PMAD-AA TURBOchannel Ethernet Module Functional Specification

Revision 1.2

Workstation Systems Engineering
Digital Equipment Corporation
100 Hamilton Avenue
Palo Alto, CA 94301

August 27, 1990

August 1990

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibilty for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Digital Equipment Coporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital or its affiliated companies.

Blank Page

Table of Contents

1. PMAD-AA TURBOchannel Ethernet Module

2. Address Map

3. Interrupts

4. Power Estimates

5. External Interface

6. Functional Description

Revision History

Date	Version	Content/Changes
29 Sep 89	1.0	Initial release
01 Nov 89	1.1	Cleaned-up version with Artemis drawings
24 Aug 90	1.2	Cleaned-up version for public release

Blank Page

1. PMAD-AA TURBOchannel Ethernet Module

This is the functional specification for the PMAD-AA Ethernet module. The Ethernet module is a single-width option card, which connects the TURBOchannel to a DIGITAL ThickWire Ethernet. The module consists of the following subsystems;

- AMD’s Local Area Network Controller for Ethernet (LANCE)
- 128 KBytes of SRAM buffer
- 32 KByte diagnostic ROM
- 32 Bytes of Ethernet Station Address ROM

The module is a slave only device and therefore, does not perform DMA on the TURBOchannel. The Ethernet interface is based upon the AMD’s LANCE chip and 32K-by-32-bit (128 KByte) network buffer. The LANCE manages transmission and reception of packets via ring descriptors and packet buffers located in the network buffer. The network buffer is time-multiplexed between the LANCE and the processor. The LANCE access the network buffer during DMA transfers; it has no access to the system memory. An AM7992 SIA implements the serial interface to a 15-pin DB connector for an external transceiver on the option card.

2. Address Map

The DECstation 5000/200 allocates the top 32 MBytes of physical address space to the I/O subsystems. The 32 MByte I/O region is further divided into 8 regions of 4 MBytes each, for each one of the eight subsystems that may exist on the IO-bus. Refer to the *KN02 Functional Specification* for further details on the bus. Table 2-1 lists these regions.

Table 2-1: DECstation 5000/200 I/O Address Space

Address Range	Size (MBytes)	Sub-Systems
0x1FC00000..0x1FFFFFFF	4.0	System support
0x1F800000..0x1FBFFFFF	4.0	System module embedded I/O slot 6
0x1F400000..0x1F7FFFFF	4.0	System module embedded I/O slot 5
0x1F000000..0x1F3FFFFF	4.0	Reserved
0x1EC00000..0x1EFFFFFF	4.0	Reserved
0x1E800000..0x1EBFFFFF	4.0	Option Connector 2
0x1E400000..0x1E7FFFFF	4.0	Option Connector 1
0x1E000000..0x1E3FFFFF	4.0	Option Connector 0

The Ethernet module may reside in any of the three option connector slots. The 4 MByte address space on the Ethernet module is further divided as shown in the Table 2-2

Table 2-2: Ethernet Module Address Map

Address Range	Subsystem
0xbb+000000..0xbb+0FFFFF	Ethernet SRAM Buffer
0xbb+100000..0xbb+17FFFF	LANCE
0xbb+1C0000..0xbb+1FFFFF	Ethernet Diagnostic ROM and Station Address ROM

The Ethernet subsystem does not occupy the entire 4 MByte region. References to portions of Ethernet subsystem address space not explicitly defined in the following sections should not be issued. "bb" is the base address of the I/O slot in which the Ethernet module resides.

The R3000 CPU is configured for little-endian byte order. All address space descriptions in this document are correspondingly little-endian.

3. Interrupts

Table 3-1 lists the I/O device connections to the R3000 interrupt inputs. The state of the interrupt signals is continually reflected in the R3000 CAUSE register at the bit position shown in the table. Note that a given interrupt signal only generates an R3000 exception if it is enabled in the STATUS register interrupt mask field, and interrupts are enabled by the STATUS<0> register bit. Note also that the interrupt signals are visible in the CAUSE register regardless of the state of the STATUS interrupt mask. That is, the operating system interrupt dispatcher must explicitly check that a given interrupt level, which is asserted in the CAUSE register, is enabled before activating that interrupt level's handler.

Table 3-1: I/O Interrupt Levels

Level	CAUSE/STATUS	Source
5	15	FPU
4	14	Reserved
3	13	Memory Error
2	12	Reserved
1	11	Real Time Clock
0	10	I/O Slots

Due to the increased number of I/O devices supported in DECstation 5000/200 systems, all I/O interrupts are merged into a single R3000 interrupt. However, the individual I/O device interrupts are visible through the system module interrupt status register. The bits in the register are as shown in the Table 3-2

Table 3-2: I/O Interrupt Status Register

Bit	I/O Device
7	Serial Lines
6	System module embedded I/O slot 6
5	System module embedded I/O slot 5
4	Reserved
3	Reserved
2	Option Connector 2
1	Option Connector 1
0	Option Connector 0

The LANCE will interrupt the CPU through the I/O interrupt at level 0, which will be visible on the R3000 CAUSE<10> register bit. The host should then read the Interrupt Status register to determine which I/O device is interrupting. The bits in the Interrupt Status register are cleared automatically when the interrupt line of the respective device is cleared by the software.

4. Power Estimates

Table 4-1 lists the power consumption estimates for the Ethernet module. The +12 volt supply is required only to power external transciever, it is not used on the module itself.

Table 4-1: Power Estimates

Supply(V)	Typical Current (A)	Max Current (A)
+5	1.25	2.5
+12	0.25	0.5

5. External Interface

The Ethernet module connects to the Thick Wire ethernet via a DB15 receptical type connector. Table 5-1 lists the pinout of the connector.

Table 5-1: Ethernet connector pinout

Pin	Signal
1	chassis
2	col
3	xmit
4	chassis
5	rcv
6	+12 ret
7	nc
8	chassis
9	~col
10	~xmit
11	chassis
12	~rcv
13	+12
14	chassis
15	nc

6. Functional Description

6.1. LANCE

The LANCE supports byte and half-word reads and writes in DMA mode and half-word reads and writes in the programmed I/O mode. The LANCE registers are word-aligned. Reads of the LANCE nominally stall the CPU for 25 cycles. Writes to the LANCE nominally complete at a rate of 23 cycles per byte. In DMA mode the LANCE can perform reads and writes of the network buffer at a rate of 20 cycles per half-word, a peak transfer rate of 2.5 MBytes/sec. During LANCE DMA, CPU access to the LANCE registers may incur an additional latency of up to 5 microseconds.

Table 6-1 lists the LANCE register addresses.

Table 6-1: LANCE Register Addresses

Address	Register
0xbb+100000	LANCE_RDP
0xbb+100004	LANCE_RAP

6.1.1. LANCE Chip Overview

The LANCE chip is a microprogrammed controller which can conduct extensive operations independently of the central processor. There are four control and status registers (CSR's) within the LANCE chip which are programmed by a processor to initialize the LANCE chip and start its independent operation. Once started, The LANCE uses its builtin DMA controller to directly access the network buffer to get additional operating parameters and to manage the buffers it uses to transfer packets to and from the Ethernet.

All references to *memory* in the following LANCE descriptions refer to the network buffer. The LANCE generates 24-bit physical addresses. Since the network buffer is only 128 KBytes in size, the high order seven bits of the LANCE address have no effect. However, software should always zero those high-order address bits for future compatibility.

The LANCE uses three structures in memory:

Initialization Block 24 bytes of contiguous memory starting on a word boundary. The initialization block is set up by the central processor and is read by the LANCE when the processor starts the LANCE's initialization process. The initialization block contains the system's network address and pointers to the receive and transmit descriptor rings; it is described below.

