DEC LANcontroller 400

DEC LANcontroller 400 Technical Manual

Order Number: EK-DEMNA-TM-001

This manual describes the DEC LANcontroller 400 (also called the DEMNA controller), which is an Ethernet/802 controller for systems that have an XMI bus. Detailed information is provided on hardware functions, installation, and diagnostics. The manual is intended for use by Digital customer service representatives, self-maintenance customers, and customer engineers and system programmers who incorporate DEC LANcontroller 400 modules into their own products or systems.

digital equipment corporation
maynard, massachusetts

First Printing, June 1990

DEBNA	PDP	VAXcluster
DEC	ULTRIX	VAXELN
DEC LANcontroller	UNIBUS	VMS
DECnet	VAX	XMI
DECUS	VAXBI	digital™

Contents

EXAMPLES

FIGURES

TABLES

PAGE xvi INTENTIONALLY LEFT BLANK

Preface

Purpose of This Manual

This manual describes the DEC LANcontroller 400, which is an Ethernet/802 controller for systems that have an XMI bus. The manual is a technical reference for installing, troubleshooting, repairing, and writing or customizing device drivers or diagnostics for the DEMNA.

The DEC LANcontroller 400 is also called the DEMNA controller. Throughout the rest of this manual, the DEC LANcontroller 400 is referred to as the DEMNA.

Intended Audience

This manual is for:

- Digital or customer personnel who install or replace the DEMNA in the field
- Customer engineers and system programmers who incorporate DEMNA modules into their own products or systems

Before Using This Manual

The reader should be familiar with the following:

- VAX architecture, as described in the *VAX Architecture Reference Manual*, EY-3459E-DP
- Ethernet network, as described in the *Ethernet Operation and Maintenance Manual*, EK-DEC-SAOP
- XMI bus, as described in the *VAX 6000-400 System Technical User's Guide*, EK–640EB–TM
- IEEE Std 802.2-1985 (ISO DIS 8802/2)
- IEEE Std 802.3-1985 (ISO DP 8802/3)

Structure of Manual

The manual has 10 chapters and 13 appendixes, which are described below.

Chapter 1 introduces the DEMNA option and its overall architecture.

Chapter 2 describes the DEMNA port and its relationship to the DEMNA port driver.

Chapter 3 describes the data flow through the DEMNA, as well as the data flow between the DEMNA and host, for both transmits and receives.

Chapter 4 defines the DEMNA registers that are visible over the XMI bus.

Chapter 5 describes the DEMNA power-up, node reset, node halt/restart, and power-down sequences.

Chapter 6 describes the DEMNA power-up self-test and how to interpret the self-test results.

Chapter 7 describes diagnostics that apply to the DEMNA module. The chapter explains how to run the ROM-based diagnostics that are on the module. It also explains how to run software diagnostics under the VAX Diagnostic Supervisor.

Chapter 8 describes the error handling performed by the DEMNA port.

Chapter 9 describes how to modify flags and parameters in the DEMNA EEPROM.

Chapter 10 describes the DEMNA console monitor program.

Appendix A summarizes the DEMNA power and environmental requirements.

Appendix B describes how to install the DEMNA.

Appendix C describes how the DEMNA accomplishes booting over the Ethernet.

Appendix D describes how to convert an Ethernet address to a DECnet address.

Appendix E describes how to read the DEMNA Ethernet address.

Appendix F describes the errors for the DEMNA ROM-based diagnostics (RBDs).

Appendix G describes the DEMNA Node-Private Registers in the DEMNA nonfatal error blocks.

Appendix H lists the device type codes of all XMI modules available at the printing of this manual.

Appendix I lists some commonly used Ethernet protocol types.

Appendix J lists some commonly used multicast addresses.

Appendix K lists some commonly used 802 SAPs and SNAP SAP protocol IDs.

Appendix L describes a program that can be used to connect to the DEMNA console monitor program if the Network Control Program (NCP) is not available.

Appendix M describes describes the types of history entries that can be displayed by the console monitor program.

Associated Documents

Related documentation includes:

- *DEC LANcontroller 400 Installation Guide*, EK–DEMNA–IN
- *DEC LANcontroller 400 Console User's Guide*, EK–DEMNA–UG
- *DEC LANcontroller 400 Programmer's Guide*, EK–DEMNA–PG
- *Ethernet Installation Guide*, EK–ETHER–IN
- *VAX 9000 Family System Maintenance Guide, Vol. 2*, EK–DA902–MG
- *VMS I/O User's Reference Manual: Part II*, AA–LA84A–TE
- *VMS Network Control Program Manual*, AA–LA50A–TE

Conventions Used

- All addresses are in hexadecimal (hex). All bit patterns are in binary notation. All other numbers are decimal unless otherwise indicated.
- Ranges are inclusive. For example, the range 0–4 includes the integers 0, 1, 2, 3, 4.
- Bits are enclosed in angle brackets (for example, <12>).
- Bit ranges are indicated by two bits in descending order separated by a colon; for example, <12:1>. Bit ranges are inclusive.
- K = kilo (1024); M = mega (1024**2); G = giga (1024 **3).
- The term "asserted" indicates that a signal line is in the true state. The term "deasserted" indicates that a signal line is in the false state.

  "Assertion" is the transition from the false to the true state. "Deassertion" is the transition from the true to the false state.

Command Notation

The following command notation is used in this manual:

Convention	Meaning
{ }	Large braces enclose lists from which you must choose one item. For example: { KERNEL USER }
. . .	Horizontal ellipsis points mean that you can repeat the item preceding the points. For example: */qualifier* . . .
. . .	Horizontal or vertical ellipsis points in an example indicate that not all the information the system would display is shown or that not all the information a user is to supply is shown.
{ }, . . .	Braces followed by a comma and horizontal ellipsis points mean that you can repeat the enclosed items one or more times, separating two or more items with commas.
[ ]	Square brackets enclose items you can omit. For example: [=*option*, . . .]
UPPERCASE characters	Language-specific reserved words and identifiers are printed in uppercase characters. However, you can enter them in uppercase, lowercase, or a combination of uppercase and lowercase characters.
italic lowercase characters	Elements you must replace according to the description in the text are printed in italic lowercase characters. However, you can enter them in lowercase, uppercase, or a combination of lowercase and uppercase characters.

CHAPTER 1

1 DEC LANcontroller 400 Module Overview

The DEC LANcontroller 400 is an intelligent, high-performance I/O controller that enables a host processor on the XMI bus to communicate with other nodes in an Ethernet/802 local area network. The DEC LANcontroller 400 is compatible with the Ethernet and IEEE 802 specifications.[1] Digital's *Systems and Options Catalog* indicates which systems support the DEMNA option.

A single XMI bus can support multiple DEC LANcontroller 400s. An XMI bus can thus connect to multiple Ethernet/802 networks. Each DEC LANcontroller 400 connects to a single network through a standard 15-pin Sub-D connector.

The DEC LANcontroller 400 is also called the DEMNA controller. Throughout the rest of this manual, the DEC LANcontroller 400 is referred to as the DEMNA.

1.1 Basic Functions

The DEMNA supports one Ethernet/IEEE 802 port, which provides the physical link layer and portions of the data link communication layer of the Ethernet and 802 protocols, as defined by the Ethernet and IEEE 802 specifications.

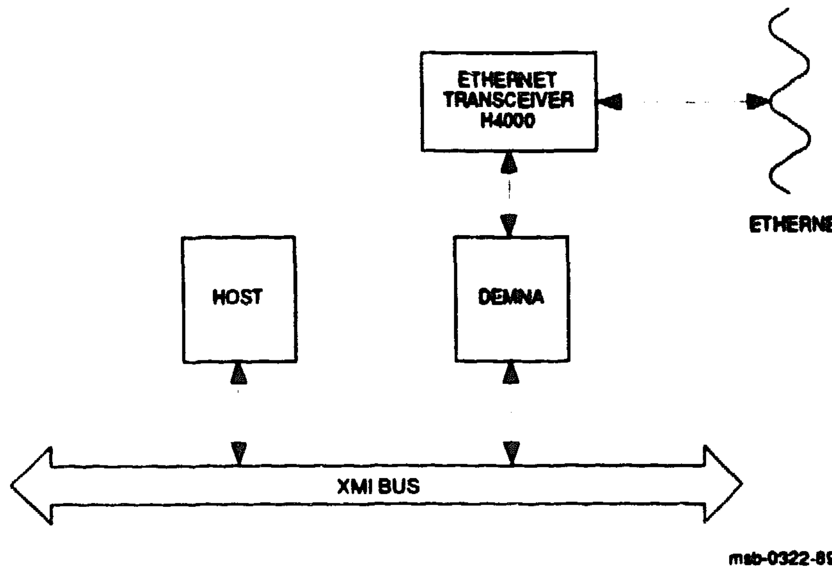
With its own onboard CVAX processor, the DEMNA can control operations independently of the host processor. The details of Ethernet transactions, including data transfer over the XMI bus, are thus transparent to the host processor (see Figure 1–1).

The onboard firmware is contained in EEPROM, which allows revised firmware to be loaded without hardware modification. The firmware can thus be easily upgraded in the field. In addition, various DEMNA operating parameters can be modified easily in the field.

The DEMNA firmware includes a console monitor program that allows a user at virtually any terminal on the network to monitor DEMNA operation and network traffic. The console monitor program can be accessed over the network or from a terminal (called the physical console) attached directly to the DEMNA.

[1] In this manual, 802 refers specifically to the CSMA/CD local area network defined in the IEEE 802.2 and 802.3 specifications (physical and data link layers).

Figure 1–1 DEMNA Module in an XMI System



The DEMNA has extensive onboard tests. On power-up or reset, the DEMNA tests itself and makes its status (pass or fail) available through LEDs on the module and through an onboard Power-Up Diagnostic (XPUD) Register. In addition, a customer service engineer may invoke other onboard diagnostics from the system console or the DEMNA physical console to test the DEMNA's logic and functionality more extensively.

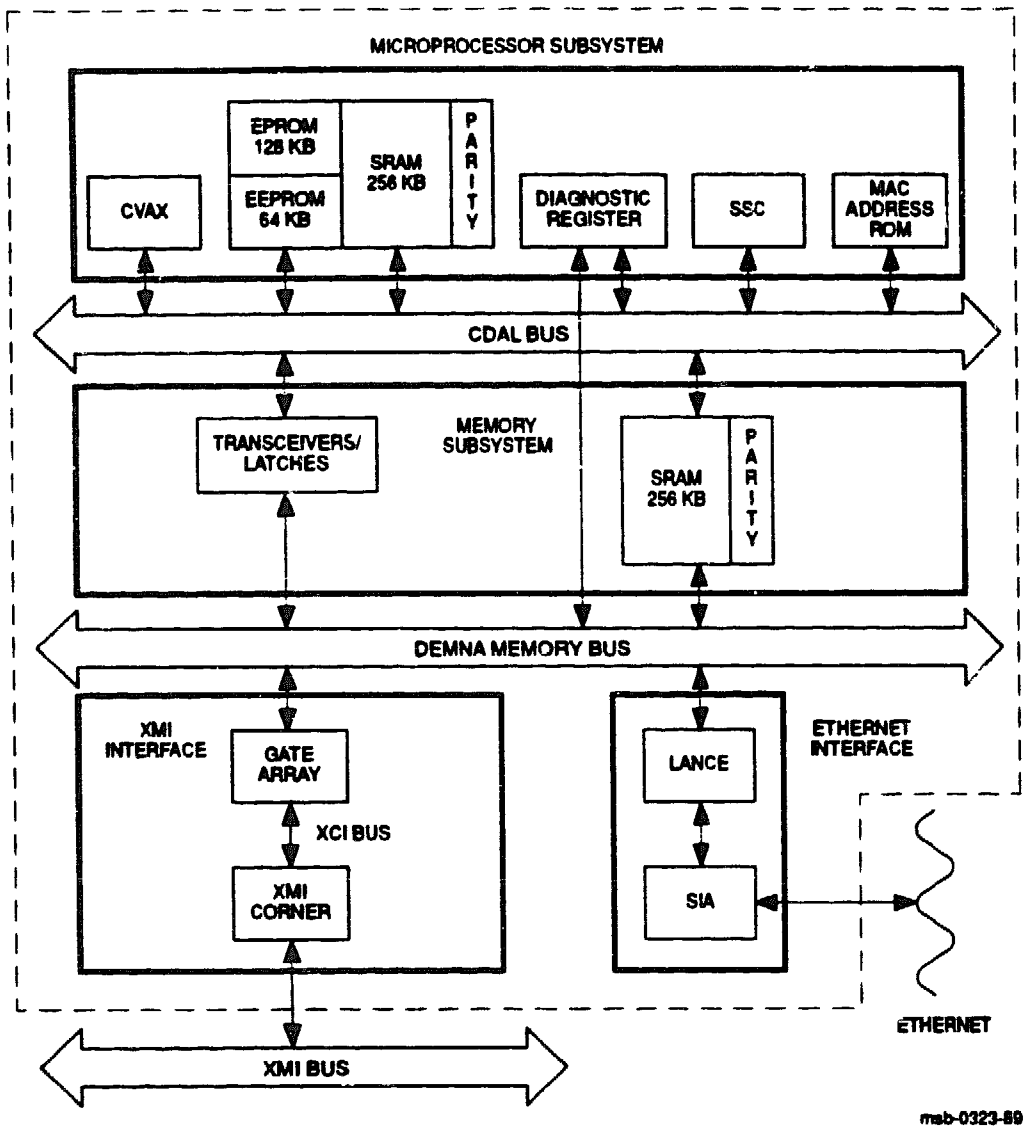
The DEMNA may participate in network boot operations. The DEMNA may be specified as the boot device by its host system or be enabled to involuntarily boot its host system on receiving a valid Boot message over the network. (See the Appendix C for further information.)

1.2 Logic Overview

The DEMNA logic is divided into the following four subsystems, as shown in Figure 1–2:

- Microprocessor subsystem
- Shared memory subsystem
- XMI interface subsystem
- Ethernet interface subsystem

Figure 1–2 DEMNA Simplified Block Diagram



1.2.1 Microprocessor Subsystem

The microprocessor subsystem performs the following major functions:

- Stores and executes the module firmware, including onboard diagnostics and the console monitor program
- Stores and supplies the module's default (Medium Access Control) Ethernet address

The microprocessor subsystem contains the following major components:

- CVAX—a 32-bit CMOS processor dedicated to running firmware. The CVAX cannot be used directly by application programs running on the host processor or by a user at the system console.

- System Support Chip (SSC)—This chip provides control logic for the microprocessor subsystem, including timers, address decode logic, internal processor registers, and a UART for connection with the DEMNA physical console.
- EEPROM and CVAX RAM—The EEPROM stores the module's operational firmware, which executes from CVAX RAM (SRAM). The EEPROM also provides a diagnostic patch area and stores history data on DEMNA failures.
- MAC Address (ENET) PROM—This PROM stores the module's default physical (Ethernet) address (DPA),[1] which is also called the Medium Access Control (MAC) address. The PROM also stores a PROM test pattern.
- EPROM—The EPROM stores a working copy of the DEMNA firmware minus the console monitor program. If the DEMNA self-test finds that the EEPROM contents are invalid, the EPROM code is loaded into CVAX RAM.
- Diagnostic Register—This register is a control/status register that controls certain low-level diagnostic operations, such as the disabling of CVAX RAM parity.

The CVAX, SSC, CVAX RAM, and Diagnostic Register connect to each other through the CDAL bus, which in turn connects to the DEMNA memory bus through latched transceivers.

1.2.2 Shared Memory Subsystem

The shared memory subsystem performs the following major functions:

- Buffers packets to and from the Ethernet interface
- Buffers transfers to and from the XMI bus
- Stores shared data structures that allow the CVAX and LANCE to communicate

The shared memory subsystem has the following major components:

- 256 Kbytes of parity-protected SRAM—The SRAM buffers Ethernet and XMI transfers and stores data structures shared by the CVAX and LANCE.
- Bus control logic—This logic controls read/write timing and read/write signals.
- DMA logic—This logic controls access to the SRAM.
- DEMNA timeout logic—This logic detects when a DMA grant on the DEMNA memory bus has been outstanding longer than the timeout period.

[1] At the request of applications starting up a protocol such as DECnet, the port driver may assign one or more alternative addresses to the DEMNA. This type of address is called an actual physical address (APA).

The SRAM is on the DEMNA memory bus, which connects to the Ethernet interface, XMI interface, and the CDAL bus. The SRAM can be accessed by the LANCE chip (Ethernet interface), CVAX, or DEMNA gate array (XMI interface). The DMA access priority for these devices is LANCE, CVAX, and gate array.

1.2.3 Ethernet Interface Subsystem

The Ethernet interface provides an interface between the DEMNA's shared memory and the Ethernet wire. The Ethernet interface performs transmits (reads) from the shared memory and receives (writes) to the shared memory.

The Ethernet interface has the following major components:

- Local Area Network Controller for Ethernet (LANCE) chip—The LANCE chip implements the microprocessor interface, performs DMA to and from the DEMNA shared memory, implements the CSMA/CD network access algorithm, does packet handling on transmits and receives, and reports errors.
- Serial Interface Adapter (SIA) chip—The SIA chip performs Manchester encoding for transmits, Manchester decoding for receives, and implements a TTL/differential signal interface between the LANCE (TTL) and the Ethernet wire (differential signals).
- Bus interface logic—This logic generates byte parity on transfers to DEMNA shared memory and checks byte parity on transfers from shared memory.

1.2.4 XMI Interface Subsystem

The XMI interface provides an interface between the DEMNA's shared memory and the XMI bus. The XMI interface performs the following major functions:

- Transfers Ethernet read and write data between DEMNA shared memory and host memory
- Performs control operations for the DEMNA CVAX (high-priority quadword XMI reads and writes to memory and longword XMI I/O reads and writes)
- Implements the DEMNA port registers
- Implements the XMI-required registers
- Implements XMI interrupt logic

The XMI interface has the following major components:

- Gate array—The gate array implements most of the XMI interface logic.
- XMI resistors, clocks, and module-decoupling capacitors.
- XMI timeout logic—This logic detects timeouts for XMI operations.

1.3 Physical Description

The DEMNA option consists of a single board. The DEMNA has the following two cables:

- An internal Ethernet cable that connects the DEMNA with the transceiver cable and provides power to an H4000 transceiver (Section B.4). This cable is not part of the DEMNA option but is included in the cabinet kits for the DEMNA.
- An internal cable for the physical console that connects the DEMNA with a terminal cable. This cable has its own cabinet kit.

An external Ethernet cable runs from the bulkhead to an Ethernet transceiver. This cable is not part of the DEMNA option and is not included in the cabinet kits for the DEMNA. The cable must be ordered separately. (See Digital's *Systems and Options Catalog* for more information.)

Table 1–1 lists the items on the DEMNA packing list. These are the items included with the DEMNA option (DEMNA-M).

Table 1–1 Packing List for DEMNA Option

Part Number	Quantity	Description
T2020	1	DEMNA module
EK-DEMNA-IN	1	*DEC LANcontroller 400 Installation Guide*
EK-DEMNA-RN	1	*DEC LANcontroller 400 Release Note*
EK-DEMNA-UG	1	*DEC LANcontroller 400 Console User's Guide*

1.3.1 Cabinet Kits

Table 1–2 lists DEMNA cabinet kits for VAX 6000 and 9000 systems. Cabinet kits must be ordered separately from the DEMNA option. For new systems not included in this table, please see Digital's *Systems and Options Catalog*.

Table 1–2 Cabinet Kits for DEMNA Options

Description	Kit Number	Contents
Internal Cable for Physical Console	CK-DEMNA-AM	I/O connector panel for VAX 6000 cabinets (74-26407-32) I/O connector panel for VAX 9000 cabinets (70-28010-01) Blank panel for I/O connector panel (74-26407-01) Internal cable for physical console (17-02168-01) *DEC LANcontroller 400 Installation Guide* (EK-DEMNA-IN) *DEC LANcontroller 400 Console User's Guide* (EK-DEMNA-UG)
VAX 6000 cabinet kit	CK-DEMNA-KD	Ethernet I/O connector panel (74-26407-41) 8-ft. internal Ethernet cable (17-01496-02) Blank panel (74-26407-01) Ethernet loopback connector (12-22196-02)

Table 1–2 (Cont.) Cabinet Kits for DEMNA Options

Description	Kit Number	Contents
VAX 9000 Model 2xx cabinet kit	CK-DEMNA-KE	Ethernet I/O connector panel (70-27894-01) 3-ft. internal Ethernet cable (17-01496-01) Ethernet loopback connector (12-22196-02)
VAX 9000 Model 4xx cabinet kit	CK-DEMNA-KM	Ethernet I/O connector panel (70-27894-01) 8-ft. internal Ethernet cable (17-01496-02) Ethernet loopback connector (12-22196-02)

1.3.2 Status LEDs

The DEMNA module has two status-indicator lights:

- One DEMNA OK LED (yellow)—the self-test LED required by the XMI bus specification
- One External Loopback LED (green)

Immediately after power-up or reset, both status-indicator lights are off. If all the tests in the self-test pass (aside from the LANCE external loopback test), the self-test lights the yellow DEMNA OK LED. If the LANCE external loopback test passes (indicating that the DEMNA can transmit and receive a loopback packet over the network), the self-test lights the green External Loopback LED.

CHAPTER 2

2 DEMNA Port Overview

This chapter provides an overview of the DEMNA port/port driver interface. For a detailed description of this interface, see the *DEC LANcontroller 400 Programmer's Guide*.

This chapter includes the following sections:

- Introduction
- Shared Data Structures
- Port Registers
- Port Commands
- Interrupts
- Port States
- Network Maintenance Operations Supported by the Port
- Packet Filtering and Validation Performed by the Port
- Port-Maintained Status

2.1 Introduction

The DEMNA implements a single Ethernet/802 port. This port provides the physical and data link communication layers of the Ethernet/802 network protocol.

The port is controlled by a single port driver running on the host. In a multiprocessor system, the port driver may run on different processors at different times; however, from the port's perspective, there is only one port driver.

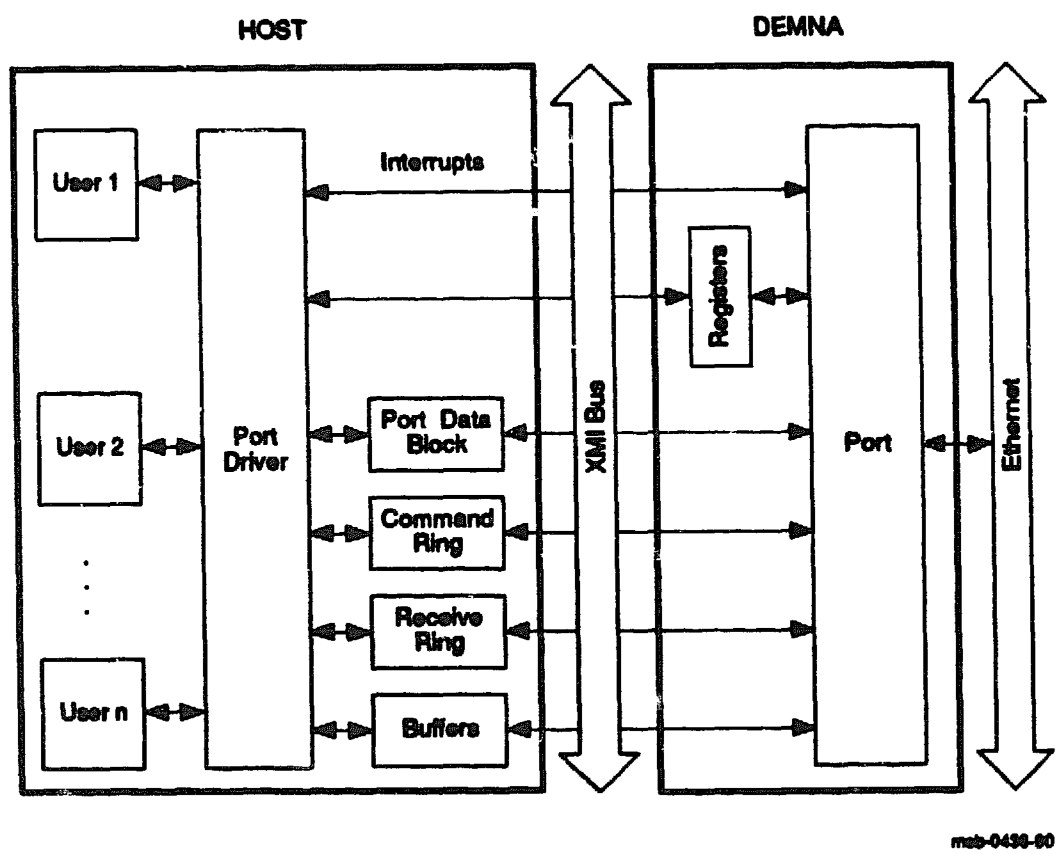
The DEMNA port has the following major features:

- A simple, logical interface
- Support for Ethernet and 802.2/802.3 protocols
- Support for up to 64 users, which may include an "unknown user" (a catchall user that transmits and receives packets for all users beyond the users defined to the port)
- 30-bit virtual-to-physical address translation, 34-bit virtual-to-physical address translation, 40-bit physical addressing, dual-segment 40-bit physical addressing (physical addressing across page boundaries)
- Multiple-buffer support (buffer chaining) for transmits and dual-buffer support for receives when dual-segment 40-bit addressing is enabled
- Packet filtering by destination address, multicast address, protocol type, SAP, GSAP, and SNAP SAP protocol identifier

- Support for multiple physical Ethernet addresses
- Maintenance commands that allow diagnostic programs to obtain detailed operating status and error status, load a new firmware image into DEMNA EEPROM, and change parameter settings in DEMNA EEPROM
- Support for Digital's Maintenance Operations Protocol (MOP) functions
- Console support for diagnostic routines and field-service functions
- A console monitor program that allows a user at virtually any terminal on the network to monitor DEMNA operation and network utilization
- Error logging

Figure 2–1 illustrates the port/port driver interface.

Figure 2–1 Port/Port Driver Interface



2.2 Shared Data Structures

The port and port driver share the following data structures, which are resident in host memory:

- Port Data Block (PDB)
- Command and receive rings
- Transmit, receive, and command buffers

These data structures provide the main means of communication and data transfer between the port and port driver.

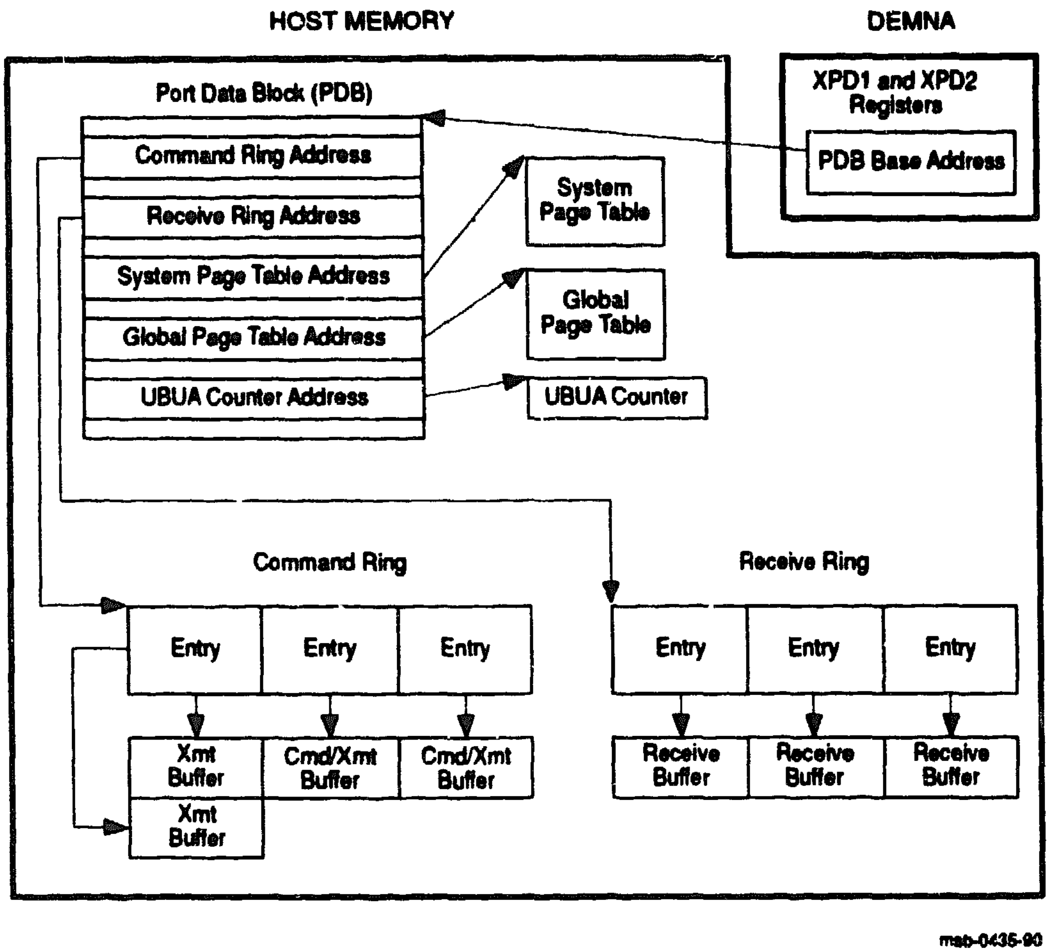
Figure 2–2 shows how the port finds the data structures resident in host memory. On port initialization, the port driver writes Port Data Registers 1 and 2 (XPD1 and XPD2) in the DEMNA with the physical base address of the Port Data Block (PDB). The PDB supplies the following:

- Base address of the System Page Table
- Base address of the Global Page Table
- Base address of the command ring
- Base address of the receive ring
- Address of the User Buffer Unavailable (UBUA) Counter

Each entry in the command and receive rings provides the base address of a buffer or, in the case of buffer-chained transmits or receives, the base addresses of multiple buffers.

The PDB also indicates to the port the addressing mode (virtual or physical) to use when accessing the data structures in host memory.

Figure 2–2 Port Access of Host Data Structures



2.2.1 Port Data Block (PDB)

The Port Data Block (PDB) is a 512-byte block aligned to a 512-byte boundary in host memory. Both the port and port driver can read and write the PDB.

Before commanding the port to initialize itself, the port driver writes the PDB with initialization parameters for the port. The PDB indicates the following to the port:

- The addressing mode (virtual or physical) that the port should use
- The location of the command and receive rings and the size of the command ring entries
- The location and size of the system page table (used only if virtual addressing is specified)
- The location of the global page table (used only if virtual addressing is specified)
- Interrupt information for both error and port interrupts, including which node(s) to interrupt, the interrupt vector to return, and the level at which to interrupt
- The location and length of the User Buffer Unavailable (UBUA) Counter, which indicates how many receive packets the port driver has discarded because no user buffers were available.

The PDB also contains the following counters:

- Potential SBUA Counter—This counter records how many times the port looked for but did not find a system buffer when attempting to deliver a receive packet to the port driver. If the port does not obtain a system buffer within 3 seconds, it discards the receive packet. The operating system can monitor this counter to optimize the number of buffers it allocates to the DEMNA.
- Actual SBUA Counter—This counter records the number of receive packets discarded by the port because (1) the port could not obtain a system buffer and did not have sufficient internal memory space to buffer the packet or (2) because the port could not obtain a system buffer within 3 seconds and the packet became stale (held longer than 3 seconds by the port).
- DOR Counter—This counter records the number of receive packets discarded by the port because the DEMNA hardware or firmware was unable to keep up with the data rate on the network.

Finally, the port writes error data to the PDB when a fatal port error occurs or after the port executes a node halt/restart (Section 5.2) from the initialized state. (When the port receives a node halt/restart in the initialized state, it assumes that it is being restarted from an error.) The port writes the PDB with a fatal error block (Section 8.2.1) that describes the fatal port error.

Figure 2–3 shows the layout of the PDB.

Figure 2-3 Port Data Block Layout

31 … 16	15 … 8	7 … 0	Offset (Hex)
Reserved		Addr Mode	0
Command Ring Address <31:0>			4
Command Ring Entry Size	0s	<39:32>	8
Receive Ring Address <31:0>			C
0s		<39:32>	10
System Page Table Length			14
System Page Table Address <31:0>			18
0s		<39:32>	1C
Global Page Table Address (Sva)			20
Host Interrupt Data XMI Node ID mask	Vector <15:2>	Level <1:0>	24
Host Error Interrupt Data XMI Node ID mask	Vector <15:2>	Level <1:0>	28
Driver UBUA Counter Address <31:0>			2C
UBUA Counter Length	0s	<39:32>	30
Potential SBUA Counter			34 38
SBUA Counter			3C 40
DOR Counter			44 48
Reserved to Port Driver (52 bytes)			4C 7C
Port Error Log Area (128 bytes)			80 FC
Reserved to Port (256 bytes)			100 1FF

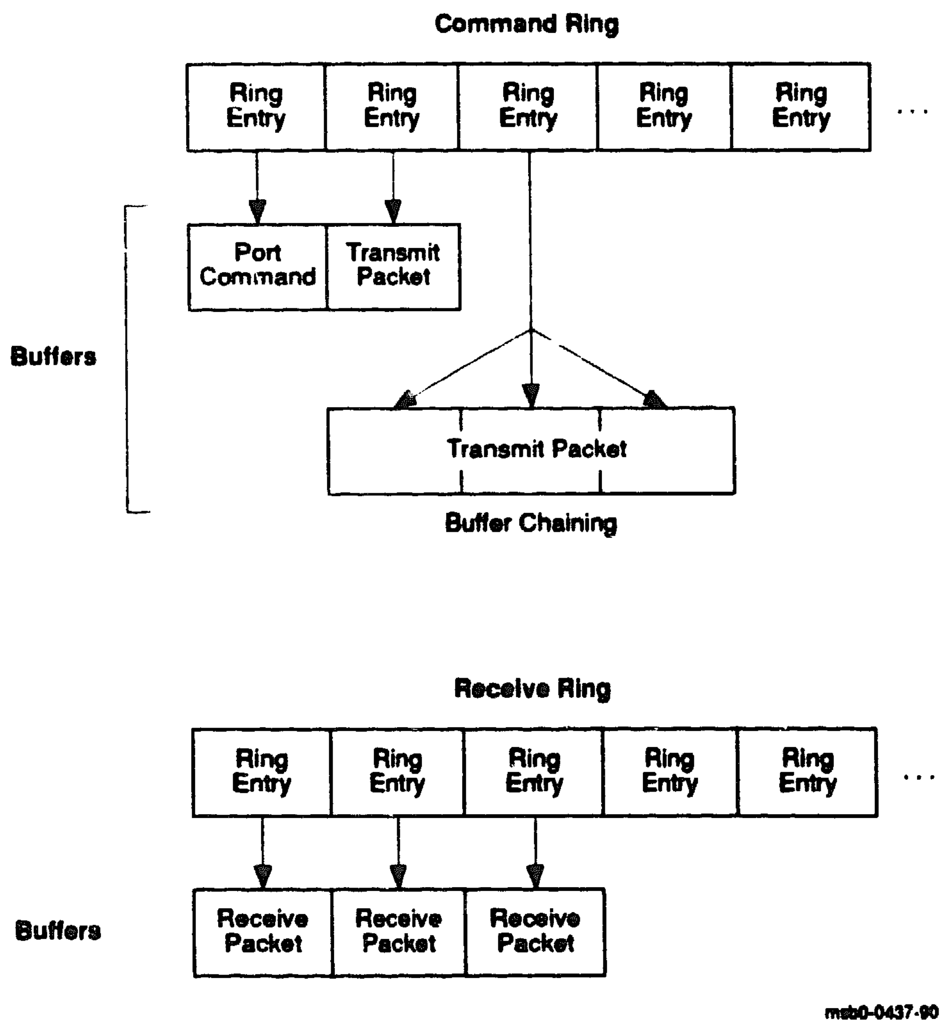
msb0430-90

2.2.2 Rings and Buffers

The command and receive rings each consist of 1024 contiguous bytes aligned to a 512-byte boundary. Each ring contains entries that point to a buffer or (for buffer-chained transmits) to a series of buffers (Figure 2–4).

Command ring buffers contain port commands or transmit packets. Receive ring buffers contain receive packets. The ring entries also contain error status on the command, transmit, or receive, as well as a user index that identifies the user on whose behalf the command or receive (but not transmit) is being performed.

Figure 2–4 Rings and Buffers



2.2.2.1 Command Ring

The port driver uses the command ring to transfer transmit packets and commands to the port as follows:

1. When the port driver has a command or transmit packet to send to the port, it obtains a command ring entry, builds the command or transmit packet in the buffer addressed by the ring entry, writes appropriate information to the ring entry, and then transfers ownership of the ring entry to the port.
2. The port copies the ring entry and buffer contents into its internal memory, executes the command or transmit, writes error status back to the command ring entry, transfers ownership of the ring entry back to the port driver, and interrupts the port driver if port interrupts are enabled.
3. The port driver returns status or error information to higher-level software and logs any errors as appropriate.

A command ring entry can point to multiple transmit buffers. This practice, which is called buffer chaining, reduces host overhead for transmits but increases port overhead for transmits. Buffer chaining can be used only for transmits—not for port commands. Each port command must reside in a single buffer.

Figure 2–5 shows the layout of a command ring entry.

Figure 2–5 Command Ring Entry Layout

31	30	29	28	27–24	23–16	15–8	7–0	Offset (Hex)
O	E	C	M	Rsvd	Error Code	No. of Buffers	User Index	0
Reserved to Port Driver								4
Buffer Segment Address 1 <31:0>								8
Segment Length 1						0s	<39:32>	C
Buffer Segment Address 2 <31:0>								10
Segment Length 2						0s	<39:32>	14
. . .								
Buffer Segment Address n <31:0>								n
Segment Length n						0s	<39:32>	n + 4

msb-0431-90

2.2.2.2 Receive Ring

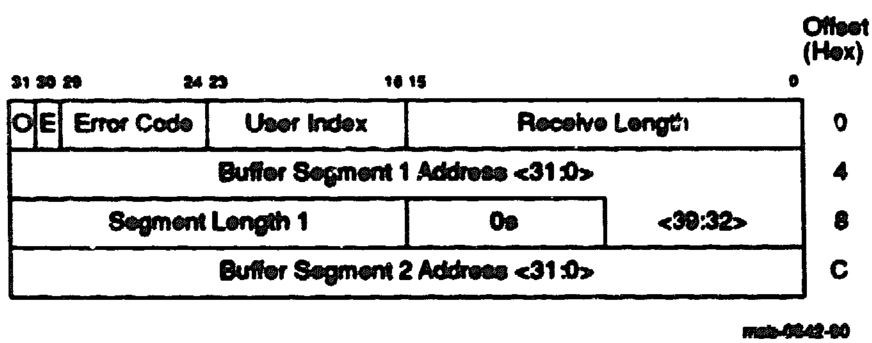
The port uses the receive ring to transfer receive packets to the port driver as follows:

1. When the port has a receive packet to send, it obtains a receive ring entry and copies the packet from internal memory to the host receive buffer addressed by the ring entry. The port also writes error status, the user index, and the packet length to the ring entry.
2. The port transfers ownership of the receive ring entry to the port driver and interrupts the port driver if port interrupts are enabled.
3. The port driver accesses the entry and the receive buffer and then transfers the receive packet, including error information provided by the port, to the appropriate higher-level user.
4. The port driver creates another receive buffer for the receive ring or, after the user has processed the receive packet, returns the original receive buffer to the receive ring.

When dual-segment 40-bit physical addressing is enabled, a receive ring entry can point to two receive buffer segments. This allows a receive packet to be buffered across a page boundary. The second buffer segment must be on a 512-byte page boundary.

Figure 2–6 shows the layout of a receive ring entry.

Figure 2–6 Receive Ring Entry Layout



2.2.2.3 Ring Processing

The port and port driver process the entries in each ring in sequential order, starting with the first entry. A ring entry can be processed only when owned by the party (port or port driver) doing the processing. The port, for example, can process only the entries that it owns. The same holds true for the port driver.

When the last entry in a ring is reached, processing starts again with the first entry (Figure 2–7). Both the port and port driver stop processing a ring on encountering an entry that is owned by the other party.

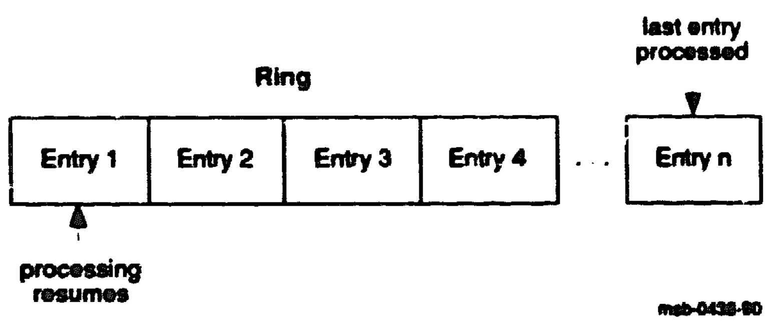
The port and port driver pass command ring entries to each other as follows:

1 After the port driver has assembled the appropriate command or transmit buffer(s) and written the necessary information to a command ring entry, it transfers ownership of the entry to the port.
2 After completing the command or transmit, the port writes the error status of the command or transmit to the ring entry and returns ownership of the entry to the port driver.
3 The port driver then checks the error status but does not transfer ownership of the entry back to the port until there is another command or transmit packet to send by means of the entry.

The port and port driver pass receive ring entries to each other as follows:

1 After assembling a receive buffer and writing the necessary information to a ring entry, the port transfers ownership of the entry to the port driver.
2 The port driver reads the ring entry, obtains the receive packet from the receive buffer, copies the receive packet to a system buffer, writes the address of a receive buffer to the ring entry (if a receive buffer is available), and transfers ownership of the ring entry back to the port.

Figure 2-7 Ring Processing



2.2.3 Port Polling of Rings

The port polls the command ring whenever the port driver writes the Port Control Poll (XPCP) Register. The port driver should write the XPCP Register once for each command or transmit packet that it places in the command ring.

The port polls the receive ring whenever the port has a receive packet to deliver to the port driver but does not own any of the receive ring entries. The port then polls the receive ring until it obtains an entry.

2.2.4 Ring Release Function

The ring release function enables the port to inform the port driver through a port interrupt that the port driver should resume processing the rings. The port does this only when it suspects that the port driver will not otherwise resume processing the rings.

The ring release function works as follows (Figure 2–8):

1. The port and port driver each maintain an internal ring release counter. The port's internal ring release counter indicates how far the port driver should have progressed through the rings. The port driver's internal ring release counter indicates how far the port driver has in fact progressed through the rings. The port and port driver both initialize their respective internal ring release counters to –1.
2. After processing all the entries that it owns in both the command and receive rings, the port driver copies its internal ring release counter to Port Data Register 2 (XPD2).
3. On detecting that the XPD2 Register has been written, the port compares the count in its internal ring release counter with the ring release count in the XPD2 Register.
4. If these counts differ, the port knows that the port driver has missed processing one or more ring entries. The port then sends a port interrupt to the port driver if port interrupts are enabled.
5. On receiving the interrupt, the port driver resumes processing of the rings.

NOTE: **The suggested order of ring processing for the port driver is as follows:**

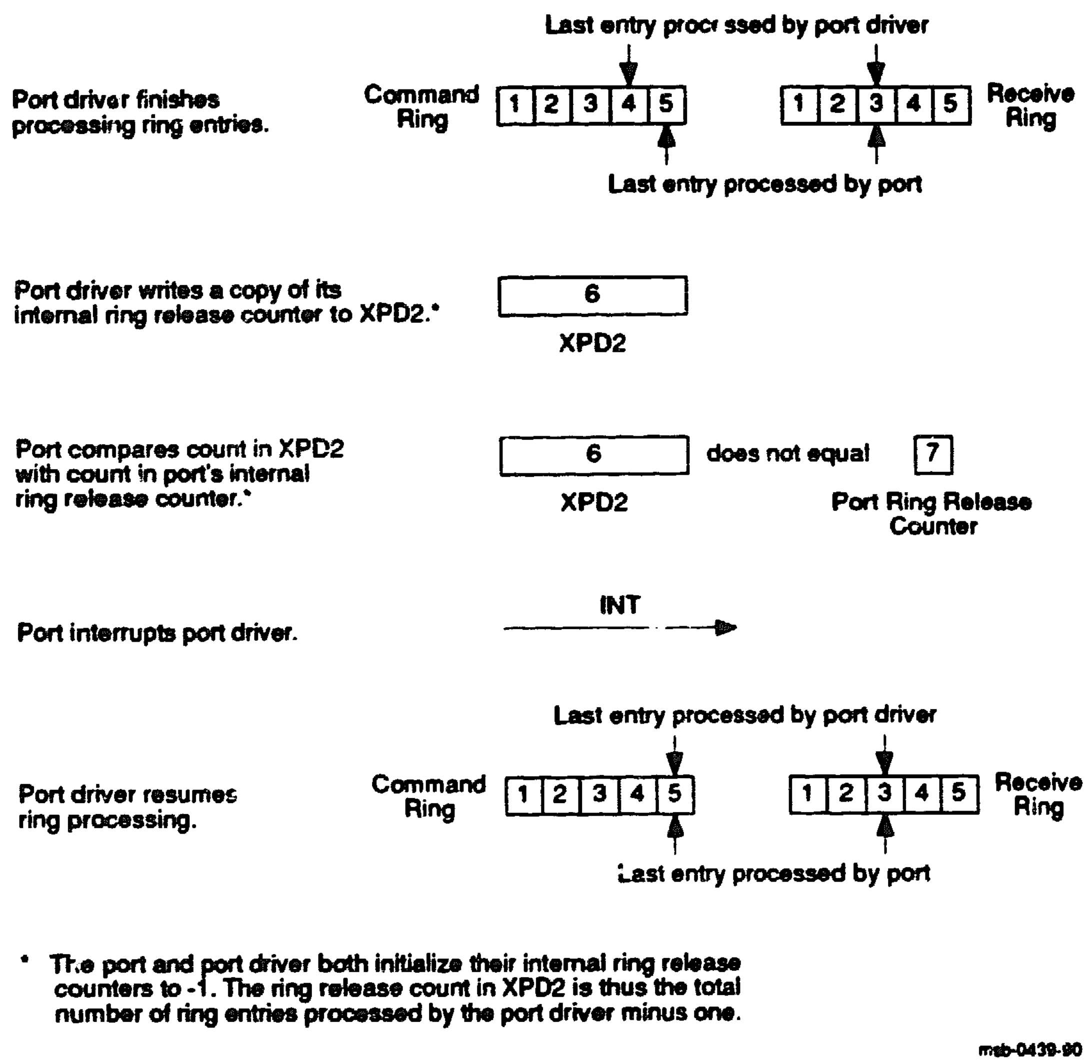
1. **Process the command ring**
2. **Process the receive ring**
3. **Write the ring release count to the XPD2 Register**

2.3 Port Registers

The port registers are used by the port and port driver to communicate with each other during normal operation of the port. The port registers are divided into the following groups:

- Port Status Register

Figure 2-8 Ring Release Example



- **Port Control Registers**
- **Port Data Registers**

Table 2-1 provides a brief description of each port register. For a detailed description of the port registers, see Section 4.3.

Table 2-1 Port Registers—Summary

Register	Mnemonic	Description
Port Status	XPST	The port writes this register to indicate a change in port state and the reason for the change. The port also writes error information to this register when a fatal port error aborts an attempted port initialization or shutdown. The port driver should thus read the Port Status Register to confirm port state transitions (after commanding the port to initialize itself, for example) and after receiving an error indication from the port.
Port Control Registers		
Port Control Initialization	XPCI	The port driver writes this register to command the port to initialize the data structures that it shares with the port driver.
Port Control Poll	XPCP	The port driver writes this register to command the port to poll the command ring.
Port Control Shutdown	XPCS	The port driver writes this register to command the port to shut down.
Port Data Registers		
Port Data Register 1 Port Data Register 2	XPD1 XPD2	The port driver writes the Port Data Registers to supply the port with the base address of the Port Data Block (PDB) before initializing the port. In addition, after finishing processing all the entries that it owns in both the command and receive rings, the port driver writes the XPD2 Register with the ring release count to let the port know how far the port driver has progressed through the rings (Section 2.2.4). The port writes the Port Data Registers to convey to the port driver the DEMNA's default MAC (Medium Access Control) address, as well as various error information.

2.4 Port Commands

In addition to the commands that it can issue to the port by writing the Port Control Registers, the port driver can issue various port commands through the command ring. To do this, the port driver builds a command buffer, obtains a command ring entry, writes the ring entry with the buffer address, and then transfers ownership of the entry to the port. The port copies the command into its internal memory, executes the command, and returns appropriate data and status to the command buffer.

The port driver can issue two types of port commands through the command ring: operation commands and maintenance commands.

The operation commands perform the following functions, which are part of the port's normal operation:

- Initializing the port's operational parameters
- Providing host-specific system ID information to the port
- Reading and clearing the port's data link counters
- Defining, redefining, and undefining users to the port

The maintenance commands perform the following maintenance functions:

- Obtaining maintenance status information, including recent fatal and nonfatal port errors and the current port status
- Loading the EEPROM image and, at various points in the load process, verifying the image being loaded
- Reading and initializing the history data stored in EEPROM
- Reading a saved MOP boot message to the DEMNA
- Reading Ethernet activity data for the most active Ethernet users and nodes
- Reading the settings of user-settable flags and parameters in EEPROM

Table 2–2 summarizes the port commands.

Table 2–2 Summary of Port Commands

Operation Commands	Description
PARAM	Sets and/or reads operation parameters.
RCCNTR	Reads and clears the port's data link counters.
RDCNTR	Reads the port's data link counters.
SYSID	Provides the port with host-specific information for the SYSID message (Section 2.7.2).
UCHANGE	Redefines an existing user to the port.
USTART	Defines a new user to the port.
USTOP	Undefines an existing user to the port.

Table 2-2 (Cont.) Summary of Port Commands

Maintenance Commands	Description
MAINT	Specifies a particular maintenance command. The specific maintenance commands are listed below.
EE$INITHISTORY	Initializes the history data in DEMNA EEPROM.
EE$READEEPROM	Reads the firmware image from DEMNA EEPROM for verification.
EE$READHISTORY	Reads the history data from DEMNA EEPROM. The history data consists of up to 31 history entries (each of which provides information on a particular error) followed by a history entry header that provides module-specific information.
EE$READIMAGE	Copies the firmware image in DEMNA memory to host memory for verification before the image is written to DEMNA EEPROM.
EE$READPARAM	Reads the user-settable flags and parameters, as well as the firmware revision number and revision date, from DEMNA EEPROM.
EE$WRITEEEPROM	Writes the firmware image in DEMNA memory to DEMNA EEPROM.
EE$WRITEIMAGE	Copies a firmware image from host memory to DEMNA memory.
EE$WRITEPARAM	Sets the user-settable flags and parameters in DEMNA EEPROM.
READ$BOOT	Reads a saved MOP Boot message from DEMNA memory.
READ$ERROR	Reads the port's recent error history, which consists of up to five fatal error blocks and up to five nonfatal error blocks.
READ$MONITOR	Reads network traffic data maintained by the DEMNA Ethernet monitor for the most active Ethernet nodes and users.
READ$SNAPSHOT	Reads a snapshot of current DEMNA status.
READ$STATUS	Reads the complete set of internal counters, including Maintenance Operation Protocol (MOP) counters, maintained by DEMNA firmware.

2.5 Interrupts

The port can generate both port interrupts and error interrupts to the port driver. These interrupts are enabled when the port driver supplies the necessary interrupt information in the Port Data Block.

2.5.1 Port Interrupts

When port interrupts are enabled, the port interrupts the port driver when any of the following occurs:

- The port transitions from the uninitialized state to the initialized state or vice versa.
- The port detects that the Port Data 2 (XPD2) Register has been written and determines that the port's internal ring release count does not match the ring release count written by the port driver to the XPD2 Register (Section 2.2.4).

2.5.2 Error Interrupts

If error interrupts are enabled, the port generates an error interrupt to the port driver whenever the port detects that the Error Summary bit in the Bus Error Register (XBER) has been written, indicating that an XMI transaction experienced a hard (transaction-aborting) error. The port does not generate error interrupts for soft (nonaborting) errors.

2.6 Port States

There are three possible port states:

- Resetting
- Uninitialized
- Initialized

2.6.1 Resetting

In the resetting state, the port is either powering up (Section 5.1), performing a node reset (Section 5.1), or performing a node/halt restart (Section 5.2).

2.6.2 Uninitialized

In the uninitialized state, the port is waiting for the port driver to write the XPCI register to command the port to initialize itself. The port cannot process user requests, transmit user packets, or receive user packets when in this state, nor can it communicate with the port driver. The only functions that the port can perform when in the uninitialized state are certain network maintenance functions (Section 2.7.5).

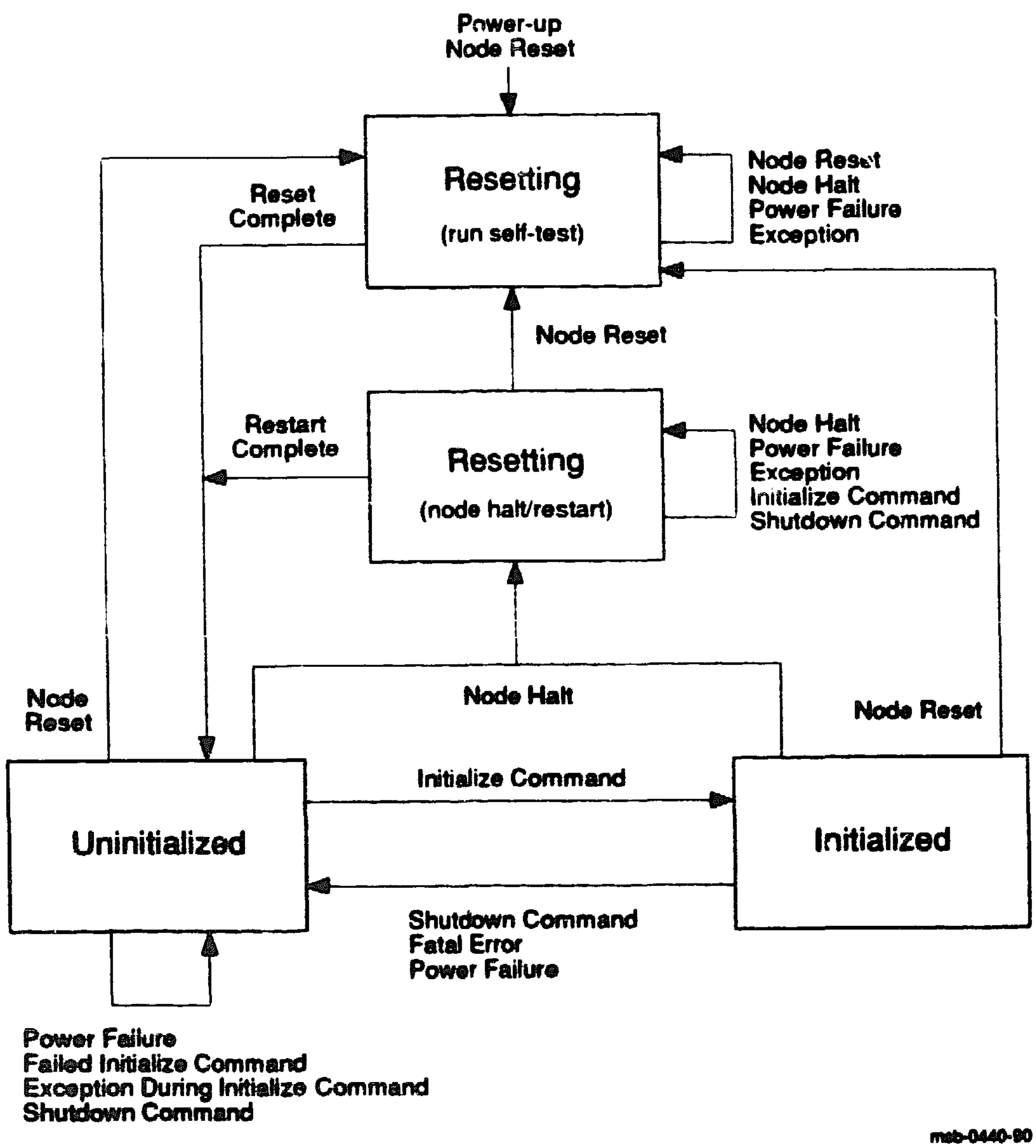
2.6.3 Initialized State

In the initialized state, the port's entire functionality is available to the port driver. In addition, the port responds to certain network maintenance messages that it receives on the Ethernet. The initialized state is thus the normal operating state for the port.

2.6.4 Port State Transitions

Figure 2–9 illustrates all the possible port state transitions and indicates the events that cause these transitions. Certain events affect only certain states. Unless otherwise indicated in Figure 2–9, the port ignores a state-transitioning event when in a particular state.

Figure 2–9 Port State Transitions



2.7 Network Maintenance Operations Supported by the Port

The port supports the following network maintenance operations:

- Processing a loopback message
- Processing remote console messages
- Participating in boot operations that involve the network
- Processing 802 Test/XID commands

These functions are fully defined in the *Digital Network Architecture Maintenance Operations Functional Specification*. The 802 Test/XID functions are also defined in the IEEE 802.2 specification.

The descriptions that follow are summary in nature. Refer to above-mentioned specifications for more detailed information.

2.7.1 Processing Loopback Messages

Network management and diagnostic programs can use a loopback test to verify the ability of a node to communicate with another node.

To comply with loopback requirements, the DEMNA port implements a loop server that responds to a loop requester on another node. In response to a valid Loop Data message sent to any physical address enabled for the port or to the loopback assistant multicast address, the port transmits a Looped Data message to the sending node or to another node specified in the message.

2.7.2 Remote Console Messages

The DEMNA port implements a console server that enables it to respond to console messages transmitted by other nodes. The port responds to the following remote console messages as indicated:

- Request ID. In response to a valid Request ID message, the port transmits a System ID message to the requesting node. The System ID message contains DEMNA-specific data automatically supplied by the port, as well as any host-specific data previously supplied by the port driver in the SYSID command.
- Request Counters. In response to a valid Request Counters message, the port transmits a Counters message that contains a copy of the port's data link counters.
- Boot. When network booting is enabled for the port, the port responds to a valid Boot message by asserting XMI RESET L, thus causing an XMI bus reset. This reset may cause the host to reboot the system (Appendix C).
- Reserve Console. In response to a valid Reserve Console message, the port reserves the DEMNA console monitor for the remote node indicated in the message.

- Release Console. In response to a valid Release Console message, the port releases the DEMNA console monitor so that it is once again available to any requesting node.
- Console Command and Poll. The Console Command and Pole message contains input data from a remote node on behalf of the DEMNA console monitor.
- Console Response and Acknowledge. In response to a valid Console Command and Pole message, the port transmits output data to a remote node in a Console Response and Acknowledge message on behalf of the DEMNA console monitor.

In addition to transmitting a System ID message in response to a valid Request ID message, the port automatically transmits two System ID messages every 8 to 12 minutes. One message has an Ethernet format and the other has an 802.3 format.

2.7.3 Boot Operations over the Network

The DEMNA plays a role in booting whenever (a) the specified device from which to boot is the DEMNA or (b) a command to boot arrives from the network. See Appendix C for a detailed description of the DEMNA's participation in booting over the network.

2.7.4 802 Test/XID Commands

In compliance with the IEEE 802.2 specification, the port responds to valid Test command messages and XID command messages directed toward Class-1 802 users defined to the port. (A Class-2 user performs Test/XID processing itself; the port simply receives and transmits packets on behalf of the user without doing any of the Test/XID processing.)

The Test command and response is a loopback test for 802 users. In response to a valid Test command message, the port changes one bit in the message to indicate that the message is a response and then retransmits the message.

The Exchange Identification (XID) command and response can be used to determine the network configuration and which network users have sufficient functionality to communicate. In response to a valid XID command Protocol Data Unit (PDU), the port transmits an XID response PDU that identifies the DEMNA node.

2.7.5 Port Response to Maintenance Commands versus Port State

Table 2–3 indicates the port's response to various network maintenance commands for each port state. The port states are listed in the leftmost column. The next four columns indicate various network maintenance commands. Each table entry includes the destination address(es) to which the port responds for the particular maintenance command(s). The address types are described in the table key. If the port does not respond to a particular type of maintenance command in a certain state, the table entry is *No Response*.

Table 2–3 Port Response to Maintenance Commands versus Port State

State	Periodic System ID	Loopback	Other MOP Commands	802 Test/XID
Resetting	No Response	No Response	No Response	No Response
Uninitialized	RCSMA	EP, LAMA	EP	No Response
Initialized	RCSMA	EP, LAMA	EP	For Class-1 users: port responds. For Class-2 users: port deliver packet to user

EP—all enabled physical addresses (one or more actual physical addresses assigned by the host through PARAM and USTART commands, or, if the host has not assigned an address, the DEMNA default physical address (DPA))
DPA—default physical address from the DEMNA's MAC address PROM
LAMA—loopback assistant multicast address
RCSMA—remote console server multicast address

2.8 Packet Filtering and Validation Performed by the Port

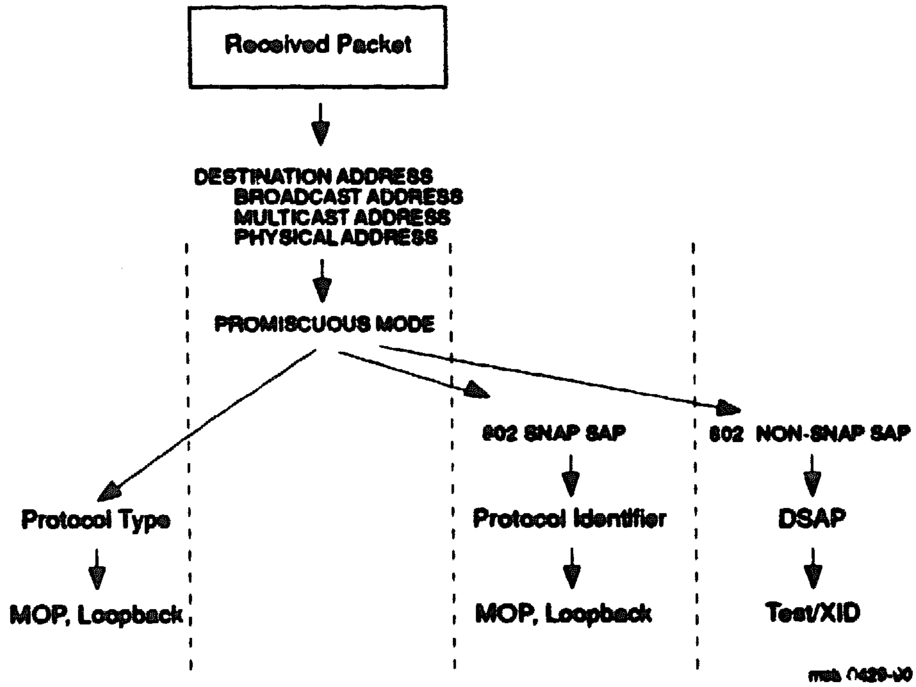
The port performs a series of filtering operations on each packet received over the network to determine whether the packet should be delivered to any user defined to the port. The port also performs a series of validation checks on each packet to determine whether the packet has a valid format.

