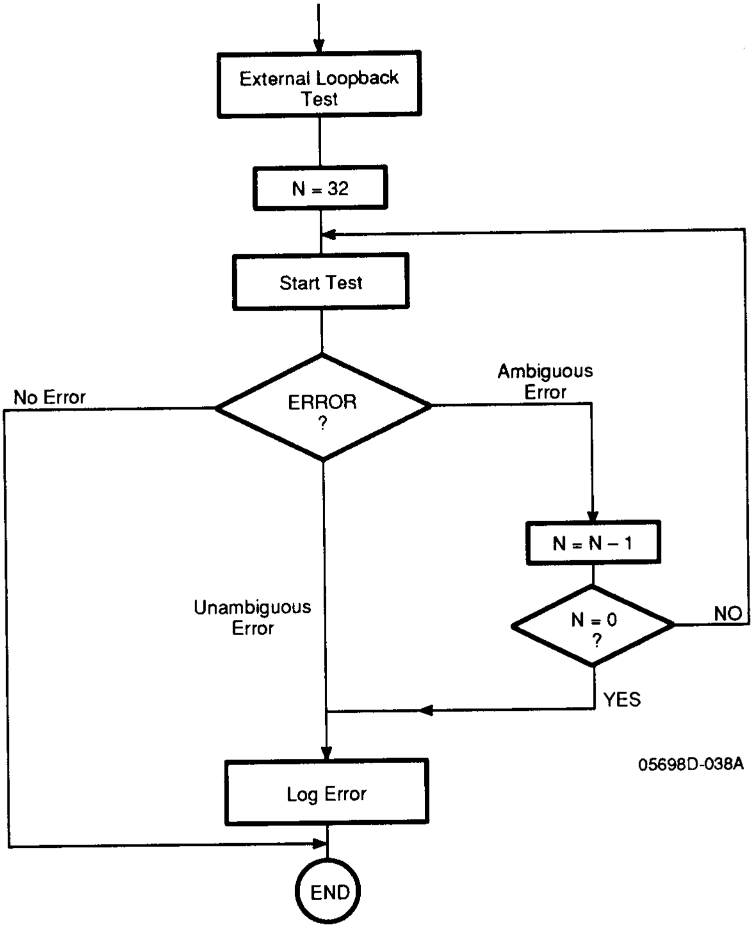
The normal operation of the LANCE is as a half-duplex device. However, to provide an on-line operational test of the LANCE, a pseudo-full duplex mode is provided. In this mode simultaneous transmission and reception of a loopback packet are enabled with the following constraints:

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.


External Loopback Test
N = 32
Start Test
ERROR ?
No Error
Ambiguous Error
N = N − 1
N = 0 ?
NO
YES
Unambiguous Error
Log Error
END
05698D-038A

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)

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

Program example in BASIC to generate the hash filter, for multicast addressing, in the 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 00 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)

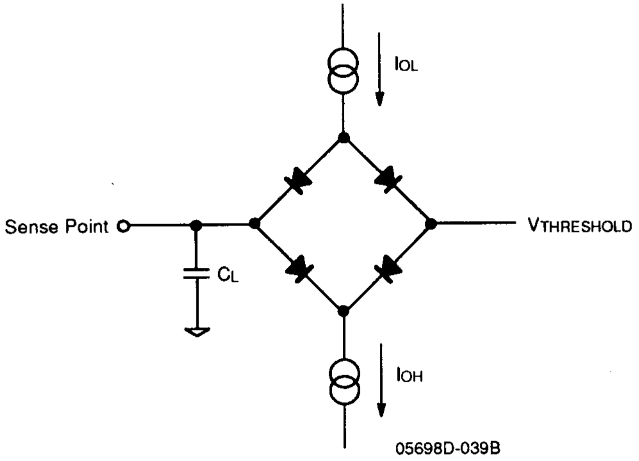
No.	Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Unit
31	t_{RDAH}	Data Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Master Read)		0			ns
32	t_{DDAS}	Data Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (Bus Master Write)		10			ns
33	t_{WDS}	Data Setup 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)

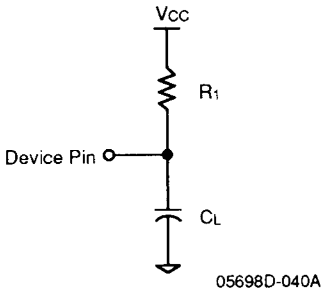
No.	Parameter Symbol	Parameter Description	Test Conditions	Min.	Typ.	Max.	Units
55	t_{SAH}	ADR Hold Time After the Rising Edge of $\overline{\text{DAS}}$ (Bus Slave)		0			ns
56	t_{SAS}	ADR Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (Bus Slave)		0			ns
57	t_{ARYD}	Delay from the Falling 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}.