Descriptor Rings two logically circular rings of buffer descriptors, one ring used by the chip receiver for incoming data and one ring used by the chip transmitter for outgoing data. Each buffer descriptor in a ring is 8 bytes long and starts on a quadword boundary. It points to a data buffer elsewhere in memory, contains the size of that buffer, and holds various status information about the buffer's contents. Buffer descriptors are described below.

Data Buffers contiguous portions of memory to buffer incoming or outgoing packets. Data buffers must be at least 64 bytes long (100 bytes for the first buffer of a packet to be transmitted) and may begin on any byte boundary. They are discussed below.

When the system is ready to begin network operation, the central processor sets up the initialization block, the receive descriptor ring, the transmit descriptor ring, and their data buffers in memory, and then starts the LANCE by writing to its CSR's. The LANCE performs its initialization process and then enters its polling loop. In this loop, it listens to the network for packets whose destination addresses are of interest and it scans the transmit descriptor ring for descriptors which have been marked by the central processor to indicate that they contain outgoing data packets. When it detects a network packet of interest, it receives and stores that packet in one or more receive buffers and marks their descriptors accordingly. When it finds a packet to be transmitted, it transmits it to the network and marks its descriptor when transmission is complete. Whenever it completes a reception or transmission (or encounters an error condition), the LANCE chip sets flags in its control and status register 0 to signal the central processor (usually by an interrupt) that it has done something of interest.

6.1.2. Programming of the LANCE

Program control of the LANCE chip is via two 16-bit read/write ports, LANCE_RAP and LANCE_RDP. These ports provide access to four 16-bit control and status registers which are named LANCE_CSR0 through LANCE_CSR3. A CSR is accessed by first writing its number into the register address port LANCE_RAP after which the contents of the CSR are read or written by accesses to the register data port LANCE_RDP. Note that registers other than LANCE_CSR0 may be accessed only while the STOP bit of LANCE_CSR0 is set.

6.1.2.1. Register Address Port (LANCE_RAP)

The LANCE_RAP register selects which of the four CSR’s is accessed via the register data port.

```
     15      14      13      12      11      10       9       8
  +-------+-------+-------+-------+-------+-------+-------+-------+--
  |                          reserved
  +-------+-------+-------+-------+-------+-------+-------+-------+-

      7       6       5       4       3       2       1       0
 -+-------+-------+-------+-------+-------+-------+-------+-------+
                    reserved                      |     CSRNO     |
--+-------+-------+-------+-------+-------+-------+-------+-------+
```

Figure 6-1: LANCE Register Address Port (LANCE_RAP)

<15:2> Reserved. Ignored on write; read as zeros.

CSRNO CSR select (bits 1:0). These read/write bits select which of the four CSR’s is accessible via the register data port. They are cleared to zero upon power-on. Values are:

Bits 1:0	Register
0 0	LANCE_CSR0
0 1	LANCE_CSR1
1 0	LANCE_CSR2
1 1	LANCE_CSR3

6.1.2.2. Register Data Port (LANCE_RDP)

The register data port is a 16-bit window through which a processor can read and write the CSR designated by the register address port LANCE_RAP.

Note that registers LANCE_CSR1, LANCE_CSR2, and LANCE_CSR3 are accessible only while the STOP bit in LANCE_CSR0 is set. If that STOP bit is clear (i.e. the LANCE chip is active), attempts to read from those CSR’s will return UNDEFINED data and attempts to write to them will be ignored. Accesses to a CSR via LANCE_RDP do not alter the register address pointer LANCE_RAP. In normal operation only LANCE_CSR0 can be accessed, so LANCE_RAP should be set to point to LANCE_CSR0 and left that way.

6.1.2.3. Control and Status Register 0 (LANCE_CSR0)

This register is used by the controlling program to start and stop the operation of the LANCE chip and to monitor its status. It is accessible to the processor via port LANCE_RDP when bits 1:0 of LANCE_RAP are set to 00. All of its bits can be read at any time and none of its bits is affected by reading the register. The effects of a write operation are described individually for each bit.

When power is applied to the system, all the bits in this register are cleared except the STOP bit which is set.

```
     15      14      13      12      11      10       9       8
  +-------+-------+-------+-------+-------+-------+-------+-------+--
  |  ERR  |  BABL |  CERR |  MISS |  MERR |  RINT |  TINT |  IDON |
  +-------+-------+-------+-------+-------+-------+-------+-------+-

      7       6       5       4       3       2       1       0
 -+-------+-------+-------+-------+-------+-------+-------+-------+
  |  INTR |  INEA |  RXON |  TXON |  TDMD |  STOP |  STRT |  INIT |
--+-------+-------+-------+-------+-------+-------+-------+-------+
```

Figure 6-2: LANCE Control and Status Register 0 (LANCE_CSR0)

ERR Error summary (bit 15). This read-only bit is one whenever any of the bits BABL, CERR, MISS, or MERR in this register are ones. Writing to this bit has no effect. It is cleared when all of the bits which set it are zero or when the STOP bit is set.

BABL Transmitter timeout error (bit 14). This bit is set when the transmitter has been on the channel longer than the time required to send the maximum length packet. It will be set after 1519 data bytes have been transmitted (the chip will continue to transmit until the whole packet is transmitted or until there is a failure before the whole packet is transmitted).

This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the ERR and INTR bits are also ones.

CERR Collision error (bit 13). This bit is set when the collision input to the chip failed to activate within 2 microseconds after a chip-initiated transmission is completed. This collision-after-transmission is a transceiver test feature. This function is also known as heartbeat or SQE (signal quality error) test.

This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the ERR bit is also one.

MISS Missed packet (bit 12). This bit is set when the receiver loses a packet because it does not own a receive buffer. The MISS bit is not valid in internal loopback mode.

This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the ERR and INTR bits are also ones.

MERR Memory error (bit 11). This bit is set when the chip attempts a DMA transfer and does not receive a ready response from the memory within 25.6 microseconds after beginning the memory cycle. When MERR is set, the receiver and transmitter are turned off (bits RXON and TXON of this register are cleared to zero).

This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the ERR and INTR bits are also ones.

RINT Receive interrupt (bit 10). This bit is set when the chip updates an entry in the receive descriptor ring for the last buffer received or when reception is stopped due to a failure.

This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the INTR bit is also one.

TINT Transmitter interrupt (bit 9). This bit is set when the chip updates an entry in the transmit descriptor ring for the last buffer sent or when transmission is stopped due to a failure.

This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the INTR bit is also one.

IDON Initialization done (bit 8). This bit is set when the chip completes the initialization process which was started by setting the INIT bit in this register. When IDON is set, the chip has read the initialization block from memory and stored the new parameters.

This bit is cleared when a one is written to it (writing a zero has no effect) or when the STOP bit is set. When this bit is one, the INTR bit is also one.

INTR Interrupt request (bit 7). This read-only bit is one whenever any of the bits BABL, MISS, MERR, RINT, TINT, or IDON in this register are ones. Writing to this bit has no effect. It is cleared when all of the bits which set it are zero or when the STOP bit is set.

When both the INTR and INEA bits in this register are set, an interrupt request is sent to the system interrupt controller.

INEA Interrupt enable (bit 6). This read/write bit controls whether the setting of the INTR bit generates an interrupt request. When both the INTR and INEA bits in this register are set, an interrupt request is sent to the processor.

This bit is set when a one is written to it. It is cleared when a zero is written to it or when the STOP bit is set.

RXON Receiver on (bit 5). This read-only bit indicates (when it is one) that the receiver is enabled. RXON is set when initialization is completed (i.e. when IDON is set, unless the DRX bit of the initialization block MODE register was one) and then the STRT bit in this register is set. Writing to this bit has no effect. RXON is cleared when either the MERR or STOP bits of this register are set.