2.9 Packet Filtering

The port filters a receive packet for each of the following parameters (Figure 2–10):

- Destination address: The destination address can be either a broadcast address, a multicast address, or a physical address.
  - If the packet's destination address is the broadcast address, the port designates the packet for delivery to all users for whom the broadcast address is enabled.
  - If the packet's destination address is a multicast address, the port designates the packet for delivery to all users for whom that particular multicast address is enabled.
  - If the packet's destination address is an enabled physical address, the packet is sent to all users who meet the other necessary criteria for reception of the packet.

Figure 2–10 Packet Filtering Performed by the Port



- Promiscuous mode: The port determines whether any users are operating in promiscuous mode.[1] If so, the port designates the packet for delivery to all users for whom promiscuous mode is enabled.
- User designator: For Ethernet packets, the user designator is the protocol type. For 802 SNAP SAP packets, the user designator is the protocol identifier (PI). For 802 non-SNAP SAP packets, the user designator is the DSAP. If the user designator is recognized by the port, the port designates the packet for delivery to all users that accept that user designator. If the user designator is not recognized by the port, the port assigns the packet to the unknown user (Section 2.10), provided that an unknown user is enabled. If the port cannot assign the packet to a user, it discards the packet, unless promiscuous mode is enabled for one or more users, in which case the port designates the packet for delivery to each promiscuous user.
- MOP, loopback, or 802 Test/XID: If the packet is a MOP command supported by the port, a loopback message, or an 802 Test/XID message; the port normally does not deliver the packet to a user (except a promiscuous user) but itself performs all MOP, loopback, and 802 Test/XID functions. However, if an 802 Test/XID packet is addressed to a user who has requested Class-2 service, the port sends

[1] When in promiscuous mode, a user receives all error-free packets received by the DEMNA. Packets that have inconsistent length errors are regarded as error-free for promiscuous users. However, a packet larger than the maximum packet length of 1518 bytes is considered to be an error packet and is therefore not delivered to a promiscuous user unless that user is designated to receive bad packets (the Bad bit is set in the USTART command that defines the user to the port).

the packet to that user, who then performs the necessary Test/XID operations.

2.10 The Unknown User

The unknown user is a special class of user enabled by setting the UNK bit in the USTART command that defines the user to the port. In effect, the unknown user is a catchall user that accepts all valid receive packets not claimed by any of the other defined users. The unknown user can thus be used to accommodate more users than the port's maximum of 64. All users beyond the 63rd user can be assigned to the unknown user. When the port receives a packet for a user it does not recognize, it assigns the packet to the unknown user.

The port driver or higher-level software is responsible for demultiplexing unknown user packets to the appropriate user. In addition, the port driver or higher-level software must perform promiscuous-mode filtering on unknown user packets if promiscuous mode is enabled for the unknown user. Finally, the port driver or higher-level software must perform Test/XID processing on unknown user packets.

NOTE: **In general, it is not useful to have more than one unknown user.**

2.11 Packet Validation

The port performs the following validation checks on each receive packet:

- Cyclic redundancy check (CRC): If the CRC value supplied in the packet does not match the CRC calculated by the port, the port increments the Receive Failures—CRC Error Counter.
- Frame too long: If the packet is longer than the maximum allowable receive packet length, the port increments the Receive Failures—Frame Too Long Counter.
- Frame length inconsistent: The port compares the actual packet length (header bytes + user data bytes + padding bytes) with the length indicated by the packet.
- Maximum receive data size: Before sending a packet to a user, the port determines whether the user data to be sent is less than or equal to the maximum receive user data size specified in the USTART command for the user. If this is the case, the port delivers the packet. If this is not the case, the port discards the packet.

Normally, the port discards any packets that do not pass the above validation checks. However, the first three validation checks can be overridden as follows:

- If the Bad flag was set in the USTART command that defined the user to the port, the port delivers to that user packets with CRC errors, frames that are too long, and/or inconsistent frame lengths.

- If the Frame Length Error Accept flag was set in a previous PARAM command, the port delivers packets with inconsistent frame lengths to users for whom the LEN bit was set in the USTART command that defined the user to the port. When the port delivers a packet with an inconsistent frame length to a user that is not an unknown or promiscuous user, it writes the Frame Length Inconsistent error code to the receive ring entry for the packet. When the port delivers such a packet to an unknown or promiscuous user, it does not write error status to the receive ring entry for the packet.

There is no override for the maximum receive data size error. However, this validation check can be effectively disabled for a particular user by supplying the largest allowable user data length for the maximum receive data size in the USTART command for the user. Any packet of legal length will then be delivered to the user, provided that all other conditions are met.

2.12 Port-Maintained Status

The port maintains operational and error status in various registers, internal data structures, and the Port Data Block (PDB).

2.12.1 Status in Registers

The port maintains the indicated status in the following registers:

- XBER: error status on XMI and loopback transactions
- XPUD: the results of the DEMNA's self-test
- XPST: port state changes and the reasons for the indicated state transitions
- XPD1 and XPD2:
  - Default (MAC) Ethernet address: The port writes XPD1 and XPD2 with the default Ethernet address from the DEMNA's MAC Address PROM.
  - Address of invalid PDB field: If the port detects an invalid PDB field during port initialization, the port writes the host address of this field (PDB base address + offset to beginning of invalid field) to XPD1.
  - Firmware PC: If the port experiences an exception or other internal error that is fatal, it writes the program counter (PC) value to XPD1.
  - Current ring offset: If the port detects an invalid command ring entry or an invalid receive ring entry, it writes the offset of the invalid entry to XPD1.
  - NI Address: the default Ethernet address from the DEMNA's MAC Address PROM

2.12.2 Status in Internal Data Structures

The port maintains the following internal status:

- Data link counters, which can be read with the RCCNTR/RDCNTR commands, the READ$STATUS command, or the READ$SNAPSHOT command. The data link counters can also be examined using the DEMNA console monitor program.
- A fatal error block that describes the last five fatal port errors and a nonfatal error block that describes the last five nonfatal port errors. These error blocks can be read with the READ$ERROR or READ$SNAPSHOT command or from the DEMNA console monitor program.
- Resource utilization information on the DEMNA and the network that can be read with the READ$STATUS or READ$SNAPSHOT command or from the DEMNA console monitor program

2.12.3 Status in the Port Data Block

The port writes a fatal error block to the Port Error Log Area of the Port Data Block (PDB) immediately after a fatal port error occurs or the port receives a node halt/restart when in the initialized state. (When the port receives a node halt/restart in the initialized state, it assumes that it is being restarted from an error.)

The PDB also contains the Potential SBUA Counter, Actual SBUA Counter, and DOR Counter, all of which record errors related to processing receive packets. (See Section 2.2.1 for a description of these counters.)

2.13 Port Driver-Maintained Status

The port driver should maintain the User Buffer Unavailable (UBUA) Counter in the Port Data Block. This is the only counter not maintained by the port. The UBUA Counter is a 16-bit or 64-bit counter that records how many receive packets the port driver had to discard because a user buffer was unavailable.

The UBUA Counter address and length are specified to the port in the Port Data Block. When the port receives a MOP Request Counters message over the network or a RCCNTR/RDCNTR command from the port driver, it reads the UBUA Counter and supplies the information to the requester.

CHAPTER 3

3 Data Flow Description

This chapter describes the data flow through the DEMNA, as well as the data flow between the DEMNA and the host, for both transmits and receives. The chapter includes the following sections:

- Transmit Data Flow
- Receive Data Flow

3.1 Transmit Data Flow

A transmit operation has the following data flow:

1 The port driver receives a user request to transmit a packet. The packet consists of one or more buffer segments. Depending on the driver design and the mapping of the buffers, the port driver may copy the buffer segment(s) to system buffers.

2 The port driver writes the address of each transmit buffer segment to the next available entry in the host command ring and relinquishes ownership of the entry.

3 The port driver writes the DEMNA Port Control Poll (XPCP) Register to inform the port that a transmit is outstanding.

4 The port periodically polls the Gate Array Control/Status (GACSR) Register to determine whether the XPCP Register has been written.

5 On discovering that the port driver has written the XPCP Register, the port performs a peek read operation through the DEMNA gate array to read the Ownership bit of the next command ring entry. This read also obtains the type of operation to be performed (in this case, a transmit operation), address translation information, and the number of transmit buffer segments involved.

6 If the port owns the entry, it performs a peek read operation to obtain the address of the host transmit buffer that contains transmit data.

7 If virtual addressing is enabled, the port translates the address of the first host transmit buffer into a physical address.

8 The port uses a datamove operation (issued through the gate array) to read the data in the host transmit buffer into a transmit buffer in DEMNA shared memory.

9 The port uses peek and datamove reads in the above fashion to locate any subsequent transmit buffers specified by the ring entry and to copy the buffer contents into DEMNA shared memory.

10 The port writes the address of the transmit buffer in shared memory to a LANCE transmit descriptor, which is also in DEMNA shared memory. The write operation also transfers ownership of the LANCE transmit descriptor to the LANCE chip.

11 The LANCE chip reads the transmit data into an internal FIFO. The LANCE builds an Ethernet packet by prefacing the data with preamble and synchronization information, calculating a CRC checksum, and appending the checksum to the end of the data. This packet building occurs while data is passing through the LANCE. Parallel data input to the LANCE is pipelined with serial data output to the SIA chip. The SIA converts the TTL inputs (NRZ DATA and READ CLOCK) into differential Manchester signals for transmission over Ethernet.

12 When the LANCE chip has completed the transmit, it writes completion status to the transmit descriptor. The write operation also transfers ownership of the LANCE transmit descriptor back to the port.

13 The port reads the completion status from the transmit descriptor and, using a peek operation, writes completion status to the command ring entry in host memory and returns ownership of the entry to the port driver. The port also updates its data link counters and error counters.

14 If the port driver has written the Port Data 2 (XPD2) Register to update the ring release counter, the port interrupts the port driver to let the port driver know that the transmit has completed.

15 The port driver reads the completion status of the transmit from the ring entry and reports the completion of the transmit to the appropriate host user.

3.2 Receive Data Flow

A receive operation has the following data flow:

1 When a carrier is present on the Ethernet cable, the SIA chip locks onto the differential Receive signals from the Ethernet wire with a phase-lock loop. The SIA converts these differential Manchester signals into TTL NRZ (non-return-to-zero) Data and a TTL Read Clock for input to the LANCE.

2 The LANCE inputs NRZ Data and the Read Clock from the SIA chip. While loading the NRZ data into an internal FIFO (first-in, first-out) memory, the LANCE requests a DMA transfer to DEMNA shared memory. On obtaining the DMA grant, the LANCE writes the FIFO contents into one or more receive buffers in shared memory and then inputs more data from the SIA chip. This pipelined input and output continues until the entire packet has been transferred.

3 After writing each receive buffer, the LANCE transfers to the port ownership of the receive descriptor that points to the buffer. (The receive descriptors are also in DEMNA shared memory.) In addition, the LANCE writes completion status for the entire packet to the receive descriptor for the last buffer used in the receive operation.

4 The port copies the first 24 bytes of the packet into CVAX RAM. This is all the port needs to filter and validate the packet.

5 The port filters the packet for destination address, user designator (Ethernet protocol type, 802 SAP, or 802 SNAP SAP), and MOP, Loopback, or Test/XID processing. The port also performs validation checks on the packet. (See Section 2.8 for a description of port filtering and validation.) The result of filtering and validation is a 64-bit mask that indicates which of the possible 64 port users are to receive a copy of the packet.

6 If one or more port users are to receive the packet, the port uses a peek operation to read the next entry in the host receive ring. If the port owns the entry, it uses another peek operation to read the address of a host receive buffer from the entry. If virtual addressing is being used, the port translates the virtual buffer address into a physical buffer address. Actually, the port normally prefetches this information and performs the address translation before it has to transfer a receive packet to the host. This reduces the time needed to transfer the receive packet to the host.

7 Using datamove operations, the port copies 192-byte sections of data at a time to the receive buffer in host memory while doing any additional address translation as needed.

8 When the transfer of the packet is complete, the port writes completion status to the receive ring entry and returns ownership of the entry to the port driver. However, if the port discovers at the end the data transfer that the packet has an error (for example, a CRC error or the packet is longer than 1518 bytes), the port does not transfer ownership of the entry to the port driver if the user does not want to receive bad packets. The corrupted packet, in this case, is written over during the next receive operation and is thereby discarded.

9 If more than one user is to receive a copy of the receive packet, the port repeats steps 6–8 for each user that receives the packet.

10 The port driver delivers each copy of the receive packet to the appropriate user. Depending on the driver design and the mapping of the buffer, the port driver may copy the receive packet to a system buffer.

CHAPTER 4

4 Registers

This chapter describes the DEMNA registers that are visible over the XMI bus. The chapter contains the following sections:

- Introduction
- XMI Registers
- Port Registers
- Power-Up Diagnostic Register

4.1 Introduction

The DEMNA's visible registers are grouped as follows:

- XMI registers, which are a standard part of the XMI bus interface
- Port registers, which are used for port/port driver communication
- Power-Up Diagnostic Register (XPUD), which indicates the results of the DEMNA self-test

All of these registers are located in the DEMNA gate array, which implements the DEMNA's primary interface to the XMI bus.

In addition to having addresses in the XMI address space, all but three of the above registers have addresses in the DEMNA node private space. The DEMNA port can thus access appropriate XMI-visible registers without having to generate transactions on the XMI bus.

Figure 4–1 is an address map of the registers.

Figure 4–1 XMI-Visible Register Locations

XMI Registers (bits 31–0)

XMI Address	XMI Registers	Node-Private Address
bb + 00	Device Register (XDEV)	2015 0000
bb + 04	Bus Error Register (XBER)	2015 0004
bb + 08	Failing Address Register (XFADR)	2015 0008
bb + 10	Communications Register (XCOMM)	2015 0010
bb + 2C	Failing Address Extension Register (XFAER)	2015 002C

Port Registers (bits 31–0)

XMI Address	Port Registers	Node-Private Address
bb + 100	Port Data Register 1 (XPD1)	2015 0100
bb + 104	Port Data Register 2 (XPD2)	2015 0104
bb + 108	Port Status Register (XPST)	2015 0108
bb + 110	Port Control Initialization Register (XPCI)	Not Accessible
bb + 114	Port Control Poll Register (XPCP)	Not Accessible
bb + 118	Port Control Shutdown Register (XPCS)	Not Accessible

Diagnostic Register (bits 31–0)

XMI Address	Diagnostic Register	Node-Private Address
bb + 10C	Power-Up Diagnostic Register (XPUD)	2015 010C

bb = Nodespace base address

msb-0455-90

4.2 XMI Registers

Table 4–1 summarizes the registers. Table 4–2 explains the codes that describe the bit types in the registers.

Table 4–1 Summary of XMI Registers

Name	Mnemonic	XMI Address	Type[1]	Description
Device	XDEV	bb + 0	RO	Indicates the XMI node's device type, functional hardware revision level, and firmware revision level.
Bus Error	XBER	bb + 4	R/W	Records XMI bus errors.

[1]With respect to the port driver

Table 4–1 (Cont.) Summary of XMI Registers

Name	Mnemonic	XMI Address	Type[1]	Description
Power-Up Diagnostic	XPUD	bb + 10C	RO	Indicates whether self-test has completed and which tests in the self-test passed.
Failing Address Failing Address Extension	XFADR XFAER	bb + 8 bb + 2C	RO	Provide information on failing XMI transactions initiated by the DEMNA.
Communications	XCOMM	bb + 10	R/W	Serves as the hardware interface between the host and the DEMNA ROM-based diagnostics.

[1]With respect to the port driver

Table 4–2 Abbreviations for Bit Types

Abbreviation	Definition
0	Initialized to logic level zero
1	Initialized to logic level one
X	Initialized to either logic state
RO	Read only
R/W	Read/write
R/W1C	Read/cleared by writing a 1
U	Undefined

Device Register (XDEV)

The Device Register identifies the DEMNA device type, hardware revision, and EEPROM firmware revision. The device type for the DEMNA is 0C03 (hexadecimal).

The port loads the Device Register on power-up or reset. The port driver reads the Device Register to determine the module device type before attempting to initialize the port.

ADDRESS

Nodespace base address + 0

31 — 16	15 — 0
Device Revision	Device Type

msb-0343-89

bits<31:16>

Name: Device Revision
Mnemonic: DREV
Type: RO, 0

Indicates the device revision of the DEMNA. The high-order byte of the field is the hardware revision number, which indicates the functional revision of the hardware. This is identical to the hardware functional revision indicated on the module label. The low-order byte of the field is the revision number of the loaded firmware. If the EEPROM firmware is loaded (which is normally the case), the firmware revision field indicates the EEPROM firmware revision. However, if the EPROM firmware is loaded (which occurs if the EEPROM fails selftest), the firmware revision field indicates the EPROM firmware revision.

The hardware revision field is decoded as follows for the first 10 hardware revisions of the DEMNA. Note that letters G and I, as well as their corresponding codes, are skipped.

Code (hex)	Hardware Revision Level
01	A
02	B
03	C
04	D
05	E
06	F

Code (hex)	Hardware Revision Level
08	H
0A	J
0B	K
0C	L

The firmware revision field is decoded as follows for the first 10 EEPROM firmware revisions for a given hardware revision level.

Code (hex)	EEPROM Firmware Revision Level
01	01
02	02
03	03
04	04
05	05
06	06
07	07
08	08
09	09
0A	10

Note that the firmware revision levels of both the EEPROM and the EPROM are zeroed when the hardware revision level changes.

Table 4–3 shows how the Device Revision field is encoded for the first three firmware revisions of hardware revision F and for the first firmware revision of hardware revision H.

Table 4–3 Example of Device Revision Field

Hex Value	Hardware Revision	Firmware Revision
0600	F	0
0601	F	1
0602	F	2
0603	F	3
0800	H	0

bits<15:0>

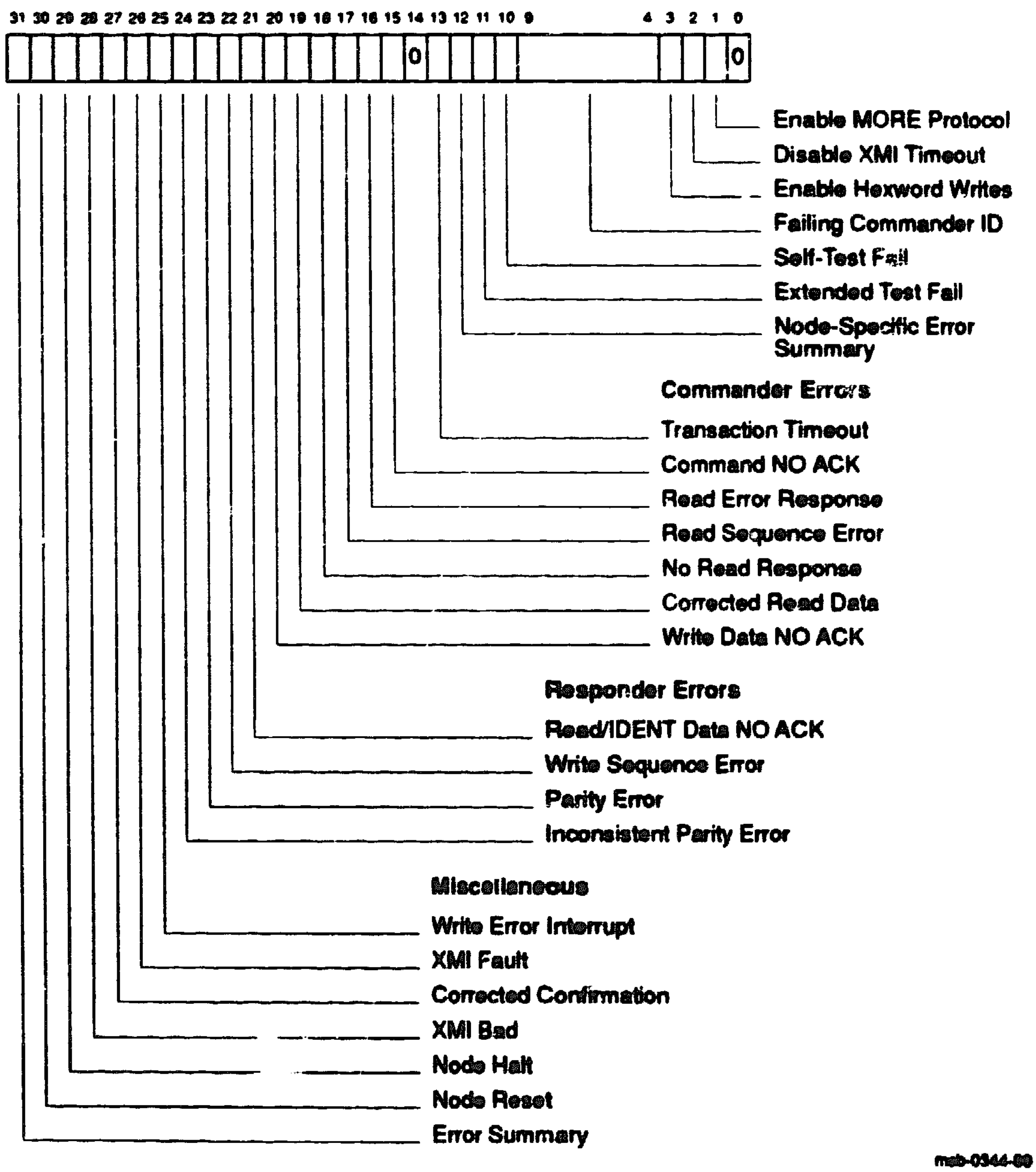
Name: Device Type
Mnemonic: XDEV
Type: RO, 0

A value of 0C03 (hex) indicates that the adapter is a DEMNA module.

Bus Error Register (XBER)

The Bus Error Register contains error status on a failed XMI transaction. This status includes the failed command, commander ID, and an error bit that indicates the type of error that occurred. This status remains locked until software resets the error bit(s).

ADDRESS *Nodespace base address + 0000 0004*



bit<31>

Name: Error Summary
Mnemonic: ES
Type: RO, 0

ES represents the logical OR of the error bits in this register. Therefore, ES asserts when any error bit asserts.

bit<30>

Name: Node Reset
Mnemonic: NRST
Type: R/W, 0

The host writes a one to NRST to cause the DEMNA to execute a complete power-up reset. This sequence is similar to the response caused by a real power-up sequence, which is triggered by the assertion and deassertion of XMI DC LO L. The DEMNA asserts XMI BAD L until the self-test completes successfully. Nodes should not access the DEMNA from the time it is reset until it completes self-test (or the maximum self-test time is exceeded).

While the DEMNA is responding to node reset, it does not access other nodes on the XMI bus. In response to a real power-up sequence (caused by XMI DC LO L), the NRST bit will be cleared. However, when set by the host to cause a node reset, the bit remains set, thus indicating to the DEMNA CVAX that the host issued a node reset. The DEMNA self-test clears this bit once the node reset has completed.

bit<29>

Name: Node Halt
Mnemonic: NHALT
Type: R/W, 0

The host sets NHALT to force the DEMNA to execute its halt sequence. The halt causes the DEMNA to go into a quiet state and retain as much state information as possible. Firmware execution jumps to the initialization code in EPROM (Boot ROM), shuts down the port, and loops on the NHALT bit. When the host clears NHALT, the DEMNA executes its restart sequence, which is identical to the power-up/reset sequence, except that the DEMNA does not execute its self-test, clear its internal data link or error counters, or clear its fatal and nonfatal error blocks.

bit<28>

Name: XMI BAD
Mnemonic: XBAD
Type: RO, 0

The DEMNA does not use this bit.

bit<27>

Name: Corrected Confirmation
Mnemonic: CC
Type: R/W1C, 0

This bit sets when the DEMNA detects a single-bit CNF error. Single-bit CNF errors are automatically corrected by the XCLOCK chip in the XMI Corner.

bit<26>

Name: XMI FAULT
Mnemonic: XFAULT
Type: RO, 0

The DEMNA does not use this bit.

bit<25>

Name: Write Error Interrupt
Mnemonic: WEI
Type: RO, 0

The DEMNA does not use this bit.

bit<24>

Name: Inconsistent Parity Error
Mnemonic: IPE
Type: RO, 0

The DEMNA does not use this bit.

bit<23>

Name: Parity Error
Mnemonic: PE
Type: R/W1C, 0

When set, indicates that the DEMNA has detected a parity error on an XMI cycle.

bit<22>

Name: Write Sequence Error
Mnemonic: WSE
Type: R/W1C, 0

When set, indicates that an XMI node attempting a write to the DEMNA aborted the write transaction due to missing data cycles. Only XMI responder nodes are required to implement this bit. If not implemented, nodes return zero.

bit<21>

Name: Read/IDENT Data NO ACK
Mnemonic: RIDNAK
Type: R/W1C, 0

When set, indicates that a Read or IDENT data cycle (GRDn) transmitted by the DEMNA has received a NO ACK confirmation.

bit<20>

Name: Write Data NO ACK
Mnemonic: WDNAK
Type: R/W1C, 0

When set, indicates that a Write data cycle (WDAT) transmitted by the DEMNA has received a NO ACK confirmation.

bit<19>

Name: Corrected Read Data
Mnemonic: CRD
Type: R/W1C, 0

When set, indicates that the node has received a CRDn read response.

bit<18>

Name: No Read Response
Mnemonic: NRR
Type: R/W1C, 0

When set, indicates that a transaction initiated by the DEMNA failed due to a read response timeout.

bit<17>

Name: Read Sequence Error
Mnemonic: RSE
Type: R/W1C, 0

When set, indicates that a transaction initiated by the DEMNA failed due to a read sequence error.

bit<16>

Name: Read Error Response
Mnemonic: RER
Type: R/W1C, 0

When set, RER indicates that a node has received a Read Error Response.

bit<15>

Name: Command NO ACK

Mnemonic: CNAK

Type: R/W1C, 0

When set, indicates that a command cycle transmitted by the DEMNA has received a NO ACK confirmation caused either by a reference to a nonexistent memory location or by a command cycle parity error.

bit<14>

Name: Reserved

Mnemonic: None

Type: RO, 0

Reserved; must be zero.

bit<13>

Name: Transaction Timeout

Mnemonic: TTO

Type: R/W1C, 0

When set, indicates that a transaction initiated by the DEMNA failed due to a transaction timeout.

bit<12>

Name: Node-Specific Error Summary

Mnemonic: NSES

Type: RO, 0

When set, NSES indicates that a node-specific error condition has been detected. The DEMNA does not use this bit.

bit<11>

Name: Extended Test Fail

Mnemonic: ETF

Type: R/W, 0

This bit is not used by the DEMNA.

bit<10>

Name: Self-Test Fail
Mnemonic: STF
Type: R/W1C, 1

When the STF bit is set, indicating that the DEMNA has not passed self-test, the DEMNA hardware asserts XMI BAD L on the XMI bus. When the STF bit is cleared, indicating that the DEMNA has passed self-test, the DEMNA deasserts XMI BAD L.

bits<9:4>

Name: Failing Commander ID
Mnemonic: FCID
Type: RO

Bits <9:6> log the commander ID of a failing transaction during a command cycle. The failing commander ID is recorded for command errors detected by XBER bits <20> and <18:13>. Bits <5:4> indicate the type of operation that failed: 00 = a failed peek or interrupt operation; 01 = a failed datamove operation.

bit<3>

Name: Enable Hexword Write
Mnemonic: EHWW
Type: RO, 0

The DEMNA does not use this bit.

bit<2>

Name: Disable XMI Timeout
Mnemonic: DXTO
Type: RW, 0

This bit enables or disables the reporting of all XMI timeouts by a commander. When this bit is set, the node will never encounter a No Read Response (NRR) error or a transaction timeout (TTO) if retries are disabled.

bit<1>

Name: Enable MORE Protocol
Mnemonic: EMP
Type: RW, 0

When cleared, prevents the gate array from asserting the More signal on the XMI bus. When set, enables the DEMNA gate array to assert the More signal.

bit<0>

Name: Reserved
Mnemonic: None
Type: RO, 0

Reserved; must be zero.

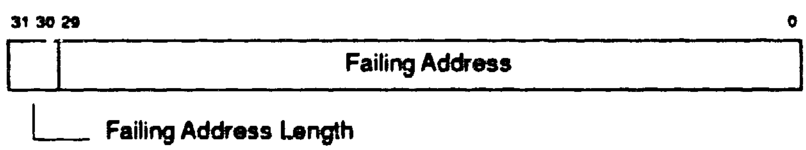
Failing Address Register (XFADR)

The Failing Address Register logs address and data length information associated with a failing XMI transaction. The register is locked when any of the following bits in the Bus Error Register (XBER) sets (these are commander errors):

- Write Data NO ACK (WDNAK), XBER<20>
- No Read Response (NRR), XBER<18>
- Read Sequence Error (RSE), XBER<17>
- Read Error Response (RER), XBER<16>
- Command NO ACK (CNAK), XBER<15>
- Transaction Timeout (TTO), XBER<13>

The XFADR is unlocked (free to latch new information) when the XBER bits that lock the register are cleared. If none of the above listed errors has occurred, the XFADR contains the address and data length of the last XMI transaction.

ADDRESS *Nodespace base address + 8*



msb-0346-89

bits<31:30>

Name: Failing Length

Mnemonic: FLN

Type: RO, 0

FLN logs the value of XMI D<31:30> during the command/address cycle of a failed XMI commander transaction. FLN loads on every C/A cycle issued by the DEMNA. It locks only after all retries of the transaction fail, and it unlocks when the error that caused the lock is cleared.

XMI D<31:30>, the Length field, is used to define the number of words in the XMI data transfer. The table below shows the Length field coding. Longword-length transactions are used only in I/O space. Quadword-, octaword-, and hexword-length transactions are used only in memory space. Hexword lengths are used only for Read or Interlock Read transactions.

XMI D<31:30> L Logic Level 31	30	Size
0	0	Hexword
0	1	Longword
1	0	Quadword
1	1	Octaword

bits<29:0>

Name: Failing Address
Mnemonic: None
Type: RO, 0

The Failing Address field logs the value of XMI D<29:0> during the command cycle of a failing transaction. Failing Address loads on every C/A cycle issued by the DEMNA. It locks only after all retries of the transaction fail, and it unlocks when the error that caused the lock is cleared.

XMI D<29:0> defines the address of an XMI read or write transaction. If an XMI transaction has a 40-bit address, the XMI D bits decode to the address as follows:

A<39>	XMI D<29>
A<38:29>	XMI D<57:48>
A<28:0>	XMI D<28:0>

The number of significant bits in the address depends on the transaction type and length.

When XMI D<29> is set, the address is a reference to I/O space. When XMI D<29> is cleared, the address is a reference to memory space.

Communications Register (XCOMM)

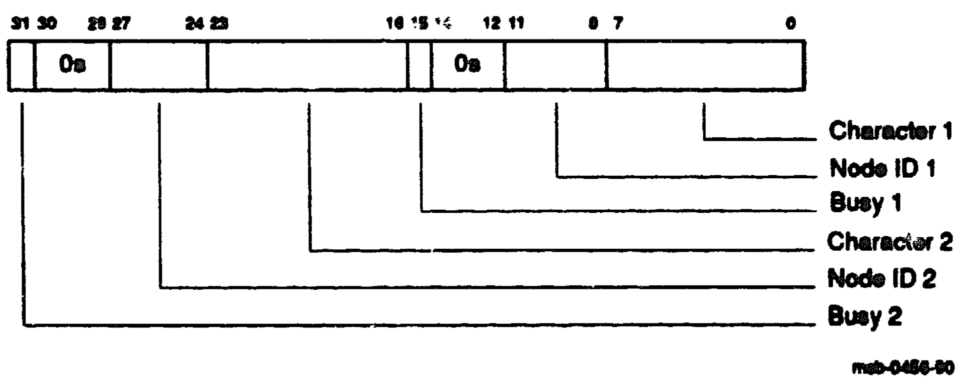
The XCOMM Register is the host's hardware interface with diagnostic routines that run on the DEMNA. The host and the DEMNA diagnostic routines communicate with each other by writing and reading one ASCII character at a time.

In addition, the XCOMM Register in the DEMNA can be used to do the following:

- Read the DEMNA default physical address
- Read the DEMNA module serial number
- Invalidate the contents of the DEMNA EEPROM
- Clear the history data in EEPROM
- Read the history data from EEPROM

ADDRESS

Nodespace base address + 10



bit<31>

Name: Busy 2
Mnemonic: None
Type: R/W

When set, indicates that the CHAR2 field contains a character that has not yet been read by the host. The host clears the bit after reading the CHAR2 field. The DEMNA cannot write the CHAR2 field until the bit is clear.

bits<30:28>

Name: Reserved
Mnemonic: None
Type: R/W, 0

Reserved; must be zeros.

---

bits<27:24>

Name: Node ID 2
Mnemonic: None
Type: R/W

Written by the DEMNA to indicate that data in the CHAR2 field is from the DEMNA.

---

bits<23:16>

Name: Character 2
Mnemonic: CHAR2
Type: R/W

Contains the character being sent from the DEMNA to the host.

---

bit<15>

Name: Busy 1
Mnemonic: None
Type: R/W

When set, indicates that the CHAR1 field contains a character that has not yet been read by the DEMNA. The port clears this bit after reading the CHAR1 field. The sending node cannot write another character to the CHAR1 field until the bit is cleared.

---

bits<14:12>

Name: Reserved
Mnemonic: None
Type: R/W, 0

Reserved; must be zeros.

---

bits<11:8>

Name: Node ID 1
Mnemonic: None
Type: R/W

Written by the sending node. Indicates the node that has sent the data in the CHAR1 field.

bits<7:0>

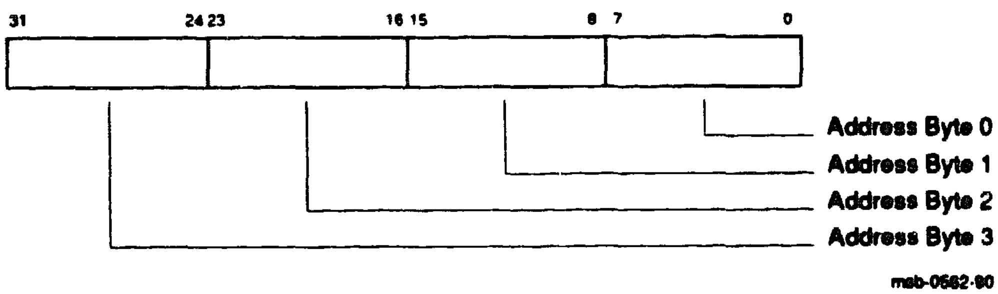
Name: Character 1
Mnemonic: CHAR1
Type: R/W

Written by the sending node with a character.

To Read the DEMNA Default Physical Address:

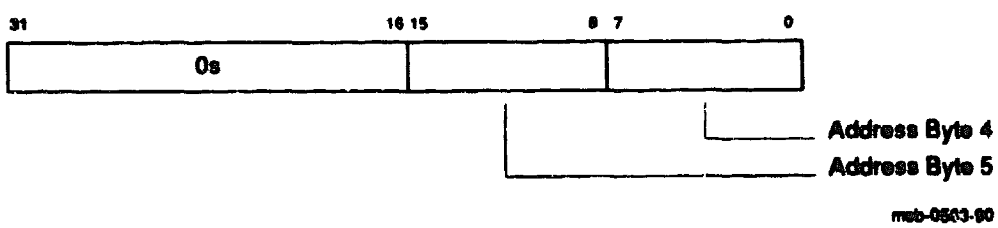
1 Deposit FFFFFFFF (hex) into the XCOMM Register.

2 Examine the XCOMM Register to obtain bytes 0–3 of the default physical address (DPA):



3 Deposit FFFFFFFE (hex) into the XCOMM Register.

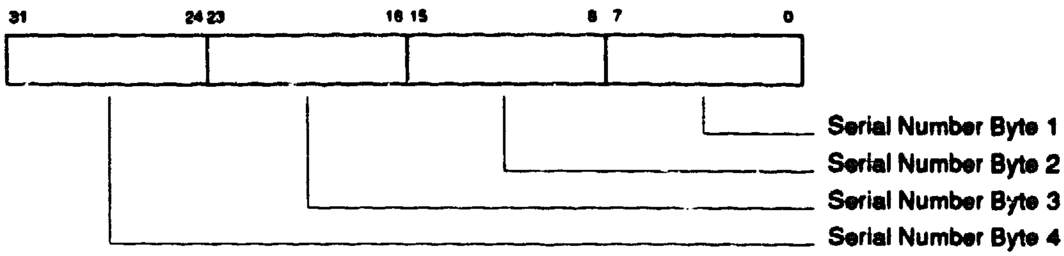
4 Read the XCOMM Register to obtain bytes 4 and 5 of the default physical address (DPA):



To Read the DEMNA Module Serial Number:

1 Deposit FFFFFFFD (hex) into the XCOMM Register.

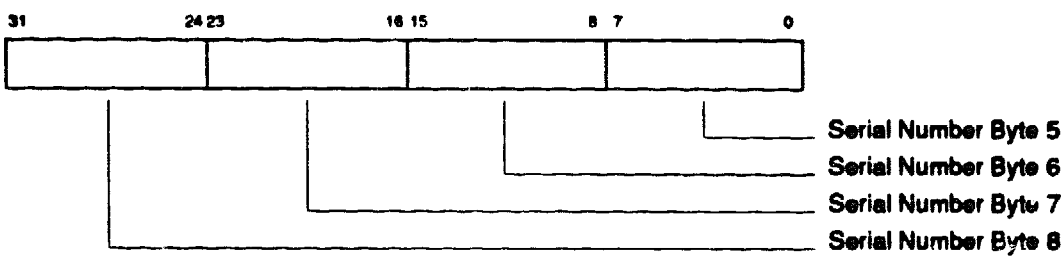
2 Examine the XCOMM Register to obtain bytes 1–4 of the module serial number:



msb-0564-90

3 Deposit FFFFFFFC (hex) into the XCOMM Register.

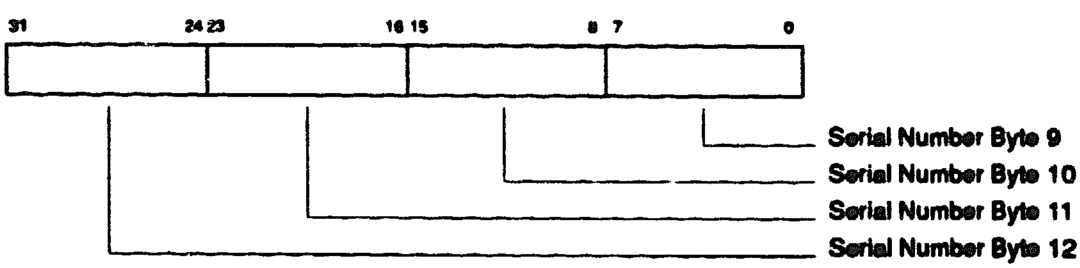
4 Read the XCOMM register to obtain bytes 5–8 of the module serial number:



msb-0565-90

5 Deposit FFFFFFFB (hex) into the XCOMM Register.

6 Read the XCOMM register to obtain bytes 9–12 of the module serial number:



msb-0566-90

To Invalidate the EEPROM Contents:

1. Deposit FFFFFFFA (hex) into the XCOMM Register.
2. Examine the XCOMM Register to obtain confirmation of the invalidation. If the EEPROM contents have been invalidated, the XCOMM Register will contain all zeros.

See Section 9.1 for information on when to use this command.

To Clear the History Data in EEPROM:

1. Deposit FFFFFFF9 (hex) into the XCOMM Register.
2. Examine the XCOMM Register to obtain confirmation that error history has been cleared. If the error history has been cleared, the XCOMM Register will contain all zeros.

To Read the History Data from EEPROM:

History data (256 longwords) can be read from EEPROM one longword at a time. To examine a longword, deposit its encoded offset into the XCOMM Register and then examine the XCOMM Register to read the longword. The offsets of history data longwords 0 through 255 are encoded with hexadecimal numbers -8 through -107, respectively.

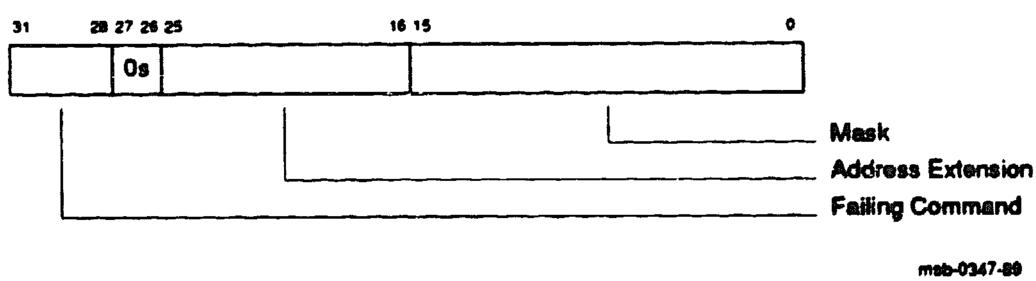
Failing Address Extension Register (XFAER)

XFAER logs the address extension, command, and mask information associated with a failed XMI commander transaction. The register is locked when any of the following bits in the Bus Error Register (XBER) sets (these are commander errors):

- Write Data NO ACK (WDNAK), XBER<20>
- No Read Response (NRR), XBER<18>
- Read Sequence Error (RSE), XBER<17>
- Read Error Response (RER), XBER<16>
- Command NO ACK (CNAK), XBER<15>
- Transaction Timeout (TTO), XBER<13>

XFAER is unlocked (free to latch new information) when the XBER bits that lock the register are cleared. If none of the above listed errors has occurred, the XFADR contains the command code, address extension, and mask of the last XMI transaction.

ADDRESS *XMI nodespace base address + 2C*



bits<31:28>

Name: Failing Command
Mnemonic: FCMD
Type: RO, 0

FCMD logs XMI D<63:60> during the C/A cycle of a failed XMI commander transaction. FCMD is loaded on every C/A cycle issued by the DEMNA but locks only if the transaction fails and unlocks when the error that caused the lock is cleared.

XMI D<63:60> is the Command field. The Command field specifies the transaction being initiated in the command cycle. The table below shows how the Command field is encoded.

XMI D<63:60> L Logic Level					
63	62	61	60	Command	Mnemonic
0	0	0	0	Reserved	
0	0	0	1	Read	READ
0	0	1	0	Interlock Read	IREAD
0	0	1	1	Reserved	
0	1	0	0	Reserved	
0	1	0	1	Reserved	
0	1	1	0	Unlock Write Mask	UWMASK
0	1	1	1	Write Mask	WMASK
1	0	0	0	Interrupt	INTR
1	0	0	1	Identify	IDENT
1	0	1	0	Reserved	
1	0	1	1	Reserved	
1	1	0	0	Reserved	
1	1	0	1	Reserved	
1	1	1	0	Reserved	
1	1	1	1	Implied Vector Interrupt	IVINTR

bits<27:26>

Name: Reserved
Mnemonic: None
Type: RO, 0

Unused; must be zero.

bits<25:16>

Name: Failing Address Extension

Mnemonic: None

Type: RO, 0

Failing Address Extension logs XMI D<57:48> during the C/A cycle of a failed XMI commander transaction or bits <38:29> of the address specified in the transaction for DMA reads and writes.

Failing Address Extension is loaded on every C/A cycle issued by the DEMNA but locks only if the transaction fails and unlocks when the error that caused the lock is cleared.

XMI D<57:48> are the extended portion of the XMI address. If an XMI transaction has a 40-bit address, the XMI D bits decode to the address as follows:

A<39>	XMI D<29>
A<38:29>	XMI D<57:48>
A<28:0>	XMI D<28:0>

bits<15:0>

Name: Failing Mask
Mnemonic: None
Type: RO, 0

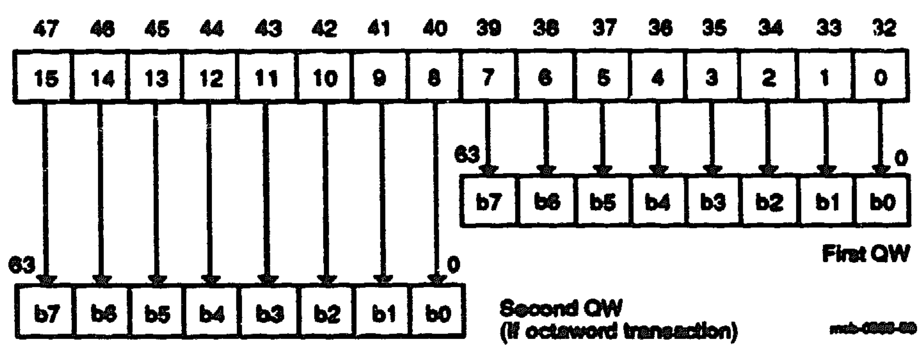
Failing Mask logs XMI D<47:32> during the C/A cycle of a failed XMI commander transaction or the write mask for DMA writes. The field is undefined for other transactions.

Failing Mask is loaded on every C/A cycle issued by the DEMNA but locks only if the transaction fails and unlocks when the error that caused the lock is cleared.

XMI D<47:32> is the Mask field, which supplies byte-level mask information for the XMI Write Mask and Unlock Write Mask transactions. During nonwrite transactions this field is a "don't care," but proper parity is still generated (see Figure 4–2).

The maximum length of a write transaction is an octaword, which requires 16 mask bits in the upper longword of the command. The mask bits define which bytes of the following write data cycles are to be written to the specified locations. For longword- and quadword-length writes, the unused mask bits (D<47:36> L and D<47:40> L, respectively) are unspecified and are ignored by responders, other than to check parity.

Figure 4–2 Mask Field Bit Assignments



4.3 Port Registers

The port registers are used by the port and port driver to communicate with each other. The port registers are grouped by function as follows:

- Port Status Register
- Port Control Registers
- Port Data Registers

The port polls the Gate Array Control/Status Register (GACSR) to determine when a port register has been written.

Table 4–4 provides a brief description of each port register. For a description of the bit-type codes for the port registers, see Table 4–2.

Table 4–4 Port Registers—Summary

Register	Mnemonic	Description
Port Status	XPST	The port writes this register to indicate a change in port state and the reason for the change. The port also writes error information to this register when a fatal error aborts an attempted port initialization or shutdown, resulting in no change of the port state.
Port Control Registers		
Port Control Initialization	XPCI	The port driver writes this register to command the port to initialize the port/port driver interface.
Port Control Poll	XPCP	The port driver writes this register to command the port to poll the command ring.
Port Control Shutdown	XPCS	The port driver writes this register to command the port to shut down.
Port Data Registers		
Port Data Register 1	XPD1	The port driver writes XPD1 and XPD2 to convey the Port Data Block base address and the ring release count (Section 2.2.4) to the port.
Port Data Register 2	XPD2	The port writes the registers to convey the DEMNA default (MAC) Ethernet address and various error information to the port driver.

Port Control Initialization Register (XPCI)

The port driver writes this register to initialize the port (to change the port state from uninitialized to initialized). Before issuing the INITIALIZE command, the port driver must write the physical address of the Port Data Block (PDB) to the Port Data Registers. The command is meaningful only when the port is in the uninitialized state. Issuing the command when the port is in the initialized state causes a port shutdown.

On receiving the INITIALIZE command, the port executes its initialization routine. At the end of this routine, the port writes the Port Status (XPST) Register with the initialized state code and with the appropriate state qualifier. If port interrupts are enabled, the port also sends an interrupt to the port driver. If initialization fails, the port state does not change, the port writes the appropriate state qualifier to the XPST Register and, if applicable, writes error information to the Port Data Registers.

The port driver should verify successful completion of port initialization by checking the XPST Register for transition of the port state from uninitialized to initialized. Typically, port initialization completes in less than 10.5 milliseconds. The port driver can thus safely use a timeout of one second.

ADDRESS

Nodespace base address + 110

msb-0457-90

bits<31:0>

Name: Write Bits

Mnemonic: None

Type: WO to port driver, U to port

The port driver may write any value to this register. Only the act of writing is significant. For fastest access, the port driver should clear the entire register with a CLRL (or equivalent) instruction.

Port Control Poll Register (XPCP)

The port driver writes this register to command the port to poll the command ring for a new entry. The port driver should write this register for each command ring entry that it creates, immediately after relinquishing ownership of the entry.

The port responds to the POLL command only if it has run out of command ring entries to process—that is, if the last command ring entry read by the port was owned by the port driver. When it does not have command ring entries to process, the port polls the Gate Array Control/Status Register (GACSR) frequently to determine whether the port driver has written the XPCP Register.

The port responds to the POLL command only when in the initialized state. Otherwise, the port ignores the command.

ADDRESS

Nodespace base address + 114

msb-0457-90

bits<31:0>

Name: Write Bits

Mnemonic: None

Type: WO to port driver, U to port

The port driver may write any value to this register. Only the act of writing is significant. For fastest access, the port driver should clear the entire register with a CLRL (or equivalent) instruction.

Port Control Shutdown Register (XPCS)

The port driver writes this register to shut down the port (to change the port state from initialized to uninitialized). The shutdown command is meaningful only when the port is in the initialized state. Issuing the command when the port is in any other state does not affect the port state. However, if the port is in the resetting or uninitialized state when issued a SHUTDOWN command, the state qualifier field of the Port Status (XPST) Register will indicate an error.

On receiving the SHUTDOWN command, the port executes its shutdown routine (Section 5.4). At the end of this routine, the port writes the Port Status (XPST) Register with the uninitialized state code and with the appropriate state qualifier. If port interrupts are enabled, the port also sends an interrupt to the port driver.

The port driver should check for successful completion of the port shutdown by checking the XPST Register for transition of the port state from initialized to uninitialized. Typically, port shutdown completes in less than 10.5 milliseconds. The port driver can thus safely use a timeout of one second.

ADDRESS

Nodespace base address + 118

31 ... 0
Write Bits

msb-0457-90

bits<31:0>

Name: Write Bits
Mnemonic: None
Type: WO to port driver, U to port

The port driver may write any value to this register. Only the act of writing is significant. For fastest access, the port driver should clear the entire register with a CLRL (or equivalent) instruction.

Port Status Register (XPST)

Whenever the port changes state, it indicates in the XPST Register the new state and the reason for transitioning to this state. In addition, the port also writes error information to this register when a fatal port error aborts an attempted port initialization or shutdown.

ADDRESS *Nodespace base address + 108*

31 — 8	7 — 0
State Qualifier	State

mab-0458-90

bits<31:8>

Name: State Qualifier
Mnemonic: None
Type: RO to port driver, WO to port

Indicates why the port is in its current state. The State Qualifier field is significant after power-up, reset, or a fatal port shutdown. The following table describes the state qualifiers and indicates the corresponding contents of the Port Data Registers.

AFTER POWER-UP OR NODE RESET

Code	Meaning	Port State	XPST	XPD1/XPD2 Contents
0	No error	Uninitialized	00000001	MAC address
1	Port failed self-test (but DEMNA still usable, if only for running diagnostics)	Uninitialized	00000101	MAC address

AFTER NODE HALT

Code	Meaning	Port State	XPST	XPD1/XPD2 Contents
0	Node Halt/Restart complete	Uninitialized	00000001	MAC address
13	Node Halt/Restart in progress	Resetting	00000D00	Unchanged

AFTER PORT INITIALIZATION				
Code	Meaning	Port State	XPST	XPD1/XPD2 Contents
0	Initialization succeeded	Initialized	00000002	Unchanged
2	Initialization failed, DEMNA failed self-test	Uninitialized	00000201	Unchanged
3	Initialization failed, invalid base address of Port Data Block (PDB) in XPD1 and XPD2	Uninitialized	00000301	Invalid base address of PDB
4	Initialization failed, contents of a PDB field not valid	Uninitialized	00000401	Address of invalid PDB field
5	Initialization succeeded but EEPROM contents are invalid. Port can execute maintenance commands for updating EEPROM.	Uninitialized	00000502	Unchanged

AFTER PORT SHUTDOWN				
Code	Meaning	Port State	XPST	XPD1/XPD2 Contents
7	Initialization attempted when port not in uninitialized state	Uninitialized	00000701	Unchanged
8	Invalid command ring	Uninitialized	00000801	Current ring offset (in bytes) in XPD1
9	Invalid receive ring	Uninitialized	00000901	Current ring offset (in bytes) in XPD1
10	Power failure	Uninitialized	00000A01	Firmware PC in XPD1
11	Unexpected firmware exception	Uninitialized	00000B01	Firmware PC in XPD1
12	Unrecoverable XMI failure, including memory error	Uninitialized	00000C01	Firmware PC in XPD1
14	Fatal firmware internal error occurred	Uninitialized	00000E01	Firmware PC in XPD1
15	Fatal firmware internal error—keep-alive counter error (the firmware was in an infinite loop)	Uninitialized	00000F01	Firmware PC in XPD1
16	Firmware update completed	Uninitialized	00001001	Unchanged

bits<7:0>

Name: State
Mnemonic: None
Type: RO to port driver, WO to port

Indicates what state the port is in.

Code (Decimal)	Port State	Description
0	Resetting	The DEMNA has just been reset or powered up and is executing its power-up or node halt/restart sequence. Maintenance Operation Protocol (MOP), loopback, and 802 Test/XID processing are disabled.
1	Uninitialized	The DEMNA has completed reset, power-up, node halt/restart, or shutdown. MOP processing and loopback processing are enabled for the default (MAC) address and for the loopback assistant multicast address. 802 Test/XID processing is disabled.
2	Initialized	The DEMNA is in its normal operating mode, in which it processes the command and receive rings. MOP, loopback, and 802 Test/XID processing are enabled for the following addresses: MOP processing—any enabled physical address, loopback processing—APA and loopback assistant multicast address, 802 Test/XID processing—APA and addresses enabled for the user.

Port Data Registers (XPD1 and XPD2)

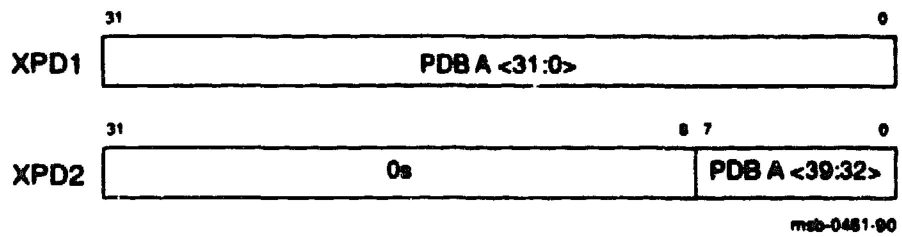
These two registers are accessed by both the port and port driver to transfer the following information:

- Port Data Block (PDB) Base Address: The port driver writes the XPD1 and XPD2 Registers with the physical address of the PDB before commanding the port to initialize itself. The port accesses the PDB at the indicated address.
- Ring release count: After processing the rings, the port driver copies its internal ring release counter to XPD2. The ring release count indicates to the port how far the port driver has proceeded through the rings (Section 2.2.4).
- Default (MAC) Ethernet address: The port writes the XPD1 and XPD2 Registers with the default Ethernet address from the DEMNA MAC Address ROM.
- Port error data: The port writes the XPD1 and XPD2 Registers with the following error data:
  - Address of invalid PDB field: If the port detects an invalid PDB field during port initialization, the port writes the physical host address of this field (PDB base address + offset to beginning of invalid field) to the XPD1 Register. This information can be used for debugging the port driver and/or the port firmware.
  - Firmware PC: If the port experiences an exception or other internal error that is fatal, it writes the program counter (PC) value to the XPD1 Register. This information can be used for debugging the port firmware.
  - Current ring offset: If the port detects an invalid command ring entry or an invalid receive ring entry, it writes the offset of the invalid entry (in bytes) to the XPD1 Register.

ADDRESS

Nodespace base address + 100, 104

Port Data Block Base Address:



xpd1 bits<31:0>

Name: Port Data Block Address
Mnemonic: None
Type: R/W to port and port driver

Bits <31:0> of the physical base address of the Port Data Block.

xpd2 bits<31:8>

Name: Reserved
Mnemonic: None
Type: R/W to port and port driver

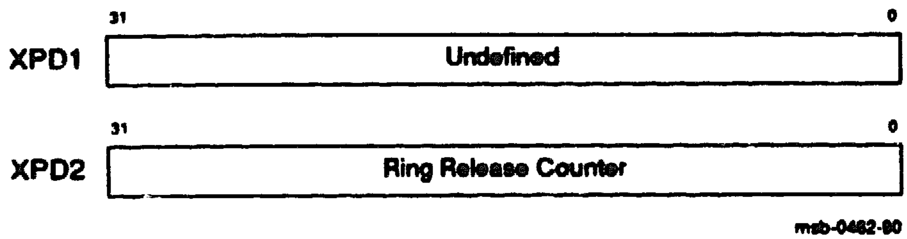
Reserved; must be zeros.

xpd2 bits<7:0>

Name: Port Data Block Address
Mnemonic: None
Type: R/W to port and port driver

Bits <39:32> of the physical base address of the Port Data Block.

Ring Release Counter:



xpd1 bits<31:0>

Name: Undefined

Mnemonic: None

Type: R/W to port and port driver

Undefined; not meaningful.

xpd2 bits<31:0>

Name: Ring Release Counter

Mnemonic: None

Type: R/W to port and port driver

Indicates the total number of command and receive ring entries that the port driver has processed since the port was last initialized. The port driver updates this counter each time that it completes processing of the rings (Section 2.2.4).

NOTE: **The port initializes XPD2 to -1 and the port driver initializes its internal ring release counter to -1. The ring release count is thus always one less than the actual number of ring entries that have been processed.**

Default Ethernet Address:

	31	16	15	0
XPD1	Ethernet Address <31:0>			
XPD2	Undefined		Ethernet Address <47:32>	

msb-[illegible]-90

xpd1 bits<31:0>

Name: Ethernet Address <31:0>
Mnemonic: None
Type: R/W to port and port driver

Bits <31:0> of the default (MAC) Ethernet address of the DEMNA. The port writes this field after power-up, node reset, or node halt/restart.

xpd2 bits<31:16>

Name: Undefined
Mnemonic: None
Type: R/W to port and port driver

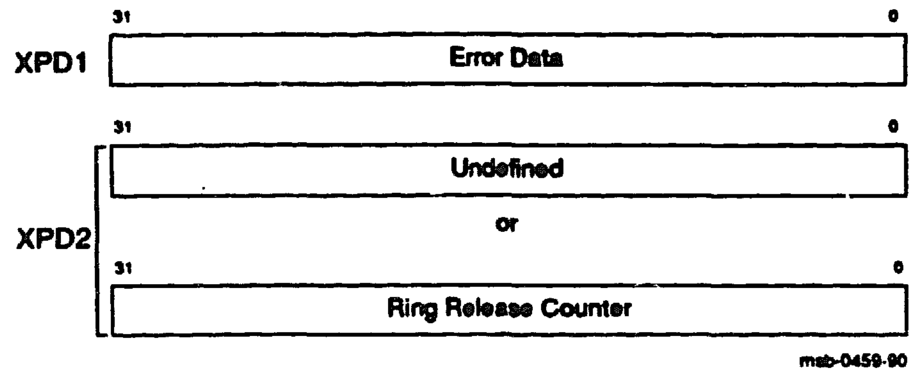
Undefined; not meaningful.

xpd2 bits<15:0>

Name: Ethernet Address <47:32>
Mnemonic: None
Type: R/W to port and port driver

Bits <47:32> of the default (MAC) Ethernet address of the DEMNA. The port writes this field after power-up, node reset, or node halt/restart.

Error Data:



xpd1 bits<31:0>

Name: Error Data
Mnemonic: None
Type: R/W to port and port driver

On detecting a fatal port error, the port writes this field to indicate the type of error that occurred. The port writes the following error data to this field:

- Invalid PDB field address: If the port detects an invalid PDB field during port initialization, it writes the physical host address of this field (PDB base address + offset to beginning of invalid field) to this field. This information can be used for debugging the port driver and/or the port firmware.
- Firmware PC: If the port firmware experiences an exception or other internal fatal error, it writes its program counter (PC) value to this field. This information can be used for debugging the port firmware.
- Current ring offset: If the port detects an invalid command ring entry or an invalid receive ring entry, it writes the offset of the invalid entry (in bytes) to this field. The address of the bad ring entry can be calculated by adding the offset to the base address of the affected ring.

NOTE: **For some errors (when the state qualifier = 2, 3, 5, 6, or 7), XPD1 may contain information from a previous write. This leftover information can be either the PDB base address or the default Ethernet address.**

When XPD1 contains error data, XPD2 is defined as follows:

---

xpd2 bits<31:0>

Name: Ring Release Counter

Mnemonic: None

Type: R/W to port and port driver

Indicates the total number of command and receive ring entries that the port driver has processed since that port was last initialized. The port driver updates this counter each time that it completes processing of both rings (Section 2.2.4).

NOTE: When XPD1 contains an invalid PDB field address, XPD2 is undefined. When XPD1 contains the firmware PC or the current ring offset, XPD2 contains the ring release counter.