IOL
Sense Point
CL
VTHRESHOLD
IOH
05698D-039B

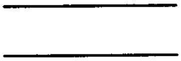
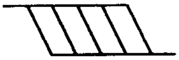
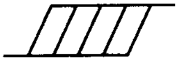
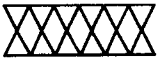
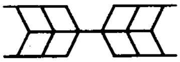
A. Normal & Three-State Outputs


VCC
R1
Device Pin
CL
05698D-040A

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

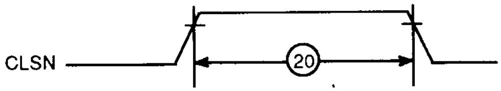
TEST OUTPUT LOADS			
Pin Name	**Test Circuit**	**R1 (kΩ)**	**CL (pF)**
All Outputs and I/O Pins except $\overline{\text{INTR}}$, $\overline{\text{HOLD}}$/$\overline{\text{BUSRQ}}$, $\overline{\text{READY}}$	A	–	100
$\overline{\text{INTR}}$, $\overline{\text{HOLD}}$/$\overline{\text{BUSRQ}}$, $\overline{\text{READY}}$	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

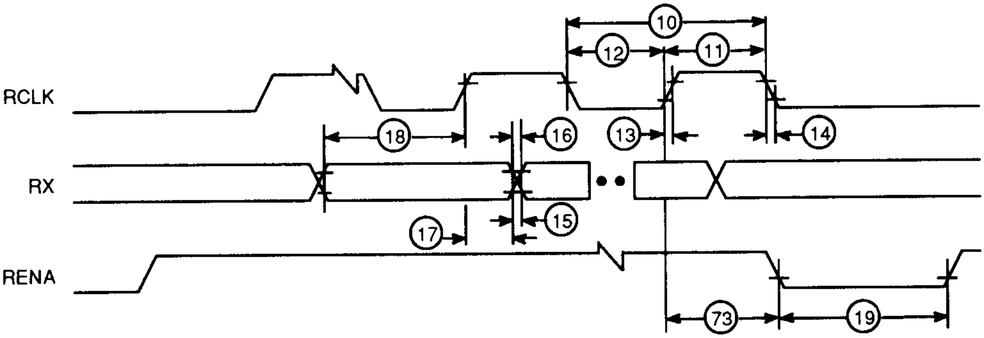
KS000010

SWITCHING WAVEFORMS (Note 1)



Serial Link Timing (Collision)

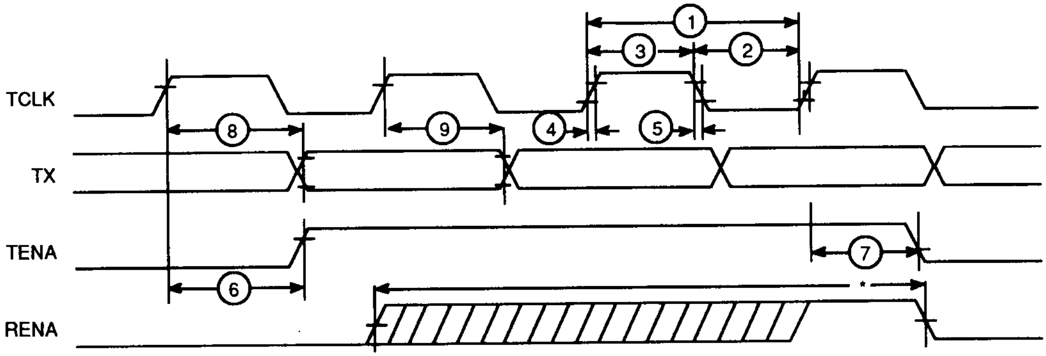
05698D-041A



05698D-042A

Serial Link Timing (Receive)