TXON Transmitter on (bit 4). This read-only bit indicates (when it is one) that the transmitter is enabled. TXON is set when initialization is completed (i.e. when IDON is set, unless the DTX bit of the initialization block MODE register was one) and then the STRT bit in this register is set. Writing to this bit has no effect. TXON is cleared when either the MERR or STOP bits of this register are set or when any of bits UFLO, BUFF, or RTRY in a Transmit Buffer Descriptor are set.

TDMD Transmit demand (bit 3). Setting this bit signals the chip to access the transmit descriptor ring without waiting for the polltime interval to elapse. This bit need not be set to transmit a packet; setting it merely hastens the chip's response to the insertion of a transmit descriptor ring entry by the host program.

This bit is set by writing a one to it (writing a zero has no effect) and is cleared by the chip when it recognizes the bit (the bit may read as one for a short time after it is set, depending upon the level of activity in the chip). TDMD is also cleared when the STOP bit is set.

STOP Stop external activity (bit 2). Setting this bit stops all external activity and clears the internal logic of the chip; this has the same effect as the electrical reset signaled upon power-on. The chip remains inactive and STOP remains set until the STRT or INIT bits in this register are set.

This bit is set by writing a one to it (writing a zero has no effect) or upon power-on. It is cleared when either INIT or STRT is set. If the processor writes ones to STOP, INIT, and STRT at the same time, STOP takes precedence and neither STRT nor INIT is set.

Setting STOP clears all the other bits in this register. After STOP has been set, the other three CSR's (LANCE_CSR1, LANCE_CSR2, and LANCE_CSR3) must be reloaded before setting INIT or STRT (note that those three registers may be accessed only while STOP is set).

STRT Start operation (bit 1). Setting this bit enables the chip to send and receive packets, perform DMA and do buffer management. The STOP bit must be set prior to setting the STRT bit (setting STRT then clears STOP).

STRT is set by writing a one to it (writing a zero has no effect). It is cleared when the STOP bit is set.

INIT Initialize (bit 0). Setting this bit causes the chip to perform its initialization process, which reads the initialization block from the memory addressed by the contents of LANCE_CSR1 and LANCE_CSR2 using DMA accesses. The STOP bit must be set prior to setting the INIT bit (setting INIT then clears STOP).

INIT is set by writing a one to it (writing a zero has no effect). It is cleared when the STOP bit is set.

The INIT and STRT bits must not be set at the same time. The proper initialization procedure is as follows:

1. Set STOP in LANCE_CSR0
2. Set up the initialization block in memory
3. Load LANCE_CSR1 and LANCE_CSR2 with the starting address of the initialization block
4. Set INIT in LANCE_CSR0
5. Wait for IDON in LANCE_CSR0 to become set
6. Set STRT in LANCE_CSR0 to begin operation

6.1.2.4. Control and Status Register 1 (LANCE_CSR1)

This read/write register is used in conjunction with LANCE_CSR2 to supply the 24-bit physical memory address of the initialization block which the chip reads when it performs its initialization process. The register is accessible to the processor via LANCE_RDP when bits 1:0 of LANCE_RAP are 01 and the STOP bit of LANCE_CSR0 is set. Its contents upon power-on are UNPREDICTABLE.

```
     15      14      13      12      11      10       9       8
  +-------+-------+-------+-------+-------+-------+-------+-------+--
  |                           IADR 15:8
  +-------+-------+-------+-------+-------+-------+-------+-------+-

      7       6       5       4       3       2       1       0
 -+-------+-------+-------+-------+-------+-------+-------+-------+
                              IADR 7:1                    |   0   |
--+-------+-------+-------+-------+-------+-------+-------+-------+
```

Figure 6-3: LANCE Control and Status Register 1 (LANCE_CSR1)

IADR Initialization block address (bits 15:0). These are the low-order sixteen bits of the (24-bit physical) byte address of the first byte of the initialization block. Note that since the block must be word-aligned, bit 0 must be zero.

6.1.2.5. Control and Status Register 2 (LANCE_CSR2)

This read/write register is used in conjunction with LANCE_CSR1 to supply the 24-bit physical memory address of the initialization block which the chip reads when it performs its initialization process. The register is accessible to the processor via LANCE_RDP when bits 1:0 of LANCE_RAP are 10 and the STOP bit of LANCE_CSR0 is set. Its contents upon power-on are UNPREDICTABLE.

```
     15      14      13      12      11      10       9       8
  +-------+-------+-------+-------+-------+-------+-------+-------+--
  |                           reserved                            |
  +-------+-------+-------+-------+-------+-------+-------+-------+-

      7       6       5       4       3       2       1       0
 -+-------+-------+-------+-------+-------+-------+-------+-------+
  |                           IADR 23:16                          |
--+-------+-------+-------+-------+-------+-------+-------+-------+
```

Figure 6-4: LANCE Control and Status Register 2 (LANCE_CSR2)

<15:8> Reserved. Write with zeros.

IADR Initialization block address (bits 7:0). These are the high-order eight bits of the (24-bit physical) byte address of the first byte of the initialization block.

6.1.2.6. Control and Status Register 3 (LANCE_CSR3)

This read/write register controls certain aspects of the *electrical* interface between the LANCE chip and the system. It must be set as indicated for each bit. The register is accessible to the processor via LANCE_RDP when bits 1:0 of LANCE_RAP are 11 and the STOP bit of LANCE_CSR0 is set. Its contents upon power-on are entirely zeros.

```
      15      14      13      12      11      10       9       8
   +-------+-------+-------+-------+-------+-------+-------+-------+--
   |                           reserved
   +-------+-------+-------+-------+-------+-------+-------+-------+-

       7       6       5       4       3       2       1       0
  -+-------+-------+-------+-------+-------+-------+-------+-------+
                  reserved                 |  BSWP |  ACON |  BCON |
 --+-------+-------+-------+-------+-------+-------+-------+-------+
```

Figure 6-5: LANCE Control and Status Register 3 (LANCE_CSR3)

<15:3> Reserved. Ignored on write; read as zeros.

BSWP Byte swap (bit 2). When this bit is set, the chip will swap the high and low bytes for DMA data transfers between the silo and bus memory in order to accommodate processors which consider bus bits 15:08 to be the least significant byte of data. This bit is read/write; it is cleared when the STOP bit in LANCE_CSR0 is set. For this system, this bit must be ZERO.

ACON ALE control (bit 1). This bit controls the polarity of the signal emitted on the chip's ALE/AS pin during DMA operation. This bit is read/write; it is cleared when the STOP bit in LANCE_CSR0 is set. For this system, this bit must be ZERO.

BCON Byte control (bit 0). This bit controls the configuration of the byte mask and hold signals on the chip's pins during DMA operation. This bit is read/write; it is cleared when the STOP bit in LANCE_CSR0 is set. For this system, this bit must be ZERO.

6.1.3. Interrupts

The LANCE chip asserts an interrupt request signal whenever the INTR and INEA bits in its control and status register 0 (LANCE_CSR0) are both ones. This signal is presented to the processor.

6.1.4. DMA Operation

The LANCE chip contains a built-in DMA controller which can transfer data directly between the chip and memory in the address range 0x0000 through 0xFFFF. The chip contains a 48-byte FIFO buffer to allow for DMA service latency and to minimize the number of request-grant arbitration cycles. When transferring large amounts of data in burst mode, the chip transfers 16 bytes per DMA request. Each word transfer requires 0.6 microseconds, so a 16-byte burst will require 4.8 microseconds.

This DMA controller is used to read the initialization block, to read and write the descriptor rings, and to read and write data buffers. Note that all the memory addresses handled by the chip are physical addresses.