Since the port driver initializes its internal ring release counter to -1, the ring release count is always one less than the actual number of ring entries that have been processed.

4.4 Power-Up Diagnostic Register

The Power-Up Diagnostic (XPUD) Register displays the results of the DEMNA self-test. The register is read-only to the port, the port driver, and other host software. For a description of the bit-type codes for the XPUD Register, see Table 4–2.

Power-Up Diagnostic Register (XPUD)

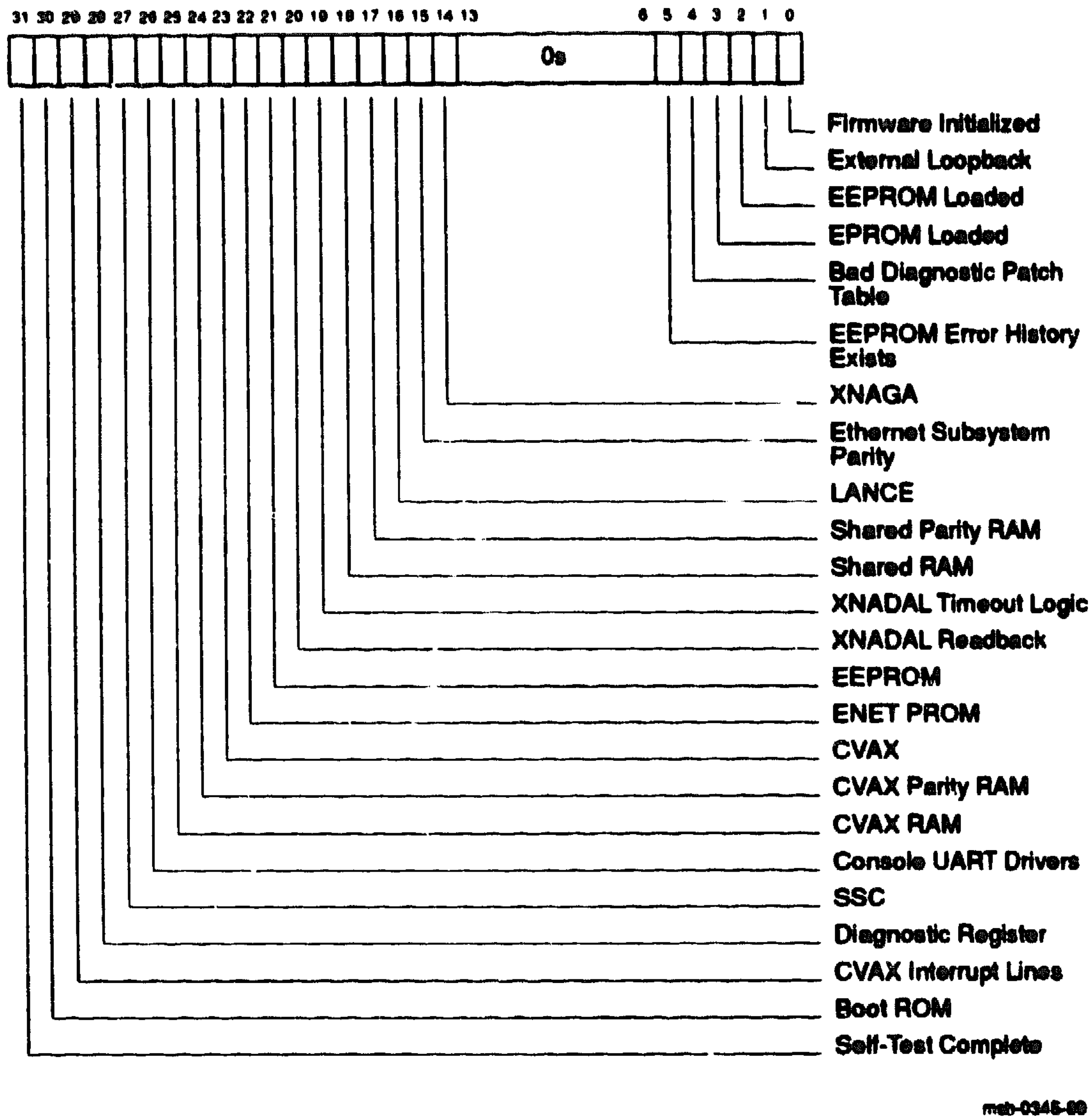
The XPUD Register displays the results of the DEMNA self-test, which runs automatically on power-up or reset. After the self-test finishes, the port driver can read the register and pass the register contents to higher-level software that can determine which DEMNA components passed self-test.

The XPUD Register is treated as follows:

- The DEMNA initializes the XPUD Register to all zeros on power-up or reset.
- When a piece of DEMNA logic passes self-test, its corresponding bit in the XPUD Register sets.
- If a piece of logic fails self-test, the corresponding bit remains cleared.

The XPUD Register of a DEMNA that passes self-test has a value of FFFFC007 (if there are no error history entries) or FFFFC027 (if there are error history entries).

ADDRESS *Nodespace base address + 10C*



bit<31>

Name: Self-Test Complete
Mnemonic: STC
Type: RO to port driver, 0

When set, indicates that the DEMNA self-test has completed and that the contents of the XPUD Register are valid. The register contents are invalid when the bit is cleared.

bit<30>

Name: Boot ROM
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the contents of the Boot ROM (also called the EPROM) are valid.

bit<29>

Name: CVAX Interrupt Lines
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the CVAX interrupt lines are not stuck (always asserted) or being driven by onboard logic.

bit<28>

Name: Diagnostic Register
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that all bits in the Diagnostic Register powered-up to the correct state and can be read and written.

bit<27>

Name: SSC
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the SSC chip can perform all its functions.

bit<26>

Name: Console UART Drivers
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the console UART drivers are functioning correctly.

bit<25>

Name: CVAX RAM
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the CVAX RAM is functioning correctly (that is, passed the CVAX RAM march test).

bit<24>

Name: CVAX Parity RAM
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the CVAX parity RAM is functioning correctly.

bit<23>

Name: CVAX
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the CVAX chip is functioning correctly.

bit<22>

Name: ENET PROM
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the contents of the ENET PROM are valid.

bit<21>

Name: EEPROM
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the contents of the EEPROM are valid.

bit<20>

Name: XNADAL Readback
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the address latches and bus transceivers for the gate array/DEMNA memory bus interface are functioning correctly.

bit<19>

Name: XNADAL Timeout Logic
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the timeout logic for the gate array/DEMNA memory bus interface is functioning correctly.

bit<18>

Name: Shared RAM
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the shared RAM is functioning correctly (that is, passed the RAM march test).

bit<17>

Name: Shared Parity RAM
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the shared parity RAM is functioning correctly.

bit<16>

Name: LANCE
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the LANCE chip can perform all its functions.

bit<15>

Name: Ethernet Subsystem Parity
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the parity circuit in the Ethernet subsystem is functioning correctly.

bit<14>

Name: XNAGA
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the gate array can perform all its functions.

bits<13:6>

Name: Reserved
Mnemonic: None
Type: RO to port driver, 0

Reserved; must be zeros.

bit<5>

Name: EEPROM Error History Exists
Mnemonic: None
Type: RO to port driver, 0

When set, indicates that the EEPROM error history has one or more entries. When cleared, indicates that there are no entries in the EEPROM error history.

---

bit<4>

Name: Bad Diagnostic Patch Table

Mnemonic: None

Type: RO to port driver, 0

When set, indicates that the diagnostic patch table in EEPROM is invalid. When cleared, indicates that this table is valid.

---

bit<3>

Name: EPROM Loaded

Mnemonic: None

Type: RO to port driver, 0

When set, indicates that the contents of the EPROM have been loaded into CVAX RAM. The EPROM contains a subset of the EEPROM code. If the EEPROM fails self-test, the contents of the EPROM are loaded into CVAX RAM. The EPROM code provides enough functionality for the CVAX to run diagnostics, update the EEPROM, and perform transmit and receive operations.

---

bit<2>

Name: EEPROM Loaded

Mnemonic: None

Type: RO to port driver, 0

When set, indicates that the contents of the EEPROM have been loaded into CVAX RAM. The EEPROM contains the operational firmware for the DEMNA.

---

bit<1>

Name: External Loopback

Mnemonic: None

Type: RO to port driver, 0

When set, indicates that the DEMNA is connected to a live Ethernet or a loopback connector and that the external loopback test has passed.

---

bit<0>

Name: Firmware Initialized

Mnemonic: None

Type: RO to port driver, 0

When set, indicates that the DEMNA firmware is initialized.

CHAPTER 5

5 Power-Up, Reset, and Shutdown

This chapter provides detailed descriptions of the DEMNA power-up, reset, and shutdown sequences. The chapter contains the following sections:

- Power-Up/Reset Sequence
- Node Halt/Restart
- Power Failure Shutdown
- Port Shutdown

5.1 Power-Up/Reset Sequence

Any of the following events causes the DEMNA to execute its power-up/reset sequence:

- System power-up: the deassertion of XMI DC LO L and XMI AC LO L.
- System reset: an XMI node asserts XMI RESET L, causing the system to cycle XMI AC LO L and XMI DC LO L through their power-down and power-up sequences.
- Node reset: the host sets the Node Reset bit in the DEMNA Bus Error (XBER) Register. DEMNA hardware logic then resets the node.

No port state is saved across a node reset, system reset, or power failure.

The power-up/reset sequence proceeds as follows:

1. During the last cycle in which XMI DC LO L is asserted, the DEMNA gate array does the following:
   a. Clears the DEMNA XMI and port registers
   b. Sets the Self-Test Fail (STF) bit in the DEMNA XBER Register, thereby asserting XMI BAD L
2. The CVAX begins firmware execution at address 2004 0000, which is the first location in boot ROM space.
3. The firmware checks the restart code to determine whether the DEMNA should perform a power-up/reset or a node halt/restart. In this case, the DEMNA continues the power-up/reset sequence.
4. The power-up/reset routine calls the DEMNA self-test, which writes results as it executes to the DEMNA Power-Up Diagnostic (XPUD) Register.
5. After all the components have been tested, the self-test reads the XPUD Register. If the DEMNA CVAX RAM failed self-test, the self-test jumps to EPROM firmware, which provides sufficient functionality to run the DEMNA ROM-based diagnostics (RBDs). If the CVAX RAM passed self-test, the self-test checks to see whether the EEPROM

contents are valid. If the EEPROM contents are valid, the self-test loads the EEPROM firmware into CVAX RAM and sets the EEPROM Loaded bit in the XPUD Register. If the EEPROM contents are invalid, the self-test loads the EPROM firmware into CVAX RAM and sets the EPROM Loaded bit in the XPUD Register.

6 The self-test routine loads the XDEV Register with 0C03 (hex), the device type for the DEMNA, and with the DEMNA functional hardware revision level and the DEMNA firmware revision level. (See Chapter 4 for a description of the XDEV Register.)

7 If all the tested DEMNA components pass, the self-test routine does the following:

   a. Lights the DEMNA OK LED

   b. Clears the STF bit in the DEMNA XBER Register, thereby deasserting XMI BAD L

   Otherwise, the above signal and LED remain in their original, powered-up states.

8 When the self-test is finished, the self-test routine sets the Self-Test Complete (STC) bit in the XPUD Register and jumps to operational firmware in RAM.

9 After self-test finishes, operational firmware, executing from RAM, does the following:

   a. Resets firmware-internal data structures, including the data link counters

   b. Sets port parameters to their default values and the DEMNA Ethernet address to the default physical address (DPA) from the MAC Address (ENET) PROM

   c. Writes the DPA from the MAC Address (ENET) PROM to the XPD1 and XPD2 Registers

   d. If self-test passed, writes the XPST Register to indicate that the port state is uninitialized (state = 1) and that no error occurred (state qualifier = 0); if self-test failed, indicates whether the firmware is still running (state qualifier = 1).

   e. While waiting for the port driver to issue an INITIALIZE command by writing the XPCI Register, processes any valid Maintenance Operation Protocol (MOP) and/or loopback packets received over Ethernet (Section 2.7).

5.2 Node Halt/Restart

The major difference between a node halt/restart and a power-up/reset is that the node halt/restart does not execute the DEMNA self-test.

The port state saved across a node/halt restart consists of the following:

- A fatal error block (Section 8.2.1) written to the Port Data Block (if the port received a node halt when in the initialized state)

- Port-internal error data that can be read with the READ$ERROR command or from the console monitor program
- DEMNA-internal counters, including data link counters that can be read by a user from the console monitor program or by the port driver with the RCCNTR/RDCNTR, READ$ERROR, READ$SNAPSHOT, and READ$STATUS commands.

The port driver initiates a node halt/restart by setting and then clearing the Node Halt (NHALT) bit in the DEMNA XBER Register. Setting the bit halts the DEMNA. Clearing the bit restarts the DEMNA.

The DEMNA responds to a node halt as follows:

1. The CVAX begins firmware execution at address 2004 0000, which is the first location in boot ROM space.
2. The firmware checks the restart code to determine whether the DEMNA should perform a power-up/reset or a node halt/restart. In this case, the DEMNA performs a node halt/restart.
3. The halt routine saves port context internally as a fatal error block.
4. The halt routine calculates a checksum for the loaded firmware (either the EEPROM firmware image or the EPROM firmware image). If the calculated checksum is incorrect, the halt/restart routine reloads the firmware. If the EEPROM firmware was loaded, the halt/restart routine reloads the EEPROM firmware. If the EPROM firmware was loaded, the halt/restart routine reloads the EPROM firmware.
5. The halt routine monitors the Node Halt bit in the DEMNA XBER Register. When the bit clears, the halt routine jumps to RAM to the restart routine.
6. If the port was in the initialized state when the Node Halt bit was set, it writes the internally saved fatal error block to the Port Error Log Area of the PDB.
7. The restart routine does the following:
   a. Resets firmware-internal data structures, including the data link counters
   b. Sets port parameters to their default values and the DEMNA Ethernet address to the default physical address (DPA) from the MAC Address (ENET) PROM
   c. Writes the DPA from the MAC Address (ENET) PROM to the XPD1 and XPD2 Registers
   d. While waiting for the port driver to issue an INITIALIZE command by writing the XPCI Register, processes any valid MOP and/or loopback packets received over Ethernet (Section 2.7).

5.3 Power Failure Shutdown

When XMI AC LO L is asserted, indicating that AC line voltage is below specification, the DEMNA executes the following power-down sequence:

1 A powerfail trap occurs on the CVAX.

2 The DEMNA executes the normal port shutdown sequence described in Section 5.4. This shutdown sequence, which lasts approximately 100 microseconds, completes before power fails.

5.4 Port Shutdown

A port shutdown occurs when the port is in the initialized state and (1) a fatal port error occurs or (2) the port driver issues a SHUTDOWN command by writing the XPCS Register.

Operational firmware executes a port shutdown as follows:

1 The port completes all in-progress datamoves (data buffer copying) to or from host memory and then stops processing. From the perspective of port users, the port simply stops.

2 The port reinitializes its internal data structures.

3 If any of the LANCE operating parameters were previously altered (such as the physical address), the port resets the LANCE. The reset occurs after any packets in the LANCE's transmit buffers have been transmitted, thus preventing the LANCE from transmitting a truncated packet with an incorrect CRC.

4 The port clears its Receive Address Filter Database, which contains all enabled multicast addresses and the actual physical address (APA) assigned by a previous PARAM command (if an APA was in fact assigned). The default physical address (DPA) is then the only remaining enabled address.

5 If the shutdown is due to a fatal port error or power failure, the port saves the port context in an internal fatal error block and then copies this block to the Port Data Block (PDB).

6 The port transitions to the uninitialized state and writes appropriate error data (if any) to the XPD1 and XPD2 Registers.

7 If the shutdown is successful and was caused by a shutdown command, the port writes the XPST Register to indicate that the port state is uninitialized (state = 1) and that no error occurred (state qualifier = 0). If the port shutdown is successful and was caused by a fatal error or a power failure, the port writes the XPST Register to indicate that the port state is uninitialized and to indicate the power failure or fatal error.

8 The port interrupts the port driver if port interrupts are enabled.

9 While waiting for the port driver to write the XPCI Register, the port processes any MOP and/or loopback packets received over Ethernet (Section 2.7).

CHAPTER 6

6 Power-Up Self-Test

The DEMNA power-up self-test consists of ROM-resident diagnostic routines that run automatically on power-up or reset. The power-up self-test verifies that the hardware at the node is operational and that the DEMNA can transmit and receive a loopback packet over the network.

Since the routines are contained in ROM, their execution requires no operating system. The self-test routines are thus stand-alone programs independent of any software environment.

6.1 How to Run Self-Test

There are several ways of running self-test for the DEMNA:

1 On system power-up—When the user powers up the host system, the DEMNA automatically runs power-up self-test. Front panel controls vary among host systems; see the system *Owner's Manual* for the specific system.

2 On XMI system reset—For VAX 6000 systems: When the user presses the reset or restart button on the host system's front panel, the system goes through its reset sequence, which causes each XMI node to run its own self-test. For VAX 9000 systems: The user can issue the following console command to reset a particular XMI card cage:

```
>>>UNJAM /XJA=n
```

where *n* is the unit number of the XJA adapter for the XMI card cage. If the XJA unit number is not supplied, the command resets all XMI card cages in the system.

3 Running self-test as a ROM-based diagnostic (RBD)—A Digital customer service engineer can invoke the self-test as an RBD from the system console of a VAX 6000 or VAX 9000 system or from the DEMNA physical console (a terminal attached directly to the DEMNA). (This is the same diagnostic that runs during power-up or reset self-test.)

Example 6–1 shows how to use VAX console commands on a VAX 6000 system to run the self-test on a DEMNA located at XMI node 3. For a description of the Z command used in this example, see the system *Owner's Manual*.

Example 6–1 Running Self-Test on a VAX 6000 System

```
>>>Z 3 [RETURN]
?33 Z connection successfully started
T/R [RETURN]
RBD3>ST 0 [RETURN]

;Selftest     3.00
;        P       3      0C03        1
;00000000 00000000 00000000 00000000 00000000 00000000 00000000
RBD3[CTRL/Z][CTRL/P]
?31 Z connection terminated by ^P
>>>
```

Example 6–2 shows how to use VAX console commands on a VAX 9000 system to run the self-test on a DEMNA at node E through XJA number 2. For a description of the Z command used in this example, see the system *Owner's Manual*.

Example 6–2 Running Self-Test on a VAX 9000 System

```
>>> Z 2E [RETURN]
[Use ^P to exit Z-mode]
T/R [RETURN]
RBD3>ST 0 [RETURN]

;Selftest     3.00
;        P       3      0C03        1
;00000000 00000000 00000000 00000000 00000000 00000000 00000000
RBD3[CTRL/Z][CTRL/P]
xxx Z connection terminated by ^P
>>>
```

If you do not know which XMI node the DEMNA is at, use the SHOW CONFIGURATION command at the system console prompt to locate the DEMNA.

Another way of locating the DEMNA nodes is to use the EXAMINE command to read the Device (XDEV) Register at each node until you find the DEMNA, which has a device type of 0C03 (hex). A module's XDEV Register is always at the module's base address.

6.2 Reporting Self-Test Results

Test results (pass or fail) are indicated by LEDs on the module and by the DEMNA Power-Up Diagnostic (XPUD) Register.

6.2.1 Self-Test Results in LEDs

There are two status-indicator lights on the module:

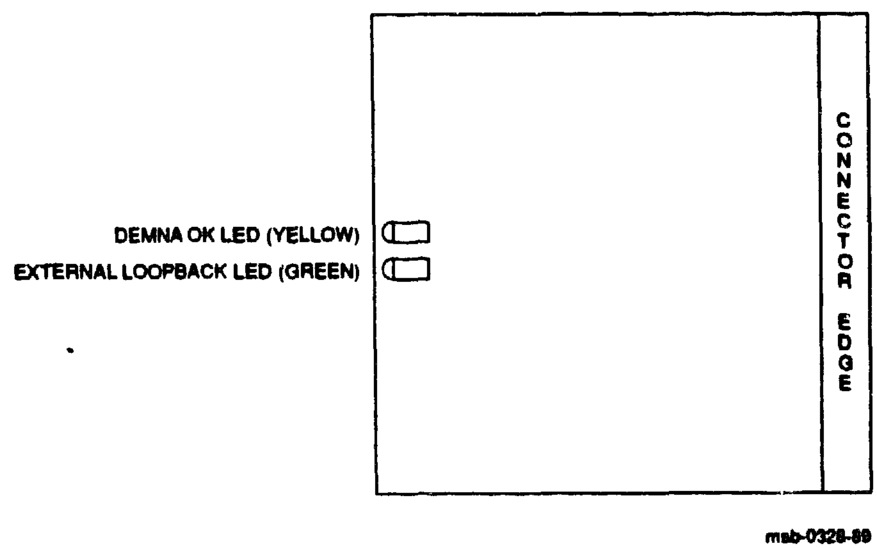
- 1 yellow DEMNA OK LED
- 1 green External Loopback LED

The location of the LEDs is shown in Figure 6–1.

The yellow DEMNA OK LED shows the status of the module after the node self-test. The green External Loopback LED indicates whether the DEMNA passed the LANCE external loopback test, which tests the DEMNA's ability to transmit and receive a loopback packet over the network.

At power-up or reset, both LEDs are off. If the DEMNA passes all the executed tests (excluding the LANCE external loopback test), the self-test lights the yellow DEMNA OK LED; otherwise, this LED remains off. If the LANCE external loopback self-test passes, the self-test lights the green External Loopback LED; otherwise, this LED remains off. (If the LEDs do not light, see Section 7.6 for a troubleshooting procedure.)

Figure 6–1 LED Locations



6.2.2 Self-Test Results in the Power-Up Diagnostic Register

The Power-Up Diagnostic (XPUD) Register indicates which tests in the self-test diagnostic passed. In addition, the Self-Test Complete (STC) bit indicates whether the self-test has completed execution. See Chapter 4 for a detailed description of the XPUD Register.

6.3 Interpreting Test Results

If the External Loopback LED fails to light, indicating that the LANCE external loopback self-test failed, this does not necessarily indicate that a DEMNA component failed self-test. The problem could be a bad cable, bad Ethernet transceiver connector, improper seating of the transceiver cable, or simply that the DEMNA is disconnected from the transceiver.[1] In any case, such an error condition prevents the DEMNA from transmitting or receiving Ethernet packets.

If the XPUD Register indicates that all of the self-test routines failed, the problem is probably the CVAX, ROM, or bus transceivers.

Self-test could also fail because of a systemwide fault. For example, a faulty power supply or missing XMI bus terminators could be the problem. Make sure that system power is OK and check for other possible systemwide faults.

After a problem discovered by the self-test has been corrected, the DEMNA LED(s) will light only if the self-test is rerun. However, if the self-test is not rerun, the DEMNA will still function properly even though the LED(s) don't light.

6.4 Tested Components

The self-test tests the following components and functions on the DEMNA module:

- CVAX
- EPROM
- EEPROM
- CVAX ROM
- All RAM
- System Support Chip (SSC)
- CVAX Interrupt Request (IRQ) lines
- Gate Array
- LANCE chip

6.5 Untested Components and Functions

The following components and functions are not tested:

- A complete CVAX instruction set
- Ethernet interface logic functions:
  - More (multiple retries of packet transmission)

[1] Note also that the External Loopback LED does not light if another test in the self-test diagnostic fails. When another self-test fails, the external loopback test is not executed.

- One (one retry of a packet transmission)
- Babble error
- Time domain reflectometry
- Late collision
- Loss of carrier

The datamove logic, which is implemented in the gate array, is tested in loopback mode only.

CHAPTER 7

7 Diagnostics and Troubleshooting

This chapter describes diagnostics that test the DEMNA module. The chapter also provides a troubleshooting procedure that uses these diagnostics. The chapter contains the following sections:

- ROM-Based Diagnostics
- How to Run ROM-Based Diagnostics from the System Console
- Console Commands and Control Characters
- Test Status and Error Reporting
- Diagnostic Error Log Reader
- Troubleshooting with ROM-Based Diagnostics
- Diagnostic Patch Mechanism
- Software Diagnostics

Two types of diagnostics are used to test the DEMNA: ROM-based diagnostics (RBDs) and software diagnostics. RBDs are resident in EPROM on the module and are invoked from the system console. Software diagnostics are loaded and run under the VAX Diagnostic Supervisor (VAX/DS). Customers must purchase software diagnostics under a separate license.

The DEMNA EPROM contains four ROM-based diagnostics (RBDs):

- DEMNA self-test, which verifies the functionality of the DEMNA module
- NI RBD, which verifies the Ethernet link from the DEMNA's Ethernet interface logic to the Ethernet transceiver
- XMI RBD, which verifies that the DEMNA can transfer data to and from host memory
- XNA RBD, which exercises the DEMNA by performing Ethernet external loopbacks and datamoves to and from host memory at the same time

Three software diagnostics are available for the DEMNA:

- EVGDB—DEMNA EEPROM Update Utility, which verifies the DEMNA firmware revision, loads a new firmware image into DEMNA EEPROM, and/or allows the user to change the setting of various flags and parameters in EEPROM
- EVDWC—NI Exerciser, which tests the installation of the host Ethernet node and checks the connectivity of all other nodes on the local network

- EVDYE—DEMNA NI Functional Diagnostic, which verifies that the DEMNA's NI port performs all the functions that the VAX/VMS port driver, EXDRIVER, may request

If the appropriate flags are set in DEMNA EEPROM (Chapter 9), errors reported by the self-test, NI RBD, XMI RBD, and/or XNA RBD are logged in DEMNA EEPROM. The first eight self-test and/or RBD errors can be logged. The error history must then be cleared to allow logging of self-test and/or RBD errors to resume.

The RBD code resident in EPROM can be patched. Patches to diagnostic code, as well as updates to the DEMNA operation firmware, are included in the DEMNA firmware image (EVGDBQ.BIN) distributed with the DEMNA EEPROM Update Utility, EVGDB. If the diagnostic code must be patched or the operation firmware must be updated, a new version of EVGDB is released.

7.1 ROM-Based Diagnostics

The tests in each category are described in detail in the following sections. Individual tests, or test groups, can be invoked with commands from the operator's console or from the DEMNA physical console. If a test fault occurs, deductive reasoning, along with a comprehensive understanding of the test routine, is required to determine which hardware element is malfunctioning.

7.1.1 DEMNA Self-Test

The DEMNA self-test , which is functionally equivalent to the DEMNA's power-up self-test, performs a basic confidence check of the module.

When run as a power-up self-test (which occurs when the system power signals, XMI AC LO L and XMI DC LO L are cycled or when the Node Reset bit is set in the DEMNA XBER Register), the self-test does the following:

- Lights the DEMNA LEDs if no self-test errors occur
- Asserts the DEMNA XMI BAD L signal and then deasserts this signal if no self-test errors occur
- Sets the Self-Test Fail bit in the DEMNA XBER Register and then clears this signal if no self-test errors occur
- Writes the test results to the DEMNA Power-up Diagnostic (XPUD) Register

However, when the self-test is invoked as an RBD from the system console, it does not provide any of the above-listed indications. Instead, it prints the test results on the system console terminal.

Table 7–1 briefly describes each test in the DEMNA self-test diagnostic. The order in which the tests are listed is the order in which they are run.

Unless otherwise specified in Table 7–1, the test execution parameters and method of fault reporting can be selected with the appropriate runtime switches (Table 7–2). If a test number is not specified with the START command used to invoke the diagnostic, all tests are executed by default. Unless halt-on-error is specified, the self-test continues running on encountering an error. This provides for maximum testing of the module.

Table 7–1 DEMNA Self-Test

Diagnostic/Test	Name	Description
T1	Boot ROM	Verifies the EPROM (Boot ROM) contents by calculating a checksum for the contents and comparing the calculated checksum to a corresponding stored checksum. The test also verifies the proper functioning of the select logic for the SSC Boot ROM space.
T2	CVAX IRQ Lines	Steps through all IPLs (interrupt priority levels) to verify that no IRQ line is stuck asserted or is being asserted by onboard logic.
T3	Diagnostic Register	Verifies that the DEMNA Diagnostic Register can be written and read.
T4	SSC Chip	Verifies that the logic in the SSC chip is functioning properly.
T5	Console UART Driver	Verifies that the drivers for the console UART are functioning properly.
T6	CVAX RAM	Verifies that each cell in CVAX RAM is uniquely accessible and that the CVAX RAM select logic and address decode logic is functioning properly.
T7	CVAX Parity RAM	Verifies that each cell in the CVAX Parity RAM is uniquely accessible and that the select logic for the CVAX Parity RAM is functioning properly.
T8	CVAX Chip	Verifies the proper functioning of certain CVAX internal processor registers (IPRs). CVAX parity generation and detection logic, and that the CVAX can perform quadword reads from CVAX RAM.
T9	ENET PROM	Verifies the contents of the ENET PROM by calculating checksums for the contents and comparing the calculated checksums to corresponding stored checksums. The test also verifies the proper functioning of the select logic for the SSC Boot ROM space.

Table 7-1 (Cont.) DEMNA Self-Test

Diagnostic/Test	Name	Description
T10	EEPROM	Verifies the contents of the EEPROM by calculating checksums for the contents and comparing the calculated checksums to corresponding stored checksums. The test also verifies the proper functioning of the select logic for the SSC Boot ROM space.
T11	XNADAL Readback	Verifies that XNADAL bus transceivers and address latches are functioning properly.
T12	XNADAL Timeout Logic	Verifies that the timeout logic for the XNADAL bus is functioning properly.
T13	Shared RAM	Verifies that each cell in shared RAM is uniquely accessible and that the shared RAM select logic and address decode logic is functioning properly.
T14	Shared Parity RAM	Verifies that each cell in the parity RAM for the shared RAM is uniquely accessible and that the select logic for the parity RAM is functioning properly.
T15	LANCE Chip	Verifies LANCE chip and CRC logic in the Ethernet interface logic. (SIA chip is not tested.)
T15	Ethernet Subsystem Parity	Verifies the proper functioning of the Ethernet subsystem parity logic, which generates parity on LANCE writes to shared RAM and checks parity on LANCE reads from shared RAM.
T17	LANCE External Loopback	Attempts to transmit and receive an Ethernet external loopback packet. If one packet is successfully transmitted and received, the firmware lights the External Loopback LED on the DEMNA. This test is the only test that can fail without causing the self-test as a whole to fail.
T18	Gate Array	Verifies the proper operation of the DEMNA gate array logic, excluding the XMI-required registers, the port registers, datamove registers, peek registers, as well as datamove and peek operations.

7.1.2 NI RBD

The Network Interconnect (NI) RBD consists of the three test routines described below. These tests verify that the Ethernet interface logic, cable, link, and transceiver are operational. If a test number is not specified when this diagnostic is selected, test 1 (external loopback on live Ethernet) is executed by default.

7.1.2.1 External Loopback on Live Ethernet Test

This test transmits and receives Ethernet packets using the LANCE's external loopback mode. The packets are transmitted onto the network, received from the network, and then compared for accuracy. The DEMNA must therefore be connected to a live network for this test to run properly.

The test does not flag as errors events that occur as part of normal network activity. Instead, the test is automatically retried if any of the following events is detected:

- Loss of carrier
- Late collision
- Retry error
- CRC error
- Framing error
- Babble error (also reported as a soft error)
- Missed packet error (also reported as a soft error)
- Overflow error (also reported as a soft error)
- Buffer error
- Collision error (loss of heartbeat)

Test 1 fails if any of the following occurs:

- 32 soft errors
- Initialization error
- Memory error
- Underflow error

The number of packets to be transmitted per test pass can be specified in decimal through the use of a command parameter. The default is 100 packets. When 0 packets are specified, packets are transmitted until [CTRL/C] is typed.

The successful completion of the test indicates that the DEMNA is properly cabled to a network and that the DEMNA, Ethernet transceiver (or DELNI), and transceiver cable are functioning properly. Note, however, that the test verifies DEMNA operation only in loopback mode.

7.1.2.2 MOP Loopback Test

The MOP loopback test transmits and receives MOP (Maintenance Operations Protocol) loopback packets. For this test to run properly, there must be at least one node on the local network that implements a MOP loop server.

During the test initialization, the DEMNA (functioning as a MOP loop requester) transmits a loopback assistance multicast address packet that requests all nodes that are MOP loop servers to identify themselves. The DEMNA uses the first such node as the server for the MOP loopback test and saves approximately the next 80 addresses of other nodes that respond. If the first MOP loop server fails, the second MOP loop server is used, and so on. If all the MOP loop servers fail, the test is aborted.

During the test, the DEMNA transmits MOP loopback packets to the MOP loop server, which transmits the packets back to the DEMNA. The DEMNA then compares each transmitted packet with its corresponding receive packet for accuracy. Note that the DEMNA is transmitting packets in normal fashion: the LANCE is not in external loopback mode but in normal operational mode.

The number of packets to be transmitted per test pass can be specified in decimal through the use of a command parameter. The default is 100 packets. When 0 packets are specified, packets are transmitted until CTRL/C is typed.

The packet length can also be specified in decimal through the use of a second command parameter. The packet length is specified in bytes within the inclusive range of 64–1518 bytes. If fewer than 64 bytes are specified, 64 bytes will be used. If more than 1518 bytes are specified, 1518 bytes will be used. If no packet size is specified, the test varies the packet size.

The first command parameter always specifies the number of packets to be transmitted per test pass. Therefore, if the user wishes to specify the packet size, he or she must also specify the number of packets to be transmitted.

The successful completion of the test indicates that the DEMNA is properly cabled to a network and that the DEMNA, Ethernet transceiver (or DELNI), and transceiver cable are functioning properly. In addition, the test indicates that the DEMNA can transmit and receive properly in normal mode (as opposed to loopback mode.)

7.1.2.3 External Loopback on Closed Ethernet Test

Like the first NI test, the external loopback on closed Ethernet test transmits and receives Ethernet packets using the LANCE's external loopback mode. However, instead of using a live network, the external loopback on closed Ethernet test requires a closed loop. The closed loop is provided by a loopback connector that must be installed either on the Ethernet connector at the system bulkhead or on the transceiver end of the transceiver cable. Since the external loopback on closed Ethernet test does not use a live network, it can be used to verify DEMNA operation apart from the network operation.

Unlike the first NI test, the external loopback on closed Ethernet test does not ignore events (such as retries and collisions) that occur as part of normal network activity. Instead, the test reports these events as errors. That is why the test must be run using a closed Ethernet loop rather than on a live Ethernet.

The number of packets to be transmitted per test pass can be specified in decimal through the use of a command parameter. The default is 100 packets. When 0 packets are specified, packets are transmitted until CTRL/C is typed.

The successful completion of the test indicates that the DEMNA is functioning properly in loopback mode. The test does not verify network operation.

7.1.3 XMI RBD

The XMI RBD, which consists of one test, verifies that the DEMNA can move data between itself and host memory. Both datamove operations and peek operations are tested. For the test to run properly, host memory must be accessible to the DEMNA.

The number of datamove and peek operations executed per test pass can be specified in decimal through the use of a command line parameter. A parameter value of *n* specifies *n* * 256 datamoves and *n* * 512 peeks. A value of 1 thus specifies 256 datamoves and 512 peeks, a value of 2 specifies 512 datamoves and 1024 peeks, and so on. The default is 256 datamoves and 512 peeks per pass. When a parameter value of 0 is specified, the DEMNA executes datamoves and peeks until [CTRL/C] is typed.

The starting address of host memory used by the test can be specified with a second command line parameter. The starting address must be page-aligned. If a nonpage-aligned address is specified, the nine least significant bits are zeroed to make the address page-aligned. The default starting address is 200 (hex). The test requires 4 Kbytes of memory from the starting address to run properly.

The first command parameter always specifies the number of datamoves and peeks executed per test pass. Therefore, if the user wishes to specify the starting address of host memory used by the test, he or she must also specify the number of datamoves and peeks to be executed.

The XMI RBD is a destructive test, since it destroys the contents of the host memory locations used to execute datamoves and peeks. The user must therefore use the /C switch when invoking the test.

Successful completion of the XMI RBD indicates that the DEMNA can properly execute datamove and peek operations at the same time. This does not imply, however, that all of host memory is functioning properly.

7.1.4 XNA RBD

The XNA RBD verifies that the DEMNA can do the following at the same time:

- Transmit and receive Ethernet packets using the LANCE's external loopback mode
- Move data between itself and host memory

The XNA RBD is essentially a combination of the external loopback with live Ethernet test (test 1) of the NI RBD, which transmits and receives loopback packets, and the XMI RBD, which performs datamove and peek operations to and from host memory.

The XNA RBD consists of a single test. As with the XMI RBD, the number of datamoves and peeks to be performed can be specified with an optional parameter p1, and the starting address of host memory can be specified with an optional parameter p2. See Section 7.1.3 for a description of these optional parameters.

The XNA RBD is a destructive test, since it destroys the contents of the host memory locations used to execute datamoves and peeks. The user must therefore use the /C switch when invoking the test.

7.2 How to Run ROM-Based Diagnostics from the System Console

The ROM-based diagnostics are stand-alone programs that can be controlled and executed from the operator's console of a VAX 6000 system or VAX 9000 system or from the DEMNA physical console.

To run the RBDs from the system console of a VAX 6000 system, do the following:

1. Enter console mode by typing the following commands at the system prompt:

```
>>>Z n [RETURN]
xxx Z connection successfully started
   T/R [RETURN]
RBDn>
```

   where: *n* is the XMI node ID of the DEMNA under test (0–F hex)

2. Use the START command along with the applicable runtime switches.
3. Terminate the diagnostics by entering the QUIT command or one of the control characters described below.

To run the RBDs from the system console of a VAX 9000 system, substitute the following for step 1 above:

1. Enter console mode by typing the following commands at the system prompt:

```
>>>Z nm [RETURN]
xxx Z connection successfully started
   T/R [RETURN]

RBDm>
```

   where:

   - *n* is the XJA unit number (0–3 hex)
   - *m* is the XMI node ID of the DEMNA under test (1–E hex)

To run the RBDs from the DEMNA physical console, enter the following command:

```
XNA>T/R [RETURN]
RBDn>
```

7.3 Console Commands and Control Characters

The ROM diagnostic console commands are as follows:

- START
- DEPOSIT
- EXAMINE

- QUIT
- SUMMARY
- XFC

Each command is described in detail in the following sections. Examples are also included. In all cases, when an illegal or invalid command is typed, the keyboard alarm sounds, a question mark is displayed, and the command-entry prompt is redisplayed as shown.

```
RBDn>Invalid Command
 ?
RBDn>
```

7.3.1 START

The START command invokes a designated test, or group of tests, within a specified diagnostic. The command format is:

```
ST[ART]n [/Sw1/Sw2/...Sw9 p1 p2]
```

n
An integer th· · pecifies the RBD to be executed as follows:

0—self-test RBD
1—NI RBD
2—XMI RBD
3—XNA RBD

Sw
Switch (see Table 7–2)

p1, p2
Parameter codes. All the NI RBD tests support one optional parameter (p1), which is an 8-digit decimal number that specifies the number of packets to be transmitted and received for each test pass. A value of zero for p1 specifies that packets be transmitted indefinitely until CTRL/C is entered. The default value for this parameter is one. The MOP loopback test of the NI RBD also supports a second optional parameter (p2), which specifies the transmit packet size. Parameter p2 must be a decimal number for 64 to 1518. The default value for parameter p2 is 64.

The XMI RBD and XNA RBD both support two optional parameters (p1 and p2). Parameter p1 specifies the number of datamove and peek operations executed per test pass. The default value for parameter p1 is 256 datamoves and 512 peeks. Parameter p2 specifies the starting address of host memory used by the test. The default value for parameter p2 is 200 (hex).

Table 7-2 START Command Switches

Switch	Name	Default	Description
/BE	Bell on Error	No bell	Sounds keyboard alarm each time an error is detected. This is useful when error printout is inhibited and loop-on-error is specified.
/C	Confirmation	No confirmation	Allows the execution of destructive tests (namely, the XMI RBD and the XNA RBD)
/DS	Disable Status	Status reports	Disables the printing of status reports.
/HE	Halt on Hard Error	Continue on hard error	Causes the diagnostic to halt on the test that detects the first hard error. (In this context, a hard error is defined as a recoverable, repeatable error—for example, a ROM checksum error. This differs from a fatal error, which is an unrecoverable fault—for example, an unexpected interrupt or exception. A fatal error always causes program abortion, regardless of the state of the /HE or /LE switch.) On detecting a hard error, the diagnostic prints an error report and a summary report, executes cleanup code, and returns to the command-entry prompt. If /HE is specified with /LE in the same command, the RBD monitor indicates an error.
/HS	Halt on Soft Error	Continue on soft error	Causes the diagnostic to halt on the test that detects the first soft error. (In this context, a soft error is defined as a recoverable error that goes away after retry.) On detecting a soft error, the diagnostic prints an error report and a summary report, executes cleanup code, and returns to the command-entry prompt. If /HS is specified with /LS in the same command, the RBD monitor indicates an error. The /HS switch is applicable only for the MOP loopback test and the external loopback on live Ethernet test of the NI RBD. The only soft error detected is a missing heartbeat.
/IE	Inhibit Errors	Error reports	Disables error reports during diagnostic execution. However, once the diagnostic finishes, error reports are printed on the console. The /IE switch, when used with the /LE switch, is useful for module repair.
/IS	Inhibit Summary	Summary reports	Disables the printing of summary reports after diagnostic execution completes.
/LE	Loop on Hard Error	Continue on error	Causes the diagnostic to loop on the test that detects the first hard error, even if the error is intermittent. Type CTRL/C, CTRL/Y, or CTRL/Z to terminate looping and return to the command-entry prompt (RBDn>). Unless suppressed with the /IE switch, error reports are printed on the console during the loop on error. In addition, a summary report is printed on the console when looping is terminated unless summary reports have been suppressed with the /IS switch.

Table 7-2 (Cont.) START Command Switches

Switch	Name	Default	Description
/LS	Loop on Soft Error	Continue on error	Causes the diagnostic to loop on the test that detects the first soft error, even if the error is intermittent. Type CTRL/C, CTRL/Y, or CTRL/Z to terminate looping and return to the command-entry prompt (RBDn>). Unless suppressed with the /IE switch, error reports are printed on the console during the loop on error. In addition, a summary report is printed on the console when looping is terminated unless summary reports have been suppressed with the /IS switch. This switch is applicable only for the MOP loopback test and the external loopback on live Ethernet test of the NI RBD. The only soft error detected is a missing heartbeat.
/P=n	Pass Count	One pass	Specifies the total number of diagnostic passes. (One pass equals one iteration of all selected tests.) A pass count of 0 selects an infinite number of passes. Type CTRL/C, CTRL/Y, or CTRL/Z to terminate a diagnostic executing an infinite number of passes.
/T=n[:m]	Test Number[s]	Unique to each diagnostic	Specifies a test or group of tests to be executed. If a group is specified, tests are executed in ascending numerical order. Both *n* and *m* are decimal values and must be within the range of test numbers for the diagnostic being executed. If an incorrect test number or incorrect range of test numbers is specified, the RBD monitor indicates an error.
/TR	Enable Trace	Trace disabled	Prints each test number when the test finishes execution, thus allowing the user to trace the progress of diagnostic testing.

Examples of START Command

1. `RBD3>ST0`

   Executes one pass of all tests of the self-test RBD.

2. `RBD3>ST0/P=0/T=4:6/BE`

   Executes tests 4–6 of the self-test RBD in loop-forever mode with bell-on-error active and trace reports disabled by default. Summary, status, and error reports are enabled by default.

3. `RBD3>ST0/P=1/TR/HE`

   Executes one pass of all tests in the self-test RBD in halt-on-error mode with trace reports enabled. Summary, status, and error reports are enabled by default.

4. `RBD3>ST0/T=8/LE/BE`

   Executes one pass, by default, of test 8 of the self-test RBD in loop-on-error mode with bell-on-error active and trace reports disabled by default. Summary, status, and error reports are enabled by default. (If test 8 fails initially, the diagnostic loops forever, even though the test may subsequently pass.) To stop, use CTRL/C.

5 `RBD3>ST1`

Executes one pass of the external loopback on live Ethernet test of the NI RBD (which is the default test executed).

6 `RBD3>ST1/T=1/P=0`

Executes the external loopback on live Ethernet test of the NI RBD for an infinite number of passes. To stop, use CTRL/C.

7 `RBD3>ST1/T=2 100`

Executes one pass of the MOP loopback test of the NI RBD, during which 100 packets will be transmitted.

8 `RBD3>ST1/T=2 0 1000`

Executes one pass of the MOP loopback test of the NI RBD, during which an infinite number of 1000-byte MOP loopback packets will be transmitted. To stop, use CTRL/C.

9 `RBD3>ST1/T=3/TR 200`

Executes one pass of the external loopback on closed Ethernet test of the NI RBD, during which 200 packets will be transmitted. Trace reports are enabled.

10 `RBD3>ST2/P=3 2 1000 /C`

Executes three passes of the XMI RBD. Each pass will perform 512 datamoves and 1024 peeks. The starting address of host memory used by the test will be 1000 (hex).

11 `RBD3>ST3 4 /C`

Executes one pass of the XNA RBD, during which 1024 datamoves and 2048 peeks will be performed. The starting address of host memory used by the test will be 200 (hex) by default.

7.3.2 DEPOSIT Command

The DEPOSIT command deposits data into XMI registers and DEMNA memory locations. The command can deposit data only to local locations (locations on the DEMNA module).

If the data length and/or address type are not specified, the defaults are the data length and/or address type used by the previous DEPOSIT or EXAMINE command. When the RBD monitor is invoked, the default address is 0, the default address type is physical, and the default data size is longword.

If the specified data size is too large for the location to be deposited, the RBD monitor ignores the command and issues an error message (?). If the specified data is too small for the location to be deposited, the RBD monitor zero-extends the data.

The DEPOSIT command has the following format.

```
D[EPOSIT] [/qualifier] [address] [data]
```

address

a 1- to 8-digit hexadecimal value or one of the following:

- +, the location immediately following the last location referenced by the previous DEPOSIT or EXAMINE command. The location referenced by + is the last location plus the current data size (byte, word, or longword)
- -, the location immediately preceding the last location referenced by the previous DEPOSIT or EXAMINE command. The location referenced by - is the last location minus the current data size (byte, word, or longword).
- *, the last location referenced by the previous DEPOSIT or EXAMINE command.

/qualifiers

/B	Defines the data size as a byte.
/L	Defines the data size as a longword. This is the default data size.
/N:*n*	Specifies a range of addresses to which the command will deposit data. The range starts with the address specified in the address field of the command and continues with the next *n* higher locations. The data specified in the data field of the command is deposited to all locations in the specified range. Note that even when the starting address is specified with "-", the next *n* addresses are always higher addresses (that is, "-" specifies only the starting address, not the direction).
/P	Defines the address space as physical memory. This is the default address type.
/W	Defines the data size as a word.

Examples of DEPOSIT Command

1 `RBD3>D/W 00001111 FFFF`

Deposit word FFFF into location 00001111.

2 `RBD3>D/B* A5`

Deposit byte A5 into the current location.

3 `RBD3>D/L/N:3 00000020 0`

Deposit 00000000 into locations 00000020, 00000024, 00000028, 0000002C.

4 `RBD3>D/W- A5A5`

Deposit word A5A5 into the previous location.

```
5    RBD3>D/B+ A5A5
     ?
```

Try to use a "byte" command to deposit an illegal byte value into the next location.

7.3.3 EXAMINE Command

The EXAMINE command displays the contents of DEMNA XMI registers and DEMNA memory locations. The command can examine only local locations (locations on the DEMNA module).

If the data length and/or address type are not specified, the defaults are the data length and/or address type used by the previous DEPOSIT or EXAMINE command. When the RBD monitor is invoked, the default address is 0, the default address type is physical, and the default data size is longword.

The EXAMINE command has the following format:

```
E[XAMINE] [/qualifier] [address]
```

address

a 1- to 8-digit hexadecimal value or one of the following.

1 +, the location immediately following the last location referenced by the previous DEPOSIT or EXAMINE command. The location referenced by + is the last location plus the current data size (byte, word, or longword).

2 -, the location immediately preceding the last location referenced by the previous DEPOSIT or EXAMINE command. The location referenced by - is the last location minus the current data size (byte, word, or longword).

3 *, the last location referenced by the previous DEPOSIT or EXAMINE command.

If no address or equivalent (+, -, *) is specified, the address used is equivalent to that specified by +.

/qualifiers

/B	Defines the data size as a byte.
/G	Defines the address space as the CVAX general purpose register set, registers 0 through 11. When /G is used, the address field must be a single hexadecimal digit between 0 and B. These hex digits correspond to the 12 GPRs.
/L	Defines the data size as a longword. This is the default data size.

/N:*n*	Specifies a range of addresses to be examined. The range starts with the address specified in the address field of the command and continues with the next *n* higher locations. Note that even when the starting address is specified with "-", the next *n* addresses are always higher addresses (that is, "-" specifies only the starting address, not the direction).
/P	Defines the address space as physical memory. This is the default address type.
/W	Defines the data size as a word.

Examples of EXAMINE Command

1
```
RBD3>E/L 20050112
P  20050112  0D25410D
```

Examine 20050112; it contains 0D25410D.

2
```
RBD3>E*
P  20050112  0D25410D
```

Examine 20050112; it contains 0D25410D.

3
```
RBD3>E/L
P  20050116  0000A5A5
```

Examine longword at 20050116; it contains 0000A5A5.

4
```
RBD3>E+
P  2005011A  000000A5
```

Examine longword at 2005011A; it contains 000000A5.

5
```
RBD3>E/W*
P  2005011A  00A5
```

Examine word at 2005011A; it contains 00A5.

6
```
RBD3>E/G A
G  0000000A  0011
```

Examine GPR 10; it contains 0011.

7.3.4 QUIT

The QUIT command sets the Node Reset bit in the DEMNA Bus Error (XBER) Register, thereby initializing the DEMNA and causing it to execute a power-up self-test. If the RBD monitor is being accessed through the host's system console, [CTRL/P] must be entered after the QUIT command to return to the console prompt. [CTRL/Z] has the same function as the QUIT command. The QUIT command has the following format:

```
QU[IT]
```

Example 7–1 Example of QUIT Command

```
RBD3> QUIT [CTRL P]
xxx 2 connection terminated by ^P
>>>
```

Example 7–2 Example of SUMMARY Command

```
RBD3> SUMMARY

;        P        5      0C03        10
;00000000 00000000 00000000 00000012 00000064 00000000 00000C80
```

7.3.5 SUMMARY

The SUMMARY command prints a summary report of the last diagnostic that was executed. If no diagnostic has been run since the RBD monitor was last invoked, the RBD monitor returns an error indication (?). Section 7.4.3 defines the fields of the summary report. The SUMMARY command has the following format:

```
SU[MMARY]
```

7.3.6 XFC

The XFC command forces a jump to the address loaded into Port Data Register 1 (XPD1). The only use for this command is to invoke the diagnostic error log reader. The XFC command has the following format:

```
XFC
```

See Section 7.5 for a description of how the XFC command is used to invoke the diagnostic error log reader.

7.3.7 Control Characters

Six control characters and one special character affect the operation of the ROM-based diagnostics. Generally, these characters are used to abort test routines, reset the DEMNA, or exit from the console mode as described in Table 7–3. To enter a control character, press and hold down the [CTRL] key and then simultaneously press the desired character key once.

Table 7–3 defines the control characters and the special character recognized by the RBD monitor.

Table 7-3 Console Mode Control Characters and Special Character

Character	When an RBD Is Executing	When the RBD Monitor Is Executing
CTRL/C	Aborts test routine, executes cleanup code, returns to RBD monitor, and reissues the RBDn> prompt. The enabled messages for the aborted test are printed on the console.	Echoes "^C", reissues RBDn> prompt.
CTRL/P	Terminates console mode and returns to the system-level prompt (>>>). If you reenter the RBD test monitor on the same node, the enabled test messages of the aborted test are printed on the console. CTRL/P is disabled when RBDs are run from the console monitor program.	Terminates console mode and returns to the system-level prompt (>>>). CTRL/Z or QUIT must be used before CTRL/P to put adapter in a known state. CTRL/P is disabled when RBDs are run from the console monitor program.
CTRL/R	Ignored.	Reprints the current command line.
CTRL/U	Ignored.	Echoes "^U", aborts current command line, and reissues RBDn> prompt.
CTRL/Y	Aborts test routine, and returns to the diagnostic prompt. Does not execute cleanup code. No test messages are printed on the console.	Echoes "^Y", reissues RBDn> prompt.
CTRL/Z	Aborts test routine, executes cleanup code, returns to RBD monitor, and reissues the RBDn> prompt. The enabled test messages of the aborted test are printed on the console. Same as CTRL/C.	Resets the DEMNA and executes power-up self-test; same as QUIT command. Use CTRL/P to return to the system console prompt (>>>).
DELETE	Ignored.	Deletes the last character of the current command line. The deleted character is printed with a preceding backslash (\). The next valid character typed by the user is substituted for the deleted character. The substituted character is also printed with a preceding backslash.

7.4 Test Status and Error Reporting

ROM-based diagnostics have three types of status and error reports:

1 Error reports
2 Status reports
3 Summary reports

The following sections describe each report in detail.

7.4.1 Error Reports

Three types of errors are reported:

- System fatal errors
- Device hard errors
- Device soft errors

A system fatal error is an error that prevents the diagnostic from running to completion. Such errors include unexpected machine checks during diagnostic execution, powerfail interrupts, and certain exceptions. A system fatal error causes the diagnostic to abort.

A device hard error is an error that prevents the tested device from being able to perform the current operation under test. A device hard error is thus an unrecoverable error. After reporting a device hard error, the diagnostic performs the action specified in the invoking command: halt on error, loop on error, or continue on error (default).

A device soft error is an intermittent hardware error that may not recur if the test is repeated. A device soft error is thus a recoverable error. After reporting a device soft error, the diagnostic performs the action specified in the invoking command: halt on error, loop on error, continue on error (default).

Figure 7–1 shows a sample error report. Table 7–4 lists the fields in each line of the report. Table 7–5 defines these fields.

Figure 7–1 Sample Error Report

```
;       F          5       0C03       18
;       HE CVAX RAM          00    T0006
;       00 55555555 5A555555 00000000 00001204 2004DF14 02
```

Table 7–4 Error Report Fields

Line	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7
1	Status	Node No.	Device Type	Passes			
2	Err Type	ASCII L	Unit No.	Test			
3	Err Code	Expected	Received	SCB	Address	PC	Err Number

Table 7–5 Error Report Field Definitions

Mnemonic	Name	Description
LINE 1		
—	Status	The test status (pass or fail). For error reports, this field always contains an F, indicating that the test failed.
NODE NO.	Node Number	The XMI node under test. The value may range from 0–F (hex).
—	Device Type	The type of module under test. 0C03 (hex) = DEMNA.
PASS	Pass Count	The execution pass at which the error was detected.
LINE 2		
ERR TYPE	Error Type	The error type: FE = system fatal error HE = hard device error SE = soft device error
ASCII L	ASCII Logic	An ASCII code that indicates the failing logic: LANCE = LANCE chip CVAX = CVAX CVAX_RAM = CVAX RAM CVAX_PAR = CVAX Parity RAM DIAG_REG = Diagnostic Register EEPROM = EEPROM ENET = ENET PROM EPROM = EPROM ETH_PAR = Ethernet parity logic INT_LIN = Interrupt lines M_CHECK = Machine check SSC = System Support Chip SETUP_ER = Setup error for the MOP loopback test of the NI RBD (no MOP loop servers available) SHR_RAM = Shared RAM SHR_PAR = Shared Parity RAM TIMEOUT = DEMNA memory bus timeout UART = Drivers for the UART UNEX_INT = Unexpected interrupt UNKNOWN = Unknown error XNADAL = DEMNA memory bus XNAGA = gate array

Table 7-5 (Cont.) Error Report Field Definitions

Mnemonic	Name	Description
LINE 2		
UNIT	Unit Number	The unit under test (always 0).
TEST	Test Number	The test that detected the error. This field always contains a "T" followed by a 4-digit decimal number that indicates the test that detected the error. A value of zero indicates that the initialization code for the test failed.
LINE 3		
ERR CODE	Error Code	The subtest of the specified test. See Appendix F for a list of error codes.
EXPCT	Expected Data	The expected data for certain types of data comparison errors.
RCVD	Received Data	The incorrect data that was received for a data comparison.
SCB	System Control Block Offset	The system control block offset through which an interrupt was expected or received.
ADD	Address	The memory location or register address at which a data comparison error or register operation error was detected.
PC	Program Counter	The PC (program counter) value in ROM at which the error was detected.
ERR NUM	Error Number	A number that allows the user to determine the particular error that occurred. See Appendix F for a description of error numbers for each RBD.

7.4.2 Status Reports

Status reports are printed during long-running tests to indicate that the diagnostic is still running and to show the status of the diagnostic. Status reports may also be printed when the LANCE reports an error that is neither a hard nor soft error but is simply a retry condition. This indicates that the LANCE is experiencing a problem (for example, due to network traffic) that is not serious enough to warrant an error report. Status reports can be printed by all of the RBDs except the self-test.

A status report consists of two lines (Figure 7-2). Table 7-6 lists the fields in each line. Table 7-7 defines the fields.

Figure 7-2 Sample Status Report

```
;       S       5       0C03       1
;       XX   NI_RBD       00    T0001
```

Table 7–6 Status Report Fields

Line	Field 1	Field 2	Field 3	Field 4
1	Status	Node No.	Device Type	Passes
2	Err Type	Diagnostic	Unit No.	Test

Table 7–7 Status Report Field Definitions

Mnemonic	Name	Description
Line 1		
—	Status	Report type. S = status report.
NODE NO.	Node Number	Node under test. The value may range from 0–F (hex).
—	Device Type	The type of module under test. 0C03 (hex) = DEMNA module.
PASS	Pass Count	Number of passes completed so far.
Line 2		
—	Err Type	Undefined field.
—	Diagnostic	An ASCII code that indicates the following:
		NI_RBD = NI diagnostic is still running.
		XMI_RBD = XMI diagnostic is still running.
		XNA_RBD = XNA diagnostic is still running.
UNIT	Unit Number	The unit under test. Always 0.
TEST	Test Number	The currently executing test.

7.4.3 Summary Reports

If not disabled (with the /IS switch), a summary report (Figure 7–3) is printed after the diagnostic has completed execution of all requested passes. In addition, the user may invoke a summary report by entering the SUMMARY command.

A summary report consists of two lines. Table 7–8 shows the summary report fields, and Table 7–9 defines the fields.

Figure 7–3 Sample Summary Report

```
;         P         5      0C03        10
; 00000000 00000000 00000000 00000012 00000064 00000000 00000C80
```

Table 7–8 Summary Report Fields

Line	Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
1	Status	Node No.	Device Type	Passes		
UNUSED	HARD ERR	SOFT ERR	SPEC 1	SPEC 2	SPEC 3	SPEC 4

Table 7–9 Summary Report Field Definitions

Mnemonic	Name	Description
LINE 1		
—	Status	The report type. P = passed, F = failed.
NODE NO.	Node Number	The node under test. The value may range from 0–F (hex).
—	Device Type	The type of module under test. 0C03 (hex) = DEMNA.
PASS	Pass Count	The number of passes executed.
LINE 2		
UNUSED	Unused	This field is unused.
HARD ERR	Hard Errors	The total number of hard errors detected.
SOFT ERR	Soft Errors	The total number of soft errors detected.
SPEC 1	Specific to diagnostic	NI and XNA RBDs—the total number of retries necessary (hex) XMI RBD—the total number of peek operations performed (high-order half of the hexadecimal count)
SPEC 2	Specific to diagnostic	NI and XNA RBDs—the total number of packets transmitted and received (hex) XMI RBD—the total number of peek operations performed (low-order half of the hexadecimal count)
SPEC 3	Specific to diagnostic	NI RBD—the total number of bytes transferred (high-order half of the hexadecimal count) XMI and XNA RBDs—the total number of datamove operations performed (high-order half of the hexadecimal count)
SPEC 4	Specific to diagnostic	NI RBD—the total number of bytes transferred (low-order half of the hexadecimal count) XMI and XNA RBDs—the total number of datamove operations performed (low-order half of the hexadecimal count)

7.5 Diagnostic Error Log Reader

The diagnostic error log reader is a program in EPROM that prints the contents of the diagnostic error log (a portion of the error history in EEPROM). When invoked, the error log reader prints a screen that indicates the DEMNA firmware revision, the DEMNA module serial number, and the number of errors logged in the diagnostic error log. The user then presses [RETURN] to view the error log entries. Each screen contains a separate entry.

The error log reader is invoked by depositing the starting address of the program, 2004C010 (hex), into Port Data Register 1 (XPD1) and then issuing the XFC command to begin execution of the program. Type [CTRL/C] to exit the error log reader and return to the RBD monitor prompt.

Example 7–3 shows a sample session with the error log reader.

Example 7–3 Using the Error Log Reader

```
RBD3> D 20150100 2004C010
RBD3> XFC


*************** DEMNA EEPROM ERROR FRAME READER V1.00 **************

EEPROM revision and date:  0600   ( 14-FEB-1990)

Module serial number:      *SG915Y8879*

Logging is currently enabled for: Selftest  NIRBD XMIRBD XNARBD

There are 2 error frames stored.







Type <CR> to continue, <CTRL/C> to abort...

---------------------- Error frame number 1 ----------------------

Sequence number:              1

Diagnostic number:            1
Diagnostic revision:       3.00
Operating mode:             RBD

XMI node number:              3
Test number:                  3
Error code:
Error number:                65
Expected data:         00000003(X)
Received data:         00000007(X)
SCB offset:            00000000(X)
Failing address:       201004A3(X)
PC at failure:         0001A762(X)
```

Example 7–3 Cont'd. on next page

Example 7-3 (Cont.) Using the Error Log Reader

```
Number of times logged:        1


Type <CR> to continue, <CTRL/C> to abort...


---------------------- Error frame number 2 ----------------------

Sequence number:              2

Diagnostic number:            0
Diagnostic revision:       3.00
Operating mode:        Power-up

XMI node number:              3
Test number:                 18
Error code:                   3
Error number:                 5
Expected data:      00000000(X)
Received data:      00800000(X)
SCB offset:         00000000(X)
Failing address:    20150004(X)
PC at failure:      20051D97(X)

Number of times logged:       2


Type <CR> to continue, <CTRL/C> to abort...

*** No more errors logged ***
```

7.6 Troubleshooting with ROM-Based Diagnostics

In general, the RBDs are used when the DEMNA self-test is unable to isolate a suspected DEMNA problem. The following is a basic procedure for using the RBDs and the software diagnostics to troubleshoot a suspected DEMNA problem. Figure 7–4 provides a flowchart of this procedure.