SWITCHING WAVEFORMS

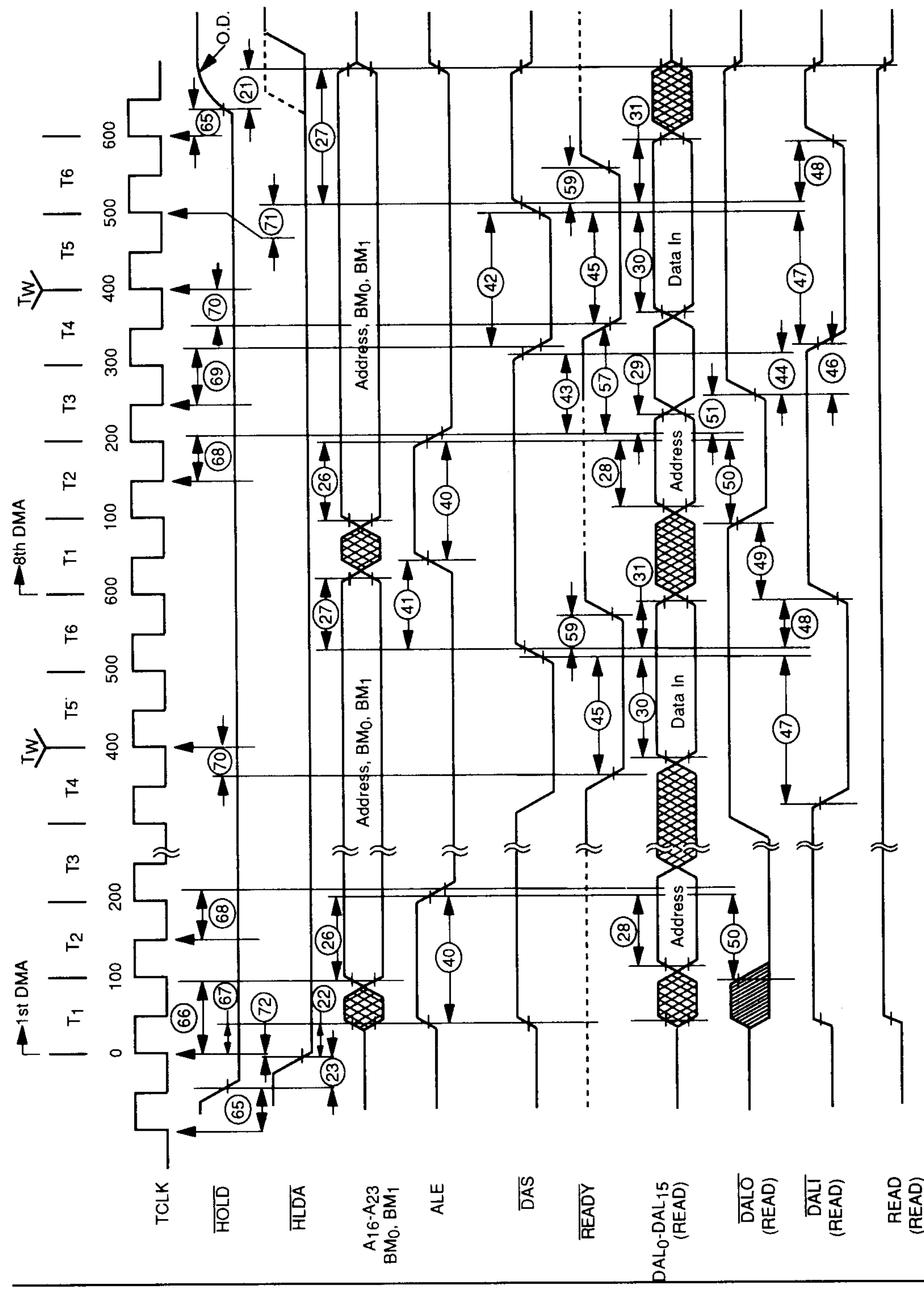


*During transmit, RENA input must be asserted (HIGH) and remain active-HIGH before TENA goes inactive (LOW). If RENA is deasserted before TENA is deasserted, LCAR will be reported in TMD3 after the transmission is completed by the LANCE.

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

Serial Link Timing (Transmit)