6.1.5. Initialization Block

When the LANCE chip is initialized (by setting the INIT bit in LANCE_CSR0), it reads a 24-byte block of data called the initialization block from main memory using DMA accesses. The physical address of the initialization block (IADR) is taken from LANCE_CSR1 and LANCE_CSR2. Since the data must be word-aligned, the low-order bit of the address must be zero. The initialization block comprises 12 16-bit words arranged as follows:

```
                         +-----------------------------+
            IADR +  0    |            MODE             |
                         +-----------------------------+
            IADR +  2    |         PADR <15:00>        |
                         +-----------------------------+
            IADR +  4    |         PADR <31:16>        |
                         +-----------------------------+
            IADR +  6    |         PADR <47:32>        |
                         +-----------------------------+
            IADR +  8    |        LADRF <15:00>        |
                         +-----------------------------+
            IADR + 10    |        LADRF <31:16>        |
                         +-----------------------------+
            IADR + 12    |        LADRF <47:32>        |
                         +-----------------------------+
            IADR + 14    |        LADRF <63:48>        |
                         +-----------------------------+
            IADR + 16    |         RDRA <15:00>        |
                         +------+----------------------+
            IADR + 18    | RLEN |  RDRA <23:16>        |
                         +------+----------------------+
            IADR + 20    |         TDRA <15:00>        |
                         +------+----------------------+
            IADR + 22    | TLEN |  TDRA <23:16>        |
                         +------+----------------------+
```

Figure 6-6: LANCE Initialization Block Layout

6.1.5.1. Initialization Block MODE Word (NIB_MODE)

The MODE word of the initialization block allows alteration of the LANCE chip's normal operation for testing and special applications. For normal operation the MODE word is entirely zero.

```
     15      14      13      12      11      10       9       8
  +-------+-------+-------+-------+-------+-------+-------+-------+--
  |  PROM |                       reserved
  +-------+-------+-------+-------+-------+-------+-------+-------+-

      7       6       5       4       3       2       1       0
 -+-------+-------+-------+-------+-------+-------+-------+-------+
    resv  |  INTL |  DRTY |  COLL |  DTCR |  LOOP |  DTX  |  DRX  |
--+-------+-------+-------+-------+-------+-------+-------+-------+
```

Figure 6-7: Initialization Block MODE Word (NIB_MODE)

PROM Promiscuous mode (bit 15). When this bit is set, all incoming packets are accepted regardless of their destination addresses.

<14:7> Reserved. Should be written with zeros.

INTL Internal loopback (bit 6). This bit is used in conjunction with the LOOP bit in this word to control loopback operation. See the description of the LOOP bit, below.

DRTY Disable retry (bit 5). When this bit is set, the chip will attempt only one transmission of a packet. If there is a collision on the first transmission attempt, a retry error (RTRY) will be reported in the transmit buffer descriptor.

COLL Force collision (bit 4). Setting this bit allows the collision logic to be tested. The chip must be in internal loopback mode for COLL to be used. When COLL is one a collision will be forced during the subsequent transmission attempt. This will result in 16 total transmission attempts with a retry error reported in LANCE_TMD3.

DTCR Disable transmit CRC (bit 3). When DTCR is zero the transmitter will generate and append a 4-byte CRC to each transmitted packet (normal operation). When DTCR is one the CRC logic is allocated instead to the receiver and no CRC is sent with a transmitted packet.

During loopback, setting DTCR to zero will cause a CRC to be generated and sent with the transmitted packet, but no CRC check can be done by the receiver since the CRC logic is shared and cannot both generate and check a CRC at the same time. The CRC transmitted with the packet will be received and written into memory following the data where it can be checked by software.

If DTCR is set to one during loopback, the driving software must compute and append a CRC value to the data to be transmitted. The receiver will check this CRC upon reception and report any error.

LOOP Loopback control (bit 2). Loopback allows the LANCE chip to operate in full duplex mode for test purposes. The maximum packet size is limited to 32 data bytes (in addition to which 4 CRC bytes may be appended). 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).

Setting LOOP to one allows simultaneous transmission and reception for a packet constrained to fit within the silo. The chip waits until the entire packet is in the silo before beginning serial transmission. The incoming data stream fills the silo from behind as it is being emptied. Moving the received packet out of the silo into memory does not begin until reception has ceased.

In loopback mode, transmit data chaining is not possible. Receive data chaining is allowed regardless of the receive buffer length. (In normal operation, the receive buffers must be 64 bytes long, to allow time for buffer lookahead.)

Valid loopback bit settings are:

LOOP	INTL	Operation
0	x	Normal on-line operation
1	0	External loopback
1	1	Internal loopback

Internal loopback allows the chip to receive its own transmitted packet without disturbing the network. The chip will not receive any packets from the network while it is in internal loopback mode.

External loopback allows the chip to transmit a packet through the transceiver out to the network cable to check the operability of all circuits and connections between the LANCE chip and the network cable. Multicast addressing in external loopback is valid only when DTCR is one (user needs to append the 4 CRC bytes). In external loopback, the chip also receives packets from other nodes.

DTX Disable transmitter (bit 1). If this bit is set, the chip will not set the TXON bit in LANCE_CSR0 at the completion of initialization. This will prevent the LANCE chip from attempting to access the transmit descriptor ring, hence no transmissions will be attempted.

DRX Disable receiver (bit 0). If this bit is set, the chip will not set the RXON bit in LANCE_CSR0 at the completion of initialization. This will cause the chip to reject all incoming packets and to not attempt to access the receive descriptor ring.

6.1.5.2. Network Physical Address (NIB_PADR)

The 48-bit physical Ethernet network node address is contained in bytes 2:7 of the initialization block. (This is a network address; it has no relationship to any memory address.)

```
           |<---IADR+6--->|<---IADR+4--->|<---IADR+2--->|

            47          32 31          16 15           0
           +--------------+--------------+--------------+
           |                                         |0|
           +--------------+--------------+--------------+
```

Figure 6-8: Network Physical Address (NIB_PADR)

The contents of NIB_PADR identify this station to the network and must be unique within the domain of the network. Its value is normally taken from the Ethernet Station Address ROM. The low-order bit (bit 0) of this address must be zero since it is a physical address.

6.1.5.3. Multicast Address Filter Mask (NIB_LADRF)

Bytes 8:15 of the initialization block contain the 64-bit multicast address filter mask. The multicast address filter is a partial filter which assists the network controller driver program to selectively receive packets which contain multicast network addresses.

```
  |<---IADR+14--->|<---IADR+12--->|<---IADR+10--->|<---IADR+8---->|

   63           48 47           32 31           16 15            0
  +---------------+---------------+---------------+---------------+
  |                                                               |
  +---------------+---------------+---------------+---------------+
```

Figure 6-9: Multicast Address Filter Mask (NIB_LADRF)

Multicast Ethernet addresses are distinguished from physical network addresses by the presence of a one in bit 0 of the 48-bit address field. If an incoming packet contains a physical destination address (bit 0 is zero), then its entire 48 bits are compared with the contents of NIB_PADR and the packet is ignored if they are not equal. If the packet contains a multicast destination address which is all ones (the broadcast address), it is always accepted and stored regardless of the contents of the multicast address filter mask.

All other multicast addresses are processed through the multicast address filter to determine whether the incoming packet will be stored in a receive buffer. This filtering is performed by passing the multicast address field through the CRC generator. The high-order 6 bits of the resulting 32-bit CRC are used to select one of the 64 bits of NIB_LADRF. (These high-order six bits represent in binary the number of the bit in NIB_LADRF) If the bit selected from NIB_LADRF is one, the packet is stored in a receive buffer; otherwise it is ignored. This mechanism effectively splits the entire domain of 2**47 multicast addresses into 64 parts, and multicast addresses falling into each part will be accepted or ignored according to the value of the corresponding bit in NIB_LADRF. The driver program must examine the addresses of the packets accepted by this partial filtering to complete the filtering task.