1. Run the DEMNA self-test. If self-test fails, reseat the module and its cables and rerun self-test. If self-test fails again, replace the DEMNA module.
2. If the DEMNA self-test passes and you still suspect a DEMNA problem, run the external loopback on live Ethernet test (test 1) of the NI RBD.
3. If test 1 of the NI RBD fails, use the external loopback on closed Ethernet test (test 3) of the NI RBD to further isolate the problem as follows:
   a. Disconnect the external Ethernet transceiver cable (BNE3) at the transceiver end.
   b. Install a loopback connector on the cable.

c. Run test 3 and observe one of the following:

   - If the test passes, the transceiver is probably bad. Replace the transceiver, reconnect the cable to the new transceiver, and rerun the test to verify proper operation. No further action is required.
   - If the test fails, one of the following is probably bad: transceiver cable, internal Ethernet cable, backplane, or DEMNA module. Go to the next step.

d. Disconnect the external transceiver cable at the system bulkhead and install a loopback connector in its place.

e. Rerun test 3 and observe one of the following:

   - If the test passes, the transceiver cable is bad. Replace the cable and rerun the test to verify proper operation. No further action is required.
   - If the test fails, one of the following is bad: internal Ethernet cable, backplane, or DEMNA module. Replace the internal Ethernet cable and install the loopback connector on the new cable. Go to the next step.

f. Rerun test 3 and observe one of the following:

   - If the test passes, the removed cable is bad. Rerun the test to verify proper operation. No further action is required.
   - If the test fails, either the DEMNA module or the backplane is bad. If the module passes self-test, it is probably good, but replace it and go to the next step.

g. Rerun test 3 and observe one of the following:

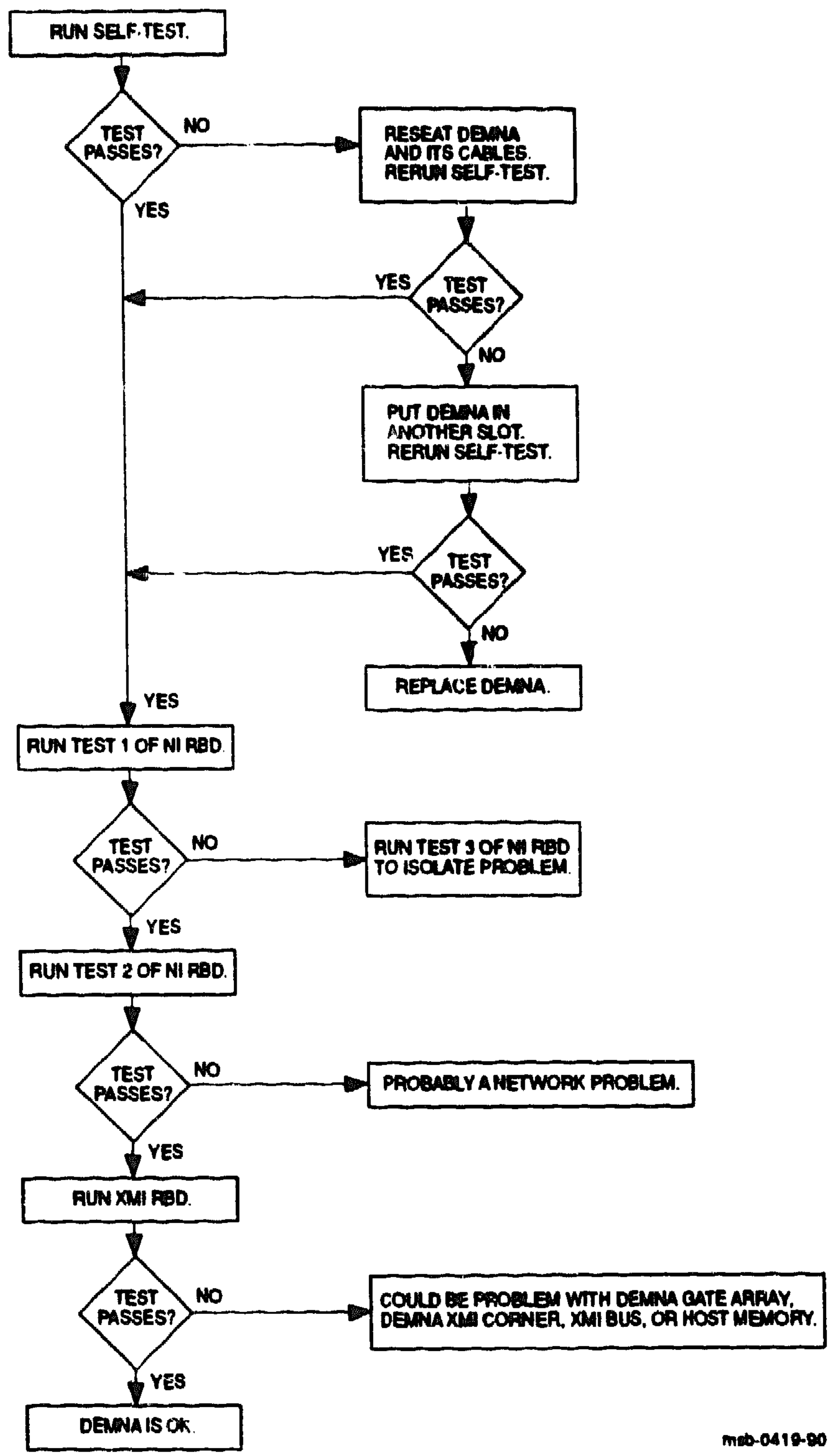
   - If the test passes, the removed DEMNA module is bad. Rerun the test to verify proper operation. No further action is required.
   - If the test fails, the backplane is bad. Install the DEMNA module in a different slot. Rerun the test to verify proper operation. Consider replacing the card cage.

4 If test 1 of the NI RBD passes and you still suspect a DEMNA problem, run the MOP loopback test (test 2) of the NI RBD to test the DEMNA's ability to communicate with another node on the network. If this test fails, the problem is probably a network problem rather than a DEMNA problem.

5 If test 2 of the NI RBD passes, run the XMI RBD to the test the DEMNA's ability to transfer data to and from host memory.

6 If the XMI RBD passes, the DEMNA is probably OK. If the XMI RBD fails, the problem is probably related to the DEMNA gate array, DEMNA XMI Corner, the XMI bus, or host memory.

Figure 7–4 Troubleshooting Flowchart for ROM-Based Diagnostics



7.7 Diagnostic Patch Mechanism

The DEMNA EEPROM contains a 2-Kbyte diagnostic patch space that holds any patches made to the diagnostic code in EPROM. When the diagnostic code is executing from EPROM, it checks at various strategic points whether there is patch code in EEPROM. If there is patch code, it is executed instead of the corresponding segment of diagnostic code in EPROM.

When the DEMNA self-test or any DEMNA RBD is run, a checksum is calculated for the diagnostic patch area in EEPROM and compared with a valid checksum. If the checksum test fails, the diagnostic patch area is declared invalid. In this case, no diagnostic patches are executed. If this checksum failure occurs during self-test, the self-test fails, and the Bad Diagnostic Patch Table bit in the Power-Up Diagnostic (XPUD) Register is set. If the checksum failure occurs when an RBD is invoked, the message

```
BAD_PATCH
```

is printed on the system console before the RBD is executed.

7.8 Software Diagnostics

In addition to ROM-based diagnostics, the DEMNA can be tested with software diagnostics that are loaded into and run from host memory. Table 7–10 lists the software diagnostics available for the DEMNA.

Table 7–10 Software Diagnostic Programs for the DEMNA

Program	Diagnostic Level	Description
EVGDB	2	A utility program that updates the firmware in the DEMNA EEPROM and/or modifies certain flags and parameters in the EEPROM.
EVDWC	2R	An exerciser that tests the installation of the host Ethernet node and all other nodes on the local Ethernet that support MOP protocol.
EVDYE	2R	A functional diagnostic that tests the functioning of the DEMNA Ethernet/802 port.

7.8.1 EVGDB Program (DEMNA EEPROM Update Utility)

EVGDB is a utility program that enables a user to update the firmware in the DEMNA EEPROM. Using EVGDB, a user can easily load a new firmware image in the DEMNA EEPROM. The firmware image is contained in a file called EVGDBQ.BIN that is distributed with EVGDB. EVGDB can also be used to modify the setting of flags and other parameters that control various aspects of DEMNA operation. Chapter 9 describes the modifiable flags and parameters in the DEMNA EEPROM and provides a detailed example of how to use EVGDB to modify these flags and parameters.

In addition to providing the above functions, EVGDB executes the DEMNA self-test on being invoked in stand-alone mode. This provides confirmation that the DEMNA is operating properly.

7.8.2 EVDWC Program (NI Exerciser)

The EVDWC program tests the installation of the host Ethernet node and checks the connectivity of all other nodes on the local Ethernet.

EVDWC has the following functions:

- Monitors MOP System ID messages on the Ethernet
- Builds a node table for the local Ethernet
- Conducts loopback testing, monitors network traffic, and permits users to view contents of network packets
- Provides online help for users

7.8.3 EVDYE Program (DEMNA NI Functional Diagnostic)

The EVDYE program makes sure that the DEMNA NI port performs all the functions that the VAX/VMS Ethernet port driver, EXDRIVER, may request.

EVDYD has 11 tests:

1 Start and Stop Users test
2 Transmit and Receive Ethernet Packets test
3 Transmit and Receive 802 Packets test
4 Transmit and Receive 802 Extended Packets test
5 Multicast Address test
6 Group SAP test
7 Promiscuous Mode test
8 CRC test
9 Receive Data test
10 Stress test
11 Get Status test

Tests 2–10 use either internal or external loopback facilities:

- Internal loopback—If you do nothing, the above tests will use the LANCE chip's internal loopback.
- External loopback—You must
    1 Install a loopback connector on the system bulkhead or the Ethernet transceiver cable.

2 Use the procedure for running EVDYE described below, with the following insertion after the line *SELECT EXm0*:

```
DS> SET EVENT FLAG 1 [RETURN]
```

7.8.4 Running EVGDB

Chapter 9 provides a detailed example of running EVGDB.

7.8.5 Running EVDWC and EVDYE

The level 2R diagnostics, EVDYD and EVDWC, are run as follows:

1 Log into the customer service account or enter the following command at the VAX/VMS system prompt:

```
$SET DEFAULT SYS$MAINTENANCE
```

2 Run the VAX Diagnostic Supervisor (VAX/VDS) with the following command:

```
$RUN filename.EXE
```

where *filename* is the executable VAX/DS file as follows:

VAX System	VAX/DS File
6000 Model 2xx/3xx	ELSAA
6000 Model 4xx	ERSAA
9000	EWSAA

3 On a VAX 6000 system, enter the following commands at the VAX/DS prompt:

```
DS>LOAD diagnostic_name
DS>ATTACH DEMNA HUB EXm0 n [RETURN]
DS>SELECT EXm0
DS>START
```

where:

diagnostic_name is the name of the diagnostic to be executed (either EVDWC or EVDYE)

m is the unit number of the DEMNA. The DEMNA with the lowest XMI node number is unit A, the DEMNA with the second lowest XMI node number is unit B, and so on.

n is the XMI node ID of the DEMNA

4 On a VAX 9000 system, enter the following commands at the VAX/DS prompt:

```
DS>LOAD diagnostic_name
DS>ATTACH XJA HUB XJAx x [RETURN]
DS>ATTACH DEMNA XJAx0 EXm0 n [RETURN]
DS>SELECT EXm0
DS>START
```

where:

diagnostic_name is the name of the diagnostic to be executed (either EVDWC or EVDYE)

m is the unit number of the DEMNA. The DEMNA with the lowest XMI node number is unit A, the DEMNA with the second lowest XMI node number is unit B, and so on.

n is the XMI node ID of the DEMNA

x is the XJA unit number (0–3)

CHAPTER 8

8 Port Error Handling

This chapter describes the types of errors recorded and reported by the port, the error blocks maintained by the port, the port's response to errors that are visible to the port driver, error logging in EEPROM, and how to restart the port from a fatal error.

The chapter includes the following sections:

- Error Types
- Error Blocks
- Error Logging in EEPROM
- Error Response
- Restarting the Port from a Fatal Error

8.1 Error Types

From the port's perspective, there are three different types of errors:

- Fatal errors: errors that cause a port shutdown
- Nonfatal errors: errors that do not cause a port shutdown
- Ethernet errors: nonfatal errors arising from Ethernet activity

8.1.1 Fatal Errors

Fatal errors include the following:

- Machine checks on the DEMNA CVAX
- Exceptions on the DEMNA CVAX
- A node halt/restart when the port is in the initialized state
- Port initialization failures
- Port driver protocol errors
- Port command failures
- Specification of more than one buffer for a port command
- Specification of an incorrect number of transmit buffers
- Other firmware-related errors, such as keep-alive timeouts
- Firmware updates
- A failed access to the command ring or receive ring that could not be recovered

If the DEMNA CVAX experiences a machine check or exception, the port executes its shutdown sequence (Section 5.4). At the end of this sequence, the port writes a fatal error block (Section 8.2.1) to the Port Error Log Area of the Port Data Block (PDB) and transitions to the uninitialized state. If the port was in the initialized state, it interrupts the port driver if port interrupts are enabled.

If the port receives a node halt/restart when in the initialized state, it executes its node halt/restart sequence (Section 5.2). At the end of this sequence, the port writes a fatal error block to the PDB. The port does not interrupt the port driver.

If port initialization fails, the port writes error status to the Port Status (XPST) Register and remains in the uninitialized state. Port initialization can fail for the following reasons:

- The DEMNA failed self-test and is not operational
- The DEMNA determines that the base address of the Port Data Block (PDB) is invalid
- The DEMNA determines that the contents of the PDB are invalid

The port driver can cause a fatal port error by attempting to initialize the port when the port is already in the initialized state. If this occurs, the port executes its shutdown sequence, writes a fatal error block to the PDB, and, if port interrupts are enabled, interrupts the port driver.

The firmware periodically makes various internal checks to ensure that it is running. One of these checks involves an internal keep-alive counter that monitors whether the firmware scheduler is running. If a keep-alive timeout occurs, indicating that the firmware scheduler is not executing properly, the port executes its shutdown sequence, writes a fatal error block to the PDB, and, if port interrupts are enabled, interrupts the port driver.

The DEMNA treats an update of the firmware image in EEPROM as a fatal error. After the firmware has been updated, the port executes its shutdown sequence, writes a fatal error block to the PDB, and, if port interrupts are enabled, interrupts the port driver.

8.1.2 Nonfatal Errors

Nonfatal errors include the following:

- Datamove and peek errors that succeeded on the first retry or that did not directly access the command ring or receive ring
- Buffer transfer failures
- Address translation errors
- Port command errors (for example, a command length error)

All of above errors are retried once. If the retry fails, a fatal error occurs.

The port uses the datamover in the DEMNA gate array to transfer data to and from the ring buffers. Datamove errors are recorded in the gate array's datamove registers and are not visible as such to the port driver. If a datamove error occurs, the port driver will detect the error through the error status written by the port to the appropriate ring entry.

Buffer transfer failures, port address translation errors, and port command failures are detected by the port and reported to the port driver through the error codes in the ring entries.

8.1.3 Ethernet Errors

Ethernet errors occur as part of normal network operation and do not cause a port shutdown. Ethernet errors include transmit and receive errors caused by activity on the Ethernet wire, as well as receive errors caused by an insufficient allocation of system and user buffers by the host. (See Section 10.7.1.1 for a description of the Ethernet error counters maintained by the port.)

The port records transmit and receive errors for all users in its internal data link counters. The user can read these counters by issuing the SHOW KNOWN LINE COUNTERS command from the Network Control Program (NCP). In addition, the user can monitor these counters with the Status and Status/Error screens available through the DEMNA console monitor program. Higher-level software, such as NCP, can read these counters by requesting the DEMNA port driver to issue the port a RCCNTR/RDCNTR or READ$STATUS command.

If the host allocates too few system (receive) buffers, the port will occasionally have to discard a receive packet because of a lack of a receive buffer. The port records such errors in its Receive Failures—SBUA Counter, which is displayed in the Status/Error screen available through the console monitor program. Higher-level software can read the counter by requesting the port driver to issue the port a RCCNTR/RDCNTR, READ$STATUS, or READ$SNAPSHOT command. In addition, a copy of the Receive Failures—SBUA Counter is located in the Port Data Block (PDB). The PDB copy of the Receive Failures—SBUA Counter is updated as often as once per second.

If the user allocates too few user buffers or does not process receive packets as fast as the port is delivering them, the port driver will have to discard a receive packet delivered to it by the port because there is no user buffer in which to deposit the packet. The port driver records such errors in the Receive Failures—User Buffer Unavailable (UBUA) Counter located in host memory. The user can monitor this counter with the Status/Error screen available through the DEMNA console monitor program. When issued a RCCNTR/RDCNTR, READ$STATUS, or READ$SNAPSHOT command by the port driver or when issued a Read Counters command over the network, the port reads this counter and includes it in the counters returned as a response to any of these commands.

8.2 Error Blocks

The port maintains two types of internal error blocks: fatal error blocks and nonfatal error blocks. The user can read these blocks with the SHOW ERROR H*n* and SHOW ERROR S*n* commands, respectively, from the console monitor program. Higher-level software can read the error blocks by requesting the port driver to issue the port a READ$ERROR command. In addition, the port writes a fatal error block to the Port Error Log Area of the Port Data Block (PDB) when a fatal port error occurs or when the port is issued a node halt/restart (Section 5.2) when in the initialized state.

The port and port driver maintain additional error status in various registers, counters, and the PDB. Section 2.12 and Section 2.13 summarize this error information.

8.2.1 Fatal Error Blocks

The port maintains two versions of the fatal error block: one for firmware updates and one for all other fatal errors. Figure 8–1 and Figure 8–2 show the layout of the fatal error blocks. Table 8–1 and Table 8–2 describe the fields in fatal error blocks.

Figure 8–1 Firmware Update Block

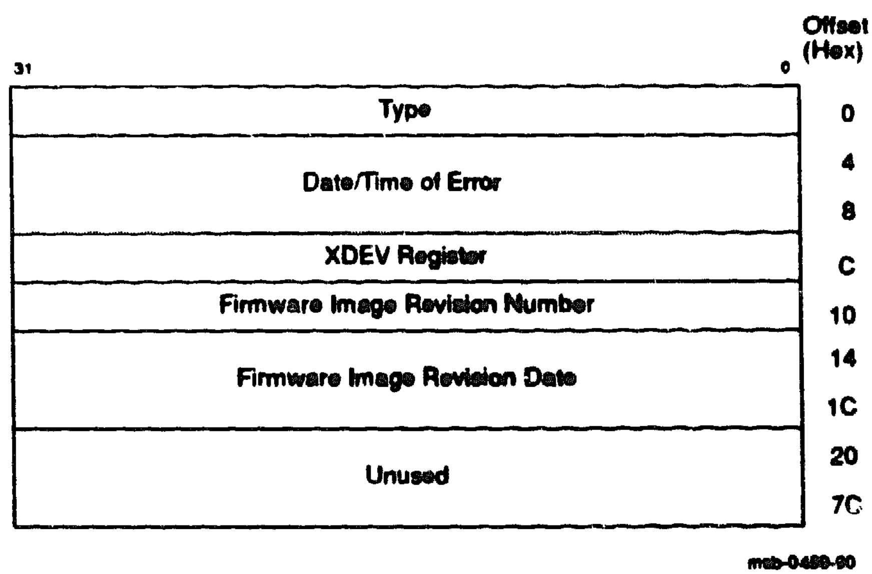


Table 8–1 Fields in Firmware Update Block

Field	Description
Type	Identifies the type of fatal error. A value of 4 indicates a firmware update.
Date/Time of Error	The date and time at which the error occurred. This value is expressed in binary absolute format. If the system date/time was specified in the PARAM command, the date/time is the base time supplied by the host plus DEMNA uptime until the error occurred. If the system date/time was not specified in the PARAM command, the base date/time defaults to 01-JAN-88. In this case, the date/time of error is the base time (01-JAN-88) plus the DEMNA uptime until the error occurred. If this field is all zeros, then no error occurred.
XDEV Register	Device Register
Firmware Image Revision Number	The revision number of the EEPROM firmware image in ASCII. This 4-character revision number is taken from the low-order byte of the XDEV Register.
Firmware Image Revision Date	Twelve ASCII characters that indicate the revision date of the EEPROM firmware image. *The month, day, and year of the revision date are indicated—for example,* 12-JUL-1989.
Unused	This field is unused.

Figure 8–2 Fatal Error Block for Other Fatal Errors

Bits 31–0	Offset (Hex)
Type	0
Date/Time of Error	4 8
R0...R12	C 3C
XBER Register	40
XFADR Register	44
XFAER Register	48
GACSR Register	4C
Diagnostic Register	50
XPST Register (before error)	54
XPD1 Register (before error)	58
XPD2 Register (before error)	5C
XPST Register (after error)	60
XPD1 Register (after error)	64
Stack Contents	68 7C

msb-0487-90

Table 8–2 Fields in Fatal Error Block for Other Fatal Errors

Field	Description	
Type	Identifies the type of fatal error. This field is encoded as follows:	
	0	No error
	1	DEMNA machine check or exception
	2	Node halt/restart when the port was in the initialized state (When the port receives a node halt/restart in the initialized state, it assumes that it is being restarted from an error.)
	3	Fatal error other than machine check or exception (for example, a port initialization failure or a keep-alive timeout)
	6	Driver error. The port driver attempts to initialize the port when the port is already in the initialized state.
Date/Time of Error	The date and time at which the error occurred. This value is expressed in binary absolute format. If the system date/time was specified in the PARAM command, the date/time is the base time supplied by the host plus the DEMNA uptime until the error occurred. If the system date/time was not specified in the PARAM command, the base date/time defaults to 01-JAN-88. In this case, the date/time of error is the base time (01-JAN-88) plus the DEMNA uptime until the error occurred. If this field is all zeros, then no error occurred.	
R0...R12	CVAX general purpose registers	
XBER Register	Bus Error Register	
XFADR	Failing Address Register	
XFAER	Failing Address Extension Register	
GACSR Register	Gate Array Control/Status Register	
Diagnostic Register	Diagnostic Register (a DEMNA-internal Register)	
XPST Register (before error)	Port Status Register before the error was reported	
XPD1 Register (before error)	Port Data 1 Register before the error was reported	
XPD2 Register (before error)	Port Data 2 Register before the error was reported	
XPST Register (after error)	Port Status Register after the error was reported	
XPD1 Register (after error)	Port Data 1 Register after the error was reported	
Stack Contents	The stack contents of the CVAX at the time of the error	

8.2.2 Nonfatal Error Blocks

The port maintains three versions of the nonfatal error block:

- One for datamove and peek errors
- One for host interrupt errors
- One for XMI errors reported in the DEMNA XBER Register

Figure 8–3 through Figure 8–5 show the layouts of the nonfatal error blocks. Figure 8–5 describes the fields in the nonfatal error blocks.

Figure 8–3 Nonfatal Error Block for Datamove and Peek Errors

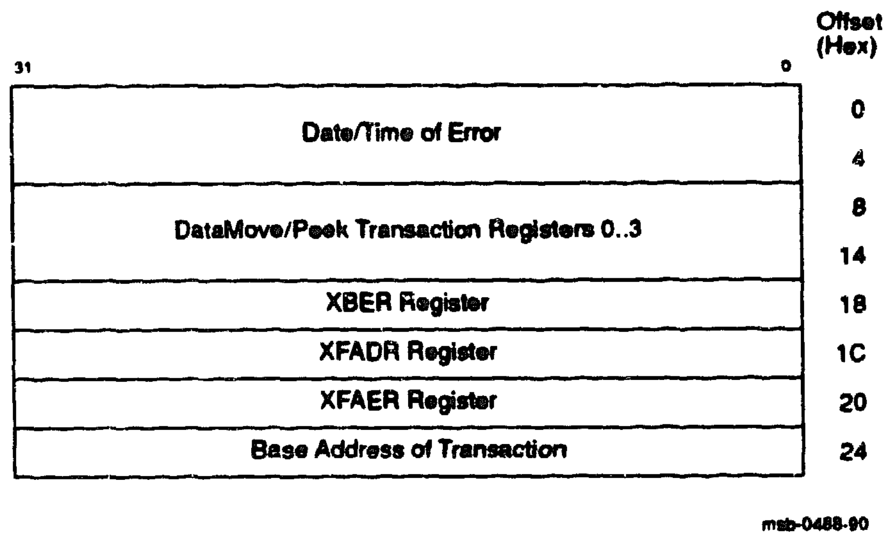


Table 8–3 Nonfatal Error Block Fields for Datamove and Peek Errors

Field	Description
Date/Time of Error	The date and time at which the error occurred. This value is expressed in binary absolute format. If the system date/time was specified in the PARAM command, the date/time is the base time supplied by the host plus DEMNA uptime until the error occurred. If the system date/time was not specified in the PARAM command, the base date/time defaults to 01-JAN-88. In this case, the date/time of error is the base time (01-JAN-88) plus the DEMNA uptime until the error occurred.

Table 8–3 (Cont.) Nonfatal Error Block Fields for Datamove and Peek Errors

Field	Description
Datamove/Peek Registers	If the error occurred during a datamove transaction, these registers are the four datamove registers (DMPOR*n*, DMCSR*n*, DMXMI*n*, and DMNPA*n*). If the error occurred during a peek transaction, these registers are the four peek registers (PKXMIL*n*, PCKMIH*n*, PKDATA*n*, and PKDATB*n*).
XBER Register	Bus Error Register
XFADR Register	Failing Address Register
XFAER Register	Failing Address Extension Register
Base Address of Transaction	The base address of the failing datamove or peek transaction.

Figure 8–4 Nonfatal Error Block for Interrupt Errors

31 … 0	Offset (Hex)
Date/Time of Error	0
	4
GACSR Register	8
GAHIR Register	C
GAIVR Register	10
GATMR Register	14
XBER Register	18
XFADR Register	1C
XFAER Register	20
GACSR Address	24

mab-0480-90

Table 8–4 Nonfatal Error Block Fields for Interrupt Errors

Field	Description
Date/Time of Error	The date and time at which the error occurred. This value is expressed in binary absolute format. If the system date/time was specified in the PARAM command, the date/time is the base time supplied by the host plus DEMNA uptime until the error occurred. If the system date/time was not specified in the PARAM command, the base date/time defaults to 01-JAN-88. In this case, the date/time of error is the base time (01-JAN-88) plus the DEMNA uptime until the error occurred.
GACSR Register	Gate Array Control and Status Register
GAHIR	Gate Array Host Interrupt Register
GAIVR	Gate Array IDENT Vector Register
GATMR	Gate Array Timer Register
XBER Register	Bus Error Register
XFADR Register	Failing Address Register
XFAER Register	Failing Address Extension Register
GACSR Address	The address of the Gate Array Control/Status Register (GACSR).

Figure 8–5 Nonfatal Error Block for XBER-Reported Errors

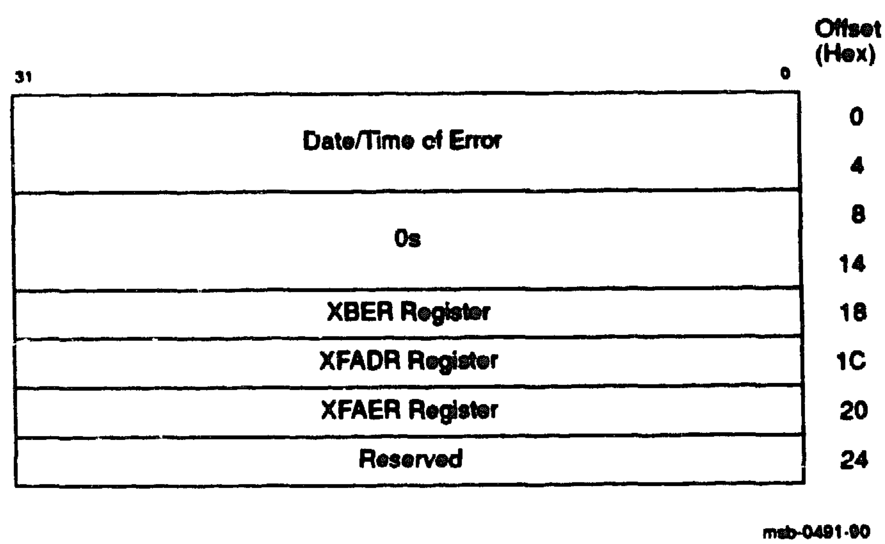


Table 8–5 Nonfatal Error Block Fields for XBER-Reported Errors

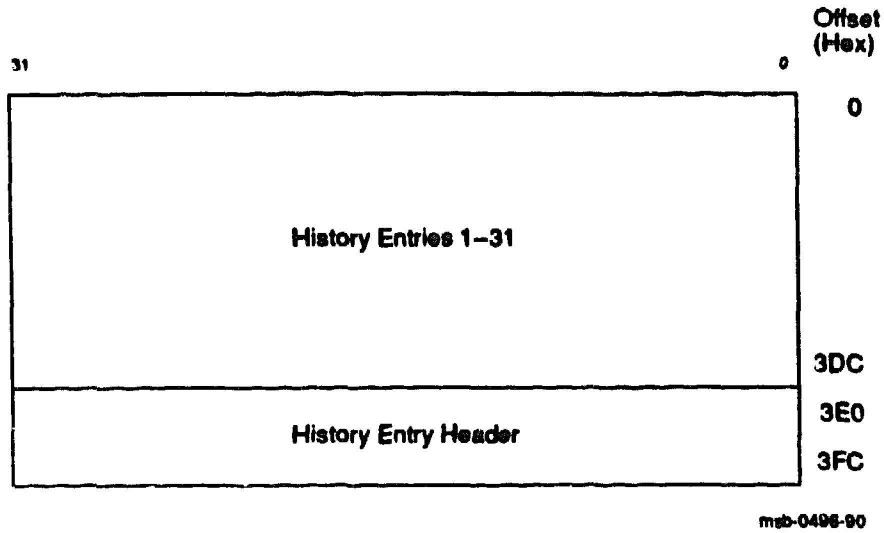
Field	Description
Date/Time of Error	The date and time at which the error occurred. This value is expressed in binary absolute format. If the system date/time was specified in the PARAM command, the date/time is the base time supplied by the host plus the DEMNA uptime until the error occurred. If the system date/time was not specified in the PARAM command, the base date/time defaults to 01-JAN-88. In this case, the date/time of error is the base time (01-JAN-88) plus the DEMNA uptime until the error occurred.
0s	Zeros; undefined
XBER Register	Bus Error Register
XFADR Register	Failing Address Register
XFAER Register	Failing Address Extension Register
Reserved	This field is reserved.

8.3 Error Logging in EEPROM

The DEMNA logs both fatal and nonfatal errors in a 1-Kbyte History Data area of EEPROM (Figure 8–6). This area contains 31 error history entries, each of which is 32 bytes long, and a history entry header, which is also 32 bytes long. Each history entry provides information specific to a particular error. The history entry header provides module-specific information, including the number of history entries written to EEPROM since the last DEMNA power-up or reset, the console password, the module serial number, and the module runtime.

The user can read the history entries with the SHOW HISTORY command from the console monitor program or by writing and reading the DEMNA Communications (XCOMM) Register. Higher-level software can read the history entries by requesting the DEMNA port driver to issue the port a READ$HISTORY command.

Figure 8–6 History Data Area in EEPROM



The DEMNA records four types of errors in the EEPROM history area:

- Fatal errors
- Nonfatal errors
- Diagnostic errors (self-test and ROM-based diagnostic errors)
- Firmware updates

Table 8–6 indicates which history entries are allocated to which error types and the conditions under which each error type is logged in EEPROM.

Table 8–6 EEPROM History Error Types

Error Type	History Entries Allocated to This Error Type	Logging Information
Fatal error	Entries 1–8	Logged immediately after a fatal error occurs. Entries are written over if more fatal errors occur than can be recorded. Logging stops after 32 fatal errors have been recorded.
Nonfatal errors	Entries 9–16	For a datamove error, peek error, or host interrupt error: logged immediately after the error occurs. For XBER-reported errors: logged after the firmware polls the XBER Register and discovers the error. Logging stops after 16 nonfatal errors have been recorded.

Table 8–6 (Cont.) EEPROM History Error Types

Error Type	History Entries Allocated to This Error Type	Logging Information
Diagnostic errors	Entries 17–24	Logged immediately after each self-test error, NI diagnostic error, XMI diagnostic error, or XNA diagnostic error, provided that error logging is enabled by the corresponding EEPROM flag. Entries are not overwritten if more diagnostic errors occur than can be recorded. Logging stops after eight diagnostic errors have been recorded.
Firmware updates	Entries 25–31	Logged after each firmware update. Entries are written over if more firmware updates occur than can be recorded. There is no limit on the number of firmware updates that are logged.

8.3.1 History Entry

The history entry has two basic formats: one for diagnostic errors and one for all other errors. Figure 8–7 and Figure 8–10 show the history entry layouts. Table 8–7 and Table 8–10 describe the history entry fields.

Figure 8–7 History Entry Format for Diagnostic Errors

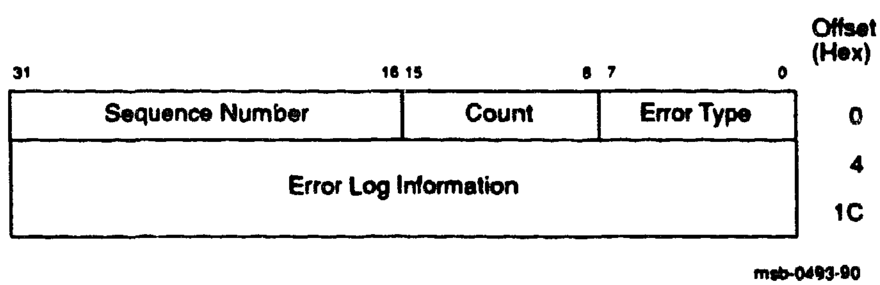


Table 8–7 Fields in History Entry for Diagnostic Errors

Field	Description	
Error type	The error type for diagnostic errors is 5.	
Count	The number of times that this type of diagnostic error has been recorded.	
Sequence #	The history entry number, which is an integer from one through 255. This number indicates the order in which the entry was logged with respect to other history entries. Lower-numbered entries were logged before higher-numbered entries.	
Error Log Information	Eight longwords that supply the following data:	
	Longword 1	See Figure 8–8 and Table 8–8
	Longword 2	See Figure 8–9 and Table 8–9

Table 8–7 (Cont.) Fields in History Entry for Diagnostic Errors

Field	Description	
	Longword 3	Expected data
	Longword 4	Received data
	Longword 5	System control block (SCB) offset
	Longword 6	Memory address
	Longword 7	Program counter (PC) at failure
	Longword 8	Reserved

Figure 8–8 Longword 1 of History Entry for Diagnostic Errors

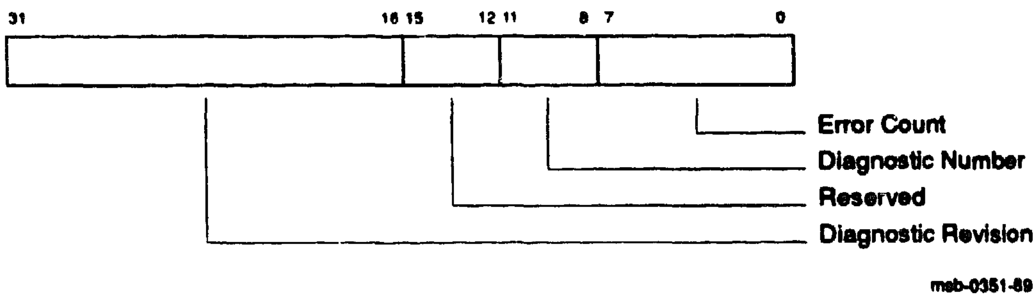


Table 8–8 Longword 1 of History Entry for Diagnostic Errors—Field Descriptions

Bits	Field	Description
31:16	Diagnostic Revision	Two ASCII numbers that indicate the revision number of the diagnostic. For example, 39 33 (ASCII) = revision 3.9.
15:12	Reserved	These bits are reserved.
11:8	Diagnostic Number	A binary field that indicates which test reported the error. 0 = self-test. 1 = NI RBD. 2 = XMI RBD. 3 = XNA RBD.
7:0	Error Count	The number of times (in binary) that this type of diagnostic error has occurred.

Figure 8–9 Longword 2 of History Entry for Diagnostic Errors

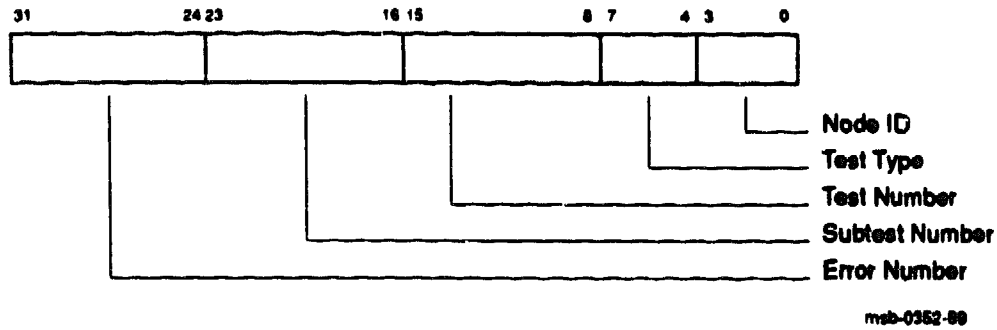


Table 8–9 Longword 2 of History Entry for Diagnostic Errors—Field Descriptions

Bits	Field	Description
31:24	Error Number	See Appendix F for a listing of error numbers.
23:16	Subtest Number	Number (hex) of the failing subtest. (See Appendix F.)
15:8	Test Number	Number of the failing test. (See Appendix F.)
7:4	Test Type	1 = power-up mode; 2 = RBD mode
3:0	Node ID	XMI node ID (hex) of the DEMNA

Figure 8–10 History Entry Format for All Other Errors

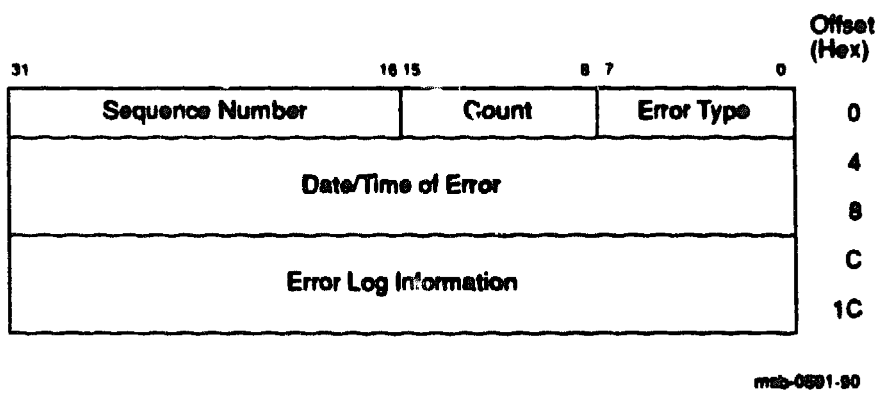


Table 8–10 Fields in History Entry for All Other Errors

Field	Description	
Error type	The error type, encoded as follows:	
	0	No error
	1	Machine check
	2	Node halt
	3	Fatal error
	4	Firmware update
	6	XBER-detected error
	7	Peek, datamove, or interrupt failed
Count	The number of times that this specific error has been recorded.	
Sequence #	The history entry number, which is a number from one through 31. This number indicates the order in which the entry was logged with respect to other history entries. Lower-numbered entries were logged before higher-numbered entries.	
Date/Time of Error	The date and time at which the error occurred. This value is expressed in binary absolute format. If the system date/time was specified in the PARAM command, the date/time is the base time supplied by the host plus the DEMNA uptime until the error occurred. If the system date/time was not specified in the PARAM command, the base date/time defaults to 01-JAN-88. In this case, the date/time of error is the base time (01-JAN-88) plus the DEMNA uptime until the error occurred.	
Error Log Information	Five longwords that supply error information specific to the error type. The error information supplied for each error type is as follows:	
		Datamove Error
	Longword 1	XBER Register
	Longword 2	XFADR Register
	Longword 3	XFAER Register
	Longword 4	DMPORn Register
	Longword 5	DMCSRn Register
		Exception
	Longword 1	Pending Port Status Register (XPST_Pending). The value that will be loaded into the XPST Register after the next state change (after error handling has been completed).
	Longword 2	Pending Port Data 1 Register (XPD1_Pending). The value that will be loaded into the XPD1 Register after the next state change (after error handling has been completed).
	Longword 3	Address of call to shutdown request.
	Longword 4	Address of exception.
	Longword 5	Exception number (offset into system control block (SCB)).

Table 8–10 (Cont.) Fields in History Entry for All Other Errors

Field	Description	
Error Information (cont.)		
		Fatal Error
	Longword 1	Pending Port Status Register (XPST_Pending). The value that will be loaded into the XPST Register after the next state change (after error handling has been completed).
	Longword 2	Pending Port Data 1 Register (XPD1_Pending). The value that will be loaded into the XPD1 Register after the next state change (after error handling has been completed).
	Longword 3	Longword 1 of the stack when the error occurred.
	Longword 4	Longword 2 of the stack when the error occurred.
	Longword 5	Longword 3 of the stack when the error occurred.
		Firmware Update
	Longword 1	XDEV Register
	Longword 2	Four ASCII numbers that indicate the DEMNA firmware revision. For example, 30313233 (ASCII) = revision 01.23.
	Longword 3–5	Firmware revision date and time (binary).
		Machine Check
	Longword 1	Pending Port Status Register (XPST_Pending). The value that will be loaded into the XPST Register after the next state change (after error handling has been completed).
	Longword 2	Pending Port Data 1 Register (XPD1_Pending). The value that will be loaded into the XPD1 Register after the next state change (after error handling has been completed).
	Longword 3	Machine check code (usually 80–83, which indicate an invalid address).
	Longword 4	Most recent memory address.
	Longword 5	Internal state information 1.
		Node Halt
	Longword 1	Pending Port Status Register (XPST_Pending). The value that will be loaded into the XPST Register after the next state change (after error handling has been completed).
	Longword 2	Pending Port Data 1 Register (XPD1_Pending). The value that will be loaded into the XPD1 Register after the next state change (after error handling has been completed).
	Longword 3	Longword 1 of the stack when the node halt occurred.
	Longword 4	Longword 2 of the stack when the node halt occurred.
	Longword 5	Longword 3 of the stack when the node halt occurred.

Table 8–10 (Cont.) Fields in History Entry for All Other Errors

Field	Description	
		No Error
	Longword 1–5	Zeros; undefined
		Peek Error
	Longword 1	XBER Register
	Longword 2	XFADR Register
	Longword 3	XFAER Register
	Longword 4	XMIL Register
	Longword 5	XMIH Register
		XBER Error
	Longword 1	XBER Register
	Longword 2	XFADR Register
	Longword 3	XFAER Register
	Longword 4–5	Zeros; undefined

8.3.2 History Entry Header

The history entry header contains error information and various operation parameters. The operation parameters can be modified with the DEMNA EEPROM Update Program, EVGDB, as described in Chapter 9. Figure 8–11 shows the layout of the history entry header. Table 8–11 describes the fields in the history entry header.

Figure 8–11 History Entry Header

31 … 16	15 … 0	Offset (Hex)
Reserved	EEPROM Flags	0
Update Count		4
Console Password		8
		C
Module Serial Number		14
		18
Runtime		1C

msb-0492-90

Table 8–11 History Entry Header—Field Descriptions

Field	Description
EEPROM Flags	Flags that control various aspects of DEMNA operation. See Figure 8–12 and Table 8–12.
Reserved	This field is reserved.
Update Count	The number of times that a history entry has been written to DEMNA EEPROM since the last DEMNA power-up or reset. If no unused history entries are available, used history entries may be written over. The update count can thus be greater than the total number of history entries in EEPROM.
Console Password	An 8-character ASCII field that indicates the password that must be used to connect to the DEMNA console monitor program.
Module Serial Number	A 12-character ASCII field that identifies the module.
Runtime	The total DEMNA uptime since the DEMNA EEPROM was initialized. The runtime is expressed in units of 524,288 seconds (6.068 days).

Figure 8–12 EEPROM Flags

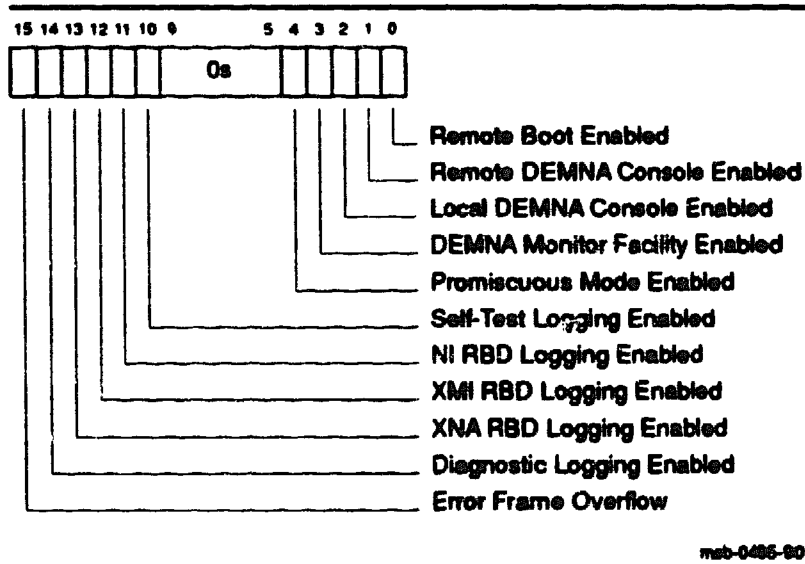


Table 8–12 EEPROM Flag Descriptions

Bits	Name	Description
15	Error Frame Overflow	When set, indicates that one more diagnostic errors could not be written to EEPROM because all the history entries allocated for diagnostic errors have been written.

Table 8–12 (Cont.) EEPROM Flag Descriptions

Bits	Name	Description
14	Diagnostic Logging Enabled	When set, enables the logging of self-test and RBD errors to EEPROM. When cleared, disables this function.
13	XNA RBD Logging Enabled	When set, enables the logging of XNA RBD errors to EEPROM. When cleared, disables this function. For XNA RBD errors to be logged, diagnostic error logging must be enabled, as well.
12	XMI RBD Logging Enabled	When set, enables the logging of XMI RBD errors to EEPROM. When cleared, disables this function. For XMI RBD errors to be logged, diagnostic error logging must be enabled, as well.
11	NI RBD Logging Enabled	When set, enables the logging of NI ROM-based diagnostic (RBD) errors to EEPROM. When cleared, disables this function. For NI RBD errors to be logged, diagnostic error logging must be enabled, as well.
10	Self-Test Logging Enabled	When set, enables the logging of self-test errors to EEPROM. When cleared, disables this function. For self-test errors to be logged, diagnostic error logging must be enabled, as well.
9:5	Reserved	Reserved. Must be zeros.
4	Promiscuous Mode Enabled	When set, the DEMNA operates by default in promiscuous mode. When cleared, the DEMNA does not operate in promiscuous mode by default. (An application can override a flag setting of No by starting up a promiscuous user.) In promiscuous mode, the DEMNA receives all packets on the network, regardless of a packet's destination. The DEMNA console monitor program uses this information to determine characteristics of the network traffic. If no users defined to the DEMNA are enabled for promiscuous mode, the DEMNA discards the packets not addressed to a DEMNA user. Otherwise, the DEMNA delivers all received packets to each DEMNA user for whom promiscuous mode is enabled. (See Section 2.9 for further information on DEMNA operation in promiscuous mode.)
3	DEMNA Monitor Facility Enabled	When set, enables operation of the DEMNA monitor facility, which monitors network operation. When cleared, disables the DEMNA monitor facility.
2	Local DEMNA Console Enabled	When set, enables the DEMNA console monitor program to be accessed from the local network node and from the DEMNA physical console. When cleared, denies access to the console monitor program from the local network node and the DEMNA physical console.
1	Remote DEMNA Console Enabled	When set, enables the DEMNA console monitor program to be accessed from a remote network node. When cleared, denies access to the console monitor program from a remote node.
0	Remote Boot Enabled	When set, enables the DEMNA to participate in remote booting over the network. When cleared, disables this function. See Appendix C for further information.

8.4 Error Response

From the perspective of the port driver, the most important aspect of the port's error response is whether the port shuts down. The port driver is alerted to a port-detected error by means of the following:

- The State Qualifier field in the XPST Register
- The error codes written by the port to the command and receive rings

- A port-generated interrupt when any of the hard error bits in the XBER Register are written (provided that port error interrupts are enabled)
- The fatal error block and counters in the Port Data Block (PDB)

In addition, the port driver may issue RCCNTR/RDCNTR, READ$STATUS, and READ$ERROR commands to read the port's data link counters, status information, and error blocks, respectively. Table 8–13 through Table 8–16 indicate the following error response to various types of DEMNA-detected errors:

- DEMNA hardware response
- DEMNA firmware response
- Port driver response and user visibility

Table 8–13 Ethernet Error Response

Error	Hardware Response	Firmware Response	Driver Response and User Visibility
Loss of Carrier	LANCE completes transmission of the packet and continues to transmit and receive packets.	Returns Transmit Failed—Loss of Carrier error to host command ring entry, increments Send Failures—Loss of Carrier counter, and continues as normal.	Status
Late Collision	LANCE does not retry this error. LANCE continues to transmit and receive packets.	Returns Transmit Failed—Late Collision error to host command ring entry, increments Send Failures—Late Collision counter, and continues as normal.	Status
Retry Error (excessive collisions)	LANCE aborts transmission of the packet and continues to transmit and receive subsequent packets.	Returns Transmit Failed—Retries Exhausted error to host command ring, increments Send Failures—Retries Exhausted counter, and continues as normal.	Status
Framing Error	LANCE continues to transmit and receive packets.	Returns Receive Failed—CRC Error to host receive ring, increments Receive Failures—Framing Error counter, discards the packet, and continues as normal.	Counters
CRC Error	LANCE continues to transmit and receive packets.	Returns Receive Failed—CRC Error to host receive ring, increments Receive Failures—CRC Error counter, discards the packet, and continues as normal.	Counters
Collision Error (heartbeat)	LANCE continues to transmit and receive packets.	Increments Send Failures—Collision Check Failure counter and continues as normal.	Counters

Key to Driver Response and User Visibility

Status—The driver returns transmit error status to the user application and continues as normal. The user application sees this error status in the I/O Status Block pointed to by the failing QIO operation (if applicable) or in device counters examined with a SHOW LINE COUNTERS command from the Network Control Program (NCP) or with a MOP (Maintenance Operations Protocol) REQUEST COUNTERS command.
Counters—The driver never notices the error. The user application sees the error in DEMNA counters by issuing a SHOW LINE COUNTERS command from NCP or by issuing a MOP REQUEST COUNTERS command.

Table 8–14 DEMNA-Internal Errors that Affect the LANCE

Error	Hardware Response	Firmware Response	Driver Response and User Visibility
Miss Error	LANCE continues to transmit and receive packets.	Increments Receive Failures—Data Overrun counter and continues as normal.	Counters
Overflow (Excess of 21-microsecond bus latency on receive DMA)	Packet is not received completely. LANCE continues to transmit and receive packets.	Increments Receive Failures—Data Overrun counter and Overflow/Underflow counter and continues as normal.	Counters
Underflow (excess of 9 microseconds on transmit DMA)	Packet is incompletely transmitted, resulting in transmission of a runt packet or packet with CRC error. The LANCE transmitter is shut off, but the LANCE receiver continues to function.	Increments Overflow/Underflow counter, restarts the LANCE, increments LANCE Restarts counter, and continues as normal.	Status
Memory Error	LANCE transmitter and receiver are shut off.	Saves internal status in a fatal error block, increments Memory Error counter, stops command and receive ring processing, and shuts down the port.	Shutdown
Parity Error (detected by external logic while LANCE is bus master)	Parity logic on the DEMNA detects a parity error and causes the LANCE to experience a memory error.	Saves internal status in a fatal error block, increments Memory Error counter, stops command and receive ring processing, and shuts down the port.	Shutdown
LANCE/Gate Array Grant Timeout (detected by the timeout logic)	DEMNA logic detects a timeout and sets the Grant Timeout bit in the DEMNA Diagnostic Register. ERR is then asserted to the CVAX, which causes the CVAX to machine check.	Resets the LANCE or gate array, saves status in a fatal error block, stops command and receive ring processing, and shuts down the port.	Shutdown
Powerfail	Nonmaskable powerfail interrupt sent to CVAX.	Saves internal status in a fatal error block, stops command and receive ring processing, and shuts down the port.	Shutdown

Key to Driver Response and User Visibility

Status—The driver returns transmit error status to the user application and continues as normal. The user application sees this error status in the I/O Status Block pointed to by the failing QIO operation (if applicable) or in device counters examined with a SHOW LINE COUNTERS command from the Network Control Program (NCP) or with a MOP (Maintenance Operations Protocol) REQUEST COUNTERS command.

Counters—The driver never notices the error. The user application sees the error in DEMNA counters by issuing a SHOW LINE COUNTERS command from NCP or by issuing a MOP REQUEST COUNTERS command.

Shutdown—The driver records a port shutdown error, returns outstanding transmits to users with a transmit error, and shuts down all users. The user application sees this error as a Circuit Down error in device counters examined with a SHOW CIRCUIT COUNTERS command from NCP. The user application sees subsequent and outstanding QIO requests returned with transmit failure status.

Table 8–15 Hardware Errors that Affect the CVAX

Error	Hardware Response	Firmware Response	Driver Response and User Visibility
Nonexistent Memory Error	The CVAX machine checks.	Saves internal status in a fatal error block, stops command and receive ring processing, and shuts down the port.	Shutdown
Internal CVAX exception (unexpected exception)	Unexpected interrupt handler is activated.	Saves internal status in a fatal error block, stops command and receive ring processing, and shuts down the port.	Shutdown
XMI Node Halt (bit 29 set in the XBER Register)	Halt asserts to the CVAX and the CVAX restarts with a restart code of 02. The CVAX starts execution at 20040000 (Boot ROM).	Running from EPROM (Boot ROM), the firmware determines that a node halt occurred and does node halt processing.	Shutdown
Fatal Parity Error Detected by the CVAX	The CVAX machine checks.	Saves current status in a fatal error block, stops command and receive ring processing, and shuts down the port.	Shutdown
Unexpected Interrupt from LANCE or Gate Array	Either the LANCE or the gate array initiates an unexpected interrupt to the CVAX. The reason is unknown.	Ignores the interrupt.	None

Key to Driver Response and User Visibility

Shutdown—The driver records a port shutdown error, returns outstanding transmits to users with a transmit error, and shuts down all users. The user application sees this error as a Circuit Down error in device counters examined with a SHOW CIRCUIT COUNTERS command from the Network Control Program (NCP). The user application sees subsequent and outstanding QIO requests returned with transmit failure status.

Table 8–16 XMI Errors

Error	Hardware Response	Firmware Response	Driver Response and User Visibility
Parity Error	The gate array sets the Parity Error bit in the DEMNA Bus Error (XBER) Register, NO ACKs the XMI transaction, and continues.	The firmware polls the XBER Register periodically to see if the Error Summary bit is set. On discovering that this bit is set, the firmware saves current status in a nonfatal error block, sends an error interrupt to the port driver if error interrupts are enabled, and proceeds normally.	Possible Crash
Write Sequence Error	The gate array sets the Write Sequence Error bit in the XBER Register, NO ACKs the XMI transaction, and continues.	The firmware polls the XBER Register periodically to see if the Error Summary bit is set. On discovering that this bit is set, the firmware saves current status in a nonfatal error block, sends an error interrupt to the port driver if error interrupts are enabled, and proceeds normally.	Possible Crash
Read/IDENT Data NO ACK	The gate array sets the Read/IDENT Data NO ACK bit in the XBER Register and continues.	The firmware polls the XBER Register periodically to see if the Error Summary bit is set. On discovering that this bit is set, the firmware saves current status in a nonfatal error block, sends an error interrupt to the port driver if error interrupts are enabled, and proceeds normally.	Possible Crash
Write Data NO ACK	The gate array sets the Write Data NO ACK bit in the XBER Register and continues.	The firmware polls the XBER Register periodically to see if the Error Summary bit is set. On discovering that this bit is set, the firmware saves current status in a nonfatal error block, sends an error interrupt to the port driver if error interrupts are enabled, and proceeds normally.	Shutdown, if retry failed. None, if retry succeeded.

Key to Driver Response and User Visibility

Shutdown—The driver records a port shutdown error, returns outstanding transmits to users with a transmit error, and shuts down all users. The user application sees this error as a Circuit Down error in device counters examined with a SHOW CIRCUIT COUNTERS command from NCP. A user application sees subsequent and outstanding QIO requests returned with transmit failure status.

Crash—The DEMNA driver machine checks if it initiated the transaction that experienced the error. This causes a system crash. If the driver did not initiate the transaction that experienced the error, the result cannot be characterized here. The system may crash.

Table 8–16 (Cont.) XMI Errors

Error	Hardware Response	Firmware Response	Driver Response and User Visibility
No Read Response	This error is always set in conjunction with another error. The gate array sets the No Read Response bit in the XBER Register and sets error status in the appropriate datamove or peek register.	The firmware polls the XBER Register periodically to see if the Error Summary bit is set. On discovering that this bit is set, the firmware saves current status in a nonfatal error block, retries the transaction, sends an error interrupt to the port driver if error interrupts are enabled, and proceeds normally.	Shutdown.
Read Sequence Error	The gate array sets the Read Sequence Error bit in the XBER Register and continues.	The firmware polls the XBER Register periodically to see if the Error Summary bit is set. On discovering that this bit is set, the firmware saves current status in a nonfatal error block, sends an error interrupt to the port driver if error interrupts are enabled, and proceeds normally.	Shutdown, if retry failed. None, if retry succeeded.
Read Error Response	The gate array sets the Read Error Response bit in the XBER Register and sets error status in the appropriate datamove or peek register.	The firmware polls the XBER Register periodically to see if the Error Summary bit is set. On discovering that this bit is set, the firmware saves current status in a nonfatal error block, retries the transaction, sends an error interrupt to the port driver if error interrupts are enabled, and proceeds normally.	Shutdown, if retry failed. None, if retry succeeded.
Command NO ACK	The gate array sets the Command NO ACK bit in the XBER Register and sets error status in the appropriate datamove or peek register.	The firmware polls the XBER Register periodically to see if the Error Summary bit is set. On discovering that this bit is set, the firmware saves current status in a nonfatal error block, retries the transaction, sends an error interrupt to the port driver if error interrupts are enabled, and proceeds normally.	Shutdown, if retry failed. None, if retry succeeded.

Key to Driver Response and User Visibility

Shutdown—The driver records a port shutdown error, returns outstanding transmits to users with a transmit error, and shuts down all users. The user application sees this error as a Circuit Down error in device counters examined with a SHOW CIRCUIT COUNTERS command from NCP. A user application sees subsequent and outstanding QIO requests returned with transmit failure status.

Crash—The DEMNA driver machine checks if it initiated the transaction that experienced the error. This causes a system crash. If the driver did not initiate the transaction that experienced the error, the result cannot be characterized here. The system may crash.

Table 8–16 (Cont.) XMI Errors

Error	Hardware Response	Firmware Response	Driver Response and User Visibility
Transaction Timeout	The gate array sets the Transaction Timeout bit in the XBER Register and sets error status in the appropriate datamove or peek register.	The firmware polls the XBER Register periodically to see if the Error Summary bit is set. On discovering that this bit is set, the firmware saves current status in a nonfatal error block, retries the transaction, sends an error interrupt to the port driver if error interrupts are enabled, and proceeds normally.	Shutdown.

Key to Driver Response and User Visibility

Shutdown—The driver records a port shutdown error, returns outstanding transmits to users with a transmit error, and shuts down all users. The user application sees this error as a Circuit Down error in device counters examined with a SHOW CIRCUIT COUNTERS command from NCP. A user application sees subsequent and outstanding QIO requests returned with transmit failure status.

Crash—The DEMNA driver machine checks if it initiated the transaction that experienced the error. This causes a system crash. If the driver did not initiate the transaction that experienced the error, the result cannot be characterized here. The system may crash.

8.5 Restarting the Port from a Fatal Error

There are three ways of restarting the port from a fatal port error:

- Issuing an INITIALIZE command to the port by writing the Port Control Initialization (XPCI) Register (after setting up the the Port Data Block and the XPD1 and XPD2 Registers)
- Causing the DEMNA to perform a node halt/restart (Section 5.2) and then issuing the DEMNA an INITIALIZE command
- Issuing a node reset to the DEMNA (Section 5.1), which causes the DEMNA to execute its power-up/reset sequence, and then issuing the DEMNA an INITIALIZE command

These methods are listed in order of increasing overhead and increasing forcefulness. For example, an INITIALIZE command involves the least overhead but is also the least forceful method. A node reset is the most forceful method but involves the most overhead. Thus, if the INITIALIZE command does not restart the DEMNA from a fatal port error, a node halt/restart should be tried, and if the node halt/restart does not work, a node reset should be tried.

CHAPTER 9

9 How to Update the DEMNA EEPROM

This chapter describes how to use the DEMNA EEPROM Update Utility (EVGDB) to update the DEMNA EEPROM. The chapter contains the following sections:

- Introduction to EVGDB
- How to Modify Flags and Parameters in EEPROM
- How to Update the Firmware in EEPROM

9.1 Introduction to EVGDB

EVGDB is a software diagnostic. Table 9–1 specifies the distribution media for EVGDB for VAX 6000 and VAX 9000 systems.