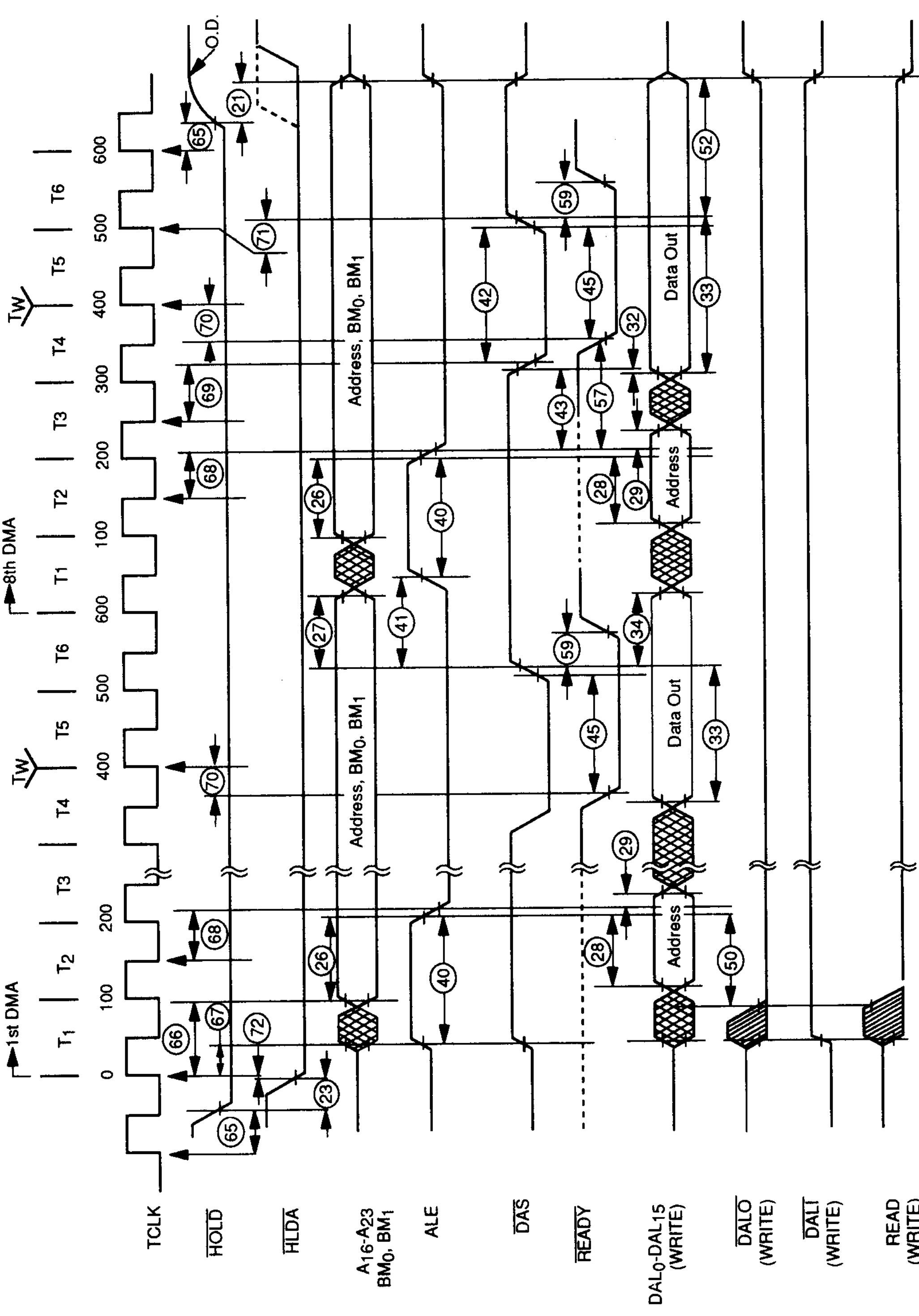
SWITCHING WAVEFORMS


TCLK
HOLD
HLDA
A16-A23 BM0, BM1
ALE
DAS
READY
DAL0-DAL15 (READ)
DALO (READ)
DALI (READ)
READ (READ)
1st DMA
8th DMA
T1
T2
T3
T4
TW
T5
T6
0
100
200
300
400
500
600
O.D.
Address, BM0, BM1
Address
Data In
21
22
23
26
27
28
29
30
31
40
41
42
43
44
45
46
47
48
49
50
51
57
59
65
66
67
68
69
70
71
72

Bus Master Read Timing (Burst DMA)

05698D-044A

SWITCHING WAVEFORMS


TCLK
HOLD
HLDA
A16-A23 BM0, BM1
ALE
DAS
READY
DAL0-DAL15 (WRITE)
DALO (WRITE)
DALI (WRITE)
READ (WRITE)
1st DMA
8th DMA
T1
T2
T3
T4
T5
T6
TW
0
100
200
300
400
500
600
O.D.
Address, BM0, BM1
Address
Data Out
21
23
26
27
28
29
32
33
34
40
41
42
43
45
50
52
57
59
65
66
67
68
69
70
71
72

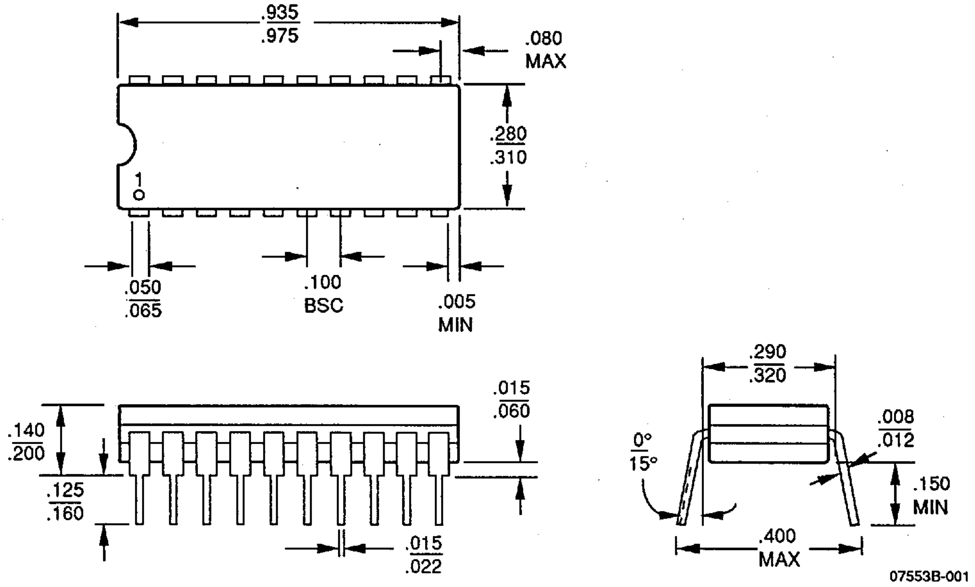
Bus Master Write Timing (Burst DMA)