6.1.5.4. Receive

Bytes 16:19 of the initialization block describe the starting address and extent of the receive descriptor ring.

```
|<-------IADR+18------->|<-------IADR+16------->|

 31  29 28  24 23     16 15                    0
+------+------+---------+----------------------+
| RLEN | resv |            RDRA             000|
+------+------+---------+----------------------+
```

Figure 6-10: Receive Descriptor Ring Pointer (NIB_RDRP)

RLEN Receive ring length (bits 31:29). This field gives the number of entries in the receive descriptor ring, expressed as a power of 2:

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

<28:24> Reserved; should be zeros.

RDRA Receive descriptor ring address (bits 23:0). This is the physical address in system memory of the first element in the ring. Since each 8-byte element must be aligned on a quadword boundary, bits 2:0 of this address must be zero.

6.1.5.5. Transmit Descriptor Ring Pointer (NIB_TDRP)

Bytes 20:23 of the initialization block describe the starting address and extent of the Transmit descriptor ring.

```
|<-------IADR+22------->|<-------IADR+20------->|

 31  29 28  24 23     16 15                    0
+------+------+---------+----------------------+
| TLEN | resv |            TDRA             000|
+------+------+---------+----------------------+
```

Figure 6-11: Transmit Descriptor Ring Pointer (NIB_TDRP)

TLEN Transmit ring length (bits 31:29). This field gives the number of entries in the transmit descriptor ring, expressed as a power of 2:

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

<28:24> Reserved; should be zeros.

TDRA Transmit descriptor ring address (bits 23:0). This is the physical address in system memory of the first element in the ring. Since each 8-byte element must be aligned on a quadword boundary, bits 2:0 of this address must be zero.

6.1.6. Buffer Management

The LANCE chip manages its data buffers by using two rings of buffer descriptors which are stored in memory: the receive descriptor ring and the transmit descriptor ring. Each buffer descriptor points to a data buffer elsewhere in memory, contains the size of that buffer, and contains status information about that buffer’s contents.

The starting location in memory of each ring and the number of descriptors in it are given to the LANCE chip via the initialization block during the chip initialization process. Each descriptor is 8 bytes long and must be aligned on a quadword boundary (the three low-order bits of its address must be zero). The descriptors in a ring are physically contiguous in memory and the number of descriptors must be a power of 2. The LANCE keeps an internal index to its current position in each ring which it increments modulo the number of descriptors in the ring as it advances around each ring.

Once started, the LANCE polls each ring to find descriptors for buffers in which to receive incoming packets and from which to transmit outgoing packets, and revises the status information in buffer descriptors as it processes their associated buffers. When polling, the LANCE is limited to looking only one ahead of the descriptor with which it is currently working. The high speed of the data stream requires that each buffer be at least 64 bytes long to allow time to chain buffers for packets which are larger than one buffer. (The first buffer of a packet to be transmitted should be at least 100 bytes to avoid problems in case a late collision is detected.)

Each descriptor in a ring is "owned" either by the LANCE chip or by the host processor; this status is indicated by the OWN bit in each descriptor. Mutual exclusion is accomplished by the rule that each device can only relinquish ownership of a descriptor to the other device, it can never take ownership; and that each device cannot change any field in a descriptor or its associated buffer after it has relinquished ownership. When the host processor sets up the rings of descriptors before starting the LANCE, it sets the OWN bits such that the LANCE will own all the descriptors in the receive descriptor ring (to be used by the LANCE to receive packets from the network) and the host will own all the descriptors in the transmit descriptor ring (to be used by the host to set up packets to be transmitted to the network).

It is recommended that software configure all receive and transmit buffers to be the maximum packet size of 1518 bytes. The 128 KByte network buffer supports 64 such receive buffers, 16 such transmit buffers, plus the receive and transmit rings and the initialization block. Even with the theoretical worst case sequence of 64 back-to-back minimum length packets, the receiver buffers will not be exhausted for 4.3 milliseconds in this configuration.

6.1.6.1. Receive Buffer Descriptor

A receive buffer descriptor comprises 4 words aligned in memory on a quadword address boundary.

```
Memory
Offset
         1 1 1 1 1 1
         5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
        +---------------+---------------+
        |                               |
 +0:    |             LADR              |
        |                               |
        +---------------+---------------+
        |O E F O C B S E|               |
        |W R R F R U T N|               |
 +2:    |N R A L C F P P|     HADR      |
        |    M O   F    |               |
        +---------------+---------------+
        |       |                       |
 +4:    |1 1 1 1|         BCNT          |
        |       |                       |
        +---------------+---------------+
        |       |                       |
 +6:    |0 0 0 0|         MCNT          |
        |       |                       |
        +---------------+---------------+
```

Figure 6-12: Receive Buffer Descriptor

LADR — Low-order buffer address (offset 0, bits 15:0). These are the low-order 16 bits of the 24-bit physical memory address of the start of the buffer associated with this descriptor. Written by the host; unchanged by the LANCE.

HADR — High-order buffer address (offset 2, bits 7:0). These are the high-order 8 bits of the 24-bit physical memory address of the start of the buffer associated with this descriptor. Written by the host; unchanged by the LANCE.

OWN Owned flag (offset 2, bit 15). This bit indicates whether the descriptor is owned by the host (OWN = 0) or by the LANCE (OWN = 1). The LANCE clears OWN after filling the buffer associated with the descriptor with an incoming packet. The host sets OWN after emptying the buffer. In each case, this must be the last bit changed by the current owner, since changing OWN passes ownership to the other party and the relinquishing party must not thereafter alter anything in the descriptor or its buffer.

ERR Error summary (offset 2, bit 14). This is the logical OR of the FRAM, OFLO, CRC and BUFF bits in this word. Set by the LANCE and cleared by the host.

FRAM Framing error (offset 2, bit 13). This bit is set by the LANCE to indicate that the incoming packet stored in the buffer had both a non-integral multiple of eight bits and a CRC error. It is cleared by the host.

OFLO Overflow error (offset 2, bit 12). This bit is set by the LANCE to indicate that the receiver has lost part or all of an incoming packet because it could not store it in the buffer before the chip's silo overflowed. Cleared by the host.

CRC Checksum error (offset 2, bit 11). This bit is set by the LANCE to indicate that the received packet has an invalid CRC checksum. Cleared by the host.

BUFF Buffer error (offset 2, bit 10). This bit is set by the LANCE when it has used all its owned receive descriptors or when it could not get the next descriptor in time while attempting to chain to a new buffer in the midst of a packet. When a buffer error occurs, an overflow error (bit OFLO) also occurs because the LANCE continues to attempt to get the next buffer until its silo overflows. BUFF is cleared by the host.

STP Start of packet (offset 2, bit 9). This bit is set by the LANCE to indicate that this is the first buffer used for this packet. Cleared by the host.

ENP End of packet (offset 2, bit 8). This bit is set by the LANCE to indicate that this is the last buffer used for this packet. When both STP and ENP are set in a descriptor, its buffer contains an entire packet; otherwise two or more buffers have been chained together to hold the packet. ENP is cleared by the host.

1111 Offset 4, bits 15:12 must be set by the host to ones. Unchanged by the LANCE.

BCNT Buffer size (offset 4, bits 11:0). This is the number of bytes in the buffer (whose starting address is in HADR and LADR) in two's complement form. Note that the minimum buffer size is 64 bytes and that the maximum required for a legal packet is 1518 bytes. Written by the host; unchanged by the LANCE.