Table 9–1 Distribution Media for EVGDB

System	Tape Name	Tape Part Number
VAX 6000 Model 2xx	VAX 6000–200 Console TK50	AQ–FJ77*–ME
VAX 6000 Model 3xx	VAX 6000–300 Console TK50	AQ–FK60*–ME
VAX 6000 Model 4xx	VAX 6000–400 Console TK50	AQ–FK87*–ME
VAX 9000	VAX9000 CNSL + UCODE Tape	AQ–PAKJ*–ME

EVGDB contains six sections, each of which performs a different set of functions as described in Table 9–2.

Table 9–2 EVGDB Sections

Section Name	Description
PARAM	Allows the user to examine and, if desired, modify the user-settable flags and parameters in EEPROM.
UPDATE	Allows the user to load a new image into the DEMNA EEPROM. Also allows the user to examine and modify (if desired) the user-settable flags and parameters in EEPROM.
VERIFY	Allows the user to load a new image from the EVGDB distribution media into the system's main memory and verify against this image the image in DEMNA EEPROM.
MFG	Allows the user to load a new image into the DEMNA EEPROM and to examine the flags and parameters in EEPROM. Clears the error log in EEPROM and initializes the flags and parameters in EEPROM.
DEFAULT	If no section is specified, the DEFAULT section is run. The DEFAULT section is identical to the UPDATE section.

Table 9-2 (Cont.) EVGDB Sections

Section Name	Description
INVAL	Allows the user to invalidate and initialize the EEPROM contents, thus causing the DEMNA to run from EPROM. The INVAL section is used to enable EEPROM updates when the normal procedure (using the UPDATE or DEFAULT sections) does not work.

EVGDB can be run in stand-alone mode under the VAX Diagnostic Supervisor (VAX/DS) or in online mode under the VAX/VMS operating system. Step 1 in Section 9.2 indicates how to run EVGDB under VAX/DS. Step 2 in Section 9.2 indicates how to run EVGDB under the VAX/VMS operating system.

EVGDB has two event flags that control which flags and parameters can be modified during a given pass of EVGDB. If no event flags are specified (the default case), only the following flags can be modified:

- Enable Remote Boot
- Enable Remote DEMNA Console
- Enable Promiscuous Mode

These flags are normally the only EEPROM flags and parameters that are accessible to the user. Table 9-3 specifies which additional flags or parameters each event flag makes accessible.

Table 9-3 EVGDB Event Flags

Event Flag	Flags or Parameter Made Accessible
1	Enable Local DEMNA Console Flag Enable DEMNA Monitor Facility Flag Enable Diagnostic Logging Flag Enable Self-Test Logging Flag Enable NI RBD Logging Flag Enable XMI RBD Logging Flag Enable XNA RBD Logging Flag
3	DEMNA Console Password

Event flags are specified with the following command at the VAX/DS prompt before EVGDB is invoked:

```
DS>SET EV flag(s)
```

where *flag(s)* is one or more event flags. Multiple event flags are separated by commas—for example: 1,2,3.

9.2 How to Modify Flags and Parameters in EEPROM

Table 9–4 describes the modifiable flags and parameters in DEMNA EEPROM.

Table 9–4 Modifiable Flags and Parameters in EEPROM

Flag/Parameter	Description
Enable Remote Boot Flag	When set to Yes, enables the DEMNA to participate in remote booting over the network. When set to No, disables this function. See Appendix C for further information on remote booting over the network.
Enable Remote DEMNA Console Flag	When set to Yes, enables the DEMNA console monitor program to be accessed from a remote network node. When set to No, denies access to the console monitor program from a remote node.
Enable Local DEMNA Console Flag	When set to Yes, enables the DEMNA console monitor program to be accessed from the local network node and from the DEMNA physical console. When set to No, denies access to the console monitor program from the local network node and the DEMNA physical console.
Enable DEMNA Monitor Facility Flag	When set to Yes, enables operation of the DEMNA monitor facility, which monitors network operation. When set to No, disables the DEMNA monitor facility.
Enable Promiscuous Mode Flag	When set to Yes, the DEMNA operates by default in promiscuous mode. When set to No, the DEMNA does not operate in promiscuous mode by default. (An application can override a flag setting of No by starting up a promiscuous user.)
	In promiscuous mode, the DEMNA receives all packets on the network, regardless of a packet's destination. If no users defined to the DEMNA are enabled for promiscuous mode, the DEMNA discards packets that are not addressed to a DEMNA user. Otherwise, the DEMNA delivers all received packets to each DEMNA user for whom promiscuous mode is enabled. (See Section 2.9 for further information on DEMNA operation in promiscuous mode.)
Enable Diagnostic Logging Flag	When set to Yes, enables the logging of self-test and RBD errors to EEPROM. When set to No, disables this function.
Enable Self-Test Logging Flag	When set to Yes, enables the logging of self-test errors to EEPROM. When set to No, disables this function. For self-test errors to be logged, diagnostic error logging must be enabled, as well.
Enable NI RBD Logging Flag	When set to Yes, enables the logging of NI ROM-based diagnostic (RBD) errors to EEPROM. When set to No, disables this function. For NI RBD errors to be logged, diagnostic error logging must be enabled, as well.

Table 9–4 (Cont.) Modifiable Flags and Parameters in EEPROM

Flag/Parameter	Description
Enable XMI RBD Logging Flag	When set to Yes, enables the logging of XMI ROM-based diagnostic (RBD) errors to EEPROM. When set to No, disables this function. For XMI RBD errors to be logged, diagnostic error logging must be enabled, as well.
Enable XNA RBD Logging Flag	When set to Yes, enables the logging of XNA ROM-based diagnostic (RBD) errors to EEPROM. When set to No, disables this function. For XNA RBD errors to be logged, diagnostic error logging must be enabled, as well.
Console Password	An 8-character ASCII field that indicates the password that must be used to connect to the DEMNA console monitor program. The default console password is *XNABOARD*.

The following procedure indicates how to use EVGDB to access all the modifiable flags and parameters in EEPROM. EVGDB event flags 1 and 3 are specified.

1 To run EVGDB under VAX/DS, do the following:

   a. Boot the VAX Diagnostic Supervisor (VAX/DS) with the console BOOT command. See the system *Owner's Manual* for a description of this command. The following is the BOOT command used on a VAX 6000 system:

      ```
      >>>BOOT/XMI:n/BI:x /R5:10 CSA1 [RETURN]
      ```

      where:

      n is the XMI node number of the DWMBA (XMI-to-VAXBI adapter)

      x is the VAXBI node number of the controller for the boot device

      The following is the BOOT command used on a VAX 9000 system:

      ```
      >>>B VDS [RETURN]
      ```

      See the *VAX 9000 Family System Maintenance Guide, Vol. 2* for further information on booting VAX/DS on a VAX 9000 system.

      Go to step 3.

2 To run EVGDB under the VAX/VMS operating system, do the following:

   a. Log into the field service account, or, at the system prompt, enter the following command:

      ```
      $SET DEFAULT SYS$MAINTENANCE
      ```

b. Run the VAX Diagnostic Supervisor (VAX/DS) with the following command:

```
$RUN filename
```

where *filename* is the executable VAX/DS file as follows:

VAX System	VAX/DS File
6000 Model 2xx/3xx	ELSAA
6000 Model 4xx	ERSAA
9000	EWSAA

3 The VAX/DS header is displayed. The following VAX/DS header is displayed on a VAX 6000 Model 2xx/3xx system:

```
                VAX DIAGNOSTIC SOFTWARE
                      PROPERTY OF
              DIGITAL EQUIPMENT CORPORATION

            ***CONFIDENTIAL AND PROPRIETARY***

Use Authorized Only Pursuant to a Valid Right-to-Use License
Copyright, Digital Equipment Corporation,  1989.  All Rights Reserved.

DIAGNOSTIC SUPERVISOR.  ZZ-ELSAA-11.7-870   1-JAN-1989 00:00:28
```

4 On a VAX 6000 system, enter the following commands at the VAX/DS prompt (DS>):

```
DS>LOAD EVGDB
DS>ATTACH DEMNA HUB EXm0 n [RETURN]
DS>SELECT ALL [RETURN]
DS>SET EV 1,3 [RETURN]
DS>START/SECTION=PARAM [RETURN]
```

where:

m is the unit number of the DEMNA. The DEMNA with the lowest XMI node number is unit A, the DEMNA with the second lowest XMI node number is unit B, and so on.

n is the XMI node number of the DEMNA

5 On a VAX 9000 system, enter the following commands at the VAX/DS prompt:

```
DS>LOAD EVGDB
DS>ATTACH XJA HUB XJAx x [RETURN]
DS>ATTACH DEMNA XJAx0 EXm0 n [RETURN]
DS>SELECT ALL [RETURN]
DS>SET EV 1,3 [RETURN]
DS>START/SECTION=PARAM [RETURN]
```

where:

m is the unit number of the DEMNA. The DEMNA with the lowest XMI node number is unit A, the DEMNA with the second lowest XMI node number is unit B, and so on.

n is the XMI node ID of the DEMNA

x is the XJA unit number (0–3)

6 When run in stand-alone mode, EVGDB runs the DEMNA self-test to verify the module operation. If self-test fails, EVGDB prints an error message and continues.

```
.. Program: EVGDB - DEMNA EEPROM Update Utility, revision 1.1, 6 tests,
   at 15:06:50.29.
Testing: _EXA0

Initiating DEMNA self-test, wait 10 seconds...
```

7 EVGDB asks you to verify that the appropriate key switch on the front panel is set to the Update position.

```
Please insure that Front Panel Switch is in Update position.
Ready [(Yes), No]
```

If the key switch is set to Update, answer Yes. If the key switch is not set to Update, set it to Update before answering Yes.

NOTE: On VAX 9000 systems, the Service Processor Access switch on the operator control panel must be set to LOCAL/SPU or REMOTE/SPU. Then issue the following command to enable EEPROM updating:

```
SET XMI_UPDATE/XMI:n ON
```

where *n* is the XMI card cage number (0–3)

8 EVGDB then asks whether you want to clear the EEPROM error log.

```
Do you wish to clear the EEPROM error log? [(No), Yes]
```

Normally, you should not clear the EEPROM error log.

9 EVGDB displays the firmware revision number and date, the module serial number, and the default settings of the parameter flags in EEPROM.

```
Reading parameters from EEPROM...

EEPROM firmware rev:      0601 04-APR-1990

DEMNA Serial Number:      *SG909T1488*

Enable Remote Boot?              (Default = No)  N
Enable Remote DEMNA console?     (Default = Yes) Y
Enable Local DEMNA console?      (Default = Yes) Y
Enable DEMNA monitor facility?   (Default = No)  N
Enable Promiscuous mode?         (Default = Yes) Y
Enable Diagnostic Logging?       (Default = Yes) Y
Enable Self-test Logging?        (Default = Yes) Y
Enable NI RBD Logging?           (Default = Yes) Y
Enable XMI RBD Logging?          (Default = Yes) Y
Enable XNA RBD Logging?          (Default = Yes) Y
```

10 EVGDB asks whether you want to modify any of the flag settings.

```
Do you wish to modify any of these parameters?  [(No), Yes]
```

11 If you answer No, the program prints the following message and then exits to the VAX/DS prompt (DS>):

```
No parameter changes made.
.. End of run, 0 errors detected, pass count is 1,
   time is  20-FEB-1990 11:14:58.77
DS>
```

If you answer Yes, EVGDB prompts you for the desired setting for each flag. EVGDB indicates the default setting and current setting for each flag. The current setting is enclosed in parentheses.

```
Enable Remote Boot?              (Default = No)   [(No), Yes]
Enable Remote DEMNA console?     (Default = Yes)  [(Yes), No]
Enable Local DEMNA console?      (Default = Yes)  [(Yes), No]
Enable DEMNA monitor facility?   (Default = No)   [(No), Ye
Enable Promiscuous Mode?         (Default = Yes)  [(Yes), No]
Enable Diagnostic Logging?       (Default = Yes)  [(Yes), No]
Enable Self-test Logging?        (Default = Yes)  [(Yes), No]
Enable NI RBD Logging?           (Default = Yes)  [(Yes), No]
Enable XMI RBD Logging?          (Default = Yes)  [(Yes), No]
Enable XNA RBD Logging?          (Default = Yes)  [(Yes), No]
```

Set the flags according to the customer's requirements.

12 The program prompts you for a new DEMNA console password:

```
Enter remote DEMNA console password (up to 8 alpha-numeric characters):
```

If you specify more than eight characters, the program displays the prompt for the console password again. If you do not specify a console password (by pressing RETURN), the default password *XNABOARD* is used. If you specify fewer than eight characters, the unspecified characters default to null characters.

13 The program asks twice whether you really want to modify the flag and parameters settings as you have indicated.

```
OK to modify EEPROM parameters?  [(No), Yes] Y

Are you sure?  [(No), Yes]
```

If you want to modify the parameters, answer Yes to both prompts.

14 EVGDB writes the modified flag and parameter settings to EEPROM, prints a completion status message, and exits to the VAX/DS prompt (DS>):

```
Writing new parameters to EEPROM...
.. End of run, 0 errors detected, pass count is 1,
   time is  20-FEB-1990 11:14:17.08
DS>
```

15 Exit VAX/VDS.

```
DS>EXIT [RETURN]
```

16 If you are on a VAX 6000 system, set the key switch to its former position (Halt or Auto Start). If you are on a VAX 9000 system, issue the following system console command to disable EEPROM updating:

```
SET XMI_UPDATE/XMI:n OFF
```

where *n* is the XMI card cage number (0–3)

and then set the Service Processor Access switch to the appropriate position.

9.3 How to Update the Firmware in EEPROM

The following procedure indicates how to use EVGDB to update the firmware in EEPROM.

1 To run EVGDB under VAX/DS, do the following:

a. Boot the VAX Diagnostic Supervisor (VAX/DS) with the console BOOT command. See the system *Owner's Manual* for a description of this command. The following is the BOOT command used on a VAX 6000 system:

```
>>>BOOT/XMI:n/BI:x /R5:10 CSA1 [RETURN]
```

where:

n is the XMI node number of the DWMBA (XMI-to-VAXBI adapter)

x is the VAXBI node number of the controller for the boot device

The following is the BOOT command used on a VAX 9000 system:

```
>>>B VDS [RETURN]
```

See the *VAX 9000 Family System Maintenance Guide, Vol. 2* for further information on booting VAX/DS on a VAX 9000 system.

Go to step 3.

2 To run EVGDB under the VAX/VMS operating system, do the following:

a. Log into the field service account, or at the system prompt, enter the following command:

```
$SET DEFAULT SYS$MAINTENANCE
```

b. Run the VAX Diagnostic Supervisor (VAX/DS) with the following command:

```
$RUN filename
```

where *filename* is the executable VAX/DS file as follows:

VAX System	VAX/DS File
6000 Model 2xx/3xx	ELSAA
6000 Model 4xx	ERSAA
9000	EWSAA

3 The VAX/DS header is displayed. The following VAX/DS header is displayed on a VAX 6000 Model 2xx/3xx system:

```
                    VAX DIAGNOSTIC SOFTWARE
                          PROPERTY OF
                 DIGITAL EQUIPMENT CORPORATION

              ***CONFIDENTIAL AND PROPRIETARY***

Use Authorized Only Pursuant to a Valid Right-to-Use License
Copyright, Digital Equipment Corporation,  1989.  All Rights Reserved.

DIAGNOSTIC SUPERVISOR.  ZZ-ELSAA-11.7-870   1-JAN-1989 00:00:28
```

4 On a VAX 6000 system, enter the following commands at the VAX/DS prompt (DS>):

```
DS>LOAD EVGDB
DS>ATTACH DEMNA HUB EXm0 n [RETURN]
DS>SELECT ALL [RETURN]
DS>START/SECTION=UPDATE [RETURN]
```

where:

m is the unit number of the DEMNA. The DEMNA with the lowest XMI node number is unit A, the DEMNA with the second lowest XMI node number is unit B, and so on.

n is the XMI node number of the DEMNA

5 On a VAX 9000 system, enter the following commands at the VAX/DS prompt:

```
DS>LOAD EVGDB
DS>ATTACH XJA HUB XJAx x [RETURN]
DS>ATTACH DEMNA XJAx0 EXm0 n [RETURN]
DS>SELECT ALL [RETURN]
DS>START/SECTION=UPDATE [RETURN]
```

where:

m is the unit number of the DEMNA. The DEMNA with the lowest XMI node number is unit A, the DEMNA with the second lowest XMI node number is unit B, and so on.

n is the XMI node ID of the DEMNA

x is the XJA unit number (0–3)

6 When run in stand-alone mode, EVGDB runs the DEMNA self-test to verify the module operation. If self-test fails, EVGDB prints an error message and continues.

```
.. Program: EVGDB - DEMNA EEPROM Update Utility, revision 1.1, 6 tests,
   at 15:06:50.29.
Testing: _EXA0

Initiating DEMNA self-test, wait 10 seconds...
```

7 EVGDB asks you to verify that the appropriate key switch on the front panel is set to the Update position.

```
Please insure that Front Panel Switch is in Update position.
Ready [(Yes), No]
```

If the key switch is set to Update, answer Yes. If the key switch is not set to Update, set it to Update before answering Yes.

NOTE: On VAX 9000 systems, the Service Processor Access switch on the operator control panel must be set to LOCAL/SPU or REMOTE/SPU. Then issue the following command to enable EEPROM updating:

```
SET XMI_UPDATE/XMI:n ON
```

where *n* is the XMI card cage number (0–3)

8 EVGDB prompts you for the name of the image file to be loaded. The default file is EVGDBQ.BIN. EVGDB then searches for the image file, loads it into the system's main memory, and prints the revision number and the revision date of the loaded image.

```
Data Image file to be loaded? <EVGDBQ.BIN>

Searching...
Load complete.

Data Image firmware rev: 0601 04-APR-1990
```

9 EVGDB asks whether you want to clear the EEPROM error log.

```
Do you wish to clear the EEPROM error log?  [(No), Yes]
```

Normally, you should not clear the EEPROM error log.

10 EVGDB displays the firmware revision number and date, the module serial number, and the default settings of the three flags in EEPROM that are accessible when no EVGDB event flags are specified. (For an example of how to use event flags to access additional EEPROM parameters, see Section 9.2.)

```
Reading parameters from EEPROM...

EEPROM firmware rev:       0601 04-APR-1990

DEMNA Serial Number:       *SG909T1488*

Enable Remote Boot?              (Default = No)  N
Enable Remote DEMNA console?     (Default = Yes) Y
Enable Promiscuous Mode?         (Default = Yes) Y
```

11 EVGDB asks whether you want to modify any of the flag settings.

```
Do you wish to modify any of these parameters?  [(No), Yes]
```

12 Answer No if you wish only to load a new firmware image. In this case, EVGDB indicates that no changes have been made to parameters in EEPROM.

```
No parameter changes made.
```

EVGDB then reads the firmware image from EEPROM into the system's main memory and compares this image with the image loaded from the data image file (EVGDBQ.BIN) specified above.

```
Reading parameters from EEPROM...
```

Normally, the image from the data image file is a newer revision than than the image read from EEPROM. If this is not the case, EVGDB indicates that the image from the data image file is older than the image read from EEPROM and asks you whether you want to load the older image into EEPROM. In this example, the image from the data image file is newer than the image from EEPROM. EVGDB writes the newer image from the data image file to DEMNA RAM, reads the image from RAM to verify its integrity, and then writes the image from RAM into DEMNA EEPROM.

```
Writing new image to RAM...
Reading image from RAM...
Writing RAM image to EEPROM...
```

EVGDB then reads the new image from EEPROM and displays the firmware revision number and date, the module serial number, and the default settings of the three flags in EEPROM.

```
Reading parameters from EEPROM...

Data Image firmware rev: 0601 04-APR-1990

EEPROM firmware rev:     0601 04-APR-1990

DEMNA Serial Number:     *SG909T1488*

Enable Remote Boot?             (Default = No)  Y
Enable Remote DEMNA console?    (Default = Yes) Y
Enable Promiscuous Mode?        (Default = Yes) Y
```

EVGDB then reads the image from EEPROM into the system's main memory to verify the image against the image from the data image file.

```
Reading EEPROM image...
Verification complete.
```

EVGDB then prints a status completion message and exits to the VAX/DS prompt:

```
.. End of run, 0 errors detected, pass count is 1,
   time is  9-APR-1990 13:30:18.50
DS>
```

13 Exit VAX/VDS.

```
DS>EXIT [RETURN]
```

14 If you are on a VAX 6000 system, set the key switch to its former position (Halt or Auto Start). If you are on a VAX 9000 system, issue the following system console command to disable EEPROM updating:

```
SET XMI_UPDATE/XMI:n OFF
```

where *n* is the XMI card cage number (0–3)

and then set the Service Processor Access switch to the appropriate position.

CHAPTER 10

10 Console Monitor Program

This chapter describes how to use the console monitor program. The chapter contains the following sections:

- Introduction
- Setup Procedure
- Invoking the Console
- Exiting the Console
- Console Commands
- Console Command Language Control Characters
- How to Use the Status Screens
- How to Use the Network Screens

10.1 Introduction

The DEMNA firmware includes a console monitor program that enables a user at virtually any terminal on the network to monitor DEMNA operation and network traffic. The console monitor program is resident in the DEMNA EEPROM and therefore does not have to be loaded into the DEMNA. Since the console monitor program runs entirely on the DEMNA, it consumes a negligible amount of host resources.

The console monitor program consists primarily of 12 interactively invoked screens (displays) that indicate current operating parameters and errors. The console dynamically monitors over 100 parameters. These parameters are updated every 3 seconds on-screen while being displayed.

In addition to displaying and updating key operational and diagnostic parameters, the console monitor program allows a user to examine the contents of DEMNA memory locations and registers.

To aid new users, the console monitor program provides online help information that describes the commands available to the user.

The console monitor program has the following security features:

- A password must be supplied when accessing the console monitor program from the network.
- Only one user at a time can access the console monitor program from a terminal on the network.
- A system manager can disable the console monitor program entirely or deny access to the program from a remote network node. The parameters that control access to the console monitor program can be changed with the DEMNA EEPROM Update Utility, as described in Chapter 9.

10.2 Setup Procedure

A user can access the console monitor program from one of three locations:

- From a terminal attached directly to the DEMNA. This terminal is referred to as the physical console.
- From a terminal at the local network node (the DEMNA's node)
- From a terminal at a remote network node (a node other than the DEMNA's)

When the console monitor program is accessed from a terminal on a network node (not from the physical console), either of the following must be used to make the actual connection:

- The Network Control Program (NCP)
- A console connection program such as the one described in Appendix L.

The above connection locations and methods require three basic setup procedures:

- For the physical console
- For a networked terminal when using NCP
- For a networked terminal when using the console connection program

10.2.1 For the Physical Console

When using the physical console to access the console monitor program, the only setup required is to connect the terminal cable to section D2 of the DEMNA's XMI slot and to set the terminal baud rate to 19.2K baud. The console monitor prompt (XNA>) will appear when the terminal is powered on.

The installation of the physical console is described in Appendix B. For information on ordering the cable, see the *Systems and Options Catalog*.

10.2.2 For a Networked Terminal When Using NCP

First, appropriate entries must be made in the network databases of the the network node from which the user will access the console monitor program. (This node can be the DEMNA's network node or a remote node). The entries are made with NCP SET/DEFINE commands.

NOTE: **DECnet must be running for NCP fields to be valid.**

The SET commands make entries in the node's temporary network database. These entries will remain valid until the system is rebooted. The DEFINE commands make entries in the node's permanent network database. These entries are retained across system reboots. (For more information on these commands, see the *VMS Network Control Program Manual*.)

The NCP commands supply the following information:

- Hardware address: the DEMNA's default Ethernet address (DPA), which is the address stored in the DEMNA MAC Address PROM
- Service password: the password required to connect to the console monitor program
- Service circuit: the circuit parameter associated with the network node from which the user will access the console monitor program. The service circuit is the value for the VIA parameter of NCP's CONNECT NODE command. For example, if the user is making the connection from a VAXstation 2000, the service circuit is SVA-0. If the user is making the connection from a MicroVAX, the service circuit is QNA-0. (See the *VMS Network Control Program Manual* for further information.)

NOTE: To use NCP SET commands, a user must have the OPER privilege. To use NCP DEFINE commands, a user must have the SYSPRV privilege.

To enhance security, the service password can be omitted from the information supplied with the NCP SET/DEFINE commands. A user must then supply the service password with the NCP CONNECT command each time that he/she wishes to connect to the console via NCP (see Section 10.3.1).

10.2.2.1 DEMNA at Remote Node

If the user is at a different network node than the DEMNA's network node, the following set of NCP SET/DEFINE commands should be used to set up the console:

```
$MC NCP
NCP>SET NODE node_name HARD ADDR address
NCP>DEF NODE node_name HARD ADDR address

NCP>SET NODE node_name SERVICE PASSWORD password
NCP>DEF NODE node_name SERVICE PASSWORD password

NCP>SET NODE node_name SERVICE CIRCUIT circuit_name
NCP>DEF NODE node_name SERVICE CIRCUIT circuit_name
```

where:

node_name is the logical name assigned to the DEMNA Ethernet node

address is the DEMNA default physical address (DPA) on the Ethernet

circuit_name is the service circuit for the system from which you are sending the commands. (Use the NCP command SHOW KNOWN CIRCUITS to determine the appropriate circuit name for your system.)

password is the password for the DEMNA console monitor program. The default password is 584E41424F415244 (hex).

Example 10–1 is a sample command sequence for setting up the console using NCP when the user is at a remote network node (for example, when the user is at node A and the DEMNA is at node B).

Example 10–1 Console Setup (DEMNA at Remote Node)

```
$MC NCP
NCP>SET NODE NODE_B HARD ADDR 08-00-3C-4F-22-22
NCP>DEF NODE NODE_B HARD ADDR 08-00-3C-4F-22-22

NCP>SET NODE NODE_B SERVICE PASSWORD 584E41424F415244
NCP>DEF NODE NODE_B SERVICE PASSWORD 584E41424F415244

NCP>SET NODE NODE_B SERVICE CIRCUIT BNA-0
NCP>DEF NODE NODE_B SERVICE CIRCUIT BNA-0
```

10.2.2.2 DEMNA at Local Node

If a user is at the same network node as the DEMNA, the user must first create a node name and assign it a valid DECnet address. The node name and DECnet address must be distinct from other node names and DECnet addresses already defined to the system. The best approach is to find a DECnet area that is not being used and simply assign a node name to an address in that area.

NOTE: **This procedure is necessary because the NCP CONNECT command does not allow a node to connect to itself. This procedure forces the CONNECT command to default to the hardware address specified in the CONNECT command.**

Use the following procedure to set up the console using NCP when the user is at the local node:

1 Use the following NCP command to determine whether the created node name is unique:

```
$MC NCP
NCP>SHOW NODE node_name
```

where *node_name* is the Ethernet node name

If the node name is unique, NCP will display a message similar to the following:

```
Node Volatile Summary as of 13-SEP-1989 14:02:52
%NCP-W-UNRCMP, Unrecognized component, Node
```

2 Use the following NCP command to determine whether the assigned DECnet address is unique:

```
NCP>SHOW NODE DECnet_address
```

where *DECnet_address* is the assigned DECnet address

NCP will display a message similar to the following:

```
Node Volatile Summary as of 13-SEP-1989 14:03:55

    Node          State     Active  Delay    Circuit    Next node
                            Links
```

If the DECnet address is unique, there will be no entry under *Node*.

3 After the node name has been created and assigned a DECnet address, use the following SET/DEFINE commands to enter the information in the network databases:

```
NCP>SET NODE DECnet_address NAME node_name
NCP>DEF NODE DECnet_address NAME node_name
```

where:

DECnet_address is the assigned DECnet address

node_name is the Ethernet node name

4 Use the following command sequence for setting up the console:

```
$MC NCP
NCP>SET NODE node_name HARD ADDR address
NCP>DEF NODE node_name HARD ADDR address
NCP>SET NODE node_name SERVICE PASSWORD 584E41424F415244
NCP>DEF NODE node_name SERVICE PASSWORD 584E41424F415244
NCP>SET NODE node_name SERVICE CIRCUIT circuit_name
NCP>DEF NODE node_name SERVICE CIRCUIT circuit_name
```

where:

node_name is the logical name that you are assigning to the DEMNA Ethernet node

address is the DEMNA default physical address (DPA) on the Ethernet

circuit_name is the service circuit for the system from which you are sending the commands. (Use the NCP command SHOW KNOWN CIRCUITS to determine the appropriate circuit name for your system.)

10.2.3 For a Networked Terminal When Using the Console Connection Program

If your system does not run NCP, use the console connection program described in Appendix L to connect to the DEMNA console monitor program.

10.3 Invoking the Console

There are two different procedures for invoking the console: one for NCP and one for the console connection program.

10.3.1 Using NCP

Once the appropriate information about the DEMNA console monitor program has been entered into the network databases of the local node, the console can be invoked with the following command sequence:

```
$MC NCP
NCP>CONNECT NODE node_name
Console connected (press CTRL/D when finished)
XNA>
```

where *node_name* is the logical name that was assigned to the DEMNA Ethernet node

If the service password was not supplied with the NCP SET/DEFINE commands when the console was set up (see Section 10.2.2), the user must supply the service password with the NCP CONNECT command:

```
NCP>CONNECT NODE node_name SERVICE PASSWORD service_password
```

Example 10–2 illustrates invoking the DEMNA console without the service password.

Example 10–2 Using NCP to Invoke the Console Monitor Program

```
$MC NCP
NCP>CONNECT NODE MYNODE
Console connected (press CTRL/D when finished)
XNA>
```

If NCP cannot connect to the console, it will return an error message. See the *VMS Network Control Program Manual* for further information.

10.3.2 Using the Console Connection Program

Once the console connection program has been assembled and linked (see Appendix L), use the following commands to invoke the console:

```
$ASSIGN Ethernet_device CONSOLE$DEVICE
$RUN CONSOLE
XNA>
```

where *Ethernet_device* is the device number for the user's Ethernet node

When a connection to the target DEMNA is established, the DEMNA console prompt (XNA>) is displayed. Example 10–3 illustrates the commands used to invoke the console monitor program with the console connection program.

Example 10–3 Using the Console Connection Program to Invoke the Console Monitor Program

```
$ASSIGN EXA0 CONSOLE$DEVICE
$RUN CONSOLE
XNA>
```

10.4 Exiting the Console

To exit the console, type [CTRL/D].

10.5 Console Commands

There are five console commands:

- BLANK
- EXAMINE
- HELP
- SHOW
- T/R

Table 10–1 summarizes these commands. Each command is then described in detail.

Table 10-1 Console Commands

Command	Parameter	Description
BLANK		Clears the screen and displays the console prompt (XNA>).
EXAMINE		Displays the contents of the next DEMNA memory location or gate array register.
EXAMINE	. (period)	Displays the contents of the current DEMNA memory location or the current gate array register.
EXAMINE	*address*	Displays the contents of the location at the specified address, where *address* is a longword address.
EXAMINE/NUMBER=*n*		Displays the contents of the next *n* longwords, where *n* is an integer.
EXAMINE/REGISTER		Displays the contents of all gate array registers.
HELP		Displays the general Help screen.
HELP	EXAMINE	Displays the Help screen for the EXAMINE command.
HELP	CONTROLCHAR	Displays the Help screen that provides definitions of all the console command language control characters.
HELP	SHOW	Displays the Help screen for the SHOW command.
SHOW	BUS	Displays the current configuration of the XMI bus.
SHOW	ERROR H*n*	Displays fatal error block *n*, where *n* is an integer from 1 through 5.
SHOW	ERROR S*n*	Displays nonfatal error block *n*, where *n* is an integer from 1 through 5.
SHOW	IMAGE	Displays the revision number and revision date of the DEMNA firmware.
SHOW	HISTORY	Displays a summary of all errors recorded in EEPROM.
SHOW	HISTORY *n*	Displays entry *n* of the EEPROM error history, where *n* is an integer from 1 to 31.
SHOW	NETWORK	Displays a continuously updated summary of network activity.
SHOW	STATUS	Displays continuously updated screens of the following: the DEMNA data link (NI) counters, statistics on the DEMNA's use of the Ethernet, error summary counters, the percentage of CVAX time used by each DEMNA firmware process, and the percentage of XMI bus traffic generated by each XMI node.
SHOW	STATUS/ERROR	Displays continuously updated screens of the DEMNA transmit, receive, and LANCE counters.
SHOW	STATUS/INTERVAL	Displays the same screen as the SHOW STATUS command. The only difference between the two screens is the time interval for which the NI counters and the Error Summary counters record events. In the Show Status screen, these counters record events from the last reset of the DEMNA module. In the Show Status/Interval screen, these counters record events starting when the SHOW STATUS/INTERVAL command is entered, when the screen is invoked by typing CTRL/E from the Status screen, or when the screen is invoked by typing CTRL/A from the Interval Status/Error screen.
SHOW	USER	Displays information about the users currently defined to the DEMNA port.
SHOW	XPUD	Displays the contents of the DEMNA Power-Up Diagnostic (XPUD) Register.
T/R		Invokes the diagnostic monitor, from which ROM-based diagnostics can be run. This command is valid only when entered from the DEMNA's physical console and when the DEMNA is in the uninitialized state. (To put the DEMNA in the uninitialized state, stop all applications that are using the DEMNA or reset the system with auto restart off.)

BLANK

The BLANK command clears the screen and displays the console prompt (XNA>).

FORMAT BLANK

EXAMPLE

```
XNA>BLANK


XNA>
```

EXAMINE

The EXAMINE command displays the contents of the specified location in DEMNA I/O or memory space.

FORMAT

EXAMINE [. (period) | *address* | /NUMBER=*n* | /REGISTER]

PARAMETERS

. (period)
Specifies that the contents of the current location be displayed.

address
Specifies that the contents of a longword location be displayed.

QUALIFIERS

/NUMBER=n
Specifies that the next *n* longwords be displayed.

/REGISTER
Specifies that the gate array registers be displayed.

restrictions

None.

DESCRIPTION

All system responses to the EXAMINE command are formatted and displayed on the console terminal.

EXAMINE	Displays contents of the next location.
EXAMINE . (period)	Displays the contents of the current location.
EXAMINE *address*	Displays the contents of the specified longword address.
EXAMINE/NUMBER=*n*	Displays the contents of the next *n* locations.
EXAMINE/REGISTER	Displays the contents of the gate array registers.

EXAMPLES

1

```
XNA>EXAMINE
 00000004/ 31303030
```

2

```
XNA>EXAMINE
 00000004/ 31303030
```

3 XNA>EXAMINE 0000FFFF

```
0000FFFF/ 00110000
```

4 XNA>EXAMINE/NUMBER=5

```
00000008/ 4A2D3132
0000000C/ 312D4E55
00000010/ 20393839
00000014/ 79706F43
00000018/ 68676972
```

5 XNA>EXAMINE/REGISTER

```
-- 08-00-2B-00-00-01 -- CVAX and GA Registers -- 01-JAN-1988 00:01:47 --

     XDEV/       01000C03          DMPOR0/   F0000800
     DiagReg/    D7A1C000          DMCSR0/   00000000
     GACSR/      30030024          DMXMI0/   00000084
     GAHIR/      00000000          DMNPA0/   00000000
     GAIVR/      00000000
     XBE/        8000A0E4          DMPOR1/   200014A0
     XPST/       00000002          DMCSR1/   00000000
     XPD1/       00010800          DMXMI1/   00000089
     XPD2/       00000001          DMNPA1/   00000224

     PKXMIL0/    0000AE74          DMPOR2/   10001480
     PKXMIH0/    40000000          DMCSR2/   0000003C
     PKDATA0/    A000019C          DMXMI2/   0000008D
     PKDATB0/    A000019D          DMNPA2/   00000234

     PKXMIL1/    00010A18          DMPOR3/   00000000
     PKXMIH1/    40000000          DMCSR3/   00000000
     PKDATA1/    00011A00          DMXMI3/   00000000
     PKDATB1/    003C0000          DMNPA3/   00000000
```

HELP

The HELP command displays information on the EXAMINE and SHOW console commands, as well as the console command language control characters.

FORMAT

HELP [*command* | **CONTROLCHAR**]

PARAMETERS

command

Specifies that help information be displayed for one of the following console commands:

- EXAMINE
- SHOW

CONTROLCHAR

Specifies that help information be displayed for the console command language control characters.

restrictions

None.

DESCRIPTION

All system responses to the HELP command are formatted and displayed on the console terminal.

HELP	Displays the general Help screen.
HELP EXAMINE	Displays the Help screen for the EXAMINE command.
HELP SHOW	Displays the Help screen for the SHOW command.
HELP CONTROLCHAR	Displays the Help screen for the console command language control characters.

EXAMPLES

1

```
XNA>HELP

Welcome to the DEMNA console.  The console is used to monitor
the Ethernet traffic, counters and internal data on this node.

The following commands are supported by the console:
 BLANK          Clear the screen
 EXAMINE        Examine DEMNA memory locations
 HELP           Display this help screen
 SHOW           Displays information about the adapter

For additional help on the following type: HELP <command>.
      EXAMINE     SHOW     CONTROLCHAR
```

2

```
XNA>HELP EXAMINE

--- EXAMINE</qualifier> <parameter> ---

Displays the data stored at the specified address (in longwords)

     EXAMINE            - Examine the next location
     EXAMINE .          - Examine the current location
     EXAMINE <address>  - Examine the given address
     EXAMINE/NUMBER=<n> - Examine the next 'n' longwords
     EXAMINE/REGISTER   - Examine the contents of Gate Array registers
```

3

```
XNA>HELP SHOW

--- SHOW <parameter> ---

The parameters are as follows:

     BUS     - Display XMI bus configuration
     ERROR   - Display error block indicated by the parameter
                Hn - Harderror block number 'n', where 0<n<6 (SH ERR H<n>)
                Sn - Softerror block number 'n', where 0<n<6 (SH ERR S<n>)
     IMAGE   - Display information about firmware image
     HISTORY - Display EEPROM history data (SH HIS) or (SH HIS <n>)
     NETWORK - Display interval network summary, continuously updating
                (Ctrl-A for accumulated network summary)
     STATUS  - Display module status, continuously updating
                /ERROR    (or Ctrl-A) - Display error counters screen
                /INTERVAL (or Ctrl-E) - Display counters from this point
                (SH STATUS) (SH STATUS/ERROR) (SH STATUS/INTERVAL)
     XPUD    - Display self test results - Power Up Diagnostic Register
     USER    - Display currently defined users
```

4

```
XNA>HELP CONTROLCHAR

--- CONTROL CHARACTERS ---

The following control characters are available:
     CTRL/A  -  During SHOW STATUS command - Switch between
                error status and regular status screens
                                      or
                During SHOW NETWORK command - Switch between accumulated
                network summary and interval network summary screens
     CTRL/D  -  Disconnect the console from the remote connection
     CTRL/E  -  During SHOW STATUS command - Switch between
                interval and accumulated counters
     CTRL/L  -  Recall previous command line
     CTRL/U  -  Clear current command line
     CTRL/W  -  Refresh the STATUS or NETWORK screen
```

SHOW

The SHOW command displays the following:

- Configuration of the XMI system that contains the DEMNA
- DEMNA fatal error blocks and nonfatal error blocks
- DEMNA firmware revision
- Error history in the DEMNA EEPROM
- Summary of network activity
- Statistical information about the DEMNA and the network
- DEMNA Power-Up Diagnostic (XPUD) Register
- Users currently defined to the DEMNA

FORMAT

```
       { BUS                              }
       { ERROR Hn                         }
       { ERROR Sn                         }
       { HISTORY [n]                      }
SHOW   { IMAGE                            }
       { NETWORK                          }
       { STATUS [ /ERROR    ]             }
       {        [ /INTERVAL ]             }
       { USER                             }
       { XPUD                             }
```

PARAMETERS

BUS

Displays the configuration of the XMI system that contains the DEMNA.

ERROR Hn

Displays the fatal error block specified by *n*, which must be an integer from 1 to 5.

ERROR Sn

Displays the nonfatal error block specified by *n*, which must be an integer from 1 to 5.

HISTORY [n]

Displays the error summary stored in the DEMNA EEPROM. If no number is supplied, a summary of all errors recorded in EEPROM is displayed. If a number is supplied, the data for only that error is displayed.

IMAGE

Displays the firmware revision number and date for both the EEPROM image and the EPROM image.

NETWORK

Displays a continuously updated summary of network activity. Statistical information is displayed for the six most active Ethernet users and the seven most active Ethernet nodes.

STATUS

Displays a continuously updated screen that includes the following:

- Statistical information on the DEMNA's use of the network
- Data link counters
- Percentage of DEMNA CPU time used by each DEMNA firmware process
- Error summary counters
- Number of DEMNA-internal transmit and receive buffers in use
- Percentage of XMI traffic generated by each XMI node
- Statistical information on the use of the entire network

STATUS/ERROR

Displays a continuously updated screen that includes the following:

- Transmit error counters
- Receive error counters
- LANCE counters
- Date and time at which various errors last occurred

STATUS/INTERVAL

Displays the same screen as the SHOW STATUS command. The only difference between the two screens is the time interval for which the NI counters and the Error Summary counters record events. In the Show Status screen, these counters record events from the last reset of the DEMNA module. In the Show Status/Interval screen, these counters record events starting when the SHOW STATUS/INTERVAL command is entered or when the Status/Interval screen is invoked by typing [CTRL/E] from the Status screen.

USER

Displays the setup parameters for each user defined to the DEMNA port.

XPUD

Displays the DEMNA Power-Up Diagnostic (XPUD) Register.

---

restrictions *None.*

DESCRIPTION

All system responses to the SHOW command are formatted and displayed on the console terminal.

SHOW BUS	Displays the configuration of the XMI system that contains the DEMNA.
SHOW ERROR H*n*	Displays fatal error block *n*, where *n* is an integer from 1 to 5.
SHOW ERROR S*n*	Displays nonfatal error block *n*, where *n* is an integer from 1 to 5.
SHOW HISTORY	Displays the error history summary stored in the DEMNA EEPROM.
SHOW HISTORY *n*	Displays the data for entry *n* in EEPROM history, where *n* is an integer from 1 to 31. This data is error-specific. See Appendix M for a description of the history entries.
SHOW IMAGE	Displays the revision number and revision date for the firmware in DEMNA EEPROM and the firmware in DEMNA EPROM.
SHOW NETWORK	Displays a continuously updated summary of network activity. There is an Interval Network screen and an Accumulated Network screen. See Section 10.8.3 for a detailed description of these two screens.
SHOW STATUS	Displays a continuously updated screen that includes the following: • Statistical information on the DEMNA's use of the network • Data link counters • The percentage of DEMNA CPU time used by each DEMNA firmware process • Error summary counters • The number of DEMNA-internal transmit and receive buffers in use • The percentage of XMI traffic generated by each XMI node • Statistical information on the use of the entire network

SHOW STATUS/ERROR	Displays a continuously updated screen that includes the following: • Transmit error counters • Receive error counters • LANCE counters • The date and time at which various errors last occurred There is an Interval Status/Error screen and an Accumulated Status/Error screen. In the Accumulated Status/Error screen, which is the default Status/Error screen invoked by the SHOW STATUS/ERROR command, the counters record events from the last reset of the DEMNA module. In the Interval Status/Error screen, the counters record events starting when the screen is first invoked by typing CTRL/E from the Accumulated Status/Error screen.
SHOW STATUS/INTERVAL	Displays the same screen as the SHOW STATUS command. The only difference between the two screens is the time interval for which the NI counters and the Error Summary counters record events. In the Show Status screen, these counters record events from the last reset of the DEMNA module. In the Show Status/Interval screen, these counters record events starting when the SHOW STATUS/INTERVAL command is entered or when the Status/Interval screen is invoked by typing CTRL/E from the Status screen.
SHOW USER	Displays the setup parameters for each user defined to the DEMNA port.
SHOW XPUD	Displays the DEMNA Power-Up Diagnostic (XPUD) Register.

EXAMPLES

1

```
XNA>SHOW BUS

-- 08-00-2B-00-00-01 -- XMI Bus Configuration -- 01-JAN-1988 00:18:07 -

Node     XDEV      Module   Revision
----   --------    ------   --------
 01    01000C03    DEMNA      1.00
 13    00008001    KA62A      0.00
 14    00014001    MS62A      0.01
```

Table 10–2 Show Bus Screen—Parameter Definitions

Parameter	Description
Ethernet address	The actual physical address (APA) of the DEMNA.
Date and time	Current date and time.
Node	XMI node number of the DEMNA.

Table 10-2 (Cont.) Show Bus Screen—Parameter Definitions

Parameter	Description
XDEV	Contents of the module's Device (XDEV) Register. See Appendix H for a listing of the device types for XMI modules.
Module	Module name.
Revision	Module revision number from the module's XDEV Register.

2

```
SHOW ERROR H1

-- 08-00-2B-00-00-01 -- Hard Error Block #  1 -- 01-JAN-1988 00:01:13 -

R0/    20135B00                              XBE/       00000380
R1/    03135B00                              XFADR/     8000B5F4
R2/    20100000                              XFAER/     10000000
R3/    00000000                              GACSR/     E0000030
R4/    00017568                              DiagReg/   0581C000
R5/    00000010                              XPST/      00000001
R6/    00000040                              XPD1/      0B2B0008
R7/    000122F8                              XPD2/      000021BB
R8/    00012310                              XPSTpnd/   00000B01
R9/    00000000                              XPD1pnd/   00008EBA
RA/    20100000                              Stack1/    00000080
RB/    00012200                              Stack2/    EFF00004
RC/    0003C800                              Stack3/    D0E0FC80
                                             Stack4/    01C07007
                                             Stack5/    00008EBA
                                             Stack6/    04010009
```

Table 10-3 Hard Error Block Screen—Parameter Definitions

Parameter	Description
Ethernet address	The DEMNA's actual physical address (APA).
Hard Error Block #	The Hard Error Block that is being displayed.
Date and time	The date and time at which the error occurred.
R0	CVAX register R0.
R1	CVAX register R1.
R2	CVAX register R2.
R3	CVAX register R3.
R4	CVAX register R4.
R5	CVAX register R5.
R6	CVAX register R6.
R7	CVAX register R7.
R8	CVAX register R8.
R9	CVAX register R9.
RA	CVAX register RA.
RB	CVAX register RB.
RC	CVAX register RC.

Table 10–3 (Cont.) Hard Error Block Screen—Parameter Definitions

Parameter	Description
XBE	Bus Error Register (also known as the XBER).
XFADR	Failing Address Register.
XFAER	Failing Address Extension Register.
GACSR	Gate Array Control/Status Register.
Diagreg	Diagnostic Register.
XPST	Port Status Register.
XPD1	Port Data 1 Register.
XPD2	Port Data 2 Register.
XPSTpnd	Pending Port Status Register. This is the value that will be loaded into the XPST Register after the next state change (after error handling has been completed).
XPD1pnd	Pending Port Data 1 Register. This is the value that will be loaded into the XPD1 Register after the next state change (after error handling has been completed).
Stack1	The first longword on the DEMNA CVAX stack.
Stack2	The second longword on the DEMNA CVAX stack.
Stack3	The third longword on the DEMNA CVAX stack.
Stack4	The fourth longword on the DEMNA CVAX stack.
Stack5	The fifth longword on the DEMNA CVAX stack.
Stack6	The sixth longword on the DEMNA CVAX stack.

8

```
SHOW ERROR S1

-- 08-00-2B-00-00-01 -- Soft Error Block #  1 -- 01-JAN-1988 00:00:00 --

Tran1/   00001440
Tran2/   0100003C
Tran3/   00FFFFFF
Tran4/   00000234
XBE/     8000A0E4
XFADR/   1FFFFE00
XFAER/   100F0000
```

Table 10–4 Soft Error Block Screen—Parameter Definitions

Parameter	Description
Ethernet address	The DEMNA's actual physical address (APA).
Soft Error Block #	The nonfatal error block that is being displayed.
Date and time	The date and time at which the error occurred.
Tran1	For a datamove error: datamove register DMPOR*n*. For a peek error: peek register PKXMIL*n*. For an interrupt error: the Gate Array Control/Status Register (GACSR). For an XBER-reported error: zeros.

Table 10–4 (Cont.) Soft Error Block Screen—Parameter Definitions

Parameter	Description
Tran2	For a datamove error: datamove register DMCSR*n*. For a peek error: peek register PKXMIH*n*. For an interrupt error: the Gate Array Host Interrupt Register (GAHIR). For an XBER-reported error: zeros.
Tran3	For a datamove error: datamove register DMXMI*n*. For a peek error: peek register PKDATA*n*. For an interrupt error: the Gate Array IDENT Vector Register (GAIVR). For an XBER-reported error: zeros.
Tran4	For a datamove error: datamove register DMNPA*n*. For a peek error: peek register PKDATB*n*. For an interrupt error: the Gate Array Timer Register (GATMR). For an XBER-reported error: zeros.
XBE	Bus Error Register (also know as the XBER).
XFADR	Failing Address Register.
XFAER	Failing Address Extension Register.

4
```
XNA>SHOW HISTORY
-- 08-00-2B-00-00-01 -- Error History Summary -- 01-JAN-1988 01:05:17 --

  #          Date          Type  Seq     #          Date          Type  Seq
 --- -------------------- ---- -----   --- -------------------- ---- -----
  1) 01-JAN-1988 00:07:06  Mck     4   17) 01-JAN-1988 00:00:00  None    0
  2) 01-JAN-1988 00:32:31  Mck     8   18) 01-JAN-1988 00:00:00  None    0
  3) 01-JAN-1988 00:19:06  Mck     6   19) 01-JAN-1988 00:00:00  None    0
  4) 01-JAN-1988 00:00:00  None    0   20) 01-JAN-1988 00:00:00  None    0
  5) 01-JAN-1988 00:00:00  None    0   21) 01-JAN-1988 00:00:00  None    0
  6) 01-JAN-1988 00:26:51  Mck     7   22) 01-JAN-1988 00:00:00  None    0
  7) 01-JAN-1988 00:00:00  None    0   23) 01-JAN-1988 00:00:00  None    0
  8) 01-JAN-1988 00:00:00  None    0   24) 01-JAN-1988 00:00:00  None    0

  9) 01-JAN-1988 00:00:00  None    0   25) 01-JAN-1988 12:59:17  FUpd    2
 10) 01-JAN-1988 00:00:00  None    0   26) 01-JAN-1988 00:00:00  None    0
 11) 01-JAN-1988 00:00:00  None    0   27) 01-JAN-1988 00:00:00  None    0
 12) 01-JAN-1988 00:00:00  None    0   28) 01-JAN-1988 00:18:12  FUpd    3
 13) 01-JAN-1988 00:00:00  None    0   29) 01-JAN-1988 00:00:00  None    0
 14) 01-JAN-1988 00:00:00  None    0   30) 01-JAN-1988 00:00:00  None    0
 15) 01-JAN-1988 00:00:00  None    0   31) 01-JAN-1988 00:00:00  None    0
 16) 01-JAN-1988 00:00:00  None    0
```

Table 10–5 History Screen—Parameter Definitions

Parameter	Description
Ethernet address	The DEMNA's actual physical address (APA).
Date and time	The current date and time.
#	The entry number. The EEPROM has 31 history entries.
Date	The date and time at which the entry was made. For the *None* entry type (no entry), the date and time are always 01-JAN-1988 00:00:00.
Type	The type of entry.

Table 10–5 (Cont.) History Screen—Parameter Definitions

Parameter	Description
Seq	A number indicating the order in which the entry was made. Lower-numbered entries were made before higher-numbered entries.

5
```
XNA>SHOW HISTORY 1

-- 08-00-2B-00-00-01 -- Error History #  1 -- 01-JAN-1988 01:05:17 --

Type:  Machine Check
Date:  01-JAN-1988 00:00:54
Number of times this event occurred:     1
Saved Data:    00000B01
               00009014
               00000080
               EFF00004
               D0E0FC80
```

Table 10–6 History Entry Screen—Parameter Definitions

Parameter	Description
Ethernet address	The DEMNA's actual physical address (APA).
Error History #	The number of the history entry.
Date and time	The current date and time.
Type	The type of error recorded. (This example shows a machine check entry.)
Date	The date and time when the history entry was logged.
Number of times this event occurred	The number of times this particular error type has occurred since error history in EEPROM was cleared.
Saved data	Error data specific to the type of history entry.

Example 5 above shows a sample report for a machine check history entry. See Appendix M for a description of the data displayed for each type of history entry.

6

```
XNA>SHOW IMAGE

-- 08-00-2B-00-00-01 -- Firmware Image Data -- 01-JAN-1988 01:08:26 --

EEPROM Firmware image date:     14-FEB-1990
EEPROM Firmware revision:              0600

EPROM Firmware image date:      14-FEB-1990
EPROM Firmware revision:               0600

Module serial number:          *SG909FF916*
```

Table 10–7 Image Screen—Parameter Definitions

Parameter	Description
Ethernet address	The DEMNA's actual physical address (APA).
Date and time	The current date and time.
EEPROM firmware image date	The firmware revision date for the image in EEPROM.
EEPROM firmware revision	The firmware revision number for the image in EEPROM.
EPROM firmware image date	The firmware revision date for the image in EPROM.
EPROM firmware revision	The firmware revision number for the image in EPROM.
Module serial number	The module serial number is a 12-character ASCII field that identifies the module.

7

```
XNA>SHOW NETWORK

-- 08-00-2B-00-00-01 -- Network -- 01-JAN-1988 01:02:41 --

                        - 2999996 usecs --   7.4% NI--  -- 00:00:06 --   1.5% NI--
# User                 Pks/Sec  Byt/Pk  %NI-Cur   Packets    Bytes(k) %NI-Tot
- ---------            -------  ------  -------   -------    -------- -------
1 60-07 NISca              328     211     6.5%      1959          49    1.1%
2 60-03*Decnet              70     155     1.0%       424           9    0.2%
3 60-04 Lat                 20     106     0.2%       109           2    0.0%
4 60-02 MopRC               14      94     0.1%        95           1    0.0%
5 80-3F LTM                  0    1490     0.0%         2           0    0.0%
6 08-00 IP                   1      98     0.0%         3           0    0.0%

# Nodes                Pks/Sec  Byt/Pk  %NI-Cur   Packets    Bytes(k) %NI-Tot
- ---------            -------  ------  -------   -------    -------- -------
1 11.111                   122     412     4.3%       796          10    0.5%
2 11.112                   119     413     4.3%       754          10    0.5%
3 AB-00-00-03-00-00         28     238     0.6%       171           0    0.0%
4 11.113                    37     143     0.5%       216           0    0.1%
5 11.114                    43      94     0.4%       254           0    0.1%
6 11.115                    39      98     0.4%       246           0    0.1%
7 11.116                    13     161     0.2%        41           0    0.0%
```

Table 10–8 Network Screen—Parameter Definitions

Parameter	Description
Ethernet address	The DEMNA's actual physical address (APA).
Date and time	The current date and time.
usecs	The length of the last interval (in microseconds) for which the following network parameters were recorded: Pks/Sec, Byt/Pk, %NI-Cur. If only one user is accessing the Network screen, the interval should be close to the nominal 3 seconds. However, if more than one user is accessing the Network screen, the interval may vary significantly from nominal.
%NI	The percentage of maximum Ethernet bandwidth (including preamble, header,[1] user data, CRC, and interpacket gaps) consumed by all nodes on the network during the last interval.
Time	The cumulative time (in seconds) for which the following network parameters have been recorded: Packets, Bytes (k), %NI-Tot.
%NI	The percentage of maximum Ethernet bandwidth (including preamble, header,[1] user data, CRC, and interpacket gaps) consumed by all nodes on the network during the cumulative time indicated in the Time field.
#	For the User column: the six network users that generated the most network traffic (reads and/or writes) during the last recording interval. The users are ranked in descending order. For the Nodes column: the seven nodes that generated the most network traffic (reads and/or writes) during the last recording interval. The nodes are ranked in descending order. When the Interval Network screen is displayed, the users and nodes are ranked according to bytes/sec (Pks/Sec * Byt/Pk) for the interval indicated by the usecs field. (Bytes/sec does not include preamble or interpacket gaps.) When the Accumulated Network screen is displayed, the users and nodes are ranked according to kilobytes (Bytes (k)) for the cumulative time indicated by the Time field. (Bytes (k) does not include preamble or interpacket gaps.)

[1]The header consists of the destination address, the source address, and the protocol type or 802 length field.

Table 10–8 (Cont.) Network Screen—Parameter Definitions

Parameter	Description
User	The user designator (protocol type, SAP, or SNAP SAP protocol identifier) for the six most active network users. A user designator is followed by the user name. A user name is supplied only for the following Ethernet users: ARP, BIOS, Bridge, DECnet, Diag, DName, DTime, ELN, Encry, Lat, LAST, Loop, MopRC, NISca, User. The 802 SNAP SAP user is identified by an asterisk (*) after the SNAP SAP protocol identifier.
Nodes	The DECnet address or Ethernet address for the seven most active network nodes.
Pks/Sec	The average number of packets transmitted or received per second per user or per node.
Byt/Pk	The average number of bytes (header[1] + user data + CRC) transmitted or received per user or per node.
%NI-Cur	The percentage of maximum Ethernet bandwidth (including preamble, header,[1] user data, CRC, and interpacket gaps) consumed by each user or node on the network during the last interval indicated by the usecs field.
Packets	The cumulative number of packets transmitted or received per user or per node.
Bytes(k)	The cumulative number of kilobytes (header[1] + user data + CRC) transmitted or received per user or per node.
%NI-Tot	The per-user or per-node percentage of maximum Ethernet bandwidth (including preamble, header,[1] user data, CRC, and interpacket gaps) consumed during the cumulative time indicated by the Time field.

[1] The header consists of the destination address, the source address, and the protocol type or 802 length field.

8

```
XNA>SHOW STATUS

-- 28.110            -- Status -- 25-OCT-1989 16:17:01 -- Uptime:   06.18:40

-- NI Statistics -------  -- NI Counters ---------  --Process---   --XMI---
Bytes/Pk ............ 85  BytesSnt ...... 46271442  Null   97.6%   0   0.0%
Bytes/Xmt .......... 103  BytesRcv ...... 43236506  Port    0.0%   1   3.3%
Bytes/Rcv ........... 72  MBytesSnt ....... 582131  Xmt-Ln  0.3%   2   5.0%
Pk/Sec .............. 34  MBytesRcv ...... 6751116  Xmt-Hs  0.0%   3   0.3%
Xmt/Sec ............. 13  PkSnt ........... 404207  Rcv-Ln  2.0%   4   1.6%
Rcv/Sec ............. 20  PkRcv ........... 526404  Rcv-Hs  0.0%   5   5.3%
MBaudRate .... 0.012362   MPkSnt ............ 9479  Cmd-Hs  0.0%   6   0.0%
Interrupts ...... 849302  MPkRcv .......... 106455  Mon     0.0%   7   0.0%
Interrupts/sec ...... 0                             Cons    0.0%   8   0.0%
                                                                   9   0.0%
-- Total NI Traffic ---   -- Error Summary -------  --Buffers--    A   0.0%
Bytes/Pk ........... 107  Xmt/Wire ............ 0  Rcv .... 0     B   0.0%
Pk/Sec ............. 595  Rcv/Wire ............ 0  Xmt .... 0     C   0.0%
ThisNI + Other = TotBaud  Rcv/Validation ...... 0                  D   0.0%
  0.2% +  5.7% =    6.0%  Rcv/NoBuffers ..... 1346  --XNA Bus---   E  84.3%
                                                    LANCE   1.6%   F   0.0%
                                                    XNAGA   0.0%
```

Table 10–9 Status Screen—Parameter Definitions

Parameter	Description
Ethernet address	The DEMNA's actual physical address (APA).

Table 10-9 (Cont.) Status Screen—Parameter Definitions

Parameter	Description
Date and time	The current date and time.
Uptime	The time since the DEMNA was last reset, expressed in hours, minutes, and seconds.
	NI Statistics
Bytes/Pk	The average number of bytes per packet (transmit or receive) during the last 3 seconds. This number includes header,[1] user data, and CRC bytes.
Bytes/Xmt	The average number of bytes per transmit packet during the last 3 seconds. This number includes header,[1] user data and CRC bytes.
Bytes/Rcv	The average number of bytes per receive packet during the last 3 seconds. This number includes header,[1] user data, and CRC bytes.
Pk/Sec	The number of packets transmitted and received per second during the last 3 seconds.
Xmt/Sec	The number of packets transmitted per second during the last 3 seconds.
Rcv/Sec	The number of packets received per second during the last 3 seconds.
MBaudRate	The megabaud rate for the DEMNA (transmits plus receives) during the last 3 seconds. The baud rate is calculated for header,[1] user data, and CRC bytes but not interpacket gaps or preamble.
Interrupts	The number of DEMNA-generated interrupts, including both error and port interrupts.
Interrupts/sec	The number of DEMNA-generated interrupts that occurred during the last 3 seconds.
	Total NI Traffic
Bytes/Pk	The average number of bytes per packet on the network during the last 3 seconds. This number includes header,[1] user data, and CRC bytes.
Pk/Sec	The average number of packets per second on the network during the last 3 seconds. This number includes all packets on the network.
ThisNI	The percentage of network bandwidth (preamble, header,[1] user data, CRC, and interpacket gaps) consumed by DEMNA-related traffic (transmits and receives) during the last 3 seconds.
Other	The percentage of network bandwidth (preamble, header,[1] user data, CRC, interpacket gaps) consumed by traffic (transmits and receives) related to other nodes during the last 3 seconds.
TotBaud	The sum of ThisNI and Other, which is equal to the percentage of network bandwidth (preamble, header,[1] user data, CRC, and interpacket gaps) consumed by all nodes during the last 3 seconds.
	NI Counters
BytesRcv	The number of user data bytes received without error. This number does not include header or CRC bytes.
BytesSnt	The number of user data bytes transmitted without error. This number does not include header or CRC bytes.
MBytesRcv	The number of user data bytes in multicast packets received without error. This number does not include header or CRC bytes.