05698D-045A

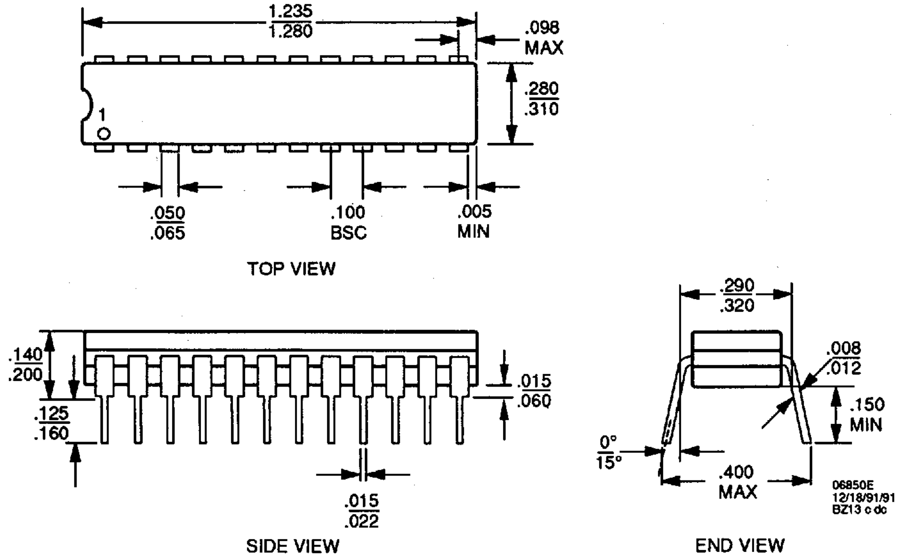
T-90-20

Physical Dimensions

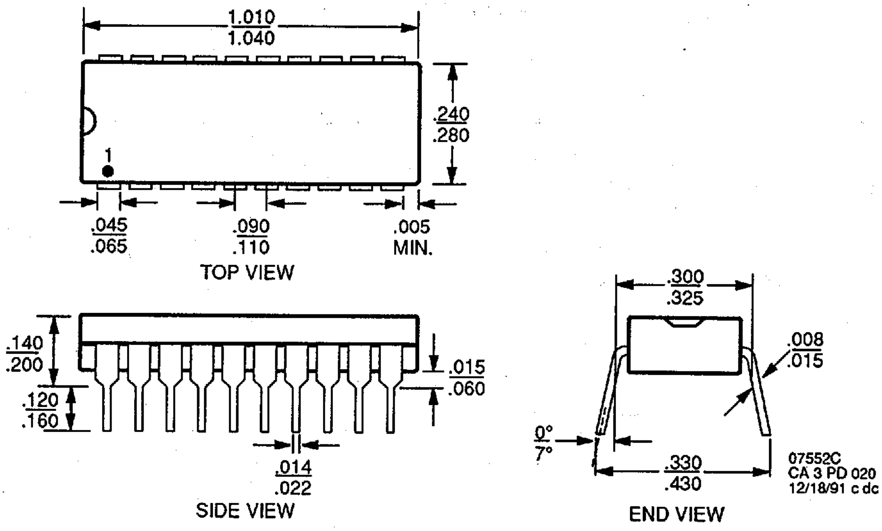
CD 020
20-Pin Ceramic DIP


.935
.975
.080
MAX
.280
.310
1
.050
.065
.100
BSC
.005
MIN
.140
.200
.125
.160
.015
.060
.015
.022
.290
.320
0°
15°
.008
.012
.150
MIN
.400
MAX
07553B-001