0000 Offset 6, bits 15:12 are reserved; they should be set to zeros by the host when it constructs the descriptor.

MCNT Byte count (offset 6, bits 11:0). This is the length in bytes of the received packet for which this is the last or only descriptor. MCNT is valid only in a descriptor in which ENP is set (last buffer) and ERR is clear (no error). Set by the LANCE and cleared by the host.

6.1.6.2. Transmit Buffer Descriptor

A transmit buffer descriptor comprises 4 words aligned in memory on a quadword address boundary.

```
Memory
Offset
         1 1 1 1 1 1
         5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
        +---------------+---------------+
        |                               |
 +0:    |             LADR              |
        |                               |
        +---------------+---------------+
        |O E r M O D S E|               |
        |W R e O N E T N|               |
 +2:    |N R s R E F P P|     HADR      |
        |    v E        |               |
        +---------------+---------------+
        |       |                       |
 +4:    |1 1 1 1|         BCNT          |
        |       |                       |
        +---------------+---------------+
        |B U r L L R|                   |
 +6:    |U F e C C T|                   |
        |F L s O A R|       TDR         |
        |F O v L R Y|                   |
        +---------------+---------------+
```

Figure 6-13: Transmit Buffer Descriptor

LADR Low-order buffer address (offset 0, bits 15:0). These are the low-order 16 bits of the 24-bit physical memory address of the start of the buffer associated with this descriptor. Written by the host; unchanged by the LANCE.

HADR High-order buffer address (offset 2, bits 7:0). These are the high-order 8 bits of the 24-bit physical memory address of the start of the buffer associated with this descriptor. Written by the host; unchanged by the LANCE.

OWN Owned flag (offset 2, bit 15). This bit indicates whether the descriptor is owned by the host (OWN = 0) or by the LANCE (OWN = 1). The host sets OWN after filling the buffer with a packet to be transmitted. The LANCE clears OWN after transmitting the contents of the buffer. In each case, this must be the last bit changed by the current owner, since changing OWN passes ownership to the other party and the relinquishing party must not thereafter alter anything in the descriptor or its buffer.

ERR Error summary (offset 2, bit 14). This is the logical OR of the LCOL, LCAR, UFLO and RTRY bits in this descriptor. Set by the LANCE and cleared by the host.

RESV Offset 2, bit 13 is reserved. The LANCE will write a zero in this bit.

MORE More retries (offset 2, bit 12). The LANCE sets this bit when more than one retry was required to transmit the packet. Cleared by the host.

ONE One retry (offset 2, bit 11). The LANCE sets this bit when exactly one retry was required to transmit the packet. Cleared by the host.

DEF Deferred (offset 2, bit 10). The LANCE sets this bit when it had to defer while trying to transmit the packet. This occurs when the network is busy when the LANCE is ready to transmit. Cleared by the host.

STP Start of packet (offset 2, bit 9). This bit is set by the host to indicate that this is the first buffer used for this packet. STP is not changed by the LANCE.

ENP End of packet (offset 2, bit 8). This bit is set by the host to indicate that this is the last buffer used for this packet. When both STP and ENP are set in a descriptor, its buffer contains an entire packet; otherwise two or more buffers have been chained together to hold the packet. ENP is not changed by the LANCE.

1111 Offset 4, bits 15:12 must be set by the host to ones. Unchanged by the LANCE.

BCNT Byte count (offset 4, bits 11:0). This is the number of bytes, in two's complement form, which the LANCE will transmit from this buffer. Note that for any buffer which is not the last of a packet, at least 64 bytes (100 bytes if it is the start of the packet) must be transmitted to allow adequate time for the LANCE to acquire the next buffer. Written by the host; unchanged by the LANCE.

NOTE: The remaining fields of the descriptor (which make up its entire fourth word) are valid only when the ERR bit in the second word has been set by the LANCE.

BUFF Buffer error (offset 6, bit 15). This bit is set by the LANCE during transmission when it does not find the ENP bit set in the current descriptor and it does not own the next descriptor. When BUFF is set, the UFLO bit (below) is also set because the LANCE continues to transmit until its silo becomes empty. BUFF is cleared by the host.

UFLO Underflow (offset 6, bit 14). This bit is set by the LANCE when it truncates a packet being transmitted because it has drained its silo before it was able to obtain additional data from a buffer in memory. UFLO is cleared by the host.

RESV Offset 6, bit 13 is reserved. The LANCE will write a zero in this bit.

LCOL Late collision (offset 6, bit 12). This bit is set by the LANCE to indicate that a collision has occurred after the slot time of the network channel has elapsed. The LANCE does not retry after a late collision. LCOL is cleared by the host.

LCAR Loss of carrier (offset 6, bit 11). This bit is set by the LANCE when the carrier-present input to the chip becomes false during a transmission initiated by the LANCE. The LANCE does not retry after such a failure. LCAR is cleared by the host.

RTRY Retries exhausted (offset 6, bit 10). This bit is set by the LANCE after 16 attempts to transmit a packet have failed due to repeated collisions on the network. (If the DRTY bit of the initialization block MODE word is set, RTRY will instead be set after only only one failed transmission attempt.) RTRY is cleared by the host.

TDR Time domain reflectometer (offset 6, bits 9:0). These bits are the value of an internal counter which is set by the LANCE to count system clocks 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; it is valid only when the RTRY bit in this word is set.

6.1.7. LANCE Operation

The LANCE chip operates independently of the host under control of its own internal microprogram. This section is a simplified description of the operation of the LANCE in terms of its principal microcode routines (these should not be confused with device driver programming in the host, which is not a part of this specification). These microcode routines make use of numerous temporary storage cells within the LANCE chip; most of these are not accessible from outside the chip but they are mentioned here when necessary to clarify the operation of the microcode.

Two such (conceptual) internal variables are of central importance: the pointers to the "current" entry in the receive descriptor ring and in the transmit descriptor ring, which are referred to below as TXP and RXP. Each of these designates the descriptor which the LANCE will use for the next operation of that type. If the descriptor designated by one of these pointers is not owned by the LANCE (the OWN bit is 0), then the LANCE can neither perform activity of that type nor advance the pointer. For the transmit ring, the LANCE will do nothing until the host sets up a packet in the buffer and sets the OWN bit in the descriptor designated by the LANCE’s TXP. (The host must keep track of the position of the TXP, since setting up a packet in some other descriptor will not be detected by the LANCE). For the receive ring, if the LANCE does not own the descriptor designated by RXP, it cannot receive a packet. In both rings, when the LANCE finishes with a descriptor and relinquishes it to the host by clearing OWN, it then advances the ring pointer (modulo the number of entries in the ring).

When the LANCE begins activity using the current descriptor (i.e. begins receiving or transmitting a packet), it may look ahead at the next descriptor and attempt to read its first three words in advance so it can chain to the next buffer in mid-packet without losing data. However, it does not actually advance its RXP or TXP until it has cleared the OWN bit in the current descriptor.

The LANCE is a very complex chip and this system specification does not attempt to cover all the details of its operation. The chip purchase specification and the chip vendor’s literature should also be consulted.

6.1.7.1. Switch Routine

Upon power on, the STOP bit is set and the INIT and STRT bits are cleared in LANCE_CSR0. The LANCE microprogram begins execution in the switch routine, which tests the INIT, STRT, and STOP bits. When the host sets either INIT or STRT, STOP is cleared. While STOP is set, if the host writes to LANCE_CSR1 and LANCE_CSR2, that data is stored for use by the initialization routine.

When the microprogram sees STOP cleared, it tests first the INIT bit and then the STRT bit. If INIT is set, it performs the initialization routine. Then if STRT is set, it begins active chip operation by jumping to the look-for-work routine. Control returns to the switch routine whenever the host again sets the STOP bit (which also clears the INIT and STRT bits). Note that the ring pointers RXP and TXP are not altered by the setting of either STOP or START; they are reset to the start of their rings only when INIT is set.