[1] The header consists of the destination address, the source address, and the protocol type or 802 length field.

Table 10-9 (Cont.) Status Screen—Parameter Definitions

Parameter	Description
MBytesSnt	The number of user data bytes in multicast packets transmitted without error. This number does not include header or CRC bytes.
PkSnt	The number of packets transmitted without error. This number includes all Xmt/Def packets (packets transmitted successfully after transmission was deferred because of Ethernet traffic), all Xmt/One packets (packets transmitted without error after a single collision-and-backoff sequence), and all Xmt/Mul packets (packets transmitted successfully on the third or subsequent attempt).
PkRcv	The number of packets received without error. This includes those packets that have passed all the port's filtering and validation checks, as well as MOP packets and loopback packets addressed to the DEMNA.
MPkSnt	The number of multicast packets transmitted without error. This number includes all Xmt/Def multicast packets (multicast packets transmitted successfully after transmission was deferred because of Ethernet traffic), all Xmt/One multicast packets (multicast packets transmitted without error after a single collision-and-backoff sequence), and all Xmt/Mul multicast packets (multicast packets transmitted successfully on the third or subsequent attempt).
MpkRcv	The number of multicast packets received without error. This includes those multicast packets that have passed all the port's filtering and validation checks, as well as MOP packets and loopback packets addressed to the DEMNA.
	Error Summary
Xmt/Wire	The sum of the following transmit errors: maximum number of retries exceeded (Rtry), lost carrier (LCar), late collision (LCol), maximum length exceeded (MLen), collision check test (CTest), transmit timeout (Timeout).
Rcv/Wire	The sum of the following receive errors: CRC error (Crc), framing error (Frame), maximum length exceeded (MLen) error, and invalid (Invalid) error.
Rcv/Validation	The number of receive packets that had one or more of the following filtering/validation errors: the packet was longer than the maximum size requested by the destination user (SizeFilter), the packet had a multicast source address (SrcMCA), an 802 packet was longer than the length implied by the packet's Length field (Long802), an 802 packet was shorter than the length implied by the packet's Length field (Short802), a nonmulticast packet was addressed to the port's physical address but had a user designator (protocol type, DSAP, SSAP, or protocol identifier) not recognized by the port (Urld).
Rcv/NoBuffers	The number of receive packets discarded due to one or more of the following resource errors: a system buffer was unavailable within 3 seconds or no internal buffering was available (Sbua), no user buffer was available (Ubua), the DEMNA port looked for but did not obtain a system buffer (NoRcvBuff), there was no available receive buffer in DEMNA memory (Missed), the DEMNA hardware or firmware was unable to keep up with the data rate (DOR).
	Process Statistics
Null	The percentage of CVAX time used by the kernel and/or scheduler (collectively called the Null process) in the last 3 seconds. The time spent in the null process is idle time.
Port	The percentage of CVAX time used by the Port firmware process in the last 3 seconds.

Table 10-9 (Cont.) Status Screen—Parameter Definitions

Parameter	Description
Xmt-Ln	The percentage of CVAX time used by the LanceXmt firmware process in the last 3 seconds.
Xmt-Hs	The percentage of CVAX time used by the HostXmt firmware process in the last 3 seconds.
Rcv-Ln	The percentage of CVAX time used by the LanceRcv firmware process in the last 3 seconds.
Rcv-Hs	The percentage of CVAX time used by the HostRcv firmware process in the last 3 seconds.
Cmd-Hs	The percentage of CVAX time used by the Command firmware process in the last 3 seconds.
Mon	The percentage of CVAX time used by the Monitor firmware process in the last 3 seconds.
Cons	The percentage of CVAX time used by the Console firmware process in the last 3 seconds.
Buffers In Use	
Rcv	The number of DEMNA-internal receive buffers currently in use. Maximum number of receive buffers = 826.
Xmt	The number of DEMNA-internal transmit buffers currently in use. Maximum number of transmit buffers = 32.
XNA Bus	
LANCE	The percentage of total XNA memory bus traffic generated by the LANCE during the last 3 seconds.
XNAGA	The percentage of total XNA memory bus traffic generated by the DEMNA gate array during the last 3 seconds.

Table 10–9 (Cont.) Status Screen—Parameter Definitions

Parameter	Description
	XMI
0 ... F	The percentage of existing XMI bus traffic generated by the corresponding XMI node (0–F) during the last 3 seconds. The combined percentages for all nodes should nominally add up to 100%. Note that these percentages pertain to existing XMI bus traffic, not to the maximum possible XMI bus traffic. For example, if XNA% = 8, then the DEMNA was consuming 8% of the existing XMI bus traffic during the 3-second interval. To roughly determine the percentage of maximum possible XMI bus traffic consumed by all XMI nodes during a 3-second interval, use the following formula: $$\frac{Pk/Sec \cdot (24 + Bytes/Pk)}{XNA\ Percentage \cdot 10^4} = XMI\ Percentage$$ For example, if Pk/Sec = 1,000, Bytes/Pk = 500, and XNA% = 8, then the total XMI bus traffic consumed by all XMI nodes during the 3-second interval was as follows: $$\frac{1000 \cdot (24 + 500)}{8 \cdot 10^4} = 6.55$$ The DEMNA thus consumed .524 % (8% of 6.55%) of the maximum possible XMI bus traffic during the 3-second interval.

5

```
XNA>SHOW STATUS/ERROR

-- 08-00-2B-00-00-01 -- Status -- 01-JAN-1988 19:02:06 -- Uptime:    19:02:06

-- Rcv Counters --------    -- Xmt Counters --------    -- Lance Counters ------
BytesRcv .... 6327084034    BytesSnt .... 6327280698    Lan/Restart .........  0
PkRcv ......... 17462507    PkSnt ......... 17465275    Lan/UOflo ...........  0
Rcv/MCAUrfd ..... 114323    Xmt/Def ............ 769    Lan/TRxoff ..........  0
Rcv/SizeFilter ......  0    Xmt/One ............ 123    Lan/Merr ............  0
Rcv/SrcMCA ..........  0    Xmt/Mul ............ 132    Lan/TxRx ............  0
Misc/Cntl ...........  0    Xmt/Rtry ...........  0     Rcv/Buffer .......... 19
Rcv/Invalid .........  8    Xmt/LCar ...........  0     Rcv/NoSTP ...........  0
Rcv/Short802 ........  4    Xmt/LCol ...........  0     -- Misc Counters ------
Rcv/Long802 ........ 136    Xmt/MLen ...........  0     Err/HostXfer ........  0
Rcv/Missed .......... 24    Xmt/CTest ..........  0     RX/NoRxBuf ..........  0
Rcv/Dor ............. 24    Xmt/Timeout ........  0     RX/XmtRngFull .......  0
Rcv/NoRcvBuf ........  0    --------------- Saved Error Data ---------------
Rcv/Stale ...........  0    Rtry at ........... None    LCol at ........... None
Rcv/Ubua ............  0    LCar at ........... None    CTst at ........... None
Rcv/Sbua ............  0    Sbua at ........... None
Rcv/Crc+Frame........  0    Crc  at ........... None
Rcv/MLen ............  0    MLen at ........... None
Rcv/Urfd .......... 2686    01-JAN-1988 19:02:05  60-02 MopRC  11.111
```

Table 10–10 Status/Error Screen—Parameter Definitions

Parameter	Description
Ethernet Address	The DEMNA's actual physical address (APA).
Date and Time	The current date and time.
Uptime	The time since the DEMNA was last reset, expressed in hours, minutes, and seconds.
Rcv Counters	
BytesRcv	The number of user data bytes received without error. This number does not include header or CRC bytes.
PkRcv	The number of packets received without error. This number includes those packets that have passed all the port's filtering and validation checks, as well as MOP packets and loopback packets addressed to the DEMNA.
Rcv/MCAUrfd	The number of multicast packets discarded because the packet's user designator (protocol type, DSAP, SSAP, or SNAP SAP protocol identifier) was not enabled for any of the users defined to the port.
SizeFilter	The number of receive packets longer than the maximum size requested by the destination user.
Rcv/SrcMCA	The number of packets received with multicast source addresses. A multicast address is illegal for source addresses.
Misc/Cnt1	Miscellaneous counter 1. An unused counter reserved for future use.
Rcv/Invalid	The number of 802 receive packets that were so short that nothing else could be determined about the packet.
Rcv/Short802	The number of 802 packets whose length was shorter than that implied by the packet's Length field.
Rcv/Long802	The number of 802 packets whose length was longer than that implied by the packet's Length field.

Table 10–10 (Cont.) Status/Error Screen—Parameter Definitions

Parameter	Description
Rcv/Missed	The number of times the LANCE reported a missed error. A missed error occurs when the LANCE discards one or more packets because no receive buffer is available in DEMNA memory.
Rcv/Dor	The number of receive packets discarded by the firmware because the DEMNA hardware or firmware was unable to keep up with the data rate.
NoRcvBuf	The number of times the port looked for but did not obtain a system buffer.
Rcv/Stale	The number of receive packets discarded because a system buffer was unavailable (SBUA) within 3 seconds.
Rcv/Ubua	The number of receive packets discarded because a user buffer was unavailable (UBUA). This counter is maintained by the port driver.
Rcv/Sbua	The number of receive packets discarded by the firmware because a system buffer was unavailable.
Rcv/Crc+Frame	The number of receive packets that had either a CRC error (Rcv/Crc) or a framing error (Rcv/Frame). A CRC error occurs when the packet CRC calculated by the LANCE does not match the CRC value specified in the packet. A framing error occurs when a packet was not framed on a byte boundary.
Rcv/MLen	The number of Ethernet receive packets whose length is longer than 1518 bytes.
Rcv/Urfd	The number of nonmulticast receive packets discarded because the user designator (protocol type, DSAP, SSAP, or protocol identifier) was not recognized by the port. (The packet was, however, addressed to a physical address enabled by the port.)
	Xmt Counters
BytesSnt	The number of user data bytes transmitted without error. This number does not include header or CRC bytes.
PkSnt	The number of packets transmitted without error. This number includes all Xmt/Def packets (packets transmitted successfully after transmission was deferred because of Ethernet traffic), all Xmt/One packets (packets transmitted without error after a single collision-and-backoff sequence), and all Xmt/Mul packets (packets transmitted successfully on the third or subsequent attempt).
Xmt/Def	The number of packets transmitted without error after transmission is delayed once. (The packet is transmitted successfully on the second attempt.)
Xmt/One	The number of packets transmitted without error after a single collision-and-backoff sequence. (The packet is transmitted successfully on the second attempt.)
Xmt/Mul	The number of packets transmitted without error after more than one collision-and-backoff sequence. (The packet is transmitted successfully on the third or subsequent attempt.)
Xmt/Rtry	The number of packets that could not be transmitted because the maximum number of transmission retries (16) was exceeded.
Xmt/LCar	The number of packets that failed transmission because the LANCE could not detect the carrier during transmission.
Xmt/LCol	The number of packets that failed transmission because of a late collision.
Xmt/MLen	The number of packets that failed transmission because the total packet length was longer than the maximum allowable size.

Table 10–10 (Cont.) Status/Error Screen—Parameter Definitions

Parameter	Description
Xmt/CTest	The number of times the LANCE did not detect the Collision Detect signal generated by the Ethernet transceiver to which the DEMNA is connected.
Xmt/Timeout	The number of times the LANCE failed to complete transmission of a packet within 800 milliseconds.
LANCE Counters	
Lan/Restart	The number of times the DEMNA firmware restarted the LANCE.
Lan/UOflo	The number of transmit underflow errors plus the number of receive overflow errors detected by the LANCE.
Lan/TRxoff	The number of times the firmware noticed that the LANCE transceiver or receiver was turned off when it should have been on.
Lan/Merr	The number of memory errors detected by the LANCE.
Lan/TxRx	The number of nonloopback receive packets whose source address is the same as the DEMNA's actual physical address (APA).
Rcv/Buffer	The number of times that the LANCE reported a buffer error in a receive buffer descriptor.
Lan/NoSTP	The number of buffer descriptors that did not have a start-of-packet indicator when the LANCE expected such descriptors.
Miscellaneous Counters	
Err/HostXfer	The number of transfer errors that occurred during a transfer to or from host memory.
RX/NoRxBuf	The number of packets that could not be transmitted in response to a MOP or loopback message because no LANCE transmit buffers were available.
RX/XmtRngFull	The number of packets that could not be transmitted in response to a MOP or loopback message because no LANCE transmit ring entries were available.
Saved Error Data	
Rtry at	The date and time at which the last Xmt/Rtry error occurred.
LCar at	The date and time at which the last Xmt/LCar error occurred.
Sbua at	The date and time at which the last Rcv/Sbua error occurred, followed by the user designator (protocol type, SAP, or SNAP SAP protocol identifier), name, and address (DECnet or Ethernet) of the node that sent the packet. The 802 SNAP SAP user is identified by an asterisk (*) after the SNAP SAP protocol identifier.
Crc at	The date and time at which the last Rcv/Crc error occurred, followed by the user designator (protocol type, SAP, or SNAP SAP protocol identifier), name, and address (DECnet or Ethernet) of the node that sent the packet. The 802 SNAP SAP user is identified by an asterisk (*) after the SNAP SAP protocol identifier. The "Crc at" field records bad-CRC packets even if such packets are not addressed to the DEMNA. The Rcv/Crc+Frame counter, however, records only packets that are addressed to the DEMNA. Thus, if the "CRC at" field displays information but no error is recorded in the Rcv/Crc+Frame counter, the detected bad-CRC packet was not addressed to the DEMNA.

Table 10-10 (Cont.) Status/Error Screen—Parameter Definitions

Parameter	Description
Mlen at	The date and time at which the last Rcv/Mlen error occurred, followed by the user designator (protocol type, SAP, or SNAP SAP protocol identifier), name, and address (DECnet or Ethernet) of the node that sent the packet. The 802 SNAP SAP user is identified by an asterisk (*) after the SNAP SAP protocol identifier.
Urld at	The date and time at which the last Rcv/Urld error occurred, followed by the user designator (protocol type, SAP, or SNAP SAP protocol identifier), name, and address (DECnet or Ethernet) of the node that sent the packet. The 802 SNAP SAP user is identified by an asterisk (*) after the SNAP SAP protocol identifier.
LCol at	The date and time at which the last Xmt/LCol error occurred.
CTst at	The date and time at which the last Xmt/CTst error occurred.

10

```
XNA>SHOW STATUS/INTERVAL

-- 28 101               -- Status -- 25-OCT-1989 [illegible]:17:01 -- Uptime:   06:18:40
                               -- Interval Counters --
-- NI Statistics -------   -- NI Counters ---------   --Process---   --XMI---
Bytes/Pk ............ 85   BytesSnt ...... 46271442   Null   97.6%   0   0.0%
Bytes/Xmt .......... 103   BytesRcv ...... 43236506   Port    0.0%   1   3.3%
Bytes/Rcv ........... 72   MBytesSnt ....... 582131   Xmt-Ln  0.3%   2   5.0%
Pk/Sec .............. 34   MBytesRcv ...... 6751116   Xmt-Hs  0.0%   3   0.3%
Xmt/Sec ............. 13   PkSnt ........... 404207   Rcv-Ln  2.0%   4   1.6%
Rcv/Sec ............. 20   PkRcv ........... 526404   Rcv-Hs  0.0%   5   5.3%
MBaudRate .... 0.012362    MPkSnt ............ 9479   Cmd-Hs  0.0%   6   0.0%
Interrupts ...... 849302   MPkRcv .......... 106455   Mon     0.0%   7   0.0%
Interrupts/sec ...... 0                               Cons    0.0%   8   0.0%
                                                                     9   0.0%
-- Total NI Traffic ---    -- Error Summary -------   --Buffers--    A   0.0%
Bytes/Pk ........... 107   Xmt/Wire ............ 0   Rcv .... 0     B   0.0%
Pk/Sec ............. 595   Rcv/Wire ............ 0   Xmt .... 0     C   0.0%
ThisNI + Other = TotBaud   Rcv/Validation ...... 0                  D   0.0%
  0.2% +  5.7% =    6.0%   Rcv/NoBuffers ..... 1346   --XNA Bus---   E  84.3%
                                                      LANCE   1.6%   F   0.0%
                                                      XNAGA   0.0%
```

The Status/Interval screen displays the same screen as the SHOW STATUS command. The only difference between the two screens is the time interval for which the NI counters and the Error Summary counters record events. In the Show Status screen, these counters record events from the last reset of the DEMNA module. In the Show Status/Interval screen, these counters record events starting when the SHOW STATUS/INTERVAL command is entered.

11

```
XNA>SHOW USER

-- 08-00-2B-00-00-01 -- User Data -- 01-JAN-1988 00:24:14 --

 #  Ptt/Sap/PI      MaxX  MinX  MaxR  Strt  Stop  Size Pad Cls Prm Bdc Unk Amc
 -  -------------   ----  ----  ----  ----  ----  ---- --- --- --- --- --- ---
 0  Eth-80-00       1514    14  1518     1     0    N   N   -   N   N   N   N
 1  802SAP-22       1514    14  1518     1     0    Y   N   1   N   N   N   N
 2  08-00-00-00-03  1514    14  1518     1     0    Y   N   1   N   N   N   N
```

Table 10–11 User Screen—Parameter Definitions

Parameter	Description
Ethernet Address	The DEMNA's actual physical address (APA).
Date and Time	The current date and time.
#	The user index.
Ptt/Sap/PI	The user designator (protocol type, SAP, or SNAP SAP protocol identifier) for the user. *Eth* indicates an Ethernet user. (See Appendix I for the most commonly used Ethernet protocol types.) *802SAP* indicates an 802 SAP user. Five 2-digit sets of hex numbers (xx-xx-xx-xx-xx) indicate an 802 SNAP SAP user.
MaxX	The maximum allowable size for transmit packets.
MinX	The minimum allowable size for transmit packets.
MaxR	The maximum receive size specified for the user.
Strt	The number of times since the last power-up or reset that the user has been started by the firmware.
Stop	The number of times since the last power-up or reset that the user has been stopped by the firmware.
Size	Indicates whether the DEMNA port expects receive packets sent to the user to have a Length field. The port uses this field to validate the packet length. Y = yes; N = no.
Pad	Indicates whether padding is enabled for the user: Y = yes; N = no.
Cls	Indicates the class (1 or 2) of an 802 user. Valid only for an 802 user.
Prm	Indicates whether the user is operating in promiscuous mode: Y = yes; N = no.
Bdc	Indicates whether the user accepts broadcast packets: Y = yes; N = no.
Unk	Indicates whether the user is the unknown user: Y = yes; N = no.
Amc	Indicates whether the user accepts packets addressed to *any* multicast address: Y = yes; N = no.

12

```
XPA>SHOW XPUD

-- 08-00-2B-00-00-01 -- Powerup Diagnostic Register -- 01-JAN-1988 01:09:35 --

Actual XPUD = FFFFC007                         Expected good value = FFFFC007

Bit     Set if...                              Bit     Set if...
---     -----------------------------          ---     -----------------------------
0  - Firmware init complete                    16 - LANCE Test passed
1  - LANCE External Loopback                   17 - Shared Parity RAM Test passed
2  - EEPROM loaded into RAM                    18 - Shared RAM March Test passed
3  - EPROM loaded into RAM                     19 - XNADAL Timeout Logic Test passed
4  - Diagnostic patch table is bad             20 - XNADAL Readback Test passed
5  - EEPROM error history exists               21 - EEPROM Test passed
6  - Reserved                                  22 - ENET PROM Test passed
7  - Reserved                                  23 - CVAX Chip Test passed
8  - Reserved                                  24 - CVAX Parity RAM Test passed
9  - Reserved                                  25 - CVAX RAM March Test passed
10 - Reserved                                  26 - Console Drivers Test passed
11 - Reserved                                  27 - SSC Test passed
12 - Reserved                                  28 - Diagnostic Register test passed
13 - Reserved                                  29 - CVAX Interrupt Lines Test passed
14 - XNAGA Test passed                         30 - Boot Rom Test passed
15 - Eth Subsystem Parity Test passed          31 - Self Test Complete
```

Table 10–12 XPUD Screen—Parameter Definitions

Parameter	Description
Date and Time	The current date and time.
Actual XPUD	The Power-Up Diagnostic (XPUD) Register contents in hexadecimal.
Expected good value	The hexadecimal value that the XPUD Register should contain if all DEMNA self-test routines passed. Note that the expected good value can also be FFFFC027 if there is an entry in the EEPROM error history.
bit 0 ... bit 31	Explanations of all significant bits in the XPUD Register.

T/R

The T/R command invokes the DEMNA diagnostic monitor, from which the DEMNA ROM-based diagnostics (RBDs) can be run. The diagnostic monitor displays the following prompt: RBD*n*, where *n* is the DEMNA's XMI node number.

This command is valid only under the following circumstances:

- When entered from the physical console attached directly to the DEMNA module
- When the DEMNA is in the uninitialized state. To put the DEMNA in the uninitialized state, stop all applications that are using the DEMNA or reset the system with auto start off.

The command cannot be used from a networked terminal (either local or remote).

FORMAT T/R

EXAMPLE

```
XNA>T/R


RBD3>
```

10.6 Console Command Language Control Characters

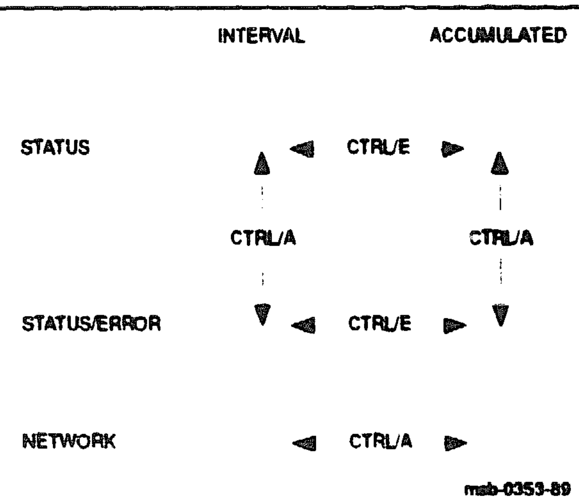
Six ASCII control characters have special meaning to the DEMNA console monitor program. Table 10–13 describes these control characters.

Table 10–13 Console Control Characters

Character	Function
CTRL/A	Alternates between a Status screen and a Status/Error screen or between an Interval Network screen and an Accumulated Network screen.
CTRL/D	Disconnects the console and exits to the system prompt. Has no effect when used on the DEMNA's physical console.
CTRL/E	Alternates between the Status screen and the Status/Interval screen or between the Interval Status/Error screen and the Accumulated Status/Error screen. Valid only when one of these screens is being displayed. If such a screen is not displayed, entering this control character invokes a Status screen.
CTRL/L	Retrieves the last console command line that was entered. The last four command lines can be retrieved.
CTRL/U	Clears the current console command line.
CTRL/W	Ref[illegible]es the screen during the display of a Status, Status/Error, Status/Interval, or Network screen. If entered when such a screen is not being displayed, clears the screen and displays a Status screen.

Figure 10–1 shows how CTRL/A and CTRL/E can be used to go from one screen to another.

Figure 10–1 Using Control Charac[illegible]s to Switch Scr[illegible]ens



10.7 How to Use the Status Screens

The three Status screens accessible through the SHOW command convey a great deal of information that may not be readily apparent. The best way to use these screens is to look at *sets* of counters rather than at individual counters in isolation. This section describes various sets of Status screen counters and suggests how to use them as diagnostic and network management tools.

In general, the Status screen counters can be divided into two main groups:

- Counters that convey error information
- Counters that indicate how resources are being used

The sets of counters within these two main groups are described below. A set of counters is described by showing a picture of the appropriate Status screen(s) with the relevant counters highlighted.

10.7.1 Error Information

Table 10–14 describes the types of errors monitored by the Status screen counters, as well as additional error information provided by these counters. There is one set of counters for each error type or class of supplemental error information.

Table 10–14 Error Information Provided by Status Screen Counters

Error Type/Error Information	Description
Ethernet errors	Errors that occur because of problems on the Ethernet wire. Examples of such errors include late collisions on transmits and CRC errors on receives.
Packet filtering/validation errors	Errors that the port discovers when filtering and performing validation checks on transmit and receive packets. Such errors include frame length errors and packets that do not filter to enabled port users.
Lack-of-resource errors	Errors that occur when there are insufficient resources to process valid receive packets. Such errors include data overrun errors and a lack of user buffers.
Time/user information	Time and/or user information pertinent to particular errors or to overall DEMNA operation.
LANCE errors	Errors recorded by the LANCE chip.
Firmware debug information	Information useful only for debugging the DEMNA firmware.

Notice that the first three error types listed above—Ethernet errors, packet filtering/validation errors, and lack-of-resource errors—trace the progress of a packet through the three major segments of a receive operation:

1. Receiving the packet from the Ethernet
2. Filtering and validating the packet
3. Transferring the packet to the port driver and higher-level users

10.7.1.1 Ethernet Error Counters

These counters record errors resulting from problems on the Ethernet wire. The Xmt/Wire and Rcv/Wire counters in the Status and Status/Interval screens are error summary counters that respectively indicate the total number of transmit and receive errors recorded by the highlighted counters in the Status/Error screen.

NOTE: The Xmt/Wire counter records Xmt/Mlen errors even though such errors are validation errors rather than Ethernet errors. The Xmt/Wire counter is used to record Xmt/Mlen errors because it is the only summary counter available for transmit errors.

Status and Status/Interval Screens

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:01:19 -- Uptime:   01:43:35

-- NI Statistics -------   -- NI Counters ---------   -- Process --   --XMI---
Bytes/Pk ............ 64  BytesSnt .... 6327255447  Null    88.0%   0   0.0%
Bytes/Xmt ........... 64  BytesRcv .... 6327084034  Port     0.7%   1   0.0%
Bytes/Rcv ........... 64  MbytesSnt ........ 20470  Xmt-Ln   2.8%   2   0.0%
Pk/Sec ............. 510  MbytesRcv ........... 0  Xmt-Hs   2.0%   3   0.0%
Xmt/Sec ............ 255  PkSnt ......... 17464886  Rcv-Ln   1.6%   4   0.0%
Rcv/Sec ............ 255  PkRcv ......... 17462507  Rcv-Hs   3.0%   5   0.0%
MBaudRate ..... 0.274471  MPkSnt ............. 230  Com-Hs   0.0%   6   0.0%
Interrupts ... 104599634  MPkRcv ............... 0  Mon      1.0%   7   0.0%
Interrupts/Sec ....  255                            Cons     0.4%   8   0.0%
                                                                    9   0.0%
-- Total NI Traffic ----  -- Error Summary -------  --Buffers--     A   0.0%
Bytes/Pk ........... 237  |Xmt/Wire ........... 0|  Rcv ....  0     B   0.0%
Pk/Sec ............ 1418  |Rcv/Wire ........... 0|  Xmt ....  1     C  11.7%
ThisNI + Other = TotBaud  Rcv/Validation ...... 1                   D   0.0%
  3.5% + 26.0% =   29.5%  Rcv/NoBuffers ....... 0  --XNA Bus---    E  89.3%
                                                    LANCE   9.3%    F   0.0%
                                                    XNAGA   0.0%
```

Status/Error Screen

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:40:45 -- Uptime:    2:23:01

-- Rcv Counters --------    -- Xmt Counters --------   -- Lance Counters ------
BytesRcv .... 6327084034    BytesSnt .... 6327280698   Lan/Restart .........  0
PkRcv ......... 17462507    PkSnt ......... 17465275   Lan/UOflo ...........  0
Rcv/MCAUrfd .........  0    Xmt/Def ............ 769   Lan/TRxoff ..........  0
Rcv/SizeFilter ......  0    Xmt/One ............ 123   Lan/Merr ............  0
Rcv/SrcMCA ..........  0    Xmt/Mul ............ 132   Lan Tx/Rx ...........  0
Misc/Cntl ...........  0   |Xmt/Rtry ............. 0|  Rcv/Buffer ..........  0
|Rcv/Invalid ........ 0|   |Xmt/LCar ............. 0|  Rcv/NoSTP ...........  0
Rcv/Short802 ........  0   |Xmt/LCol ............. 0|  -- Misc Counters ------
Rcv/Long802 .........  0    Xmt/MLen ............. 0   Err/HostXfer ........  0
Rcv/Missed ..........  0   |Xmt/CTest ............ 0|  RX/NoRxBuf ..........  0
Rcv/Dor .............  0   |Xmt/Timeout .......... 0|  RX/XmtRngFull .......  0
Rcv/NoRcvBuf ........  0    ---------------- Saved Error Data ----------------
Rcv/Stale ...........  0    Rtry at ............ None  LCol at ........... None
Rcv/Ubua ............  0    LCar at ............ None  CTst at ........... None
Rcv/Sbua ............  0    Sbua at ............ None
|Rcv/Crc+Frame ...... 0|    Crc  at ............ None
|Rcv/Mlen ........... 0|    Mlen at ............ None
Rcv/Urfd ............  1    01-AUG-1989 08:02:05  60-02 MopRC 11.111
```

msb-0329-89

10.7.1.2 Filtering/Validation Error Counters

These counters record filtering and validation errors detected by the DEMNA port. The Rcv/Validation counter in the Status and Status/Interval screens indicates the total number of filtering and validation errors recorded by the highlighted counters in the Status/Error screen.

NOTE: **Although Xmt/Mlen errors are validation errors, they are recorded in the Xmt/Wire counter in the Status and Status/Interval screens since Xmt/Wire is the only summary counter for transmit errors.**

Status and Status/Interval Screens

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:01:19 -- Uptime:   01:43:35

-- NI Statistics -------   -- NI Counters ---------  -- Process --    --XMI---
Bytes/Pk ............ 64   BytesSnt .... 6327255447  Null     88.0%   0   0.0%
Bytes/Xmt ........... 64   BytesRcv .... 6327084034  Port      0.7%   1   0.0%
Bytes/Rcv ........... 64   MbytesSnt ........ 20470  Xmt-Ln    2.8%   2   0.0%
Pk/Sec ............. 510   MbytesRcv ............ 0  Xmt-Hs    2.0%   3   0.0%
Xmt/Sec ............ 255   PkSnt ......... 17464886  Rcv-Ln    1.6%   4   0.0%
Rcv/Sec ............ 255   PkRcv ......... 17462507  Rcv-Hs    3.0%   5   0.0%
MBaudRate ..... 0.274471   MPkSnt ............. 230  Com-Hs    0.0%   6   0.0%
Interrupts ... 104599634   MPkRcv ............... 0  Mon       1.0%   7   0.0%
Interrupts/Sec .... 255                              Cons      0.4%   8   0.0%
                                                                      9   0.0%
-- Total NI Traffic ----   -- Error Summary -------  --Buffers--      A   0.0%
Bytes/Pk ........... 237   Xmt/Wire ............ 0   Rcv ....  0      B   0.0%
Pk/Sec ............ 1418   Rcv/Wire ............ 0   Xmt ....  1      C  11.7%
ThisNI + Other = TotBaud   Rcv/Validation ...... 1                    D   0.0%
  3.5% + 26.0% =   29.5%   Rcv/NoBuffers ....... 0   --XNA Bus---     E  89.3%
                                                     LANCE    9.3%    F   0.0%
                                                     XMAGA    0.0%
```

Status/Error Screen

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:40:45 -- Uptime:    2:23:01

-- Rcv Counters --------   -- Xmt Counters --------  -- Lance Counters ------
BytesRcv .... 6327084034   BytesSnt .... 6327280698  Lan/Restart .........  0
PkRcv ......... 17462507   PkSnt ......... 17465275  Lan/UOflo ...........  0
Rcv/MCAUrfd .........  0   Xmt/Def ............ 759  Lan/TRxoff ..........  0
Rcv/SizeFilter ......  0   Xmt/One ............ 123  Lan/Merr ............  0
Rcv/SrcMCA ..........  0   Xmt/Mul ............ 132  Lan Tx/Rx ...........  0
Misc/Cntl ...........  0   Xmt/Rtry ...........   0  Rcv/Buffer ..........  0
Rcv/Invalid .........  0   Xmt/LCar ...........   0  Rcv/NoSTP ...........  0
Rcv/Short802 ........  0   Xmt/LCol ...........   0  -- Misc Counters ------
Rcv/Long802 .........  0   Xmt/MLen ...........   0  Err/HostXfer ........  0
Rcv/Missed ..........  0   Xmt/CTest ..........   0  RX/NoRxBuf ..........  0
Rcv/Dor .............  0   Xmt/Timeout ........   0  RX/XmtRngFull .......  0
Rcv/NoRcvBuf ........  0   ---------------- Saved Error Data ----------------
Rcv/Stale ...........  0   Rtry at ........... None  LCol at ........... None
Rcv/Ubus ............  0   LCar at ........... None  CTst at ........... None
Rcv/Sbus ............  0   Sbus at ........... None
Rcv/Crc+Frame .......  0   Crc  at ........... None
Rcv/Mlen ............  0   Mlen at ........... None
Rcv/Urfd ............  1   01-AUG-1989 08:02:05  60-02 MopRC 11.111
```

mab-0330-89

10.7.1.3 Lack-of-Resource Counters

These counters record errors resulting from insufficient resources for processing valid receive packets. The Rcv/Nobuffers counter in the Status and Status/Interval screens indicates the total number of lack-of-resource errors recorded by the highlighted counters in the Status/Error screen.

NOTE: The Rx/NoRxBuf and Rx/XmtRngFull errors are not recorded in the Rcv/NoBuffers counter even though they are lack-of-resource errors. Rx/NoRxBuf and Rx/XmtRngFull errors pertain only to non-host related packets (specifically, MOP loopback packets).

Status and Status/Interval Screens

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:01:19 -- Uptime:   01:43:35

-- NI Statistics -------   -- NI Counters ---------  -- Process --    --XMI---
Bytes/Pk ............ 64   BytesSnt .... 6327255447  Null     88.0%   0   0.0%
Bytes/Xmt ........... 64   BytesRcv .... 6327084034  Port      0.7%   1   0.0%
Bytes/Rcv ........... 64   MbytesSnt ........ 20470  Xmt-Ln    2.8%   2   0.0%
Pk/Sec ............. 510   MbytesRcv ............ 0  Xmt-Hs    2.0%   3   0.0%
Xmt/Sec ............ 255   PkSnt ......... 17464886  Rcv-Ln    1.6%   4   0.0%
Rcv/Sec ............ 255   PkRcv ......... 17462507  Rcv-Hs    3.0%   5   0.0%
MBaudRate ..... 0.274471   MPkSnt ............. 230  Com-Hs    0.0%   6   0.0%
Interrupts ... 104599634   MPkRcv ............... 0  Mon       1.0%   7   0.0%
Interrupts/Sec ....  255                             Cons      0.4%   8   0.0%
                                                                      9   0.0%
-- Total NI Traffic ----   -- Error Summary -------  --Buffers--      A   0.0%
Bytes/Pk ........... 237   Xmt/Wire ............. 0  Rcv ....  0      B   0.0%
Pk/Sec ............ 1418   Rcv/Wire ............. 0  Xmt ....  1      C  11.7%
ThisNI + Other = TotBaud   Rcv/Validation ....... 1                   D   0.0%
  3.5% + 26.0% =   29.5%   Rcv/NoBuffers ........ 0  --XMA Bus---     E  89.3%
                                                     LANCE    9.3%    F   0.0%
                                                     XNAGA    0.0%
```

Status/Error Screen

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:40:45 -- Uptime:    2:23:01

-- Rcv Counters --------   -- Xmt Counters --------  -- Lance Counters ------
BytesRcv .... 6327084034   BytesSnt .... 6327280698  Lan/Restart .........  0
PkRcv ......... 17462507   PkSnt ......... 17465275  Lan/UOflo ...........  0
Rcv/MCAUrfd .........  0   Xmt/Def ............. 769 Lan/TRxoff ..........  0
Rcv/SizeFilter ......  0   Xmt/One ............. 123 Lan/Merr ............  0
Rcv/SrcMCA ..........  0   Xmt/Mul ............. 132 Lan Tx/Rx ...........  0
Misc/Cntl ...........  0   Xmt/Rtry .............  0 Rcv/Buffer ..........  0
Rcv/Invalid .........  0   Xmt/LCar .............  0 Rcv/NoSTP ...........  0
Rcv/Short802 ........  0   Xmt/LCol .............  0 -- Misc Counters ------
Rcv/Long802 .........  0   Xmt/MLen .............  0 Err/HostXfer ........  0
Rcv/Missed ..........  0   Xmt/CTest ............  0 RX/NoRxBuf ..........  0
Rcv/Dor .............  0   Xmt/Timeout ..........  0 RX/XmtRngFull .......  0
Rcv/NoRcvBuf ........  0   ---------------- Saved Error Data ----------------
Rcv/Stale ...........  0   Rtry at ........... None  LCol at ........... None
Rcv/Ubus ............  0   LCar at ........... None  CTst at ........... None
Rcv/Sbus ............  0   Sbus at ........... None
Rcv/Crc+Frame .......  0   Crc  at ........... None
Rcv/Mlen ............  0   Mlen at ........... None
Rcv/Urfd ............  1   01-AUG-1989 08:02:05  60-02 MopRC 11.111
```

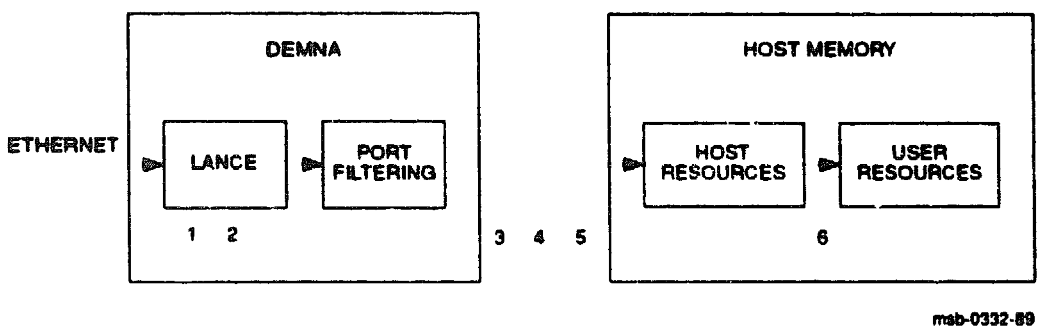
mab-0331-89

As Figure 10–2 indicates, the lack-of-resource error counters are ordered as follows with respect to the data flow of a receive operation:

1 Rcv/Missed—missed packet
2 Rcv/DOR—data overrun
3 Rcv/NoRcvBuf—no receive buffer (potential SBUA)
4 Rcv/Sbua—system buffer unavailable
5 Rcv/Stale—stale packet
6 Rcv/Ubua—user buffer unavailable

Errors 1 and 2, which involve receiving the packet from the Ethernet, are detected by the LANCE chip. Errors 3 through 5, which involve transfer of the packet from the port to the port driver, are detected by the port. Error 6, which involves transfer of the packet from the port driver to the user, is detected by the port driver.

Figure 10–2 Ordering of Lack-of-Resource Errors



10.7.1.4 Time/User Fields

These fields provide time and user information pertinent to particular errors or to overall DEMNA operation. The headers for each screen indicate (among other things) the uptime for the DEMNA module. The time/user fields in the Status/Error screen indicate when the last instance of a particular receive error occurred and which network node transmitted the receive packet.

Status and Status/Interval Screens

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:01:19 -- Uptime:   01:43:35

-- NI Statistics -------     -- NI Counters ---------   -- Process --    --XMI---
Bytes/Pk ............ 64     BytesSnt .... 6327255447   Null     88.0%   0   0.0%
Bytes/Xmt ........... 64     BytesRcv .... 6327084034   Port      0.7%   1   0.0%
Bytes/Rcv ........... 64     MbytesSnt ........ 20470   Xmt-Ln    2.8%   2   0.0%
Pk/Sec ............. 510     MbytesRcv ............ 0   Xmt-Ns    2.0%   3   0.0%
Xmt/Sec ............ 255     PkSnt ......... 17464886   Rcv-Ln    1.6%   4   0.0%
Rcv/Sec ............ 255     PkRcv ......... 17462507   Rcv-Ns    3.0%   5   0.0%
MBaudRate ..... 0.274471     MPkSnt ............. 230   Com-Ns    0.0%   6   0.0%
Interrupts ... 104599634     MPkRcv ............... 0   Mon       1.0%   7   0.0%
Interrupts/Sec ....  255                                Cons      0.4%   8   0.0%
                                                                         9   0.0%
-- Total NI Traffic ----     -- Error Summary -------   --Buffers--      A   0.0%
Bytes/Pk ........... 237     Xmt/Wire ............ 0    Rcv ....  0      B   0.0%
Pk/Sec ............ 1418     Rcv/Wire ............ 0    Xmt ....  1      C  11.7%
ThisNI + Other = TotBaud     Rcv/Validation ...... 1                     D   0.0%
  3.5% + 26.0% =   29.5%     Rcv/NoBuffers ....... 0    --XMA Bus---     E  89.3%
                                                        LANCE    9.3%    F   0.0%
                                                        XNAGA    0.0%
```

Status/Error Screen

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:40:45 -- Uptime:    2:23:01

-- Rcv Counters --------     -- Xmt Counters --------    -- Lance Counters ------
BytesRcv .... 6327084034     BytesSnt .... 6327280698    Lan/Restart .........  0
PkRcv ......... 17462507     PkSnt ......... 17465275    Lan/UOflo ...........  0
Rcv/MCAUrfd .........  0     Xmt/Def ............ 769    Lan/TRxoff ..........  0
Rcv/SizeFilter ......  0     Xmt/One ............ 123    Lan/Merr ............  0
Rcv/SrcMCA ..........  0     Xmt/Mul ............ 132    Lan Tx/Rx ...........  0
Misc/Cntl ...........  0     Xmt/Rtry ............. 0    Rcv/Buffer ..........  0
Rcv/Invalid .........  0     Xmt/LCar ............. 0    Rcv/NoSTP ...........  0
Rcv/Short802 ........  0     Xmt/LCol ............. 0    -- Misc Counters ------
Rcv/Long802 .........  0     Xmt/MLen ............. 0    Err/HostXfer ........  0
Rcv/Missed ..........  0     Xmt/CTest ............ 0    RX/NoRxBuf ..........  0
Rcv/Dor .............  0     Xmt/Timeout .......... 0    RX/XmtRngFull .......  0
Rcv/NoRcvBuf ........  0     ---------------- Saved Error Data ----------------
Rcv/Stale ...........  0     Rtry at ........... None    LCol at ........... None
Rcv/Ubus ............  0     LCar at ........... None    CTst at ........... None
Rcv/Sbus ............  0     Sbus at ........... None
Rcv/Crc+Frame .......  0     Crc  at ........... None
Rcv/Mlen ............  0     Mlen at ........... None
Rcv/Urfd ............  1     01-AUG-1989 08:02:05  60-02 MopRC 11.111
```

msb-0333-89

10.7.1.5 LANCE Counters

These counters, which are copies of LANCE chip counters, record LANCE-related events. The count in the Lan/Restart counter can increase rapidly when the DEMNA's loopback mode (a diagnostic function) is turned on or off or when the port driver enables and disables promiscuous mode for a port user when the Enable Promiscuous Mode flag in DEMNA EEPROM (see Chapter 9) is set to No.

Status/Error Screen

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:40:45 -- Uptime:     2:23:01

-- Rcv Counters --------   -- Xmt Counters --------   -- Lance Counters ------
BytesRcv .... 6327084034   BytesSnt .... 6327280698   Lan/Restart .........  0
PkRcv ......... 17462507   PkSnt ......... 17465275   Lan/UOflo ...........  0
Rcv/MCAUrfd .........  0   Xmt/Def ............. 769   Lan/TRxoff ..........  0
Rcv/SizeFilter ......  0   Xmt/One ............. 123   Lan/Merr ............  0
Rcv/SrcMCA ..........  0   Xmt/Mul ............. 132   Lan Tx/Rx ...........  0
Misc/Cntl ...........  0   Xmt/Rtry ............   0   Rcv/Buffer ..........  0
Rcv/Invalid .........  0   Xmt/LCar ............   0   Rcv/NoSTP ...........  0
Rcv/Short802 ........  0   Xmt/LCol ............   0   -- Misc Counters ------
Rcv/Long802 .........  0   Xmt/MLen ............   0   Err/HostXfer ........  0
Rcv/Missed ..........  0   Xmt/CTest ...........   0   RX/NoRxBuf ..........  0
Rcv/Dor .............  0   Xmt/Timeout .........   0   RX/XmtRegFull .......  0
Rcv/NoRcvBuf ........  0   ---------------- Saved Error Data ---------------
Rcv/Stale ...........  0   Rtry at ........... None  LCol at ........... None
Rcv/Ubus ............  0   LCar at ........... None  CTst at ........... None
Rcv/Sbus ............  0   Sbus at ........... None
Rcv/Crc+Frame .......  0   Crc  at ........... None
Rcv/Mlen ............  0   Mlen at ........... None
Rcv/Urfd ......... .   1   01-AUG-1989 08:02:05  60-02 MopRC 11.111
```

msb-0334-89

10.7.1.6 Firmware Debug Counters

These two counters are useful only for debugging DEMNA firmware and thus do not convey useful information to DEMNA users.

Status/Error Screen

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:40:45 -- Uptime:     ?:23:01

-- Rcv Counters --------     -- Xmt Counters --------     -- Lance Counters ------
BytesRcv .... 6327084034     BytesSnt .... 6327280698     Lan/Restart .........  0
PkRcv ......... 17462507     PkSnt ......... 17465275     Lan/UOflo ...........  0
Rcv/MCAUrfd .........  0     Xmt/Def ............ 769     Lan/TRxoff ..........  0
Rcv/SizeFilter ......  0     Xmt/One ............ 123     Lan/Merr ............  0
Rcv/SrcMCA ..........  0     Xmt/Mul ............ 132     Lan Tx/Rx ...........  0
Misc/Cntl ...........  0     Xmt/Rtry ...........   0     Rcv/Buffer ..........  0
Rcv/Invalid .........  0     Xmt/LCar ...........   0     Rcv/NoSTP ...........  0
Rcv/Short802 ........  0     Xmt/LCol ...........   0     -- Misc Counters ------
Rcv/Long802 .........  0     Xmt/MLen ...........   0     Err/HostXfer ........  0
Rcv/Missed ..........  0     Xmt/CTest ..........   0     RX/NoRxBuf ..........  0
Rcv/Dor .............  0     Xmt/Timeout ........   0     RX/XmtRngFull .......  0
Rcv/NoRcvBuf ........  0     ---------------- Saved Error Data ----------------
Rcv/Stale ...........  0     Rtry at ........... None     LCol at ........... None
Rcv/Ubus ............  0     LCar at ........... None     CTst at ........... None
Rcv/Sbus ............  0     Sbus at ........... None
Rcv/Crc+Frame .......  0     Crc  at ........... None
Rcv/Mlen ............  0     Mlen at ........... None
Rcv/Urfd ............  1     01-AUG-1989 08:02:05  60-02 MopRC 11.111
```

msb-0335-89

10.7.2 Resource Utilization Information

The counters that provide information on resource utilization can be divided into the following two sets:

- Data density counters
- DEMNA resource counters

10.7.2.1 Data Density Counters

These counters provide the following information about the data density on the Ethernet wire:

- The NI Statistics counters in the Status and Status/Interval screens indicate the data density per packet and per second, as well as the total (transmit and receive) baud rate for the DEMNA.
- The NI Counters in the Status and Status/Interval screens indicate the total number of bytes and packets transmitted and received by the DEMNA. Subtotals are provided for multicast packets.
- The Total NI Traffic section of the Status and Status/Interval screens shows how much network traffic the DEMNA and the other network nodes are generating.
- The data density counters in the Status/Error screen indicate the total number of bytes and packets transmitted and received by the DEMNA. Additional Xmt counters indicate how many packets were deferred or transmitted successfully after a collision-and-backoff sequence.

Status and Status/Interval Screens

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:01:19 -- Uptime:   01:43:35

-- NI Statistics -------   -- NI Counters ---------   -- Process --     --XMI---
Bytes/Pk ........... 64    BytesSnt .... 6327255447   Null    88.0%     0   0.0%
Bytes/Xmt .......... 64    BytesRcv .... 6327084034   Port     0.7%     1   0.0%
Bytes/Rcv .......... 64    MbytesSnt ........ 20670   Xmt-Ln   2.8%     2   0.0%
Pk/Sec ............ 510    MbytesRcv ........... 0    Xmt-Hs   2.0%     3   0.0%
Xmt/Sec ........... 255    PkSnt ......... 17464886   Rcv-Ln   1.6%     4   0.0%
Rcv/Sec ........... 255    PkRcv ......... 17462507   Rcv-Hs   3.0%     5   0.0%
%BaudRate ..... 0.274471   MPkSnt ............. 230   Com-Hs   0.0%     6   0.0%
Interrupts ... 104599634   MPkRcv ............... 0   Mon      1.0%     7   0.0%
Interrupts/Sec ....  255                              Cons     0.4%     8   0.0%
                                                                        9   0.0%
-- Total NI Traffic ----   -- Error Summary -------   --Buffers--       A   0.0%
Bytes/Pk .......... 237    Xmt/Wire ............ 0    Rcv ....  0       B   0.0%
Pk/Sec ........... 1418    Rcv/Wire ............ 8    Xmt ....  1       C  11.7%
ThisNI + Other = TotBaud   Rcv/Validation ...... 1                      D   0.0%
  3.5% + 26.0% =   29.5%   Rcv/NoBuffers ....... 0    --XNA Bus---      E  89.3%
                                                      LANCE    9.3%     F   0.0%
                                                      XNAGA    0.0%
```

Status/Error Screen

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:40:45 -- Uptime:    2:23:01

-- Rcv Counters --------   -- Xmt Counters --------   -- Lance Counters ------
BytesRcv .... 6327084034   BytesSnt .... 6327280698   Lan/Restart ........   0
PkRcv ......... 17462507   PkSnt ......... 17465275   Lan/UOflo ..........   0
Rcv/MCAUrfd .........  0   Xmt/Def ............ 769   Lan/TRxoff .........   0
Rcv/SizeFilter ......  0   Xmt/One ............ 123   Lan/Merr ...........   0
Rcv/SrcMCA ..........  0   Xmt/Mul ............ 132   Lan Tx/Rx ..........   0
Misc/Cntl ...........  0   Xmt/Rtry ............  0   Rcv/Buffer .........   0
Rcv/Invalid .........  0   Xmt/LCar ............  0   Rcv/NoSTP ..........   0
Rcv/Short802 ........  0   Xmt/LCol ............  0   -- Misc Counters ------
Rcv/Long802 .........  0   Xmt/MLen ............  0   Err/HostXfer .......   0
Rcv/Missed ..........  0   Xmt/CTest ...........  0   RX/NoRxBuf .........   0
Rcv/Dor .............  0   Xmt/Timeout .........  0   RX/XmtRngFull ......   0
Rcv/NoRcvBuf ........  0   ---------------- Saved Error Data ----------------
Rcv/Stale ...........  0   Rtry at ........... None   LCol at ........... None
Rcv/Ubus ............  0   LCar at ........... None   CTst at ........... None
Rcv/Sbus ............  0   Sbus at ........... None
Rcv/Crc+Frame .......  0   Crc  at ........... None
Rcv/Mlen ............  0   Mlen at ........... None
Rcv/Urfd ............  1   01-AUG-1989 08:02:05  60-02 MopRC 11.111
```

msb-0336-89

10.7.2.2 DEMNA Resource Counters

These counters provide the following information on the use of DEMNA resources:

- The total number of DEMNA-generated host interrupts and the frequency of such interrupts
- What percentage of CPU time the DEMNA spends executing a particular firmware process
- The number of DEMNA-internal transmit and receive buffers in use
- The percentage of XNA memory bus traffic generated by the LANCE chip and the gate array
- The percentage of existing XMI bus traffic generated by each XMI node

Status and Status/Interval Screens

```
-- 08-00-2B-00-00-01 -- Status -- 01-AUG-1989 19:01:19 -- Uptime:   01:43:35

-- NI Statistics -------   -- NI Counters ---------   -- Process --    --XMI---
Bytes/Pk ............ 64   BytesSnt .... 6327255447   Null      88.0%   0   0.0%
Bytes/Xmt ........... 64   BytesRcv .... 6327084034   Port       0.7%   1   0.0%
Bytes/Rcv ........... 64   MbytesSnt ........ 20470   Xmt-Ln     2.8%   2   0.0%
Pk/Sec ............. 510   MbytesRcv ............ 0   Xmt-Hs     2.0%   3   0.0%
Xmt/Sec ............ 255   PkSnt ......... 17464866   Rcv-Ln     1.6%   4   0.0%
Rcv/Sec ............ 255   PkRcv ......... 17462507   Rcv-Hs     3.0%   5   0.0%
MBaudRate ..... 0.274471   MPkSnt ............. 230   Com-Hs     0.0%   6   0.0%
Interrupts ... 104599634   MPkRcv ............... 0   Mon        1.0%   7   0.0%
Interrupts/Sec .... 255                               Cons       0.4%   8   0.0%
                                                                        9   0.0%
-- Total NI Traffic ----   -- Error Summary -------   --Buffers--       A   0.0%
Bytes/Pk ........... 237   Xmt/Wire ............. 0   Rcv ....   0      B   0.0%
Pk/Sec ............ 1418   Rcv/Wire ............. 0   Xmt ....   1      C  11.7%
ThisNI + Other = TotBaud   Rcv/Validation ....... 1                     D   0.0%
  3.5% + 26.0% =   29.5%   Rcv/NoBuffers ........ 0   --XNA Bus---      E  89.3%
                                                      LANCE     9.3%    F   0.0%
                                                      XNAGA     0.0%
```

msb-0337-89

10.8 How to Use the Network Screens

This section describes how to use the Network screens. The following topics are discussed:

- Users versus nodes
- Interval parameters versus accumulated parameters
- Interval screen versus Accumulated screen

10.8.1 Users versus Nodes

The two leftmost columns of the Network screen are used to list the six most active users and seven most active nodes. The user designator (protocol type, SAP, or SNAP SAP protocol identifier) is listed for each of the six users. The user name is also supplied for certain commonly used Ethernet users. The DECnet or Ethernet address is listed for each of the seven nodes. (See Appendix I for a listing of commonly used Ethernet protocol types, Appendix J for a listing of commonly used Ethernet addresses, and Appendix K for a listing of commonly used SAP assignments and SNAP protocol ID assignments.)

Network Screen

```
-- 08-00-2B-00-00-01 -- Network -- 01-AUG-1989 10:50:45 --

                          - 7,99996 usecs --   7.4% NI--   -- 00:00:06 --   1.5% NI--
 #  User                    Pks/Sec  Byt/Pk  %NI-Cur     Packets      Bytes(k) %NI-Tot
 -  ---------               -------  ------  -------     -------      -------- -------
 1  60-07 NISca                 328     211     6.5%        1959            49    1.1%
 2  60-03 Decnet                 70     155     1.0%         424             9    0.2%
 3  60-04 Lat                    20     106     0.2%         109             2    0.0%
 4  60-02 MopRC                  14      94     0.1%          95             1    0.0%
 5  80-3F LTM                     0    1490     0.0%           2             0    0.0%
 6  08-00 IP                      1      98     0.0%           3             0    0.0%

 #  Nodes                   Pks/Sec  Byt/Pk  %NI-Cur     Packets      Bytes(k) %NI-Tot
 -  ---------               -------  ------  -------     -------      -------- -------
 1  11.111                      122      41     4.3%         796            10    0.5%
 2  11.112                      119      41     4.3%         754            10    0.5%
 3  AB-00-03-00-00-01            28      23     0.6%         171             0    0.0%
 4  11.113                       37      14     0.5%         216             0    0.1%
 5  11.114                       43       9     0.4%         254             0    0.1%
 6  11.115                       39      91     0.4%         246             0    0.1%
 7  11.116                       13     161     0.2%          41             0    0.0%
```

msb-0338-89

The parameters in the top table of the Network screen pertain to the users. For example, in the Network screen below, the parameters on line 1 of the top table pertain to the NISca user, the parameters on line 2 pertain to the DECnet user, and so on.

Network Screen

```
-- 08-00-2B-00-00-01 -- Network -- 01-AUG-1989 10:50:45 --

                          - 2999996 usecs --   7.4% NI--  -- 00:00:06 --   1.5% NI--
 #  User               Pks/Sec  Byt/Pk  %NI-Cur   Packets    Bytes(k) %NI-Tot
 -  ----------         -------  ------  -------   -------    -------- -------
 1  60-07 NISca            328     211     6.5%      1959          49    1.1%
 2  60-03 Decnet            70     155     1.0%       424           9    0.2%
 3  60-04 Lat               20     105     0.2%       109           2    0.0%
 4  60-02 MopRC             14      94     0.1%        95           1    0.0%
 5  80-3F LTM                0    1490     0.0%         2           0    0.0%
 6  08-00 IP                 1      98     0.0%         3           0    0.0%

 #  Node               Pks/Sec  Byt/Pk  %NI-Cur   Packets    Bytes(k) %NI-Tot
 -  ----------         -------  ------  -------   -------    -------- -------
 1  11.111                 122     412     4.3%       796          10    0.5%
 2  11.112                 119     413     4.3%       754          10    0.5%
 3  AB-00-03-00-00-01       28     238     0.6%       171           0    0.0%
 4  11.113                  37     143     0.5%       216           0    0.1%
 5  11.114                  43      94     0.4%       254           0    0.1%
 6  11.115                  39      98     0.4%       246           0    0.1%
 7  11.116                  13     161     0.2%        41           0    0.0%
```

msb-0339-89

The parameters in the bottom table of the Network screen pertain to the nodes. For example, in the example screen below, the parameters on line 1 of the bottom table pertain to the node at DECnet address 11.111, the parameters on line 2 pertain to the node at DECnet address 11.112, and so on.

Network Screen

```
-- 08-00-2B-00-00-01 -- Network -- 01-AUG-1989 10:50:45 --

                        - 2999996 usecs --   7.4% NI--   -- 00:00:06 --    1.5% NI--
#  User                  Pks/Sec  Byt/Pk  %NI-Cur      Packets       Bytes(k) %NI-Tot
-  ----------            -------  ------  -------      -------       -------- -------
1  60-07 NISca               328     211     6.5%         1959             49    1.1%
2  60-03 Decnet               70     155     1.0%          424              9    0.2%
3  60-04 Lat                  20     106     0.2%          109              2    0.0%
4  60-02 MopRC                14      94     0.1%           95              1    0.0%
5  80-3F LTM                   0    1490     0.0%            2              0    0.0%
6  08-00 IP                    1      98     0.[illegible]%            3              0    0.0%

#  Nodes                 Pks/Sec  Byt/Pk  %NI-Cur      Packets       Bytes(k) %NI-Tot
-  ----------            -------  ------  -------      -------       -------- -------
1  11.111                    122     412     4.3%          796             10    0.5%
2  11.112                    119     413     4.3%          754             10    0.5%
3  AB-00-03-00-00-01          28     238     0.6%          17[illegible]              0    0.0%
4  11.113                     37     143     0.5%          216              0    0.1%
5  11.114                     43      94     0.4%          254              0    0.1%
6  11.115                     39      98     0.4%          246              0    0.1%
7  11.116                     13     161     [illegible]           41              0    0.0%
```

msb-0340-89

10.8.2 Interval Parameters versus Accumulated Parameters

The interval parameters (see the shaded area in the figure below) are recorded for the interval indicated in the usecs field. (If only one user is accessing the Network screen, the interval should be very close to the nominal 3 seconds. However, if more than one user is accessing the Network screen, the interval may vary significantly from nominal.) The interval parameters are valid only for the indicated interval. The parameter values are updated approximately every 3 seconds.

Network Screen

```
-- 08-00-2B-00-00-01 -- Network -- 01-AUG-1989 10:50:45 --

                          - 2999996 usecs --   7.4% NI--   -- 00:00:06 --   1.5% NI--
#  User                      Pks/Sec  Byt/Pk  %NI-Cur     Packets     Bytes(k) %NI-Tot
-  ---------                 -------  ------  -------     -------     -------- -------
1  60-07 NISca                   328     211     6.5%        1959           49    1.1%
2  60-03 Decnet                   70     155     1.0%         424            9    0.2%
3  60-04 Lat                      20     106     0.2%         109            2    0.0%
4  60-02 MopRC                    14      94     0.1%          95            1    0.0%
5  80-3F LTM                       0    1490     0.0%           2            0    0.0%
6  08-00 IP                        1      98     0.0%           3            0    0.0%

#  Nodes                     Pks/Sec  Byt/Pk  %NI-Cur     Packets     Bytes(k) %NI-Tot
-  ---------                 -------  ------  -------     -------     -------- -------
1  11.111                        122     412     4.3%         796           10    0.5%
2  11.112                        119     413     4.3%         754           10    0.5%
3  AB-00-[illegible]-00-00-01     28     238     0.6%         171            0    0.0%
4  11.113                         37     143     0.5%         216            0    0.1%
5  11.114                         43      94     0.4%         254            0    0.1%
6  11.115                         39      98     0.4%         246            0    0.1%
7  11.116                         13     161     0.2%          41            0    0.0%
```

msb-0341-89

The accumulated parameters (see the shaded area of the figure below) are a cumulative record started when the Network screen is displayed. The accumulated parameters are valid for the time indicated in the Time field. The parameter values are updated approximately every 3 seconds.