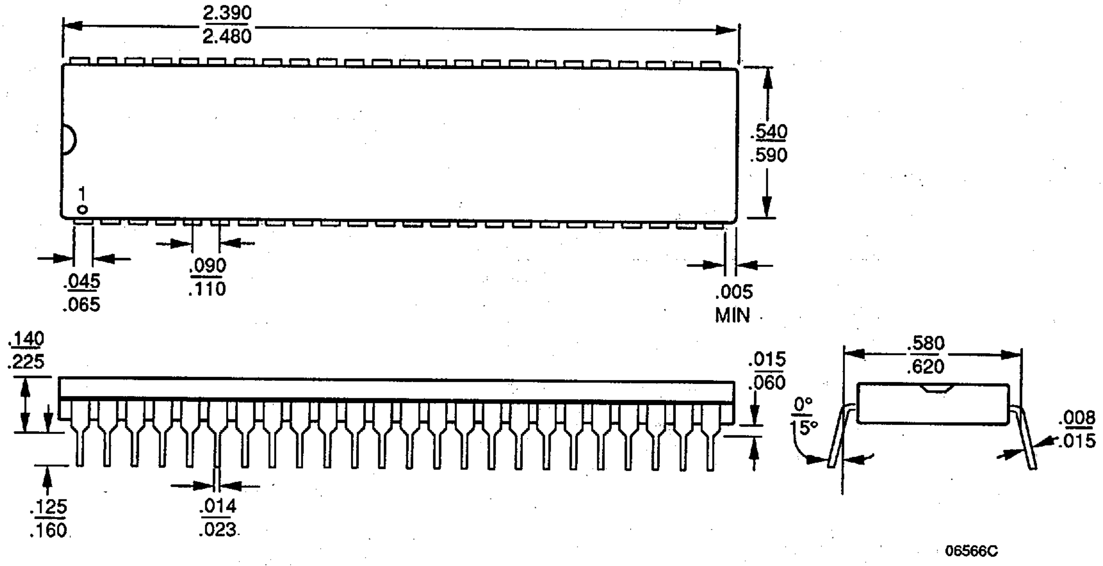
CD3024
24-Pin (Slim) Ceramic DIP


1.235
1.280
.098
MAX
.280
.310
1
.050
.065
.100
BSC
.005
MIN
TOP VIEW
.140
.200
.125
.160
.015
.060
.015
.022
SIDE VIEW
.290
.320
.008
.012
.150
MIN
0°
15°
.400
MAX
06850E
12/18/91/91
BZ13 c dc
END VIEW

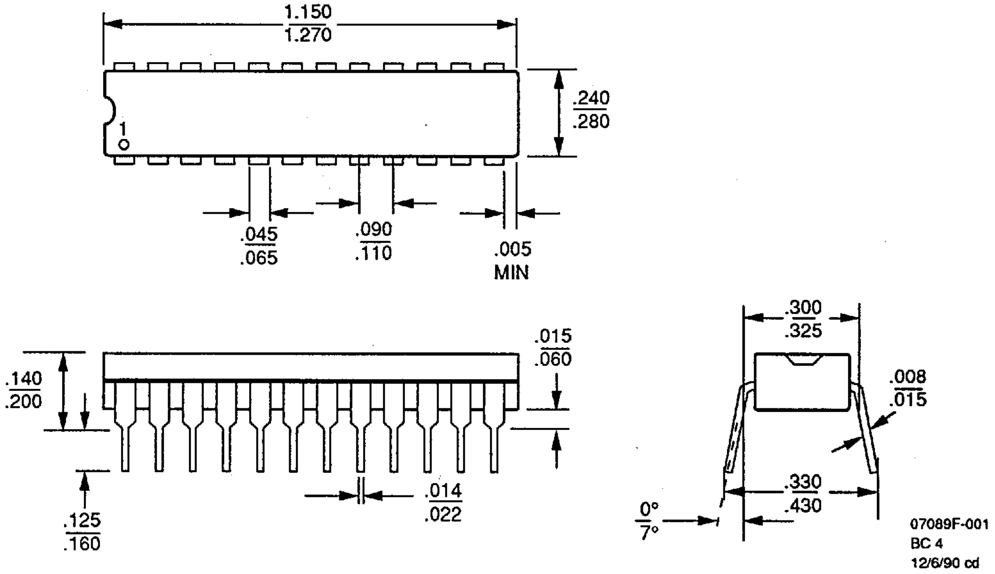
PD 020
20-Pin Plastic DIP


1.010
1.040
.240
.280
1
.045
.065
.090
.110
.005
MIN.
TOP VIEW
.140
.200
.120
.160
.015
.060
.014
.022
SIDE VIEW
.300
.325
.008
.015
0°
7°
.330
.430
END VIEW
07552C
CA 3 PD 020
12/18/91 c dc

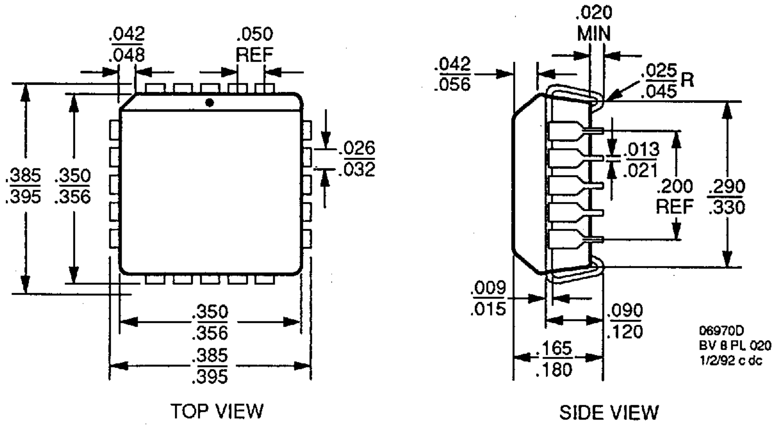
PD 048
48-Pin Plastic DIP


2.390
2.480
.540
.590
1
.045
.065
.090
.110
.005
MIN
.140
.225
.015
.060
.125
.160
.014
.023
.580
.620
0°
15°
.008
.015
06566C