6.1.7.2. Initialization Routine

The initialization routine is called from the switch routine when the latter finds the INIT bit set. It reads the initialization block from the memory addressed by LANCE_CSR1 and LANCE_CSR2 and stores its data within the LANCE chip. This routine also sets the ring pointers RXP and TXP to the start of their rings (i.e. to point to the descriptor at the lowest memory address in the ring).

6.1.7.3. Look-for-work Routine

The look-for-work routine is executed while the LANCE is active and looking for work. It is entered from the switch routine when the STRT bit is set, and is returned to from the receive and transmit routines after they have received or transmitted a packet.

This routine begins by testing whether the receiver is enabled (bit RXON of LANCE_CSR0 is set). If so, it tries to have a receive buffer available for immediate use when a packet addressed to this system arrives. It tests its internal registers to see whether it has already found a receive descriptor owned by the LANCE and, if not, calls the receive poll routine to attempt to get a receive buffer.

Next the routine tests whether the transmitter is enabled (bit TXON of LANCE_CSR0 is set). If so, it calls the transmit poll routine to see whether there is a packet to be transmitted and to transmit it if so.

If there was no transmission and the TDMD bit of LANCE_CSR0 is not set, the microprogram delays 1.6 milliseconds and then goes to check the receive descriptor status again. If a packet was transmitted or the host has set TDMD, the delay is omitted so that multiple packets will be transmitted as quickly as possible.

If at any point in this routine the receiver detects an incoming packet whose destination address matches the station’s physical address, is the broadcast address, or passes the multicast address filter (or if the PROM bit of NIB_MODE is set), the receive routine is called.

6.1.7.4. Receive Poll Routine

The receive poll routine is called whenever the receiver is enabled and the LANCE needs a free buffer from the receive descriptor ring. The routine reads the second word of the descriptor designated by RXP and, if the OWN bit in it is set, reads the first and third words also.

6.1.7.5. Receive Routine

The receive routine is called when the receiver is enabled and an incoming packet’s destination address field matches one of the criteria described above. The routine has three sections: initialization, lookahead, and descriptor update.

In initialization, the routine checks whether a receive ring descriptor has already been acquired by the receive poll routine. If not, it makes one attempt to get the descriptor designated by RXP (if OWN is not set in it, MISS and ERR are set in LANCE_CSR0 and the packet is lost). The buffer thus acquired is used by the receive DMA routine to empty the silo.

In lookahead, the routine reads the second word of the next descriptor in the receive ring and, if the OWN bit is set, reads the rest of the descriptor and holds it in readiness for possible data chaining.

The descriptor update section is performed when either the current buffer is filled or the packet ends. If the packet ends but its total length is less than 64 bytes, it is an erroneous "runt packet" and is ignored: no status is posted in the descriptor, RXP is not moved, and the buffer will be reused for the next incoming packet (this is why a receive buffer must be at least 64 bytes long; otherwise the runt might be detected after advancing RXP).

If the packet ends (with or without error), the routine writes the packet length into MCNT, sets ENP and other appropriate status bits and clears OWN in the current descriptor, and sets RINT in LANCE_CSR0 to signal the host that a complete packet has been received. Then it advances RXP and returns to the look-for-work routine.

If the buffer is full and the packet has not ended, chaining is required. The routine releases the current buffer by writing status bits into its descriptor (clearing OWN and ENP, in particular), makes current the next descriptor data acquired in the lookahead section, advances RXP, and goes to the lookahead section to prepare for possible additional chaining. Note that RINT is not set in LANCE_CSR0, although the host would find OWN cleared if it looked at the descriptor, and it could begin work on that section of the packet, since the mutual exclusion rule prevents the LANCE from going back and altering it.

6.1.7.6. Receive DMA Routine

The receive DMA routine is invoked asynchronously by the chip hardware during execution of the receive routine whenever the silo contains 16 or more bytes of incoming data or when the packet ends and the silo is not empty. It executes DMA cycles to drain data from the silo into the buffer designated by the current descriptor.

6.1.7.7. Transmit Poll Routine

The transmit poll routine is called by the look-for-work routine to see whether a packet is ready for transmission. It reads the second word of the descriptor designated by TXP and tests the OWN bit. If OWN is zero, the LANCE does not own the buffer and this routine returns to its caller. If OWN is set, the routine tests the STP bit, which should be set to indicate the start of a packet. If STP is clear, this is an invalid packet; the LANCE sets its OWN bit to return it to the host, sets TINT in LANCE_CSR0 to notify the host, and advances TXP to the next transmit descriptor. If both OWN and STP are set, this is the beginning of a packet, so the transmit poll routine reads the rest of the descriptor and then calls the transmit routine to transmit the packet. During this time the chip is still watching for incoming packets from the network and it will abort the transmit operation if one arrives.

6.1.7.8. Transmit Routine

The transmit routine is called from the transmit poll routine when the latter finds the start of a packet to be transmitted. This routine has three sections: initialization, lookahead, and descriptor update.

In initialization, the routine sets the chip’s internal buffer address and byte count from the transmit descriptor, enables the transmit DMA engine, and starts transmission of the packet preamble. It then waits until the transmitter is actually sending the bit stream (including possible backoff-and-retry actions in case of collisions).

In lookahead, the transmit routine test the current descriptor to see whether it is the last in the packet (the ENP bit is set). If so, no additional buffer is required so the routine waits until all the bytes from the current packet have been transmitted. If not, the routine attempts to get the next descriptor and hold it in readiness for data chaining, and then waits until all the bytes from the current buffer have been transmitted.

Descriptor update is entered when all the bytes from a buffer have been transmitted or an error has occurred. If there is no error and the buffer was not the last of the packet, the pre-fetched descriptor for the next buffer is made current for use by the transmit DMA routine. The routine writes the appropriate status bits and clears the OWN bits in the current descriptor and advances TXP. If this was the last buffer in the packet, the routine sets the TINT bit in LANCE_CSR0 to notify the host and returns to the look-for-work routine; otherwise it goes back to the lookahead section in this routine.

6.1.7.9. Transmit DMA Routine

The transmit DMA routine is invoked asynchronously by the chip hardware during execution of the transmit routine whenever the silo has 16 or more empty bytes. It executes DMA cycles to fill the silo with data from the buffer designated by the current descriptor.

6.1.7.10. Collision Detect Routine

This routine is invoked asynchronously by the chip hardware during execution of the transmit routine when a collision is detected on the network. It ensures that the "jam" sequence is transmitted, then backs up the chip's internal buffer address and byte count registers, waits for a pseudo-random backoff time, and then attempts the transmission again. If 15 retransmission attempts fail (a total of 16 attempts), it sends the microcode to the descriptor update routine to report an error in the current transmit descriptor (bits RTRY and ERR are set).

6.1.7.11. LANCE Programming Notes

1. The interrupt signal is simply the OR of the interrupt-causing conditions. If another such condition occurs while the interrupt signal is already asserted, there will not be another active transition of the interrupt signal and the interrupt request bit in INT_REQ will not be set again. An interrupt service routine should use logic similar to the following to avoid losing interrupts:

   - Read LANCE_CSR0 and save the results.
   - Clear the interrupt enable bit INEA in the saved copy.
   - Write LANCE_CSR0 with the saved copy. This will make the interrupt signal false because INEA is clear and will clear all the write-one-to-reset bits such as RINT, TINT and the error bits; it will not alter the STRT, INIT or STOP bits nor any interrupt-cause bits which came true after LANCE_CSR0 was read.
   - Write LANCE_CSR0 with only INEA to enable interrupts again.
   - Service all the interrupt and error conditions indicated by the flags in the saved copy.
   - Exit from the interrupt service routine.