Network Screen

```
-- 08-00-2B-00-00-01 -- Network -- 01-AUG-1989 10:50:45 --

                       - 2999996 usecs --   7.4% NI-- | -- 00:00:06 --    1.5% NI-- |
#  User                 Pks/Sec  Byt/Pk  %NI-Cur      | Packets   Bytes(k) %NI-Tot   |
-  ---------            -------  ------  -------      | -------   -------- -------   |
1  60-07 NISca              328     211    6.5%       |    1959         49    1.1%   |
2  60-03 Decnet              70     155    1.0%       |     424          8    0.2%   |
3  60-04 Lat                 20     106    0.2%       |     109          2    0.0%   |
4  60-02 MopRC               14      94    0.1%       |      95          1    0.0%   |
5  80-3F LTM                  0    1490    0.0%       |       2          0    0.0%   |
6  08-00 IP                   1      98    0.0%       |       3          0    0.0%   |
                                                      |                             |
#  Nodes                Pks/Sec  Byt/Pk  %NI-Cur      | Packets   Bytes(k) %NI-Tot   |
-  ---------            -------  ------  -------      | -------   -------- -------   |
1  11.111                   122     412    4.3%       |     796         10    0.5%   |
2  11.112                   119     413    4.3%       |     784         10    0.5%   |
3  AB-00-03-00-00-01         28     238    0.6%       |     171          0    0.0%   |
4  11.113                    37     143    0.5%       |     216          0    0.1%   |
5  11.114                    43      94    0.4%       |     294          0    0.1%   |
6  11.115                    39      98    0.4%       |     246          0    0.1%   |
7  11.116                    13     161    0.2%       |      41          0    0.0%   |
```

msb-0342-89

10.8.3 Interval Screen versus Accumulated Screen

In the Interval Network screen, which is the default Network screen, the most active users and nodes are determined by network traffic (transmits and receives) during the interval indicated by the usecs field. Thus, the users and nodes displayed on the left of the screen are ranked for the interval, not for the cumulative time. The Interval screen should therefore be used to examine the most active users and nodes at 3-second intervals.

The Accumulated Network screen can be accessed only by typing [CTRL/A] when the Interval Network screen is displayed. The Accumulated screen looks exactly like the Interval screen (with the exception of the label *Accumulated* below the date and time); however, in the Accumulated screen, the users and nodes are ranked for the cumulative time indicated in the Time field. The Accumulated screen should thus be used to examine the most active users and nodes for an interval longer than 3 seconds.

To alternate between the Interval Network screen and the Accumulated Network screen, type [CTRL/A].

APPENDIX A

A Power and Environmental Requirements

This appendix provides the power and environmental requirements for the DEMNA.

Table A–1 Power Requirements

VDC	Amperes
+5	7.75
+12	0.60 (including power to H4000 transceiver)
-12	0.10

Table A–2 Environmental Requirements

	Operating Environment
Temperature	5°C to 50°C (41°F to 122°F)
Humidity	10% to 95% with maximum wet bulb of 32°C (89.6°F) and minimum dew point of 2°C (36°F) noncondensing
Altitude	To 2.4 km (8,000 ft)

	Storage Environment
Temperature	-40°C to 66°C (-40°F to 151°F)
Humidity	To 95% noncondensing
Altitude	To 9.1 km (30,000 ft)

APPENDIX B

B Installation

This chapter explains how to install the DEMNA option into a system that has an XMI bus. The installation procedure has three major parts:

- Hardware Installation
- Verification of Hardware Installation
- Verification of DEMNA Operation in Network

It is very important to perform all parts of the installation procedure. Do not skip any part of the procedure.

The installation procedure does not describe how to install a.. Ethernet transceiver. For instructions on installing an Ethernet tr^nsceiver, see the installation guide for the host computer system.

WARNINGS: **POWER OFF—Shut off system power and disconnect the system power cord before performing any procedure in this chapter.**

WEAR ESD WRIST STRAP—You must wear an antistatic wrist strap that is connected to the processor cabinet whenever you work inside the cabinet.

USE CONDUCTIVE CONTAINERS—Whenever you remove a circuit board from an XMI card cage, place it in a conductive container.

B.1 Hardware Installation

When installing a DEMNA in a VAX 6000 system, use the procedure in Section B.1.1. When installing a DEMNA in a VAX 9000 system, use the procedure in Section B.1.2.

B.1.1 Hardware Installation in a VAX 6000 System

Up to three DEMNAs can be installed in an XMI card cage in a VAX 6000 system.

The following steps describe how to install the DEMNA hardware in a VAX 6000 system:

1 Power down the host computer system by:

   a. Turning the Power switch ^ the Off position

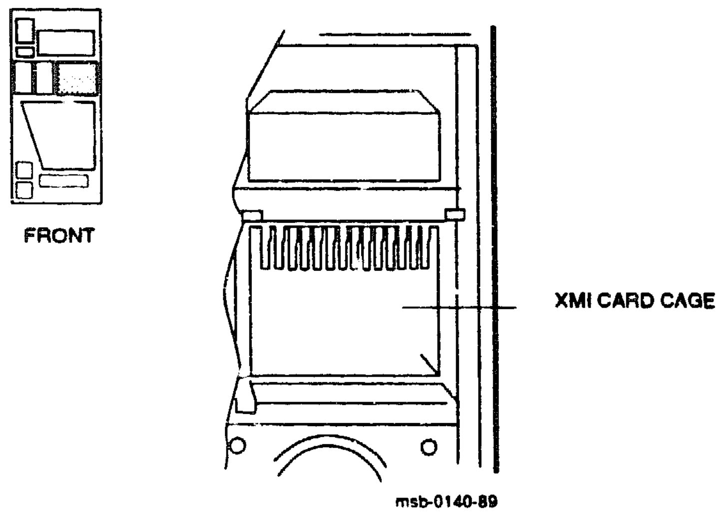
   b. Setting the system circuit breaker in the rear to Off

2 Open front door of the cabinet.

3 Put on the ESD wrist strap that is attached to the system chassis. This grounds you and thus prevents you from damaging the electronic components by discharging static electricity.

4 Locate the XMI card cage. Figure B–1 shows the location of the XMI card cage in a VAX 6000 system.

Figure B–1 Location of XMI Card Cage in a VAX 6000 System



5 Remove the door on the front of the XMI card cage.

6 Determine the slot into which the DEMNA should be installed. In a VAX 6000 system, the DEMNA can be put into any of the following slots: 1–4, B–E (see Figure B–2). In general, the DEMNA should be put in the highest-numbered slot available within the ranges specified above. CPUs are usually put in lower-numbered slots.

7 Lift the lever to open the chosen slot.

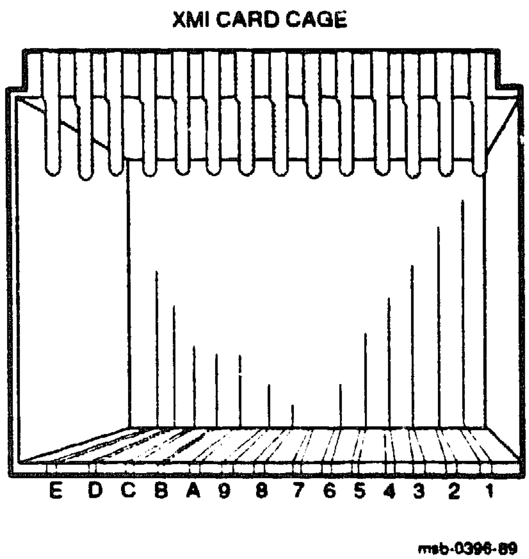
8 Slide the DEMNA module into the slot until it stops: this is a zero-insertion-force card cage.

9 Close the locking lever.

10 Replace the door on the front of the card cage.

11 Install an I/O connector panel (74-26407-41) for an Ethernet connector over one of the bulkhead cutouts. If you are replacing the first Ethernet controller in the system, this is unnecessary.

Figure B-2 XMI Card Cage Slots



12 Install the internal Ethernet cable as follows:

a. Connect the P1 connector of the internal Ethernet cable to backplane segment E2 for the DEMNA slot (see Figure B-3). The connector is right-side-up when the key on the connector is on the right. The P1 connector is not uniquely keyed for backplane segment E2. It is thus possible to insert the connector into the wrong backplane segment.

CAUTION: **Do not connect the DEMNA to the network until you have verified the DEMNA installation. If the DEMNA cabling is connected to the wrong slot, an arbitrary signal may be output on the transmit line, which might bring down the entire network.**

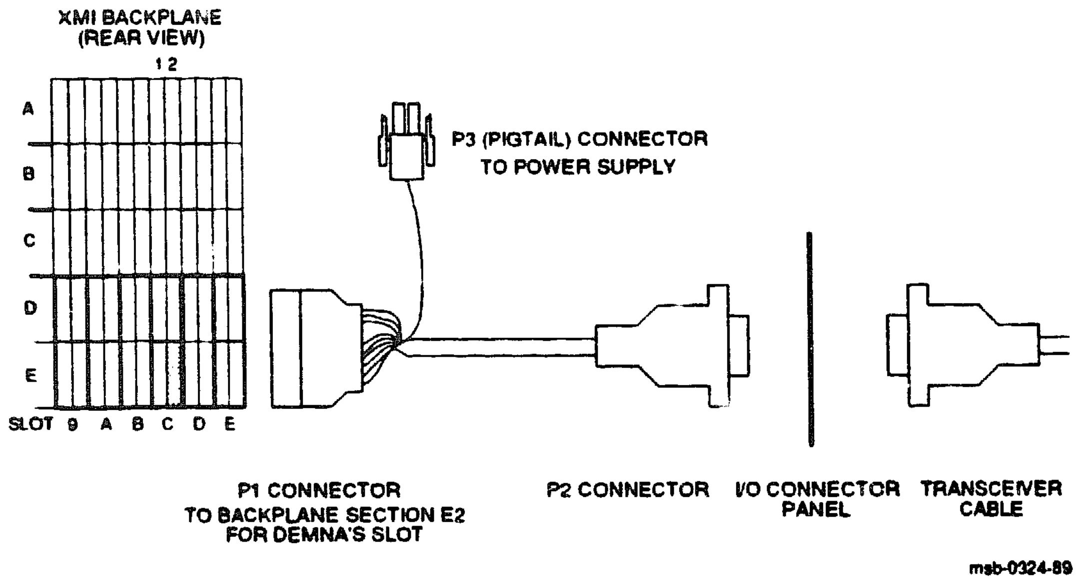
b. Attach the P2 connector of the internal Ethernet cable to the I/O connector panel. (If you are replacing the first Ethernet controller in a VAX 6000 system, connect the P2 connector to the system interconnect panel. The P2 connector plugs into the rear of the Ethernet connector on the panel. Figure B-4 shows the system interconnect panel for a VAX 6000 Model 400 system.)

c. Connect the pigtail connector (P3) from the internal Ethernet cable to a +15V 2-prong connector (J2) from any of the H7214 power supplies. These power supplies are located in the rear of the cabinet.

NOTE: All the connectors from the power supply may already be used. In this case, the external transceiver cable must not be connected directly to an H4000 transceiver, a DESTA (a thin-wire box), or a DECOM broadband transceiver—none of which has its own power supply. The transceiver cable may, however, be connected to one of the following devices, each of which has its own power supply and which, in turn, may be connected to an H4000:

- **A DELNI**
- **A DEMPR (a thin-wire version of the DELNI)**
- **A DEBET (a bridge)**

Figure B–3 Internal Ethernet Cable Connections



13 If a physical console is going to be used, install the internal cable for the physical console as follows:

a. Install an I/O connector panel (74-26407-32) for the cable over one of the bulkhead cutouts.

b. Connect the P1 connector of the cable to backplane segment D2 for the DEMNA slot (see Figure B–5). The connector is right-side-up when the key on the connector is on the right. The P1 connector is not uniquely keyed for backplane segment D2. It is thus possible to insert the connector into the wrong backplane segment.

c. Attach the P2 connector of the cable to the I/O connector panel.

Figure B–4 System Interconnect Panel—VAX 6000 Model 400 System

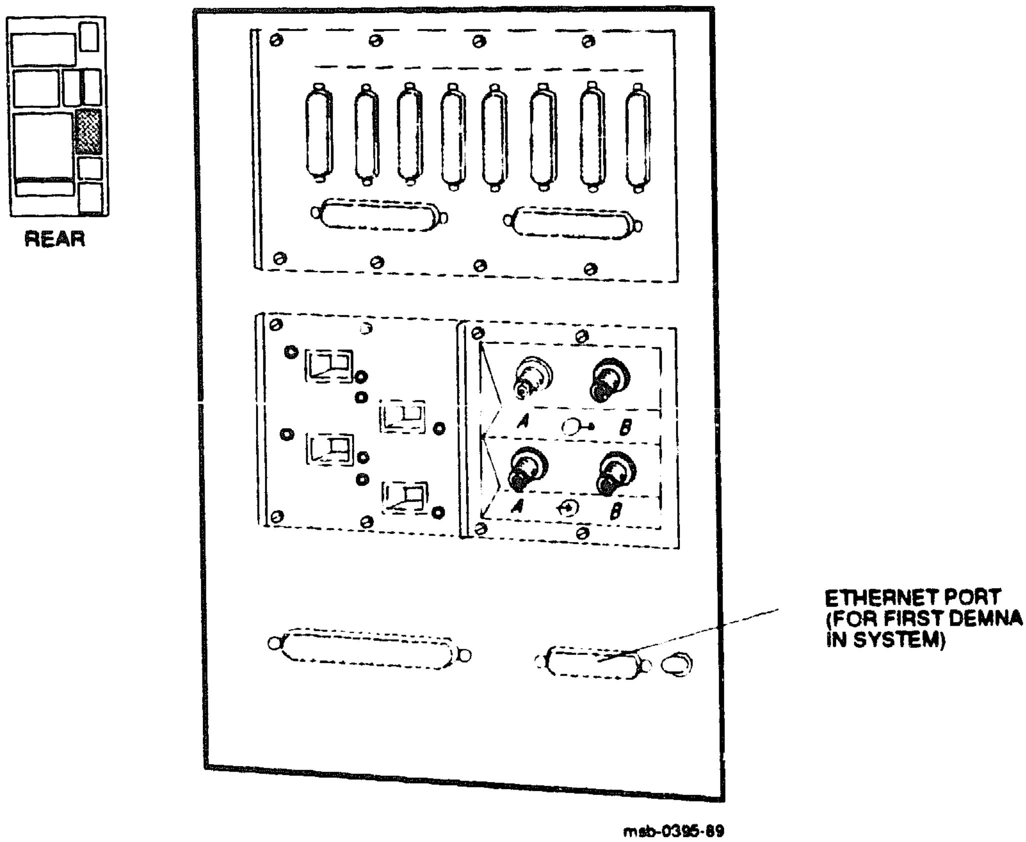
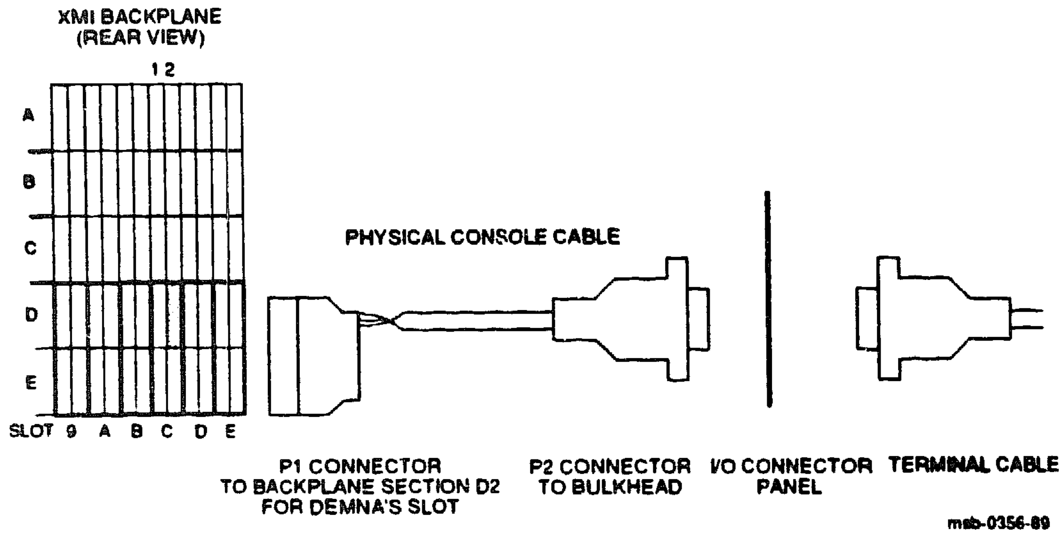


Figure B–5 Internal Cable for Physical Console



B.1.2 Hardware Installation in a VAX 9000 System

Up to four DEMNAs can be installed per XMI card cage in a VAX 9000 system.

The following steps describe how to install the DEMNA hardware in a VAX 9000 system:

1 Power down the host computer system by:

   a. Turning the Power switch to the Off position

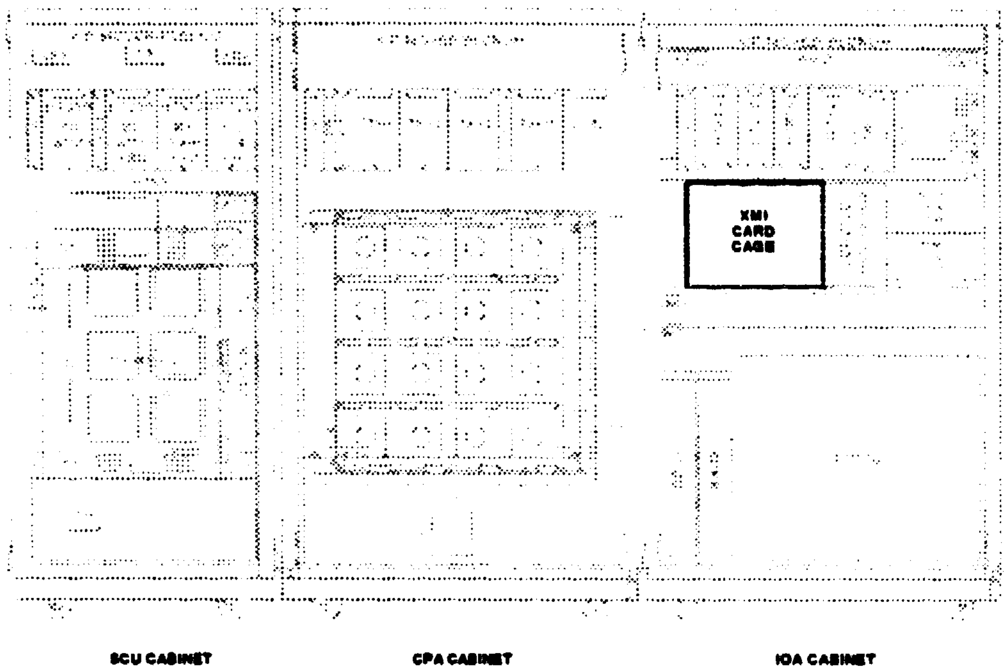
   b. Setting the appropriate system circuit breaker(s) to Off

2 Open front door of the appropriate cabinet.

3 Put on the ESD wrist strap that is attached to the system chassis. This grounds you and thus prevents you from damaging the electronic components by discharging static electricity.

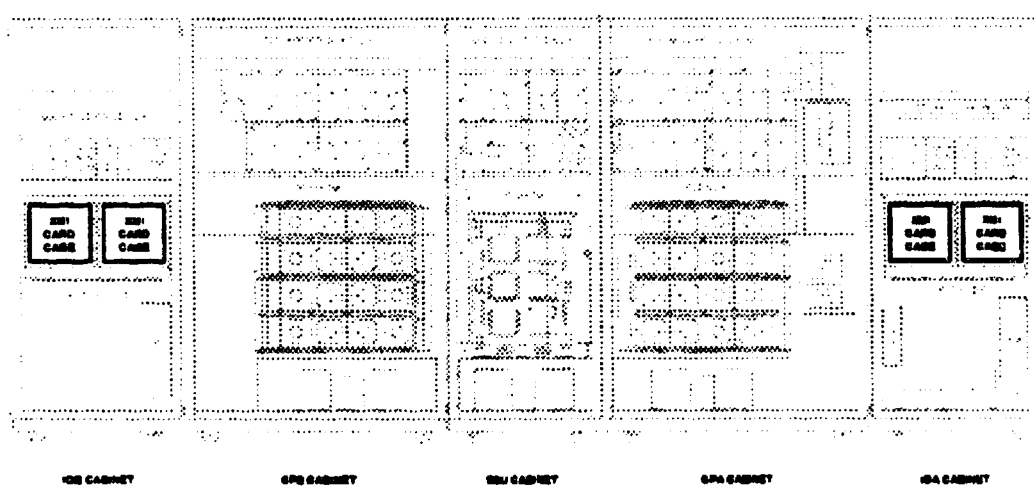
4 Locate the XMI card cage into which the DEMNA is to be installed. Figure B–6 shows the location of the XMI card cage in a VAX 9000 Model 2xx. Figure B–7 shows the location of the XMI card cage in a VAX 9000 Model 4xx.

Figure B-6 Location of XMI Card Cage in a VAX 9000 Model 2xx System



msb-0650-89

Figure B-7 Location of XMI Card Cages in a VAX 9000 Model 4xx System



msb-0651-89

5 Open the door on the front of the appropriate XMI card cage.

6 Determine the slot into which the DEMNA should be installed (see Figure B-8). In a VAX 9000 system, the DEMNA can be put into any XMI slot except slot 7 or 8.

7 Lift the lever to open the chosen slot.

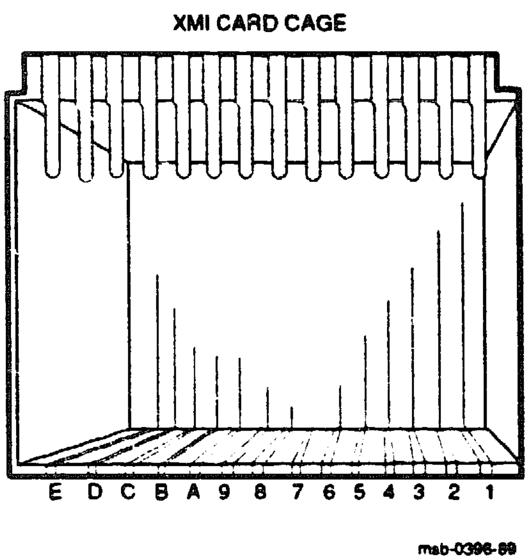
8 Slide the DEMNA module into the slot until it stops: this is a zero-insertion-force card cage.

9 Close the locking lever.

10 Close the door on the front of the XMI card cage.

11 Install an I/O connector panel (70-27894-01) for an Ethernet connector over one of the bulkhead cutouts. (If you are installing the second DEMNA in the cabinet, this is unnecessary, since the I/O connector panel accommodates two DEMNAs.)

Figure B-8 XMI Card Cage



12 Install the internal Ethernet cable as follows:

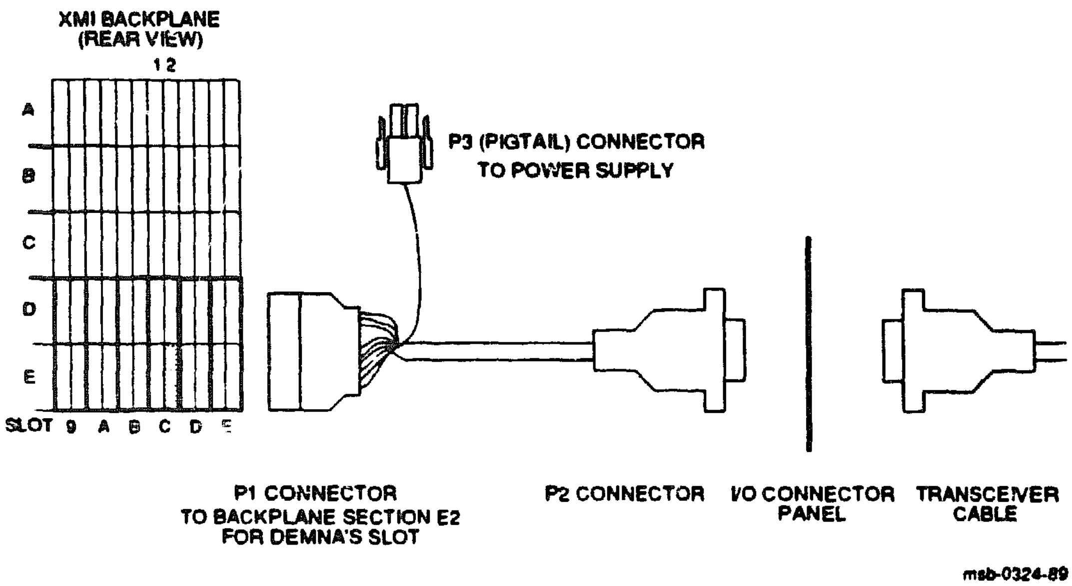
a. Connect the P1 connector of the internal Ethernet cable to backplane segment E2 for the DEMNA slot (see Figure B-9). The connector is right-side-up when the key on the connector is on the right. The P1 connector is not uniquely keyed for backplane segment E2. It is thus possible to insert the connector into the wrong backplane segment.

CAUTION: Do not connect the DEMNA to the network until you have verified the DEMNA installation. If the DEMNA cabling is connected to the wrong slot, an arbitrary signal may be output on the transmit line, which might bring down the entire network.

b. If you are installing the second DEMNA in the cabinet, remove the plate over the second cutout of the I/O connector panel.

c. Attach the P2 connector of the internal Ethernet cable to the I/O connector panel.

Figure B–9 Internal Ethernet Cable Connections

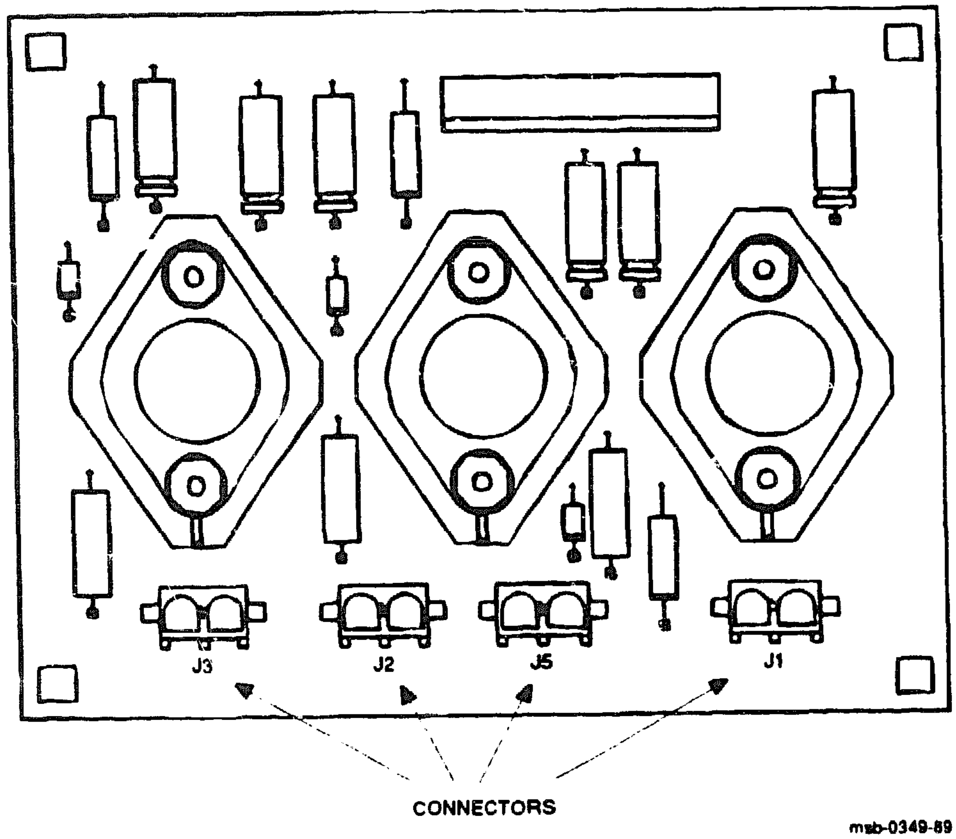


d. Connect the pigtail connector (P3) from the internal Ethernet cable to a +15V 2-prong connector on the power distribution adapter for the XMI card cage (see Figure B–10). The J2 connector on the power distribution adapter is connected to the main power supply (H7214). The J1, J3, J5 connectors are connected to the auxiliary power supply. Table B–1 summarizes the power connections for VAX 9000 systems.

Table B–1 Power Connection for Internal Ethernet Cable

System	Maximum Number of DEMNAs Per Card Cage	Power Connection
VAX 9000 Model 2xx	4	Any connector on the power distribution adapter (part no. 54-19045-01) located in the rear of the cabinet on the left-hand rails
VAX 9000 Model 4xx	4	Any connector on the power distribution adapter (part no. 54-19045-01) located in the rear of the cabinet on the right-hand rails

Figure B-10 Power Distribution Adapter—VAX 9000 Systems



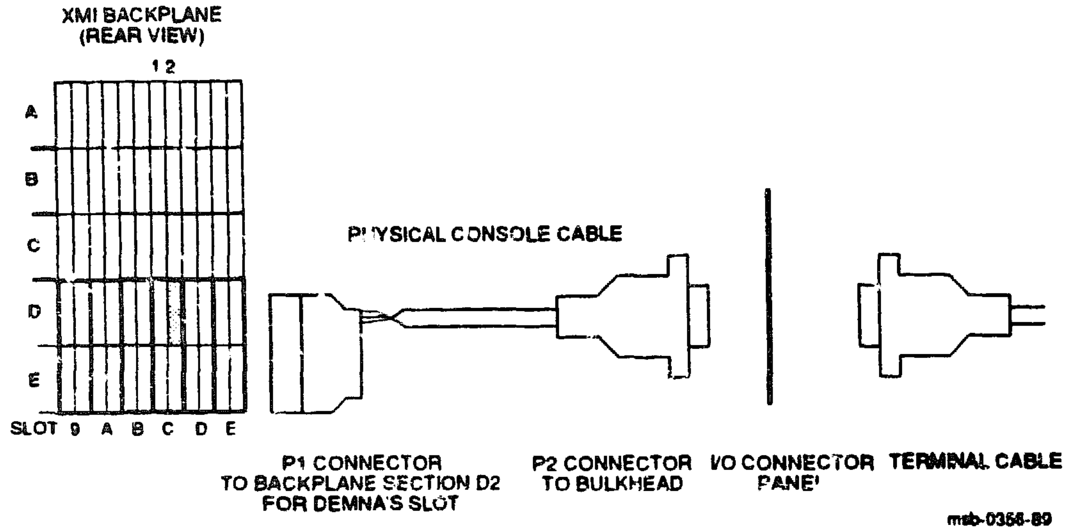
13 If a physical console is going to be used, install the internal cable for the physical console as follows:

a. Install an I/O connector panel (70-28010-01) for the cable over one of the bulkhead cutouts.

b. Connect the P1 connector of the cable to backplane segment D2 for the DEMNA slot (see Figure B-11). The connector is right-side-up when the key on the connector is on the right. The P1 connector is not uniquely keyed for backplane segment D2. It is thus possible to insert the connector into the wrong backplane segment.

c. Attach the P2 connector of the cable to the I/O connector panel.

Figure B–11 Internal Cable for Physical Console



B.2 Verification of Hardware Installation

CAUTION: It is very important that you verify that the DEMNA is properly installed. If the DEMNA is improperly installed, you may bring down the entire network when you connect the DEMNA to the network.

Follow these steps to verify that the DEMNA is properly installed:

1. Connect the loopback connector (part number 12-22196-02) to the Ethernet connector (P2 connector) of the internal Ethernet cable.
2. Set the appropriate system circuit breaker(s) to On.
3. Turn on the system power. This causes each module in the system to execute its self-test.
4. Verify that the green LED on the loopback connector is lit, indicating that the pigtail (P3) connector of the internal Ethernet cable is properly connected and is supplying +12V for the transceiver. If the LED is not lit, power down the system, make sure that the pigtail connector is connected to the right power connection, and reseat the pigtail connector. Continue with step 2.
5. Verify that the DEMNA passes both self-test (the yellow DEMNA OK LED lights) and the LANCE external loopback test (the green External Loopback LED lights). (See Section 6.2.1 for further information on the DEMNA LEDs.)
6. If the self-test and/or external loopback test fails, check to see that the DEMNA module is properly seated in the card cage and that all three connectors of the internal Ethernet cable are properly installed.

7 If the module continues to fail self-test, swap in a different DEMNA module if one is available. You can also try installing the module in a different slot.

8 If the module still fails self-test, run the DEMNA ROM-based diagnostics (RBDs), which are described in Chapter 7.

B.3 Verification of DEMNA Operation in Network

Proper operation of the DEMNA was verified up to the system bulkhead (Ethernet connector) in Section B.2. Now, follow these steps to verify that the DEMNA can communicate with other network nodes:

1 Connect the external transceiver cable to the P2 (Ethernet) connector of the internal Ethernet cable (available at the system bulkhead) or, for the first Ethernet controller in a VAX 6000 system, to the Ethernet connector on the system interconnect panel.

2 Boot the operating system.

3 Configure the network database and start the network software.

4 If the system is unable to communicate over the network, verify that the network software is installed and configured properly.

5 If the network software is properly installed and configured and the system is still unable to communicate over the network, shut down the system and check the transceiver and the transceiver cable as follows:

   a. Disconnect the external Ethernet transceiver cable (BNE3) at the transceiver end.

   b. Install the loopback connector (12-22196-02) on the cable.

   c. Run the DEMNA self-test and observe one of the following:

      - If the External Loopback LED on the DEMNA module lights, the transceiver is bad. Replace the transceiver, reconnect the cable to the new transceiver, and rerun the self-test to verify proper operation. No further action is required.

      - If the External Loopback LED on the DEMNA module does not light, the transceiver cable is bad and/or the P3 power connection for the internal Ethernet cable is not good. First, check the LED on the loopback connector. If the LED did not light, reseat the P3 connector and rerun the self-test. If the LED on the loopback connector is lit, replace the transceiver cable.

B.4 Internal Ethernet Cable

The internal Ethernet cable connects to the XMI backplane at the DEMNA slot and provides Ethernet signals and power (if required) to an H4000 transceiver or other network interface device. See Table 1–2 for part numbers.

The cable has three connectors (Figure B–3): P1, P2, and P3. The P1 connector connects to segment E2 on the XMI backplane opposite the DEMNA slot. The P2 connector is an industry-standard Ethernet connector that connects to an I/O connector panel on the bulkhead or to the rear of the Ethernet connector on the system interconnect panel in a VAX 6000 system. The P3 (pigtail) connector is a +15V direct-current power connection that supplies power to a device (such as an H4000 transceiver, a DESTA thin-wire box, or a DECOM broadband transceiver) that does not have its own power supply. The pigtail connector should be plugged in regardless of the type of transceiver.

Figure B–12 shows the P1 connector pinouts of the internal Ethernet cable. Table B–2 describes these pinouts. Figure B–13 shows the pinouts for the P2 connector of the internal Ethernet cable, and Table B–3 describes these pinouts.

Figure B–12 P1 Connector Pinouts of Internal Ethernet Cable

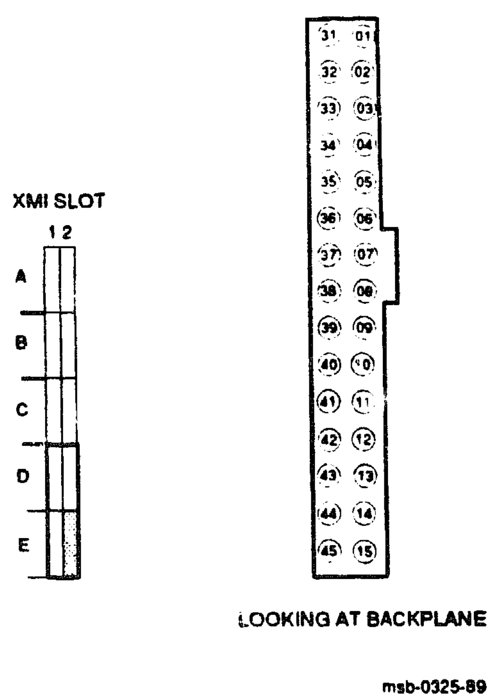
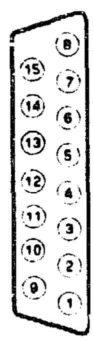


Table B-2 P1 Connector Pinouts of Internal Ethernet Cable

Pin	Signal	Description
E01–E04	Unconnected	
E05–E09	Logic Ground	
E10 E11	Ethernet Collision L Ethernet Collision H	Differential collision detect signals from the Ethernet bus.
E12 E13	Ethernet Receive L Ethernet Receive H	Differential receive signals from the Ethernet bus.
E14 E15	Ethernet Transmit L Ethernet Transmit H	Differential transmit signals to the Ethernet bus.

Figure B-13 P2 Connector Pinouts of Internal Ethernet Cable



msb-0326-89

Table B-3 P2 Connector Pinouts of Internal Ethernet Cable

Pin	Signal	Description
1	Shield	
2 9	Collision Presence H Collision Presence L	Differential signals that indicate a failure of the collision detection logic
3 10	Transmit H Transmit L	Differential transmit signals to the Ethernet bus
4	Reserved	
5 12	Receive H Receive L	Differential receive signals from the Ethernet bus
6	Power Return	Power return line
7	Reserved	
8	Reserved	
11	Reserved	
13	Power	

Table B-3 (Cont.) P2 Connector Pinouts of Internal Ethernet Cable

Pin	Signal	Description
14	Reserved	
15	Reserved	

B.5 External Transceiver Cable

The external transceiver cable runs from the Ethernet connection provided at the bulkhead or system interconnect panel to an Ethernet transceiver, such as an H4000 baseband transceiver, DECOM broadband transceiver, or DELNI local network interconnect. The cable is ordered as a separate item.

B.6 Internal Cable for Physical Console

The internal cable for the physical console connects the XMI backplane at the DEMNA slot to the system bulkhead and provides a connector for a terminal cable. The cable is ordered as a separate item.

The cable has two connectors: P1 and P2 (see Figure B–5). The P1 connector connects to segment D2 on the XMI backplane opposite the DEMNA slot. The P2 connector is a standard 25-pin Sub-D connector that connects to the bulkhead and is used as a connector for a terminal cable.

Figure B–14 shows the P1 connector pinouts for the internal cable for the physical console. Table B–4 describes these pinouts. Figure B–15 shows the P2 connector pinouts for the internal cable for the physical console. Table B–5 describes these pinouts.

Figure B-14 P1 Connector Pinouts of Internal Cable for Physical Console

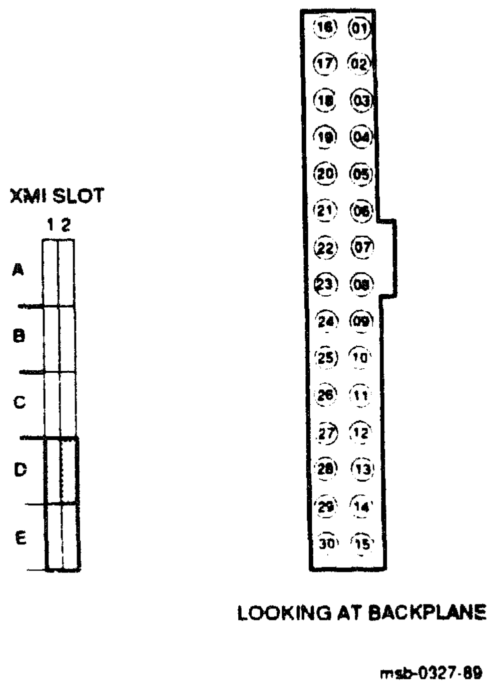
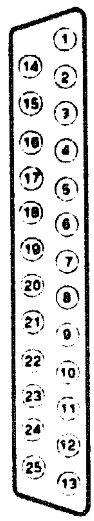


Table B-4 P1 Connector Pinouts of Internal Cable for Physical Console

Pin	Signal
D01	Transmit
D02	Receive
D03	Logic Ground
D04–D30	Unconnected

Figure B–15 P2 Connector Pinouts of Internal Cable for Physical Console



msb-0348-89

Table B–5 P2 Connector Pinouts of Internal Cable for Physical Console

Pin	Signal
1	Unconnected
2	Transmit
3	Receive
4-6	Unconnected
7	Logic Ground
8–25	Unconnected

B.7 Removal

To remove a DEMNA module, follow these steps:

1. Power down the host computer system by:
   a. Turning the Power switch to the Off position
   b. Setting the appropriate system circuit breaker(s) to Off
2. Open the appropriate cabinet.
3. Make sure you are wearing an ESD wrist strap that is attached to the system chassis.
4. Open or remove the door on the front (module-insertion side) of the XMI card cage that contains the DEMNA to be removed.
5. Locate the desired card cage slot.
6. Lift the lever to open the slot.
7. Slide the module out of the card cage slot.
8. Put the module into a conductive container.
9. Close the locking lever.
10. If another DEMNA will not be installed:
    a. Close or replace the door on the front of the XMI card cage from which the DEMNA was removed.
    b. Remove the cables from the slot that contained the DEMNA.
    c. Close the cabinet.

APPENDIX C

C Bootstrapping with the DEMNA

Most host systems in which the DEMNA resides can bootstrap their operating systems either locally (from disk or tape) or remotely (from an Ethernet network). (See the documentation for your operating system to determine whether your operating system supports booting over Ethernet.)

Host systems can boot voluntarily (from a command that a user types at the system console) or involuntarily (from a command to boot that arrives via the network).

Whether voluntarily or involuntarily, the host system boots from a specified device. The device is usually specified by the user in a console command, and the booting software then reads this parameter.

The DEMNA plays a role in booting whenever (a) the specified device from which to boot is the DEMNA or (b) a command to boot arrives from the network.

This appendix describes these two bootstrap roles:

1. The system is instructed to boot an image from the network (via the DEMNA) instead of from a disk.
2. A remote system sends the DEMNA a command to boot involuntarily (that is, the DEMNA's host system should reboot itself.)

C.1 Network Booting

The DEMNA may be specified as the boot device either explicitly by a console command or by the default setting of the auto restart function. The boot then proceeds as follows:

1. A bootstrap running on the local host initializes the port.
2. The bootstrap transmits a Request Program message to the load assistant multicast address. This message, which is defined by Digital's Maintenance Operation Protocol (MOP), requests the node that has the image to be downloaded to identify itself.
3. The bootstrap receives an Assistance Volunteer MOP message, which identifies the node that is volunteering to supply the load image.
4. The bootstrap transmits a Request Program MOP message to the volunteering node.
5. The bootstrap receives a Memory Load MOP message from the volunteering node. This message contains a section of the load image. The bootstrap writes the received image data to the appropriate location in host memory.

6 The bootstrap transmits a Request Memory Load MOP message to request the next section of image data and to indicate the status of the previous section of image data.

7 Steps 5 and 6 are repeated until the bootstrap receives a Memory Load with Transfer Address message.

8 The bootstrap transmits a final Request Memory Load MOP message to indicate that the final block was received correctly.

9 Host program execution jumps to the starting address of the downloaded program.

A user on the DEMNA's host system can request a boot by using the system console and typing the console B (BOOT) command at the system prompt (>>>). The following examples indicate the BOOT command for VAX 6000 and VAX 9000 systems.

On a VAX 6000:

```
>>> B EX
```

On a VAX 9000:

```
>>> B /XMI:mn MNA
```

where:

m is the XMI node number of the DEMNA

n is the unit number of the XJA adapter for the XMI card cage

C.2 Involuntary Booting

In an involuntary boot, a remote node sends the DEMNA a MOP Boot message. This operation is typically used for booting a system that is remotely located from an operator.

If the DEMNA port is in the uninitialized state, it responds to the Boot message if the following three conditions are met:

- The packet containing the Boot message is addressed to the DEMNA's default physical address (DPA).
- The boot verification code in the Boot message matches the DEMNA boot verification code, which is the console password in EEPROM. The default console password is 584E41424F415244 (hex).
- The Enable Remote Boot flag in EEPROM is set.

If the port is in the initialized state (which is the normal case), it responds to the Boot message if the following four conditions are met:

- The packet containing the Boot message is addressed to any of the DEMNA's enabled physical addresses.
- The boot verification code in the Boot message matches the DEMNA boot verification code, which may be assigned by the port driver through a PARAM command.

- The Enable Remote Boot flag in EEPROM is set.
- The DEMNA's Enable Boot Messages flag was set by the port driver (via the PARAM command) to enable involuntary booting over the network.

In response to a valid Boot message, the DEMNA asserts XMI RESET L on the XMI bus, which causes an XMI system reset. If auto restart is enabled for the local system, console boot software running on the local host boots the system from the default boot device. If auto restart is disabled, the console prompts for operator input from the console terminal before continuing the boot. Note that the boot device in either case may be a local disk (or tape) or the network as described in Section C.1.

APPENDIX D

D How to Convert an Ethernet Address to a DECnet Address

An Ethernet address is converted to a DECnet address as follows:

1. Take the two low-order bytes of the Ethernet address and swap them so that the low-order byte precedes the next-to-low-order byte.
2. Convert the hex value of the two bytes into a decimal number.
3. Divide the decimal number by 1024.
4. The quotient is the DECnet area number.
5. The remainder is the DECnet node number.

For example, the Ethernet address AA-00-04-00-00-26 is converted to a DECnet address as follows:

1. Swap the two low-order bytes of the address to get the hex value 2600.
2. Convert 2600 (hex) to the decimal number 9728.
3. Divide 9728 by 1024 to get a quotient of 9 and a remainder of 512.
4. The DECnet area number is 9.
5. The DECnet node number is 512.

The Ethernet address AA-00-04-00-00-26 converts to DECnet address 9.512, which references DECnet node 512 in DECnet area 9.

APPENDIX E

E How to Read the DEMNA Ethernet Address

The DEMNA's default Ethernet address, which is also called the default physical address (DPA), is stored in the DEMNA MAC address (ENET) PROM. The DEMNA uses the DPA as its Ethernet address unless the operating system assigns it a DECnet address.

E.1 Systems with DECnet

If DECnet is running on your system, invoke the Network Control Program (NCP) and use the following commands to display the DEMNA's DPA:

```
$ MC NCP [RETURN]
NCP>TELL node SHOW KNOWN LINE CHARACTERISTICS
```

where *node* is the name of the Ethernet node at which the DEMNA resides

The line characteristics of the selected node are displayed in a format similar to the following:

```
Known Line Volatile Characteristics as of 26-APR-1989 16:06:41

Line = MNA-0

Receive buffers          = 6
Controller               = normal
Protocol                 = Ethernet
Service timer            = 4000
Hardware address         = 08-00-2B-09-CD-F3
Device buffer size       = 1198
```

The hardware address is the DEMNA's DPA.

E.2 From the Console Monitor Program or the ROM-Based Diagnostic Monitor

When running the console monitor program or when at the ROM-based diagnostic (RBD) monitor, you can read the DEMNA DPA from the DEMNA ENET PROM as follows:

1 Examine address 20007000 to read the first four bytes of the DPA.

2 Examine address 20007004 to read the last two bytes of the DPA.

Example E–1 illustrates this procedure from the console monitor program.

Example E–1 Examining the DEMNA DPA from the DEMNA Console Monitor Program or the ROM-Based Diagnostic Monitor

```
XNA>E 20007000                ! Read first four bytes of DPA from
                              ! ENET PROM.

     20007000/ 092B0008

XNA>E 20007004                ! Read last two bytes of DPA from
                              ! ENET PROM.
     20004004  0000F3CD
```

The above Ethernet address bytes are transmitted in the following order (left-to-right) over the network: 08-00-2B-09-CD-F3.

E.3 VAX 6000 System

If you are on a VAX 6000 system, use the SHOW ETHERNET console command to display the DEMNA DPA as follows:

```
>>>SHOW ETHERNET

XMI:3 08-00-2B-09-CD-F3
```

The command displays the XMI node number of the DEMNA and the DEMNA DPA.

If the SHOW ETHERNET command cannot find the DEMNA, you can read the DEMNA DPA by depositing and examining the DEMNA's XMI Communications (XCOMM) Register as indicated below:

```
>>> D XCOMM_address FFFFFFFF
>>> E XCOMM_address
>>> D XCOMM_address FFFFFFFE
>>> E XCOMM_address
```

where *XCOMM_address* is the address of the DEMNA's XCOMM Register. The XCOMM Register is at address BB + 10, where address (BB) is the base address of the DEMNA nodespace computed (in hex) as follows:

21800000 + (80000 * XMI_ID)

where XMI_ID is the DEMNA's XMI node ID

Example E–2 shows how to examine the DPA of a DEMNA at XMI node 3. The XDEV Register is examined first to confirm that the module being examined is a DEMNA (device type = 0C03).

Example E–2 Examining the DEMNA DPA on a VAX 6000

```
>>>E 21980000                  ! Examine XDEV Register

   P 21980000   06010C03

>>>D 21980010   FFFFFFFF       ! Deposit FFFFFFFF into XCOMM Register
>>>E 21980010                  ! Examine XCOMM Register to obtain
                               ! first four bytes of DPA
   P 21980010   092B0008

>>>D 21980010   FFFFFFFE       ! Deposit FFFFFFFE into XCOMM Register
>>>E 21980010                  ! Examine XCOMM Register to obtain
                               ! last two bytes of DPA
   P 21980010   0000F3CD
```

The above Ethernet address bytes are transmitted in the following order (left-to-right) over the network: 08-00-2B-09-CD-F3.

E.4 VAX 9000 System

If you are on a VAX 9000 system, you can read the DEMNA DPA by depositing and examining the XCOMM Register as described in Section E.3. The XCOMM Register is at address BB + 10, where address (BB) is the base address of the DEMNA nodespace computed (in hex) as follows:

20000000 + (XJA_ID * 800000) + (XMI_ID * 8000)

where:

XJA_ID is the XJA unit number

XMI_ID is the DEMNA's XMI node number

Example E–3 shows how to examine the DPA of a DEMNA at XMI node 4 through XJA number 2. The XDEV Register is examined first to confirm that the module being examined is a DEMNA (device type = 0C03).

Example E–3 Examining the DEMNA DPA on a VAX 9000

```
>>>E 21020000                  ! Examine XDEV Register

   P 21020000   06010C03

>>>D 21020010   FFFFFFFF       ! Deposit FFFFFFFF into XCOMM Register
>>>E 21020010                  ! Examine XCOMM Register to obtain
                               ! first four bytes of DPA
   P 21020010   092B0008
>>>D 21020010   FFFFFFFF       ! Deposit FFFFFFFE into XCOMM Register
>>>E 21020010                  ! Examine XCOMM Register to obtain
                               ! last two bytes of DPA
   P 21020010   0000F3CD
```

The above Ethernet address bytes are transmitted in the following order (left-to-right) over the network: 08-00-2B-09-CD-F3.

APPENDIX F

F Error Descriptions for ROM-Based Diagnostics

This appendix describes the errors for the DEMNA ROM-based diagnostics (RBDs). The DEMNA has four RBDs:

- Self-test RBD
- NI RBD
- XMI RBD
- XNA RBD

Table F–1 Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 1—Boot ROM Test
1	EPROM checksum error.
2	Machine check while accessing EPROM.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 2—CVAX Interrupt Lines Test
1	Unexpected interrupt.
90	Unexpected machine check.
	Test 3—Diagnostic Register Test
1	Diagnostic Register incorrectly initialized on power-up.
2	Diagnostic Register compare error.
3	Machine check while accessing the Diagnostic Register.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 4—SSC Test
Subtest 1—Programmable Bus Timeout Subtest	
1	Bus timeout machine check not taken on read to nonexistent memory.
2	Bus Timeout Register not as expected after bus timeout.
3	Zeros not returned on a read to a nonexistent GPR.
4	Bus Timeout Register not as expected after bus timeout.
90	Unexpected machine check.
91	Unexpected interrupt.

Table F–1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 4—SSC Test
Subtest 2—Configuration Register Subtest	
1	Configuration Register compare error.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 3—Programmable Address Decoders Subtest	
1	Programmable address decoder was incorrectly initialized on power-up.
2	Programmable Address Decoder Register compare error.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 4—Output Port Subtest	
1	Output port was incorrectly initialized on power-up.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 5—Console UART Subtest	
1	TXCS Ready bit not set as expected.
2	TXCS Ready bit not set after TXDB loaded.
3	RXCS Done bit not set when expected.
4	RXDB not as expected after loopback transmit.
5	RXCS Done bit not cleared after RXDB was read.
6	TXCS Ready bit not set after TXDB loaded.
7	RXCS Done bit not set when expected.
8	Loopback character was transmitted too fast or slow for the particular baud rate.
9	RXDB not as expected after loopback transmit.
10	RXCS Done bit not cleared after RXDB was read.
11	TXCS Read bit not set after TXDB loaded.
12	RXDB Register not as expected after overrun error was forced.
13	RXDB Register not as expected after overrun error was forced.
90	Unexpected machine check.
91	Unexpected interrupt.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 4—SSC Test
Subtest 6—Programmable Timers Subtest	
1	Programmer Timer Register not initialized correctly on power-up.
2	Programmable Timer Register compare error.
3	TIR not loaded correctly when transfer set in TCR.
4	TNIR corrupted when transfer set in TCR.
5	TNIR does not increment when single set in TCR.
6	TNIR corrupted when single set in TCR.
7	Interrupt bit is not set or is not the only bit set in TCR upon overflow.
8	TIR not reloaded upon overflow.
9	TNIR corrupted when Interrupt set in TCR.
10	TNIR not loaded correctly when Transfer and Stop bits set in TCR.
11	TNIR corrupted when Transfer and Stop bits set in TCR.
12	TIR didn't increment when Single and Stop bits set in TCR.
13	TNIR corrupted when Single and Stop bits set in TCR.
14	Interrupt bit not set in TCR on overflow when Stop bit set in TCR.
15	TIR not reloaded upon overflow when Stop bit set in TCR.
16	TNIR corrupted when Interrupt and Stop bits set in TCR.
17	Interrupt bit not set in TCR when Run bit set, after timeout.
18	Interrupt bit not set in TCR after first interrupt, after timeout.
19	TIR still increments after Run cleared in TCR.
20	Interrupt bit not set in TCR after Run bit set when Stop is set, after timeout.
21	Interrupt and Stop bits are not set or are not the only bits set on overflow.
22	TIR not cleared on overflow when Stop bit set in TCR.
23	TIR not cleared on overflow when Stop bit set in TCR after waiting.
24	Timer accuracy failed.
25	Interrupt bit not set in TCR when Run bit set, after timeout.
26	Error bit in TCR not set after missed overflow, after timeout.
27	Error, Interrupt, and Run bits are not set or are not the only bits set after missed overflow error.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 7—Interval Timer Subtest	
1	Interval timer not initialized correctly on power-up.
90	Unexpected machine check.
91	Unexpected interrupt.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 4—SSC Test
Subtest 8—I/O Bus Reset Register Subtest	
1	SSC I/O Bus Reset Register accessed as an SSC Register, not initialized correctly upon power-up.
2	SSC I/O Bus Reset Register accessed as an EPR, not initialized correctly upon power-up.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 9—TOY Clock Subtest	
1	SSC TOY Clock Register accessed as an SSC Register, not initialized correctly upon power-up.
2	SSC TOY Clock Register, accessed as an EPR, not initialized correctly upon power-up.
3	TOY clock did not increment.
4	TOY clock did not increment after first increment.
5	TOY clock did not stop after it was cleared.
6	TOY clock did not stay stopped after it was cleared.
7	TOY clock did not increment when it was started after it was stopped.
8	TOY clock did not increment when it was started after it was stopped.
9	TOY clock accuracy failed.
10	TOY clock did not stop after it was cleared.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 10—Break/Halt Logic Subtest	
1	Ready bit in TXCS Register not set.
2	Halt taken on break when Control-P Enable bit set in Configuration Register.
3	RXDB not as expected after break.
4	Ready bit in TXCS Register not set.
5	Halt taken on break when Secure Console bit set in Diagnostic Register.
6	RXDB not as expected after break.
7	Ready bit in TXCS Register not set.
8	Halt taken on break when in halt-protected space.
9	RXDB not as expected after break.
10	Ready bit in TXCS Register not set.
11	Expected halt not taken.
12	RXDB not as expected after halt.
13	Ready bit in TXCS Register not set.
14	Halt taken on Control-P when Control-P Enable bit not set.
15	RXDB not as expected after receipt of Control-P.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 4—SSC Test
16	Ready bit in TXCS Register not set.
17	Halt not taken on Control-P when Control-P Enable set.
18	RXDB not as expected after receipt of Control-P.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 11—Interrupt Subtest	
1	Ready bit in TXCS Register not set after TXDB loaded.
2	Expected UART transmit interrupt did not occur.
3	Ready bit in TXCS Register not set after TXDB loaded.
4	Done bit in RXCS Register not set after TXDB loaded for loopback transmit.
5	Expected UART receive interrupt did not occur.
6	RXDB Register not as expected after loopback transmit.
7	Done bit in RXCS Register not cleared after RXDB was read.
8	Interrupt bit in TCR not set after Run bit set, after timeout.
9	Interrupt not taken when Interrupt bit set in TCR.
10	TCR not as expected after interrupt.
11	Interval timer did not interrupt, after timeout.
12	Interval timer did not interrupt after first interrupt, after timeout.
13	Interval timer did not interrupt after second interrupt, after timeout.
14	Interval timer accuracy failed.
15	Interval timer interrupted after it was stopped.
16	Interrupt bit in TCR not set after Run bit set, after timeout.
17	SSC interrupt occurred when IVD bit was set in the Configuration Register.
18	SSC interrupt occurred after IVD bit cleared. The SSC internal interrupt request should have been canceled.
90	Unexpected machine check.
91	Unexpected interrupt.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 5—Console Drivers Test
1	Ready bit in TXCS Register not set after TXDB loaded.
2	Done bit in RXCS Register not set after TXDB loaded for loopback transmit.
3	RXDB Register not as expected after loopback transmit.
4	Done bit in RXCS Register not cleared after RXDB Register was read.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 6—CVAX RAM March Test
1	Contents of current location does not equal initial RAM pattern.
2	Contents of current location does not equal second RAM pattern.
3	Machine check taken while reading or writing a CVAX RAM location.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 7—CVAX Parity RAM Test
1	A parity bit is not as expected.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 8—CVAX Chip Test
1	IPR (0, 1, 2 or 3) bit is stuck
2	Bit in PCBB Register is stuck.
3	ASTLVL Register not initialized correctly on power-up.
4	ASTLVL Register cannot be written to its power-up value.
5	SISR Register not initialized correctly on power-up.
6	SISR Register cannot be written to its power-up value.
7	CADR Register not initialized correctly on power-up.
8	MSER Register not initialized correctly on power-up.
9	MSER Register cannot be written to its power-up value.
10	MAPEN Register not initialized correctly on power-up.
11	MAPEN Register cannot be written to its power-up value.
12	SID Register not initialized correctly on power-up.
13	Machine check caused by unexpected parity error on CVAX RAM read.
14	Machine check not taken for expected parity error on CVAX RAM read.
15	MSER not as expected after machine check.
16	Machine check caused by unexpected parity error on CVAX RAM read.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 8—CVAX Chip Test
17	Data compare error on first longword of quadword read.
18	Data compare error on second longword of quadword read.
19	Machine check caused by bus timeout when reading a quadword.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 9—ENET PROM Test
1	First 6 bytes of the ENET PROM are all zero (null address).
2	Machine check taken while accessing the ENET PROM.
3	Low bit of the ENET PROM address is a 1 (multicast address).
4	Byte 0 of stored checksum not equal to byte 0 of calculated checksum.
5	Byte 1 of stored checksum not equal to byte 1 of calculated checksum.
6	Second copy of address and checksum not equal to first copy.
7	Third copy of address and checksum not equal to first copy.
8	First copy of test pattern not as expected.
9	Second copy of test pattern not as expected.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 10—EEPROM Test
1	Calculated checksum not equal to the stored checksum.
2	Machine check while accessing EEPROM.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 11—XNADAL Readback Test
1	XNADAL Bus is hung (Grant Timeout bit is set in the Diagnostic Register).
2	The LANCE has an outstanding grant to the XNADAL Bus (LANCE Grant Status bit cleared in Diagnostic Register).
3	XNADAL Bus is hung (Grant Timeout bit is set in the Diagnostic Register) after LANCE taken off the bus (Hard Reset to the LANCE bit cleared in Diagnostic Register).
4	The LANCE has an outstanding grant to the XNADAL Bus (LANCE Grant Status bit cleared in Diagnostic Register) after LANCE taken off the bus (Hard Reset to the LANCE bit cleared in Diagnostic Register).
5	The gate array has an outstanding grant to the XNADAL Bus (XNAGA Grant Status bit cleared in Diagnostic Register).

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 11—XNADAL Readback Test
6	XNADAL Bus is hung (Grant Timeout bit is set in the Diagnostic Register) after gate array taken off the bus (Hard Reset to the XNAGA bit cleared in Diagnostic Register).
7	The gate array has an outstanding grant to the XNADAL Bus (XNAGA Grant Status bit cleared in Diagnostic Register) after gate array taken off the bus (Hard Reset to the XNAGA bit cleared in Diagnostic Register).
8	XNADAL loopback failed.
9	XNADAL loopback failed.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 12—XNADAL Timeout Logic Test
1	Expected timeout did not occur (Grant Timeout bit not cleared in Diagnostic Register).
2	Expected timeout did not occur due to LANCE grant (LANCE Grant Status bit not cleared in Diagnostic Register).
3	Grant Timeout bit not set in Diagnostic Register after timeout cleared.
4	LANCE Grant Status bit not set in Diagnostic Register after timeout cleared.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 13—Shared RAM March Test
1	Contents of current location does not equal initial RAM pattern.
2	Contents of current location does not equal second RAM pattern.
3	Machine check taken while reading or writing a shared RAM location.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 14—Shared Parity RAM Test
1	A parity bit not as expected.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 15—LANCE Test
Subtest 1—Register Access Subtest	
1	LANCE RAP not initialized correctly upon power-up.
2	Could not make the LANCE RAP point to CSR0.
3	LANCE CSR0 not initialized correctly upon power-up.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 15—LANCE Test
4	Could not initialize LANCE CSR0 to its power-up state.
5	Could not make the LANCE RAP point to CSR1.
6	Could not make the LANCE RAP point to CSR2.
7	Could not make the LANCE RAP point to CSR3.
8	LANCE CSR3 not initialized correctly upon power-up.
9	LANCE CSR1 has stuck bit.
10	LANCE CSR2 has stuck bit.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 2—Initialization Subtest	
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE Interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 3—Transmit/Receive Subtest	
1	Packet not transmitted before timeout (TINT bit not set in CSR0).
2	Packet not received before timeout (RINT bit not set in CSR0).
3	Unexpected bit set in CSR0 (possible error bit).
4	CRC longword transmitted by the LANCE is incorrect.
63	TMD0 not as expected.
64	TMD1 not as expected.
65	TMD2 not as expected.
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
71	Receive buffer does not equal transmit buffer.
90	Unexpected machine check.
91	Unexpected interrupt.