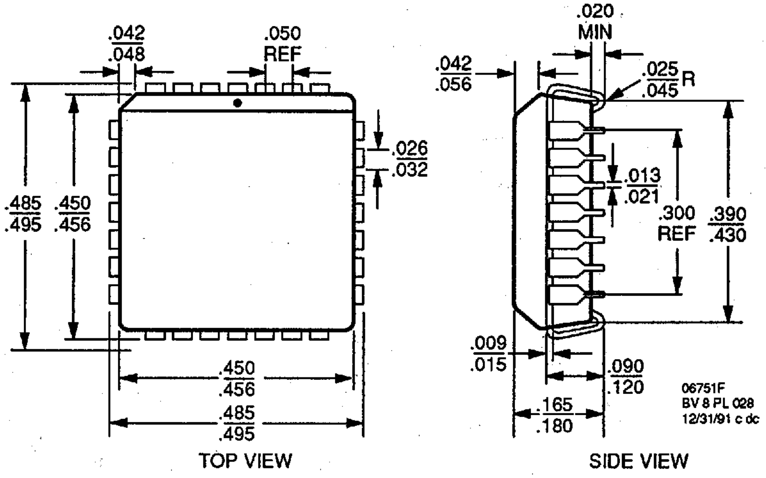
PD3024
24-Pin (Slim) Plastic DIP


1.150
1.270
.240
.280
1
.045
.065
.090
.110
.005
MIN
.015
.060
.140
.200
.125
.160
.014
.022
.300
.325
.008
.015
.330
.430
0°
7°
07089F-001
BC 4
12/6/90 cd

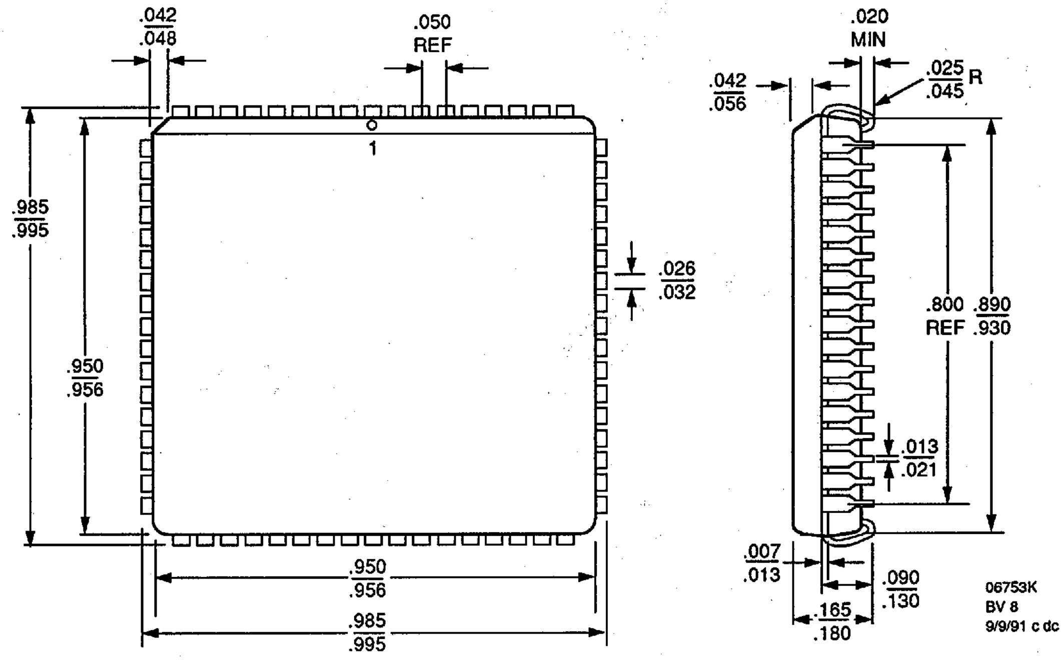
PL 020
20-Pin Plastic Leaded Chip Carrier


.042
.048
.050
REF
.385
.395
.350
.356
.026
.032
.350
.356
.385
.395
TOP VIEW
.020
MIN
.042
.056
.025
.045
R
.013
.021
.200
REF
.290
.330
.009
.015
.090
.120
.165
.180
SIDE VIEW
06970D
BV 8 PL 020
1/2/92 c dc