   1. An interrupt is signaled to the host only when the last buffer of a multibuffer (chained) packet is received or transmitted. However, the OWN bit in each descriptor is cleared as soon as the LANCE has finished with that portion of the packet, and the mutual exclusion rule makes it safe for the host to process such a descriptor and its buffer.

   2. When a transmitter underflow occurs (UFLO is set in a transmit descriptor because the silo is not filled fast enough), the LANCE will turn off its transmitter and the LANCE must be restarted to turn the transmitter back on again. This can be done by setting STOP in LANCE_CSR0 and then setting STRT in LANCE_CSR0 (DTX will still be clear in the chip's internal copy of NIB_MODE). It is not necessary to set INIT to reread the initialization block.

      Note that setting STOP will immediately terminate any reception which is in progress. If the status of a receive descriptor has been updated and its OWN bit is now clear, then the contents of its buffer are valid. If the incoming packet was chained into more than one buffer, however, the packet is only valid if its last buffer has been completed (the one with the ENP bit set).

3. The network transceiver requires up to five seconds after power on to become stable. Self-test routines must delay at least this time before attempting to use the controller for either internal or external testing.

4. The LCAR flag (loss of carrier) may be set in the transmit descriptor when a packet is sent in internal loopback mode. When the LANCE is operating in internal loopback mode and a transmission is attempted with a non-matching address, the LANCE will correctly reject that packet. If the next operation is an internal loopback transmission without first resetting the LANCE, the packet will not be sent and LCAR will be set in the transmit descriptor for that packet. The receive descriptor will still be owned by the LANCE. To avoid this problem, the LANCE should be reinitialized after each internal loopback packet.

5. The ONE flag is occasionally set in a transmit descriptor after a late collision. The LANCE does not attempt a retransmission even though ONE may be set. The host should disregard ONE if the LCOL flag is also set.

6. The chip’s internal copy of LANCE_CSR1 may become invalid when the chip is stopped. The LANCE_CSR1 and LANCE_CSR2 registers should always be loaded prior to setting INIT to initialize the LANCE chip.

7. Attempting an external loopback test on a busy network can cause a silo pointer misalignment if a transmit abort occurs while the chip was preparing to transmit the loopback packet. The resulting retransmission may cause the transmitter enable circuit to hang, and the resulting illegal length transmission must be terminated by the jabber timer in the transceiver. It is unlikely that there will be a corrupted receive buffer because the reception that caused the transmit abort will usually not pass address recognition.

   Since external loopback is a controlled situation it is possible to implement a software procedure to detect a silo pointer misalignment problem and prevent continuous transmissions. Since the test is being done in loopback the exact length and contents of the receive packet are known; thus the software can determine whether the data in the receive buffer has been corrupted.

   On transmission the diagnostic software should allow up to 32 retries before a hard error is flagged. This is not to say that 32 errors are allowed for each condition; the sum of all errors encountered in the test should not exceed 32. The diagnostic software should expect to get a transmit done interrupt with 1 millisecond of passing the transmit packet to the LANCE. If this does not occur, it should reset the LANCE and retry the test. This prevents a continuous transmission (babble) longer than the longest legal packet in case the LANCE has become hung.

8. When the chip is in internal loopback mode and a CRC error is forced, a framing error will also be indicated along with the CRC error. In external loopback, when a CRC error is forced only that error is indicated; a framing error is indicated only if the LANCE actually receives extra bits.

9. When transmit data chaining, a BUFF error will be set in the current transmit descriptor if a late collision or retry error occurred while the LANCE was still transmitting data from the previous buffer. The BUFF error in this case is an invalid error indication and should be ignored. BUFF is valid only when UFLO is also set.

10. When the host program sets up a packet for transmission in chained buffers, it should set the OWN bits in all the transmit buffers *except the first one* (i.e. the one containing the STP bit), and then as its last act set the OWN bit in the first descriptor. Once that bit is set, the LANCE will start packet transmission and may encounter an underflow error if the subsequent descriptors for the packet are not available.

11. Do not set INIT and STRT in LANCE_CSR0 at the same time. After stopping the chip, first set INIT and wait for IDON, then set STRT. If both are set at once, corrupt transmit or receive packets can be generated if RENA becomes true during the initialization process.

12. Since neither the LANCE nor the network buffer support parity bits, it is recommended that operating system software always calculate and verify the software checksums present in the packets of higher-level network protocols.

6.2. Network Buffer

The network buffer is a 32K-by-32-bit SRAM buffer that supports byte, half-word, tribyte, and word reads and writes from the processor side. The ~bm[3..0] field in the I/O bus is latched with the address and only the appropriate bytes are written into the network buffer. Upon reads the whole word is supplied, and the ~bm field is ignored, with the R3000 performing the appropriate byte masking and alignment. From the LANCE side the network buffer supports byte and half-word reads and writes in the DMA mode.

Reads of the buffer nominally stall the CPU for 9 cycles; a peak read bandwidth of 11.1 MBytes/second. Writes to the buffer nominally complete at a rate of 6 cycles per word; a peak write bandwidth of 16.67 MBytes/second. Buffer access during LANCE activity may increase the access latency by up to 7 additional cycles.

6.3. Ethernet Diagnostic ROM

The Ethernet Diagnostic ROM is located at 0xbb+1C0000. The ROM is 32K-by 8-bits and is word-aligned. The data is presented on the least significant byte of the data bus. A write to this address space will not return ready, thus causing a timeout on the bus. A read of the ROM will nominally stall the CPU for 13 cycles. The diagnostic ROM is in a socket.

6.4. Ethernet Station Address ROM

The Ethernet Station Address ROM (ESAR) is located at 0xbb+1C0000. The ROM is 32 by 8-bits and is word-aligned. The data is presented on byte 2, bits 16 through 23 of the data bus. The address space for ESAR is the same as the diagnostic ROM. A write to this address space will not return ready, thus causing a timeout on the bus. A read of the ROM will nominally stall the CPU for 13 cycles. The ESAR ROM is in a socket. Table 6-1 lists the addresses of the ESAR.

Table 6-1: Ethernet Station Address ROM Addresses

Address	Content			Value
0xbb+1C0002	Address	Octet	0	
0xbb+1C0006	"	"	1	
0xbb+1C000A	"	"	2	
0xbb+1C000E	"	"	3	
0xbb+1C0012	"	"	4	
0xbb+1C0016	"	"	5	
0xbb+1C001A	Chksum	Octet	1	
0xbb+1C001E	"	"	2	
0xbb+1C0022	"	"	2	
0xbb+1C0026	"	"	1	
0xbb+1C002A	Address	Octet	5	
0xbb+1C002E	"	"	4	
0xbb+1C0032	"	"	3	
0xbb+1C0036	"	"	2	
0xbb+1C003A	"	"	1	
0xbb+1C003E	"	"	0	
0xbb+1C0042	Address	Octet	0	
0xbb+1C0046	"	"	1	
0xbb+1C004A	"	"	2	
0xbb+1C004E	"	"	3	
0xbb+1C0052	"	"	4	
0xbb+1C0056	"	"	5	
0xbb+1C005A	Chksum	Octet	1	
0xbb+1C005E	"	"	2	
0xbb+1C0062	TEST	Pattern	0	FF
0xbb+1C0066	"	"	1	00
0xbb+1C006A	"	"	2	55
0xbb+1C006E	"	"	3	AA
0xbb+1C0072	"	"	4	FF
0xbb+1C0076	"	"	5	00
0xbb+1C007A	"	"	6	55
0xbb+1C007E	"	"	7	AA