Table F–1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 15—LANCE Test
Subtest 4—CRC Logic Subtest	
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
2	Packet not transmitted before timeout (TINT bit not set in CSR0).
3	Packet not received before timeout (RINT bit not set in CSR0).
4	Unexpected bit set in CSR0 (possible error bit).
5	Received message byte count (RMD3) is incorrect.
6	Packet not transmitted before timeout (TINT bit not set in CSR0).
7	Packet not received before timeout (RINT bit not set in CSR0).
8	Unexpected bit set in CSR0 (possible error bit).
9	Expected CRC error not indicated in RMD1.
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE Interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
63	TMD0 not as expected.
64	TMD1 not as expected.
65	TMD2 not as expected.
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
71	Receive buffer does not equal transmit buffer.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 5—Collision Detect Logic Subtest	
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
2	Expected Retry error not received (RTRY bit in TMD3 not set).
3	Unexpected bit set in CSR0 (possible error).
4	Error bit not set in TMD1 on expected retry error, or other error bits set.
5	Retry bit not set in TMD3 on expected retry error, or other error.
6	LANCE relinquished ownership of receive ring entry on expected retry error. The LANCE should not have received this packet and should therefore still own the receive ring entry.
7	The receive message byte count is not cleared, indicating that the LANCE received a packet on the expected retry error.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 15—LANCE Test
8	Initialization did not complete before timeout (IDON bit not set in CSR0).
9	Expected Retry error not received (RTRY bit in TMD3 not set).
10	Unexpected bit set in CSR0 (possible error bit).
11	Error bit not set in TMD1 on expected retry error, or other error bits set.
12	Retry bit not set in TMD3 on expected retry error, or other error bits set.
13	LANCE relinquished ownership of receive ring entry on expected retry error. The LANCE should not have received this packet and should therefore still own the receive ring entry.
14	The receive message byte count is not cleared, indicating that the LANCE received a packet on the expected retry error.
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE Interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
63	TMD0 not as expected.
64	TMD1 not as expected.
65	TMD2 not as expected.
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 6—Promiscuous Mode Subtest	
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
2	Packet not transmitted before timeout (TINT bit not set in CSR0).
3	Packet not received before timeout (RINT bit not set in CSR0).
4	Unexpected bit set in CSR0 (possible error bit).
5	CRC longword transmitted by the LANCE is incorrect.
6	Initialization did not complete before timeout (IDON bit not set in CSR0).
7	Packet not transmitted before timeout (TINT bit not set in CSR0).
8	Packet not received before timeout (RINT bit not set in CSR0).
9	Unexpected bit set in CSR0 (possible error bit).
10	CRC longword transmitted by the LANCE is incorrect.
11	Initialization did not complete before timeout (IDON bit not set in CSR0).
12	Packet not transmitted before timeout (TINT not set in CSR0).

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 15—LANCE Test
13	Misaddressed packet was received.
14	Unexpected bit set in CSR0 (possible error bit).
15	LANCE relinquished ownership of receive ring entry on misaddressed packet. The LANCE should not have received this packet and should therefore still own the receive ring entry.
16	The receive message byte count is not cleared, indicating that the LANCE received the misaddressed packet.
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
63	TMD0 not as expected.
64	TMD1 not as expected.
65	TMD2 not as expected.
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
71	Receive buffer does not equal transmit buffer.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 7—DTX Logic Subtest	
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
2	Packet was transmitted (TINT bit set in CSR0) even though the transmitter was turned off.
3	Unexpected bit set in CSR0 (possible error bit).
4	LANCE relinquished ownership of transmit ring entry even though it should not have transmitted the packet since the transmitter was turned off.
5	LANCE relinquished ownership of receive ring entry on a packet that was never transmitted. The LANCE should not have received this packet and should therefore still own the receive ring entry.
6	The receive message byte count is not cleared, indicating that the LANCE received the non-transmitted packet.
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE Interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
63	TMD0 not as expected.
64	TMD1 not as expected.
65	TMD2 not as expected.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 15—LANCE Test
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 8—DRX Logic Subtest	
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
2	Packet not transmitted before timeout (TINT not set in CSR0).
3	Packet was received (RINT set in CSR0) even though the receiver was turned off.
4	Unexpected bit set in CSR0 (possible error bit).
5	LANCE relinquished ownership of receive ring entry on a packet that it should not have received and should therefore still own the receive ring entry.
6	The receive message byte count is not cleared, indicating that the LANCE received a packet that it should not have received.
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE Interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
63	TMD0 not as expected.
64	TMD1 not as expected.
65	TMD2 not as expected.
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 9—Logical Address Filter Subtest	
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
2	Packet not transmitted before timeout (TINT bit not set in CSR0).
3	Packet not received before timeout (RINT bit not set in CSR0).
4	Unexpected bit set in CSR0 (possible error bit).

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 15—LANCE Test
5	CRC longword transmitted by the LANCE is incorrect.
6	Initialization did not complete before timeout (IDON bit not set in in CSR0).
7	Packet not transmitted before timeout (TINT bit not set in CSR0).
8	Packet was received that should have been rejected by the logical address filter.
9	Unexpected bit set in CSR0 (possible error bit).
10	LANCE relinquished ownership of receive ring entry on a packet that it should not have received and should therefore still own the receive ring entry.
11	The receive message byte count is not cleared, indicating that the LANCE received a packet that it should not have received.
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE Interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
63	TMD0 not as expected.
64	TMD1 not as expected.
65	TMD2 not as expected.
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
71	Receive buffer does not equal transmit buffer.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 10—Missed Packet Subtest	
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
2	Packet not transmitted before timeout (TINT bit not set in CSR0).
3	Packet was received (RINT bit set in CSR0) even though no receive ring entries were available.
4	Unexpected bit set in CSR0 (possible error bit).
5	LANCE had either error or STP and ENP set in RMD1.
6	The receive message byte count is not cleared, indicating that the LANCE received a packet that it should not have received.
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE Interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
63	TMD0 not as expected.
64	TMD1 not as expected.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 15—LANCE Test
65	TMD2 not as expected.
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
71	Receive buffer does not equal transmit buffer.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 11—Broadcast Mode Subtest	
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
2	Packet not transmitted before timeout (TINT bit not set in CSR0).
3	Packet not received before timeout (RINT bit not set in CSR0).
4	Unexpected bit set in CSR0 (possible error bit).
5	CRC longword transmitted by the LANCE is incorrect.
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE Interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
63	TMD0 not as expected.
64	TMD1 not as expected.
65	TMD2 not as expected.
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
71	Receive buffer does not equal transmit buffer.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 12—Hard Reset Subtest	
1	Expected machine check not taken when LANCE register written when LANCE taken off the XNADAL bus.
2	Bus Timeout Register not as expected after machine check.
3	LANCE RAP not initialized correctly upon power-up.

Table F–1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 15—LANCE Test
4	Could not make the LANCE RAP point to CSR0.
5	LANCE CSR0 not initialized correctly upon power-up.
6	Could not initialize LANCE CSR0 to its power-up state.
90	Unexpected machine check.
91	Unexpected interrupt.
	Ethernet Subsystem Parity Test
1	Initialization did not complete before timeout (IDON bit not set in CSR0).
2	Expected memory error not received (MERR bit in CSR0 not set).
3	CSR0 not as expected after the memory error.
4	Initialization did not complete before timeout (IDON bit not set in CSR0).
5	Packet was not transmitted before timeout (TINT bit not set in CSR0).
6	Packet was not received before timeout (RINT bit not set in CSR0).
7	CSR0 not as expected after TINT and RINT.
8	LANCE calculated CRC is incorrect.
9	Machine check taken on receive buffer read, indicating that the Ethernet parity logic did not write good parity.
60	Extra bits set in CSR0 after initialization.
61	IDON and INTR bits not cleared in CSR0 after writing IDON.
62	LANCE Interrupt bit set in Diagnostic Register after source of interrupt (IDON bit) was cleared.
63	TMD0 not as expected.
64	TMD1 not as expected.
65	TMD2 not as expected.
66	TMD3 not as expected.
67	RMD0 not as expected.
68	RMD1 not as expected.
69	RMD2 not as expected.
70	RMD3 not as expected.
71	Receive buffer does not equal transmit buffer.
90	Unexpected machine check.
91	Unexpected interrupt.
	Test 17—LANCE External Loopback Test
—	No errors reported by this test. The test provides only a pass or fail status indication.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 18—Gate Array Test
Subtest 1—XNA Side Register Access Subtest	
1	Gate array register not initialized correctly upon power-up.
2	Machine check accessing gate array register.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 2—XNA Side Register Write Subtest	
1	Gate array register bit not written correctly.
2	Machine check while writing gate array register.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 3—Peek Subtest	
1	Peek timed out.
2	Gate Array Busy bit not cleared in GACSR.
3	The Done bit is not set or is not the only bit set in PKXMIH.
4	PKXMIL was corrupted during the peek.
5	PKDATA does not contain the correct data.
6	PKDATB does not contain the correct data.
7	Correct bit(s) not set/cleared in Diagnostic Register.
8	XBER reported an error.
9	Diagnostic Register bits not cleared after Peek Done bit was cleared.
10	Peek timed out.
11	The Done and Write bits are not set or are not the only bits set in PKXMIH.
12	PKXMIL was corrupted during the peek.
13	PKDATA does not contain the correct data.
14	PKDATB does not contain the correct data.
15	XPD1 not written correctly.
16	Correct bit(s) not set/cleared in Diagnostic Register.
16	XBER reported an error.
17	Diagnostic Register bits not cleared after Peek Done bit was cleared.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 4—XMI Side Register Access Subtest	
1	Peek timed out.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 18—Gate Array Test
2	The Done bit is not set or is the only bit set in PKXMIH0.
3	XBER reported an error.
4	Peek timed out.
5	The Done and Write bits are not set or are not the only bits set in PKXMIH0.
6	XBER reported an error.
7	The correct register written bit is not set or is not the only bit set in the GACSR.
8	Port Register Written bit not set in Diagnostic Register.
9	Port Register Written bit not cleared in Diagnostic Register after Peek Done bit was cleared.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 5—Datamove Subtest	
1	Datamove timed out.
2	Gate Array Busy bit not cleared in GACSR.
3	DMCSR not cleared.
4	Datamove Done bit not set in Diagnostic Register.
5	XBER reported an error.
6	Shared memory was not written correctly.
7	Shared memory was not written correctly.
8	Datamove Done bit not cleared in Diagnostic Register.
9	Zero-byte datamove read timed out.
10	DMCSR not cleared.
11	XBER reported an error.
12	Shared memory was written during zero-byte datamove.
13	Shared memory was written during zero-byte datamove.
14	Zero-byte datamove write timed out.
15	Write bit is not set or is not the only bit set in DMCSR.
16	XBER reported an error.
17	PKDATA0 was written during zero-byte datamove loopback write.
18	PKDATB0 was written during zero-byte datamove loopback write.
19	Datamove timed out.
20	Gate Array Busy bit not cleared in GACSR.
21	Write and Interrupt Enable bits are not set or are not the only bits set in DMCSR.
22	Datamove Done bit not set in Diagnostic Register.
23	XBER reported an error.
24	PKDATA was not written correctly during loopback datamove.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 18—Gate Array Test
25	PKDATA was not written correctly during loopback datamove.
26	Datamove Done bit not cleared in Diagnostic Register after the Interrupt Enable bit was cleared in DMCSR.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 6—Parity Subtest	
1	Datamove timed out.
2	Error and Parity error not set in DMCSR.
3	Datamove Done bit set in Diagnostic Register.
4	XBER reported an error.
5	PKDATA should not have been written.
6	PKDATB should not have been written.
7	Datamove timed out.
8	Write bit is not set or is not the only bit set in DMCSR.
9	XBER reported an error.
10	PKDATA was not written correctly.
11	PKDATB was not written correctly.
12	Datamove timed out.
13	DMCSR not cleared.
14	Datamove Done bit set in Diagnostic Register.
15	XBER reported an error.
16	Shared Memory was not written correctly.
17	Shared Memory was not written correctly.
18	Machine check taken while reading Shared Memory that was written by the gate array.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 7—Initialization Subtest	
1	GACSR Busy bit not set.
2	Init bit in GACSR did not clear after initialization.
3	Gate array register was altered on initialization when it should not have been.
4	Gate array register was altered on initialization when it should not have been.
5	Gate array register was not initialized correctly.
6	Gate array register was not initialized correctly.
90	Unexpected machine check.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 18—Gate Array Test
91	Unexpected interrupt.
Subtest 8—Error Subtest	
1	Expected parity error not received in XBER.
2	Parity error not cleared in XBER when Force Bad Parity bit cleared.
3	Diagnostic Register peek error not set.
4	Diagnostic Register peek error not cleared.
5	Peek timed out.
6	Expected error not received in PKXMIH.
7	PKXMIL was corrupted during the peek.
8	PKDATA was written during the error peek.
9	PKDATB was written during the error peek.
10	Correct bit(s) not set/cleared in Diagnostic Register.
11	XBER didn't report the expected error.
12	Diagnostic Register bits not cleared after Peek Done bit was cleared.
13	XBER error not cleared when Force RSE bit cleared.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 9—Interrupt Subtest	
1	Peek timed out.
2	Port Register Written bit not set in Diagnostic Register.
3	Port Register Written interrupt not taken.
4	Datamove timed out.
5	Datamove error.
6	Datamove Done bit not set in Diagnostic Register.
7	Datamove Done interrupt not taken.
90	Unexpected machine check.
91	Unexpected interrupt.
Subtest 10—Halt Subtest	
1	Halt taken when in halt-protected space.
2	Expected halt not taken.
3	Node Reset interrupt taken when not expected.
4	Node Reset interrupt taken when not expected.
90	Unexpected machine check.

Table F-1 (Cont.) Error Numbers for Self-Test RBD (ST0)

Error Number	Description
	Test 18—Gate Array Test
91	Unexpected interrupt.
Subtest 11—Hard Reset Subtest	
1	Gate array register not initialized correctly on hard reset.
90	Unexpected machine check.
91	Unexpected interrupt.

Table F-2 Error Numbers for NI RBD (ST1)

Error Number	Description
	Test 1—External Loopback on Live Ethernet
1	Packet not transmitted (too many LANCE hangs in a row).
2	Packet not received (too many LANCE silo-pointer misalignment errors in a row).
60	Extra bits set in CSR0 after initialization.
61	MERR bit set in CSR0.
62	BABL bit set in CSR0.
63	MISS bit set in CSR0.
64	CERR bit set in CSR0.
68	UFLO bit set in TMD3.
69	BUFF bit set in TMD3.
70	RTRY bit set in TMD3 (too many RTRYs in a row). BUFF set in TMD3.
71	LCAR bit set in TMD3 (too many LCARs in a row).
72	LCOL bit set in TMD3 (too many LCOLs in a row).
73	Too many silo-pointer misalignment bug induced errors in a row due to invalid message byte count (RMD3).
74	OFLO bit set in RMD1 (too many OFLOs in a row).
75	CRC bit set in RMD1 (too many CRCs in a row).
76	FRAM bit set in RMD1 (too many FRAMs in a row).
77	BUFF bit set in RMD1 (too many BUFFs in a row).
78	Too many silo-pointer misalignment bug induced errors in a row due to buffer-compare errors.
90	Unexpected machine check.
91	Unexpected interrupt.

Table F-2 (Cont.) Error Numbers for NI RBD (ST1)

Error Number	Description
	Test 2—MOP Loopback Test
1	No nodes responding to MOP loopback packets.
2	Packet not transmitted.
3	No more nodes responding to MOP loopback packets.
60	Extra bits set in CSR0 after initialization.
61	MERR bit set in CSR0.
62	BABL bit set in CSR0.
63	MISS bit set in CSR0.
64	CERR bit set in CSR0.
68	UFLO bit set in TMD3.
69	BUFF bit set in TMD3.
70	RTRY bit set in TMD3 (too many RTRYs in a row). BUFF set in TMD3.
71	LCAR bit set in TMD3 (too many LCARs in a row).
72	LCOL bit set in TMD3 (too many LCOLs in a row).
73	Too many silo-pointer misalignment bug induced errors in a row due to invalid message byte count (RMD3).
74	OFLO bit set in RMD1 (too many OFLOs in a row).
75	CRC bit set in RMD1 (too many CRCs in a row).
76	FRAM bit set in RMD1 (too many FRAMs in a row).
77	BUFF bit set in RMD1 (too many BUFFs in a row).
78	Too many silo-pointer misalignment bug induced errors in a row due to buffer-compare errors.
90	Unexpected machine check.
91	Unexpected interrupt
	Test 3—External Loopback on Closed Ethernet Test
1	Packet not transmitted.
2	Packet not received.
60	Extra bits set in CSR0 after initialization.
61	MERR bit set in CSR0.
62	BABL bit set in CSR0.
63	MISS bit set in CSR0.
65	DEF bit set in TMD1.
66	ONE bit set in TMD1.
67	MORE bit set in TMD1.
68	UFLO bit set in TMD3.
69	BUFF bit set in TMD3.
70	RTRY bit set in TMD3 (too many RTRYs in a row). BUFF set in TMD3.

Table F-2 (Cont.) Error Numbers for NI RBD (ST1)

Error Number	Description
	Test 3—External Loopback on Closed Ethernet Test
71	LCAR bit set in TMD3 (too many LCARs in a row).
72	LCOL bit set in TMD3 (too many LCOLs in a row).
73	Too many silo-pointer misalignment bug induced errors in a row due to invalid message byte count (RMD3).
74	OFLO bit set in RMD1 (too many OFLOs in a row).
75	CRC bit set in RMD1 (too many CRCs in a row).
76	FRAM bit set in RMD1 (too many FRAMs in a row).
77	BUFF bit set in RMD1 (too many BUFFs in a row).
78	Too many silo-pointer misalignment bug induced errors in a row due to buffer-compare errors.
90	Unexpected machine check.
91	Unexpected interrupt.

Table F-3 Error Numbers for XMI RBD (ST2)

Error Number	Description
	Test 1—Datamove/Peek Test
1	Datamove timeout.
2	Expected node reset interrupt not received.
60	Datamove write error.
61	Error reported in XBER.
62	Datamove read error.
63	Error reported in XBER.
64	Datamove read data not equal to datamove write data.
65	Peek write timeout after retries.
66	Peek write error.
67	Error reported in XBER.
68	Peek write timeout after retries.
69	Peek write error.
70	Error reported in XBER.
71	Peek read timeout after retries.
72	Peek read error.
73	Error reported in XBER.
74	Peek read data not equal to peek write data.
75	Peek read data not equal to peek write data.
76	Peek read timeout after retries.
77	Peek read error.

Table F-3 (Cont.) Error Numbers for XMI RBD (ST2)

Error Number	Description
	Test 1—Datamove/Peek Test
78	Error reported in XBER.
79	Peek read data not equal to peek write data.
80	Peek read data not equal to peek write data.
91	Unexpected interrupt.
92	Sanity timer expired. The gate array is hung.

Table F-4 Error Numbers for XNA RBD (ST3)

Error Number	Description
	Test 1—Module Exerciser Test
60	Extra bits set in CSR0 after initialization.
61	MERR bit set in CSR0.
62	BABL bit set in CSR0.
63	MISS bit set in CSR0.
64	CERR bit set in CSR2.
68	UFLO bit set in TMD3.
69	BUFF bit set in TMD3.
70	RTRY bit set in TMD3 (too many RTRYs in a row). BUFF set in TMD3.
71	LCAR bit set in TMD3 (too many LCARs in a row).
72	LCOL bit set in TMD3 (too many LCOLs in a row).
73	Too many silo-pointer misalignment bug induced errors in a row due to invalid message byte count (RMD3).
74	OFLO bit set in RMD1 (too many OFLOs in a row).
75	CRC bit set in RMD1 (too many CRCs in a row).
76	FRAM bit set in RMD1 (too many FRAMs in a row).
77	BUFF bit set in RMD1 (too many BUFFs in a row).
78	Too many silo-pointer misalignment bug induced errors in a row due to buffer-compare errors.
90	Unexpected machine check.
91	Unexpected interrupt.
92	Datamove timed out.
93	Ethernet packet not transmitted (too many LANCE hangs in a row).
94	Ethernet packet not received (too many silo-pointer misalignment errors in a row).
95	Datamove write or read error.
96	Error reported in XBER.
97	Datamove read data not equal to datamove write data.

APPENDIX G

G Node-Private Registers in Nonfatal Error Blocks

This appendix provides bit maps of the DEMNA registers that are not visible over the XMI bus but that are included in the DEMNA nonfatal error blocks. (See Section 8.2.2 for more information on the nonfatal error blocks.)

The registers are grouped as follows:

- Datamove Registers
- Peek Registers
- Gate Array Registers
- Diagnostic Register

G.1 Datamove Registers

The gate array has four sets of datamove registers numbered 0 through 3. Each set corresponds to a separate datamove transaction. Up to four datamove transactions can be outstanding at a time.

Each set of datamove registers contains the following registers:

- Datamove Port Address Register (DMPOR*n*)
- Datamove Control/Status Register (DMCSR*n*)
- Datamove XMI Address Register (DMXMI*n*)
- Datamove Next Page Address Register (DMNPA*n*)

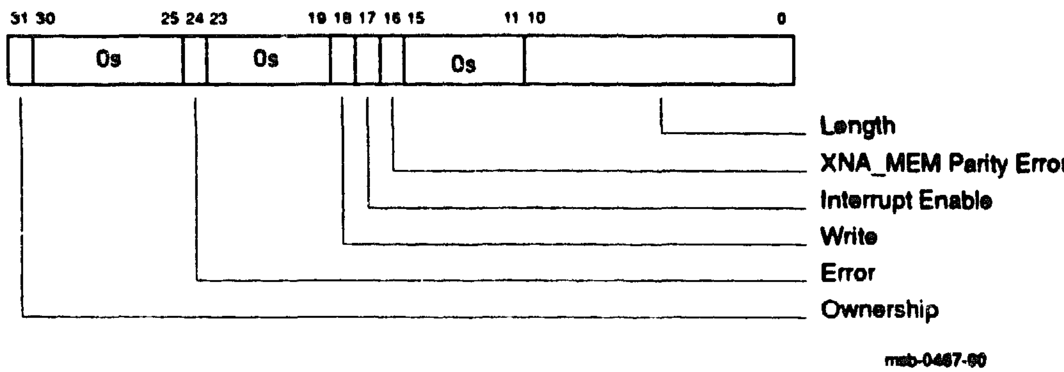
where *n* is an integer from 0 through 3 that indicates the set to which the register belongs.

G.1.1 Datamove Port Address Register (DMPOR*n*)

31 23	22 18	17 0
DMXMIL <8:0>	0s	Port Address <17:0>

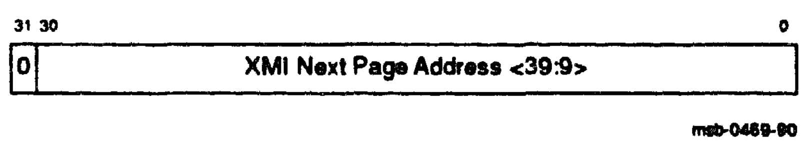
msb-0466-90

G.1.2 Datamove Control/Status Register (DMCSR*n*)



G.1.3 Datamove XMI Address Register (DMXMI*n*)

G.1.4 Datamove Next Page Address Register (DMNPA*n*)



G.2 Peek Registers

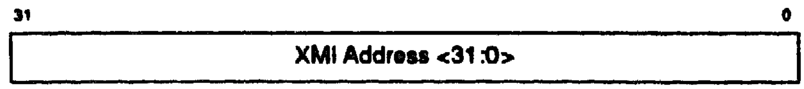
The gate array has two sets of peek registers numbered 0 and 1. Each set corresponds to a separate peek transaction. Up to two peek transactions can be outstanding at a time.

Each set of peek registers contains the following registers:

- Peek XMI Low Address Register (PKXMIL*n*)
- Peek XMI High Address Register (PKXMIH*n*)
- Peek Data A Register (PKDATA*n*)
- Peek Data B Register (PKDATB*n*)

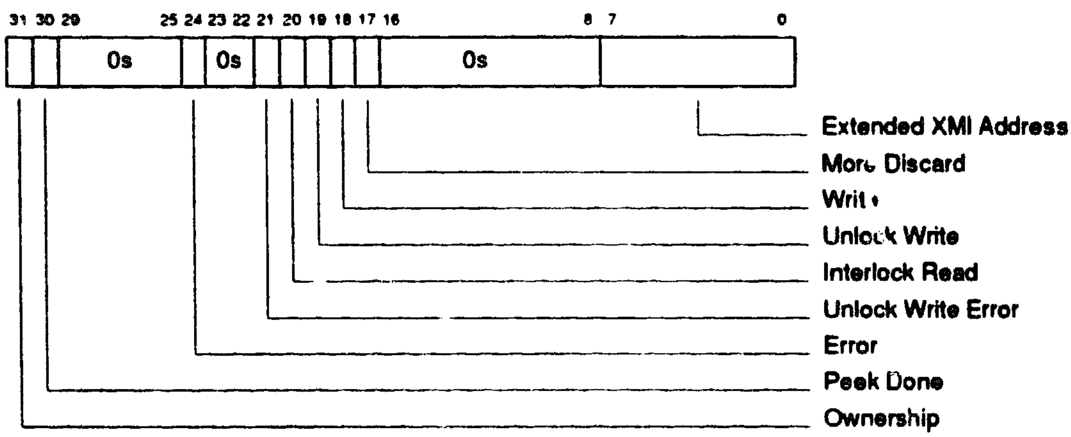
where *n* is 0 or 1, indicating the set to which the register belongs.

G.2.1 Peek XMI Low Address Register (PKXMIL*n*)



msb-0473-90

G.2.2 Peek XMI High Address Register (PKXMIH*n*)



msb-0475-90

G.2.3 Peek Data A Register (PKDATA*n*)

msb-0474-90

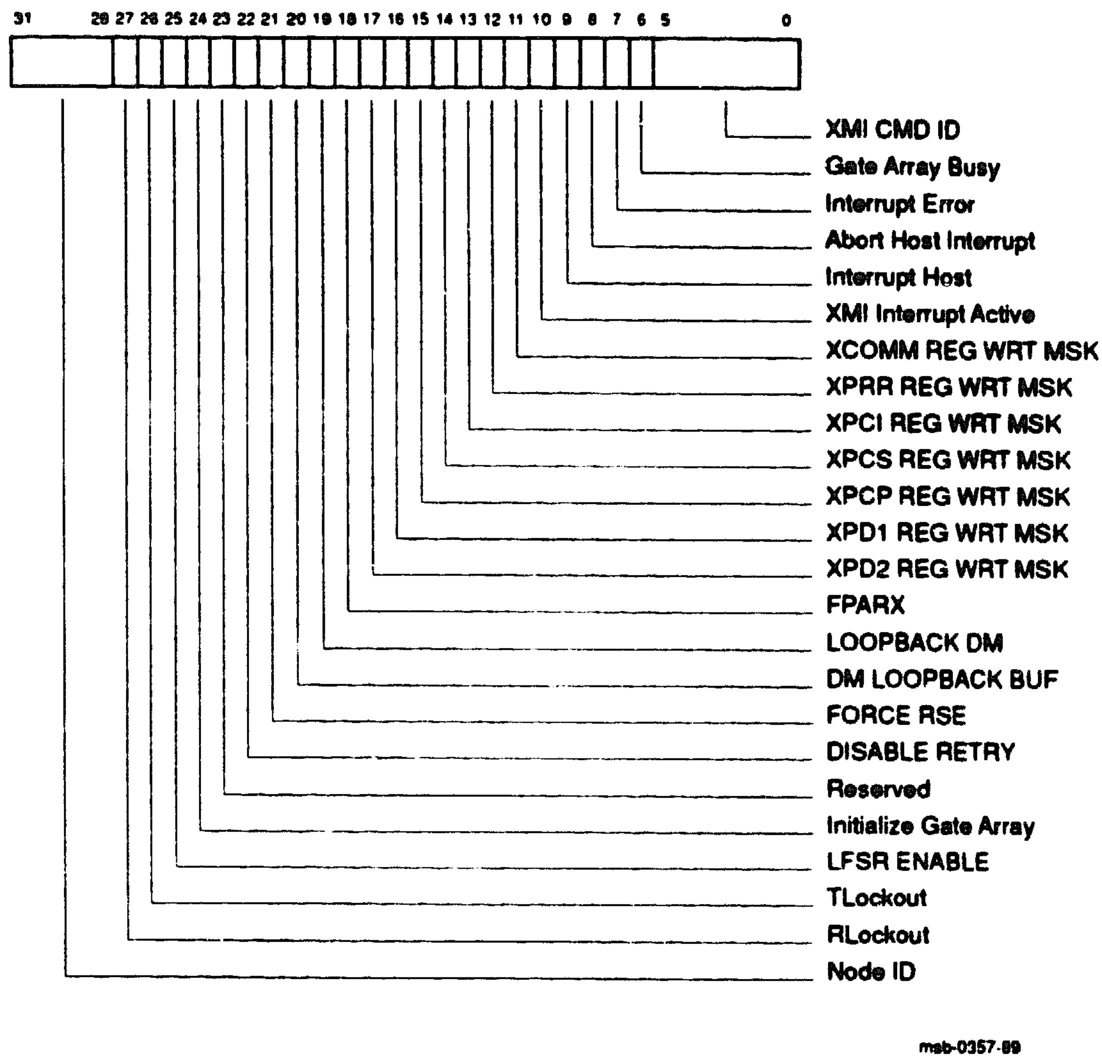
G.2.4 Peek Data B Register (PKDATB*n*)

G.3 Gate Array Registers

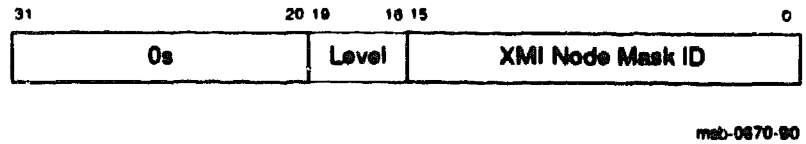
The gate array registers control gate array functions, provide gate array status information, and are used in host interrupt operations. The gate array registers are as follows:

- Gate Array Control/Status Register (GACSR)
- Gate Array Host Interrupt Register (GAHIR)
- Gate Array IDENT Vector Register (GAIVR)
- Gate Array Timer Register (GATMR)

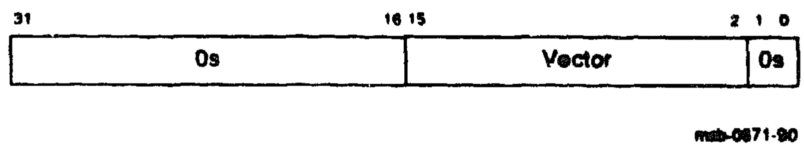
G.3.1 Gate Array Control/Status Register (GACSR)



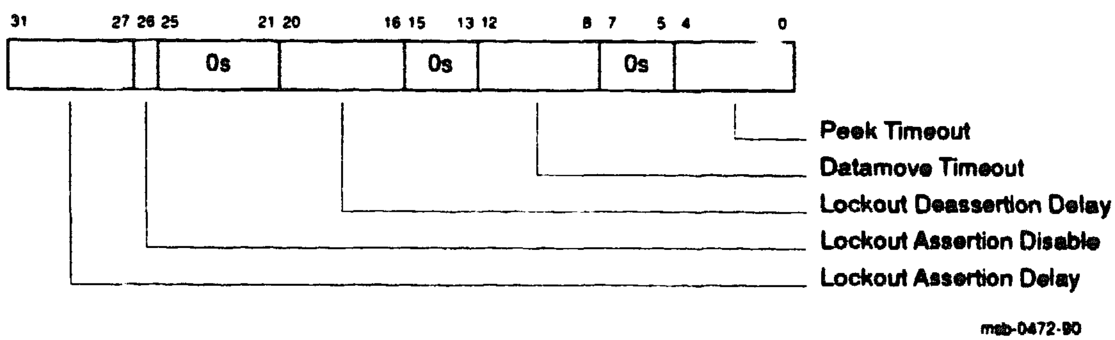
G.3.2 Gate Array Host Interrupt Register (GAHIR)



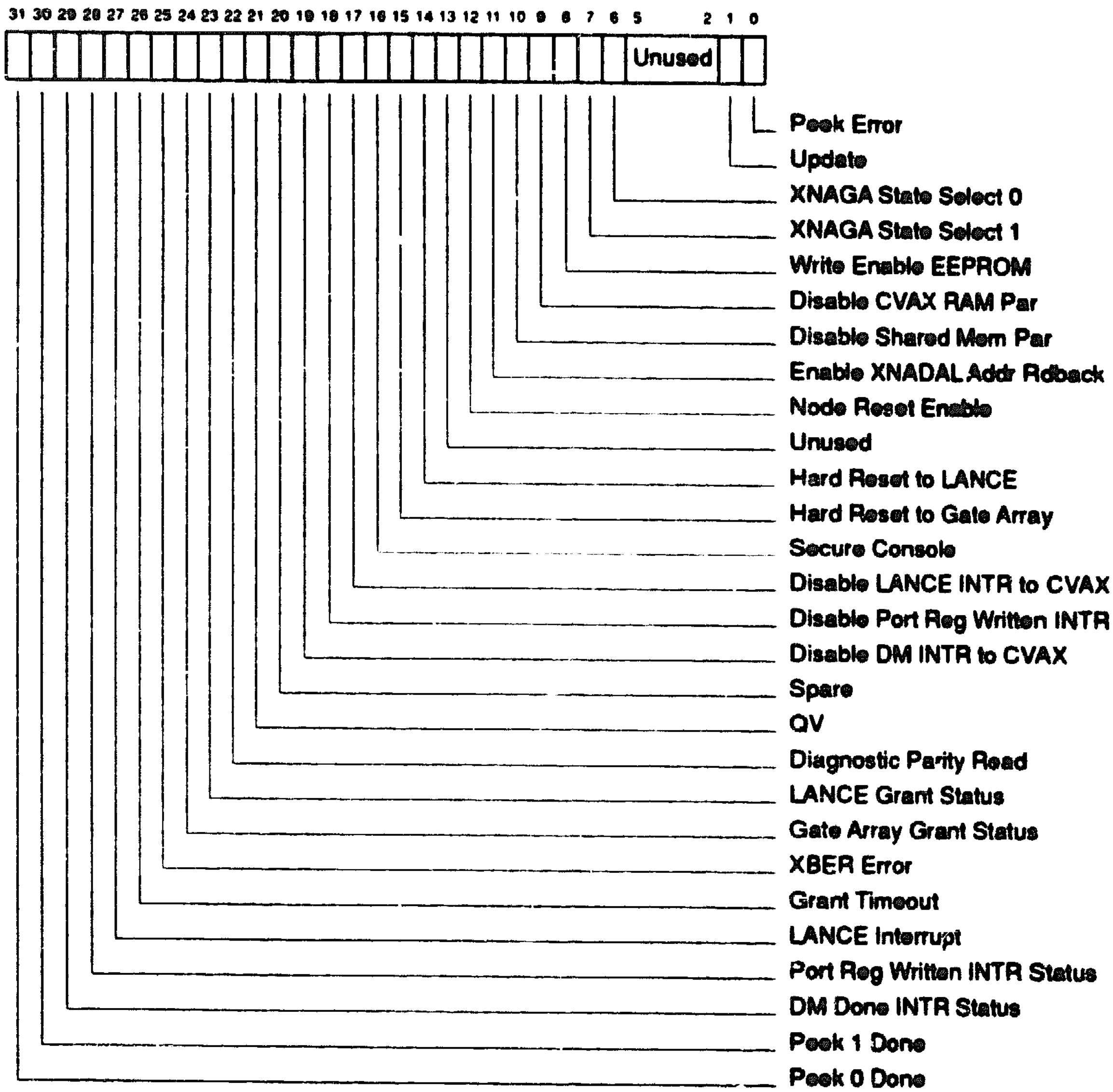
G.3.3 Gate Array IDENT Vector Register (GAIVR)



G.3.4 Gate Array Timer Register (GATMR)



G.4 Diagnostic Register



msb-0522-90

APPENDIX H

H Device Type Codes for XMI Modules

Table H–1 lists the device type codes for XMI modules available at the printing of this manual.

Table H–1 Device Type Codes for XMI Modules

Code	Device	Function
0C03	DEMNA	Ethernet/802 controller
0C05	CIXCD	CI interface adapter
0C22	KDM70	Disk and tape controller
1001	XJA	XMI-to-SCU adapter
2001	DWMBA/A	XMI-to-VAXBI adapter (unmapped)
2002	DWMBB/A	+3.3V XMI-to-VAXBI adapter (mapped)
4001	MS62A	Memory module
8001	KA62A	VAX 6000 Model 200 CPU
8001	KA62B	VAX 6000 Model 300 CPU
8081	KN58A/A	DECsystem 5800 CPU
8082	KA64A	VAX 6000 Model 400 CPU

APPENDIX I

I Ethernet Protocol Types

Table I–1 lists the only cross-company (universally administered) Ethernet protocol type. Table I–2 lists the Ethernet protocol types assigned by Digital.

Table I–1 Cross-Company Ethernet Protocol Type

Protocol Type	Description
90-00	Ethernet loopback

Table I–2 Digital's Ethernet Protocol Types

Protocol Type	Description
60-01	DNA Dump/Load (MOP)
60-02	DNA Remote Console (MOP)
60-03	DNA Routing
60-04	Local Area Transport (LAT)
60-05	Diagnostics
60-06	Customer use
60-07	System Communication Architecture (SCA)
80-38	Bridge
80-3B	VAXELN
80-3C	DNA Naming Service
80-3D	CSMA/CD Encryption
80-3F	LAN Traffic Monitor
80-40	NetBios emulator (PCSG)
80-42	Reserved

The protocol types 00–00 through 05–DC are reserved so that 802.3 format frames can be distinguished from Ethernet format frames. Use of these protocol types in Ethernet format frames is incompatible with correct operation of the CSMA/CD Data Link.

APPENDIX J

J Ethernet Addresses

Table J–1 lists the cross-company (universally administered) Ethernet multicast addresses. Table J–2 lists the Ethernet multicast addresses assigned by Digital. Table J–3 lists the Ethernet physical addresses assigned to Digital prototypes, parts, or units. Table J–4 lists the address blocks assigned to other organizations but used in Digital products.

Table J–1 Cross-Company Multicast Addresses

Multicast Address	Description
01-80-C2-00-00-00	IEEE 802.1d Bridge group address
01-80-C2-00-00-0X	IEEE 802.1d Reserved (always filtered by bridges)
01-80-C2-00-00-10	IEEE 802.1d All LANs Bridge Management group address
01-80-C2-00-00-11	IEEE 802.1e Load Server group address
01-80-C2-00-00-12	IEEE 802.1e Loadable Device group address
09-00-2B-00-00-04	ISO 9542 End System Hello
09-00-2B-00-00-05	ISO 9542 Intermediate System Hello
CF-00-00-00-00-00	Loopback Assistance
FF-FF-FF-FF-FF-FF	Broadcast

Table J–2 Digital's Multicast Addresses

Multicast Address	Description
AA-00-00-01-00-00	DNA Dump/Load Assistance (MOP)
AA-00-00-02-00-00	DNA Remote Console (MOP)
AB-00-00-03-00-00	DNA Level 1 Routing Layer routers
AB-00-00-04-00-00	DNA Routing Layer end nodes
AB-00-04-00-XX-XX	Customer use
AB-00-04-01-XX-XX	System Communication Architecture (SCA)
09-00-2B-00-00-02	VAXELN
09-00-2B-00-00-03	LAN Traffic Monitor
09-00-2B-00-00-06	CSMA/CD Encryption
09-00-2B-00-00-07	NetBios Emulator (PCSG)
09-00-2B-00-00-0F	Local Area Transport (LAT)
09-00-2B-01-00-00	All bridges
09-00-2B-01-00-01	All local bridges
09-00-2B-02-00-00	DNA Level 2 Routing Layer routers
09-00-2B-02-01-00	DNA Naming Service Advertisement
09-00-2B-02-01-01	DNA Naming Service Solicitation

Table J–3 Digital's Physical Addresses

Physical Address	Description
AA-00-04-00-XX-XX	DECnet Phase IV station addresses
AA-00-03-00-XX-XX	UNA prototype
AA-00-03-01-XX-XX	DEUNA products
AA-00-03-02-XX-XX	Miscellaneous assignments
AA-00-03-02-00-00	H4000-TA Ethernet Transceiver Tester
AA-00-03-03-XX-XX	NI20 products
08-00-2B-0X-XX-XX	PROM 23-365A1-00
08-00-2B-1X-XX-XX	PROM 23-365A1-00
08-00-2B-22-00-00	Bridge management

Table J–4 Other Physical Addresses

Physical Address	Description
00-00-69-02-XX-XX	DTQNA, Concord Communications Inc.

APPENDIX K

K SAP Assignments and SNAP Protocol ID Assignments

Table K–1 lists the cross-company (universally administered) SAP assignments. No SAPs are assigned by Digital. Table K–2 lists the SNAP protocol IDs (PIDs) assigned by Digital. There are no cross-company (universally administered) SNAP PIDs.

Table K–1 Cross-Company SAP Assignments

SAP	Description
03	LLC sublayer management function group SAP (IEEE 802.1b)
FF	Global DSAP
00	Null SAP
02	LLC sublayer management function individual SAP (IEEE 802.1b)
06	ARPAnet IP
0E	PROWAY (IEC 955) network management and initialization
42	IEEE 802.1d (ISO 10038) transparent bridge protocol
4E	EIA RS-511 Manufacturing Message Service
7E	ISO 8208 (X.25 over IEEE 802.2 type 2 LLC)
8E	PROWAY (IEC 955) active station list maintenance
AA	SNAP SAP
FE	ISO Network Layer entity

Table K–2 Digital's SNAP Protocol IDs

Protocol ID	Description
08-00-2B-60-01	DNA Dump/Load (MOP)
08-00-2B-60-02	DNA Remote Console (MOP)
08-00-2B-60-03	DNA Routing
08-00-2B-60-04	Local Area Transport (LAT)
08-00-2B-60-05	Diagnostics
08-00-2B-60-06	Customer use
08-00-2B-60-07	System Communication Architecture (SCA)
08-00-2B-80-3B	VAXELN
08-00-2B-80-3C	DNA Naming Service
08-00-2B-80-3D	CSMA/CD Encryption
08-00-2B-80-3F	LAN Traffic Monitor
08-00-2B-80-40	NetBios emulator (PCSG)
08-00-2B-90-00	MOP LAN Loopback protocol

APPENDIX L

L Console Connection Program

This appendix describes a VAX MACRO (assembly language) program called *console.mar*. This program can be used to access the DEMNA console monitor program if the Network Control Program (NCP) is not available. If NCP is available, use the procedure described in Section 10.2.2 to access the DEMNA console monitor program.

L.1 Introduction

The console connection program uses the Maintenance Operations Protocol (MOP) console carrier mechanism to connect to the target DEMNA console. The MOP specification defines the data structures and handshaking conventions used to establish a console connection and pass data back and forth between a user terminal and the DEMNA console. Data is transferred in character I/O mode—that is, one or more ASCII characters is transferred per packet.

The console connection program also makes use of Queue I/O (QIO) structures and commands for data transfer. The QIO interface is described in the *VMS I/O User's Reference Manual. Part II*.

The console connection program defines the following structures:

- A console packet (according to MOP specifications)
- A QIO channel for the Ethernet connection
- A QIO channel for the terminal connection
- A transmit buffer
- A receive buffer

The console connection routine has the following major steps:

1 Establish a connection with the target DEMNA console by sending a MOP Reserve Console message to the target DEMNA. If no response is received within the timeout period, the program aborts.

2 Check for valid user input (a terminated ASCII character) from the terminal QIO channel.

3 If there is valid input from the terminal channel, put it in a MOP Console Command and Poll message and send it to the target DEMNA. If there is no valid input, send a Command and Poll message that has no data.

4 Check for valid DEMNA input from the Ethernet QIO channel.

5 If there is a Console Response and Acknowledge message from the Ethernet channel, print the user data on the terminal. If there is no such message, proceed to the next step.

6 Hibernate (remain idle) for 1/40 second.

7 Loop to step 2 and proceed.

If an error occurs during any part of the program, the program aborts.

L.2 User-Supplied Parameter Values

The user must supply two parameter values to the program:

- The default physical address (DPA) of the target DEMNA. This address must be supplied as six consecutive 2-digit sets of hexadecimal numbers in the RemoteNode data structure defined at the end of the program. The order of the sets of hex numbers should be the order in which the address bytes are transmitted on the network.
- The Ethernet device number. The user supplies the Ethernet device to the program through an ASSIGN command before running the program.
- The console password. If the console password for the DEMNA has been changed, then the password (Vercode) in the program (under MOP parameters) must be changed. Otherwise, the default password supplied in the program must be used.

L.3 Running the Program

Before it can be run, the program must be compiled and linked as follows:

```
$MACRO CONSOLE
$LINK CONSOLE
```

Thereafter, the program is run as follows:

```
$ASSIGN Ethernet_device CONSOLE$DEVICE
$RUN CONSOLE
XNA>
```

where *Ethernet_device* is the device number for user's Ethernet node

When a connection to the target DEMNA is established, the DEMNA console prompt (XNA>) is displayed. Example L–1 shows the commands used to compile, link, and run the console connection program.

Example L–1 Compiling, Linking, and Running the Console Connection Program

```
$MACRO CONSOLE
$LINK CONSOLE
$ASSIGN EXA0 CONSOLE$DEVICE
$RUN CONSOLE
XNA>
```

L.4 Exiting the Program

Exit the program by typing [CTRL/D] or [CTRL/Y].

L.5 Program Listing

```
;****************************************************************
;*                                                              *
;*  DIGITAL ASSUMES NO RESPONSIBILITY TO SUPPORT THE            *
;*  SOFTWARE DESCRIBED IN THIS MODULE, NOR TO ANSWER            *
;*  INQUIRIES ABOUT IT.                                         *
;*                                                              *
;*  THIS SOFTWARE MODULE IS PART OF A TEMPLATE WHICH MAY        *
;*  REQUIRE CUSTOMER MODIFICATIONS TO WORK IN ALL               *
;*  CIRCUMSTANCES.                                              *
;*                                                              *
;****************************************************************

.TITLE CONSOLE - Connect to a node via MOP Console Carrier

;****************************************************************
; System Library Calls
;****************************************************************

.LIBRARY "SYS$LIBRARY:LIB.MLB"

        $IODEF                          ; Define QIO symbols
        $NMADEF                         ; Define network
                                        ;  symbols

;****************************************************************
; Error Macro Definition
;****************************************************************

.MACRO  $ERROR  ?L0                      ; Begin macro definition

        blbs    R0,L0                   ; If no error, proceed
        brw     Exit                    ; If error, exit program
L0:

.ENDM                                   ; End macro definition
```

```
;*************************************************************
; Console connection routine
;*************************************************************

.PSECT  $CODE, PAGE, SHR, NOWRT, PIC

.ENTRY  Console,^M<>

;------------------------------------------------------------
; Assign and start up QIO channels. Send Reserve Console
; message to target DEMNA.
;------------------------------------------------------------

; Assign the channels (terminal and Ethernet)

; Assign terminal channel

        $ASSIGN_S DEVNAM=TermDev,CHAN=TermChan
        $ERROR                          ; Exit on error

; Assign Ethernet channel

        $ASSIGN_S DEVNAM=NIDev,CHAN=NIChan
        $ERROR                          ; Exit on error

; Start the MOP protocol type

        $QIOW_S FUNC=#<IO$_SetMode!IO$M_Ctrl!IO$M_Startup>,-
                CHAN=NIChan,-           ; Select NI channel
                P2=#SetParmDsc          ; Specify MOP protocol
        $ERROR                          ; Exit on error

; Connect to the console. If there is no response, the console
; is reserved or disabled, an incorrect Ethernet device was
; assigned, or an incorrect RemoteNode Address was supplied.

        movb    #13,XmtData             ; Set function code to
                                        ;  "reserve console"
        movq    VerCode,XmtData + 1     ; Set verification code
        $QIOW_S FUNC=#IO$_Writevblk,-   ; Send Reserve Console msg
                CHAN=NIChan,-           ;  Select NI channel
                P1=XmtData,-            ;  Packet data pointer
                P2=#9,-                 ;  9-byte length
                P5=#ConNode             ;  Console node address
        movl    #5,KeepAlive            ;  Initialize keep-alive
                                        ;   count

;------------------------------------------------------------
; Loop forever, passing data between user terminal and DEMNA
; console.
;------------------------------------------------------------

; Obtain terminal input

0$:     $QIOW_S FUNC=#IO$_READVBLK!IO$M_TIMED!IO$M_NOECHO!IO$M_NOFILTR,-
                CHAN=TermChan,-         ; Select terminal channel
                P1=XmtData+2,-          ; Input data location
                P2=#253,-               ; Maximum length
                P3=#0,-                 ; Flush input buffer
                IOSB=QioIOSB            ; Status location
        movzwl  QioIOSB + 2,R6          ; Get input data length
        blbc    QioIOSB,1$              ; SS$Normal? No, skip terminal
                                        ;  input and send blank packet.
        incl    R6                      ; Yes, append termination
                                        ;  character
1$:     beql    2$                      ; Any data?
        cmpb    XmtData + 2,#4          ; Yes, check for disconnect
        bneq    2$                      ;  (CTRL/D typed).
        brw     Exit                    ; If CTRL/D, exit

; Issue Console Command and Poll message.
```

```
2$:     movw    #17,XmtData                 ; Set MOP code; clear flags
        addl2   #2,R6                       ; Include code/flags bytes
        $QIOW_S FUNC=#IO$_Writevblk,-       ; Send the message
                CHAN=NIChan,-               ;   Select NI channel
                P1=XmtData,-                ;   Packet data pointer
                P2=R6,-                     ;   R6 contains length
                P5=#ConNode                 ;   Console node address

; Receive Console Response and Acknowledge message.

        $QIOW_S FUNC=#IO$_Readvblk!IO$M_Now,-  ; Attempt rcv
        CHAN=NIChan,-                       ;  Select NI channel
        P1=RcvData,-                        ;  Packet data pointer
        P2=#500,-                           ;  Maximum length
        IOSB=QioIOSB                        ;  Status return
        blbc    R0,4$
        blbs    QioIOSB,3$                  ; Got packet? No, skip rcv.
        decl    KeepAlive                   ; Decrement keep-alive cntr
        bgtr    4$                          ; Keep alive expired?
        brw     Exit_NoResponse             ; Yes, exit
3$:     movl    #5,KeepAlive                ; Reset keep-alive counter
        cmpb    RcvData,#19                 ; Console Response/Ack rec'd?
        bneq    4$                          ; No, hibernate.

; Print terminal output

        movzwl  QioIOSB + 2,R6              ; Get length of rcv packet
        subl2   #2,R6                       ; Subtract code/flags byte
        bleq    4$                          ; Any data? No, hibernate
        $QIOW_S FUNC=#IO$_WRITEVBLK,-       ; Yes, copy data to terminal.
                CHAN=TermChan,-             ;  Specify terminal
                P1=RcvData+2,-              ;  Location of output data
                P2=R6                       ;  Output length

; Hibernate for 1/10 second

 $      movl    #-1000000,TimeHib           ; Set hibernation time
        movl    #-1,TimeHib + 4             ;  to 1/10 second
        $SCHDWK_S DAYTIM = TimeHib          ; Schedule wake-up call
        $HIBER_S                            ; Hibernate
        brw     0$                          ; Loop to beginning
                                            ;  of routine

;-----------------------------------------------------------
; Done
;-----------------------------------------------------------

Exit:   $Exit_S R0                          ; Exit status of a QIO
                                              command

        Exit_NoResponse:                    ; Write "No response"
        $QIOW_S CHAN=TermChan,-             ;  to terminal
        FUNC=#IO$_WRITEVBLK,-
        IOSB=QioIOSB,-                      ; General IOSB
        P1=Disconnect,-                     ; Message
        P2=#13                              ; Message length
        $Exit_S                             ; Exit
```

```
;*************************************************************
; Data structure definitions
;*************************************************************

.PSECT $DATA, PAGE, PIC, CON, LCL, NOSHR, NOEXE, RD, WRT, NOVEC

;-------------------------------------------------------------
; Device descriptors
;-------------------------------------------------------------

TermDev:      .ascid  /SYS$INPUT/          ; Terminal device
                                           ;  = SYS$INPUT
TermChan:     .long                        ; Terminal channel

NIDev:        .ascid  /Console$Device/     ; NI device =
                                           ;  Console$Device
NIChan:       .long                        ; NI channel

XmtData:      .blkb       512              ; Transmit (xmt) buffer
RcvData:      .blkb       512              ; Receive (rcv) buffer

QioIOSB:      .blkl       2                ; General IOSB

;-------------------------------------------------------------
; Start up MOP parameters
;-------------------------------------------------------------

SetParm:      .word   NMA$C_PCLI_BFN       ; Number of rcv buffers
              .long   4
              .word   NMA$C_PCLI_PAD       ; Padding value
              .long   NMA$C_STATE_ON
              .word   NMA$C_PCLI_PTY       ; Protocol type = 60-02
              .long   ^X0260

SetParmDsc::      .long           SetParmDsc-SetParm
                  .address        SetParm

; Miscellaneous data

ConNode::     .byte   ^X08,^X00,^X2B,-     ; Ethernet address of
                      ^X00,^X00,^X01       ;  target node
VerCode:      .ascii  /DRAOBANX/           ; Verification code
TimeHib:      .quad                        ; Hibernation time
KeepAlive:    .long                        ; Keep-alive counter
Disconnect:   .byte   13                   ; Disconnect message
              .ascii  /No response/
              .byte   13

.END Console
```

APPENDIX M

M History Entry Formats

This appendix describes the types of history entries that can be displayed by the console monitor program. A history entry must be one of the following error types:

- Datamove—An error specific to a datamove operation.
- Exception—A firmware exception.
- Fatal error—A fatal port error.
- Firmware update—An update to the controller firmware.
- Machine check—A firmware machine check.
- Node halt—The controller executed a node halt.
- No error—No error has been logged to this entry.
- Peek—An error specific to a peek operation.
- Self-test error—One or more of the tests in the self-test failed.
- XBER—One of the hard error bits in the XMI Bus Error Register (XBER) was set.

Figure M–1 shows the general format of a history entry. Table M–1 through Table M–4 describe the history entry fields.

Figure M–1 History Entry Format

```
-- 08-00-2B-00-00-01 -- Error History #  1 -- 01-JAN-1988 01:05:17 --

Type:   Machine Check
Date:   01-JAN-1988 00:00:54
Number of times this event occurred:    1
Saved Data:    00000B01
               00009014
               00000080
               EFF00004
               D0E0FC80
```

Table M–1 History Entry—Parameter Definitions

Parameter	Description
Ethernet address	The DEMNA's actual physical address (APA).
Error History #	The number of the history entry.

Table M–1 (Cont.) History Entry—Parameter Definitions

Parameter	Description
Date and time	The current date and time. If the system date/time was specified in the PARAM command, the date/time is the base time supplied by the host plus DEMNA uptime until the error occurred. If the system date/time was not specified in the PARAM command, the base date/time defaults to 01-Jan-88. In this case, the date/time of error is the base time (01-Jan-88) plus the DEMNA uptime until the error occurred.
Type	The type of error recorded.
Date	The date and time when the history entry was logged.
Number of times this event occurred	The number of times this particular error type occurred.
Saved data (5–7 longwords)	The meaning of these longwords is specific to the error type. See Table M–2.

Table M–2 Saved Data Definitions

	Datamove Error
Longword	**Description**
1	XBER Register
2	XFADR Register
3	XFAER Register
4	DMPOR Register
5	DMCSR Register

	Exception
Longword	**Description**
1	Pending Port Status Register (XPST_Pending). The value that will be loaded into the XPST Register after the next state change (after error handling has been completed).
2	Pending Port Data 1 Register (XPD1_Pending). The value that will be loaded into the XPD1 Register after the next state change (after error handling has been completed).
3	Address of call to shutdown request.
4	Address of exception.
5	Exception number (offset into system control block (SCB)).

Table M–2 (Cont.) Saved Data Definitions

Fatal Error	
Longword	**Description**
1	Pending Port Status Register (XPST_Pending). The value that will be loaded into the XPST Register after the next state change (after error handling has been completed).
2	Pending Port Data 1 Register (XPD1_Pending). The value that will be loaded into the XPD1 Register after the next state change (after error handling has been completed).
3	Longword 1 of the stack when the error occurred.
4	Longword 2 of the stack when the error occurred.
5	Longword 3 of the stack when the error occurred.

Firmware Update	
Longword	**Description**
1	XDEV Register
2	Four ASCII numbers that indicate the DEMNA firmware revision. For example, 30 31 32 33 (ASCII) = revision 01.23.
3–5	Firmware revision date and time (binary).

Machine Check	
Longword	**Description**
1	Pending Port Status Register (XPST_Pending). The value that will be loaded into the XPST Register after the next state change (after error handling has been completed).
2	Pending Port Data 1 Register (XPD1_Pending). The value that will be loaded into the XPD1 Register after the next state change (after error handling has been completed).
3	Machine check code (usually 80–83, which indicate an invalid address).
4	Most recent memory address.
5	Internal state information 1.

Table M–2 (Cont.) Saved Data Definitions

Node Halt

Longword	Description
1	Pending Port Status Register (XPST_Pending). The value that will be loaded into the XPST Register after the next state change (after error handling has been completed).
2	Pending Port Data 1 Register (XPD1_Pending). The value that will be loaded into the XPD1 Register after the next state change (after error handling has been completed).
3	Longword 1 of the stack when the node halt occurred.
4	Longword 2 of the stack when the node halt occurred.
5	Longword 3 of the stack when the node halt occurred.

No Error

Longword	Description
1	Zeros; undefined
2	Zeros; undefined
3	Zeros; undefined
4	Zeros; undefined
5	Zeros; undefined

Peek Error

Longword	Description
1	XBER Register
2	XFADR Register
3	XFAER Register
4	XMIL Register
5	XMIH Register

Self-Test Error

Longword	Description
1	Expected data
2	Received data
3	System control block (SCB) offset
4	Memory address
5	Program counter (PC) at failure
6	See Figure M–2 and Table M–3
7	See Figure M–3 and Table M–4

Table M-2 (Cont.) Saved Data Definitions

XBER	
Longword	**Description**
1	XBER Register
2	XFADR Register
3	XFAER Register
4	Zeros; undefined
5	Zeros; undefined

Figure M-2 Self-Test Entry—Longword 6 of Saved Data

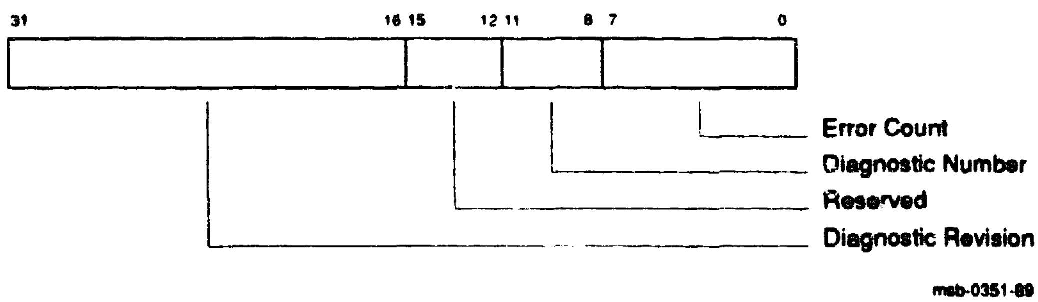


Table M-3 Self-Test Entry—Longword 6 of Saved Data

Bits	Field	Description
31:16	Diagnostic Revision	Two ASCII numbers that indicate the revision level of the diagnostic. For example, 39 33 (ASCII) = revision 3.9.
15:12	Reserved	These bits are reserved.
11:8	Diagnostic Number	A binary field that indicates which test reported the error. 0 = self-test. 1 = NI RBD. 2 = XMI RBD. 3 = XNA RBD.
7:0	Error Count	The number of times (in binary) that this type of diagnostic error occurred.

Figure M–3 Self-Test Entry—Longword 7 of Saved Data

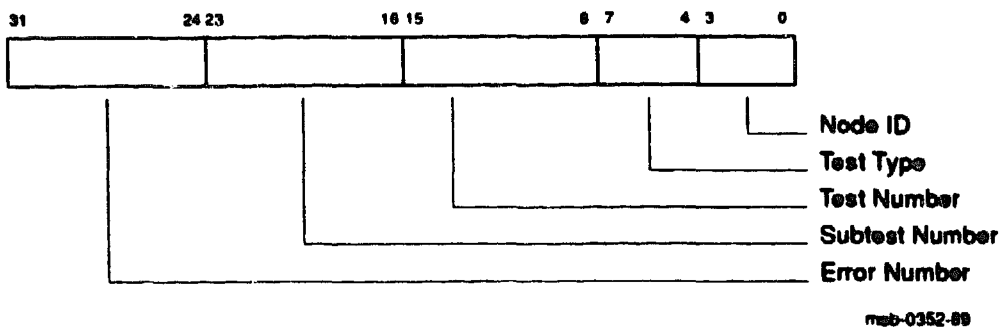


Table M–4 Self-Test Entry—Longword 7 of Saved Data

Bits	Field	Description
31:24	Error Number	See Appendix F.
23:16	Subtest Number	Number (hex) of the failing subtest. (See Appendix F.)
15:8	Test Number	Number (hex) of the failing test. (See Appendix F.)
7:4	Test Type	1 = power-up mode; 2 = RBD mode
3:0	Node ID	XMI node ID (hex) of the DEMNA

INDEX

Index

E

F

G

H

I

K

L

M

N

P

Q

R

S