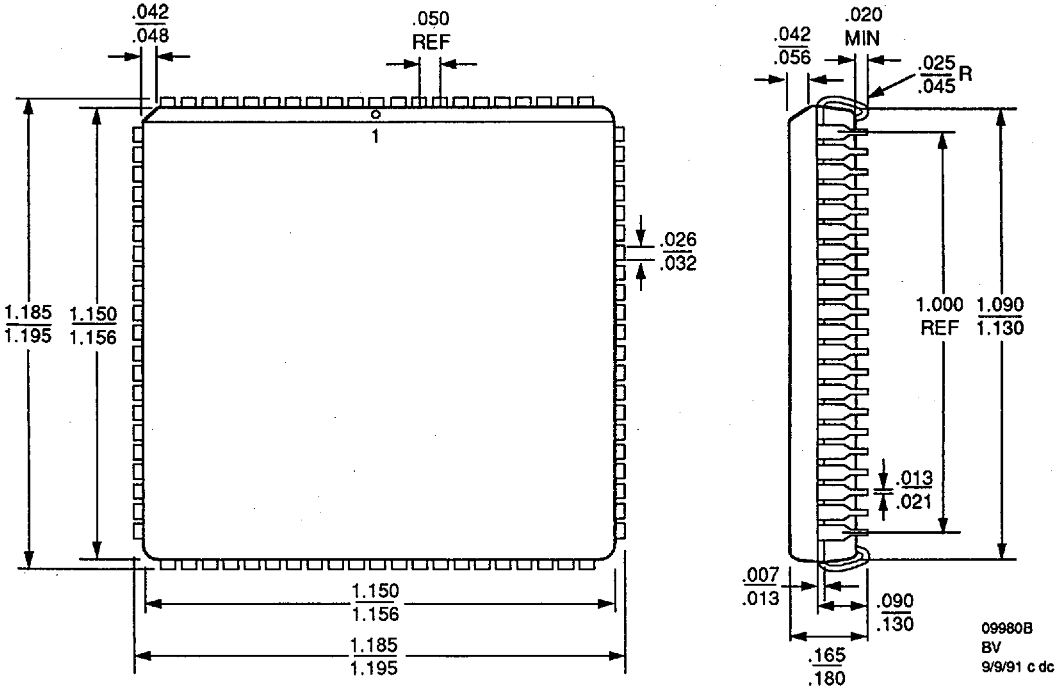
PL 028
28-Pin Plastic Leaded Chip Carrier


.042
.048
.050
REF
.026
.032
.485
.495
.450
.456
.450
.456
.485
.495
TOP VIEW
.020
MIN
.042
.056
.025
.045
R
.013
.021
.300
REF
.390
.430
.009
.015
.090
.120
.165
.180
06751F
BV 8 PL 028
12/31/91 c dc
SIDE VIEW

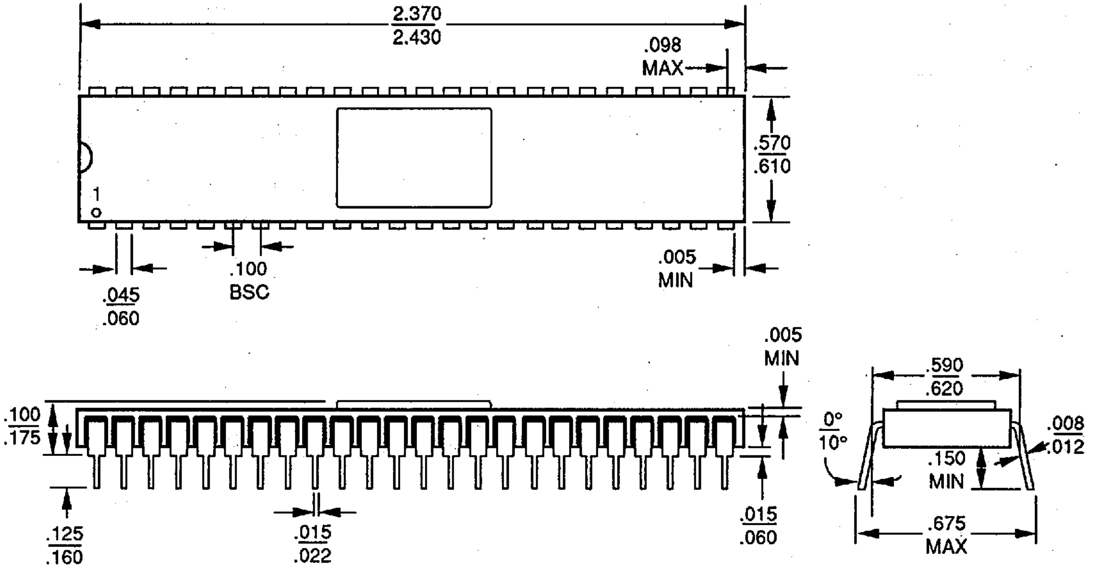
PL 068
68-Pin Plastic Leaded Chip Carrier


.042
.048
.050
REF
1
.985
.995
.950
.956
.026
.032
.950
.956
.985
.995
.020
MIN
.042
.056
.025
.045
R
.800
REF
.890
.930
.013
.021
.007
.013
.090
.130
.165
.180
06753K
BV 8
9/9/91 c dc

PL 084
84-Pin Plastic Leaded Chip Carrier


.042
.048
.050
REF
1
.026
.032
1.185
1.195
1.150
1.156
1.150
1.156
1.185
1.195
.042
.056
.020
MIN
.025
.045
R
1.000
REF
1.090
1.130
.013
.021
.007
.013
.090
.130
.165
.180
09980B
BV
9/9/91 c dc

SD 048
48-Pin Sidebrazed Ceramic DIP


2.370
2.430
.098
MAX
.570
.610
1
.045
.060
.100
BSC
.005
MIN
.005
MIN
.100
.175
.125
.160
.015
.022
.015
.060
.590
.620
0°
10°
.150
MIN
.008
.012
.675
MAX

07644C
AW 25
8/14/91 c dc

14287B