
DEC LANcontroller
200

DEC LANcontroller 200 Installation Guide

Order Number: EK-DEBNI-IN-002

The DEC LANcontroller 200 adapter (also known as the DEBNI controller) is an Ethernet/802 controller for systems that have a VAXBI bus. This guide is intended for use by DIGITAL customer service representatives and self-maintenance customers who install the DEC LANcontroller 200.

digital equipment corporation
maynard, massachusetts

First Printing, May 1989
Revised, April 1990

Contents

FIGURES

TABLES

PAGE vi INTENTIONALLY LEFT BLANK

Preface

Purpose of This Manual

This manual describes how to install the DEC LANcontroller 200 adapter. The DEC LANcontroller 200 provides an interface between an Ethernet/IEEE 802 local area network and a system that has a VAXBI bus.

This manual also describes the adapter's self-test and gives advice for troubleshooting.

The DEC LANcontroller 200 is also known as the DEBNI controller. Throughout the rest of this manual, the DEC LANcontroller 200 is referred to as the DEBNI.

Intended Audience

This manual is for DIGITAL and customer personnel who install or replace the DEBNI in the field.

Document Structure

This manual has three chapters and five appendixes, which are described below:

Chapter 1 briefly describes the DEBNI module: its functions and what you should have received.

Chapter 2 describes installation of the module.

Chapter 3 describes the module's self-test and how to interpret the results.

Appendix A gives environmental requirements for the DEBNI module.

Appendix B gives register information.

Appendix C describes booting with the DEBNI module.

Appendix D explains how to read the DEBNI's hardware Ethernet address.

Appendix E explains how to upgrade a DEBNA module to a DEBNI module.

Associated Documents

The DEBNI is one of a family of processors, memories, and adapters that use the VAXBI bus. The *VAXBI Options Handbook* (EB-32255-46) provides a technical summary of all VAXBI modules.

Other related documentation includes:

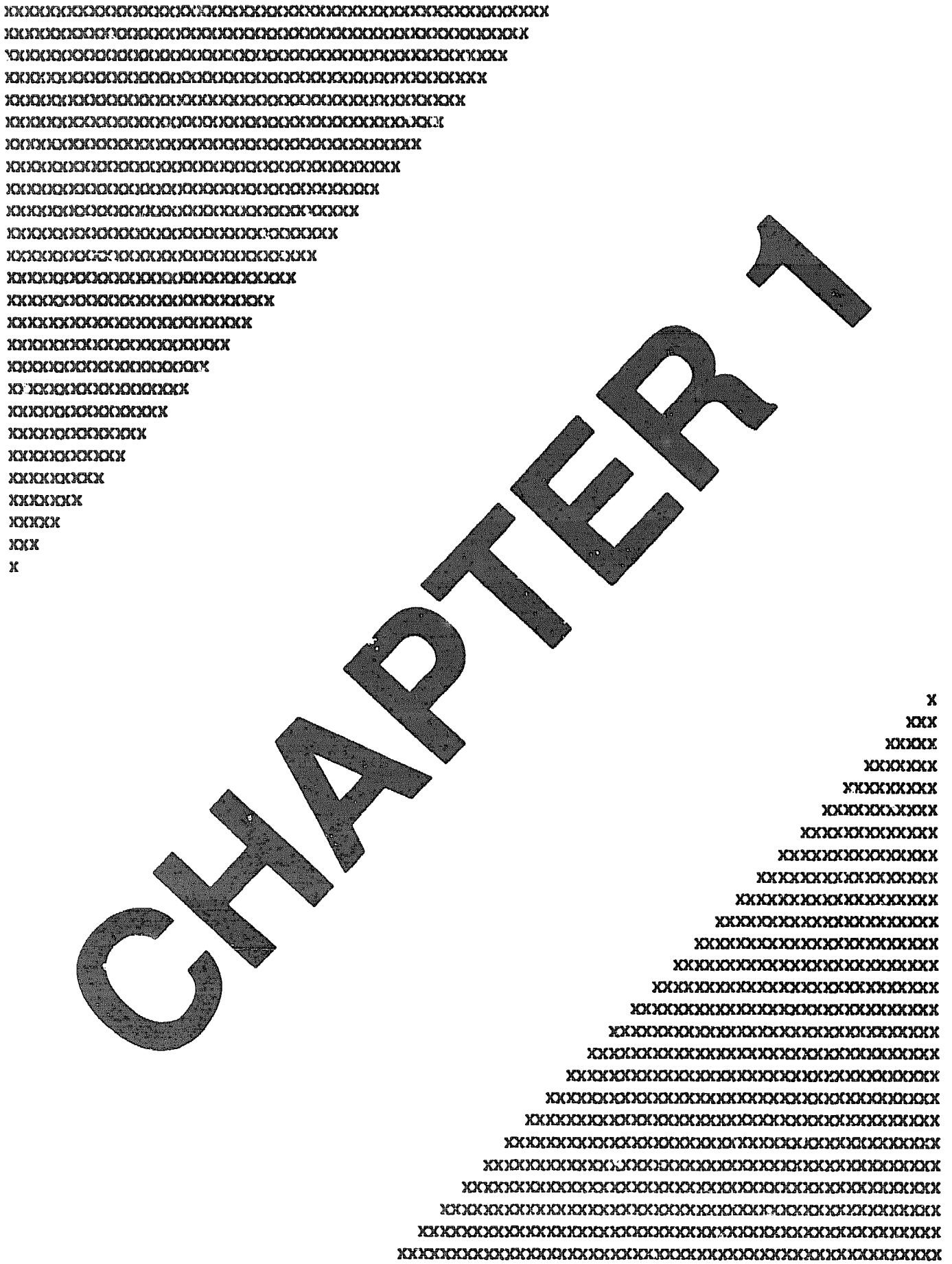
DEC LANcontroller 200 Technical Manual, EK-DEBNI-TM

DEC LANcontroller 200 Programmer's Guide, EK-DEBNI-PG

Ethernet Installation Guide, EK-ETHER-IN

Conventions

- 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.


CHAPTER 1

1 Introduction

The DEC LANcontroller 200 is an intelligent I/O controller that interfaces an Ethernet local area network to a VAXBI bus. The DEC LANcontroller 200 is compatible with the Ethernet and IEEE 802 specifications [1] and is the standard Ethernet interface for VAX 6xxx and 8xxx systems. The *Systems and Options Catalog* indicates which systems support the DEC LANcontroller 200 option.

The DEC LANcontroller 200 has a board designation of T1034-YA. It is also possible to upgrade the older DEBNA Ethernet controller, which has a board designation of T1034-00, to a DEC LANcontroller 200 by replacing four ROMs (see Appendix E). The converted version of the DEC LANcontroller 200 has a board designation of T1034-YA but can be distinguished from the nonconverted version of the DEC LANcontroller 200 by the module revision number.

The DEC LANcontroller 200 is also called the DEBNI controller. Throughout the rest of this manual, the DEC LANcontroller 200 is referred to as the DEBNI.

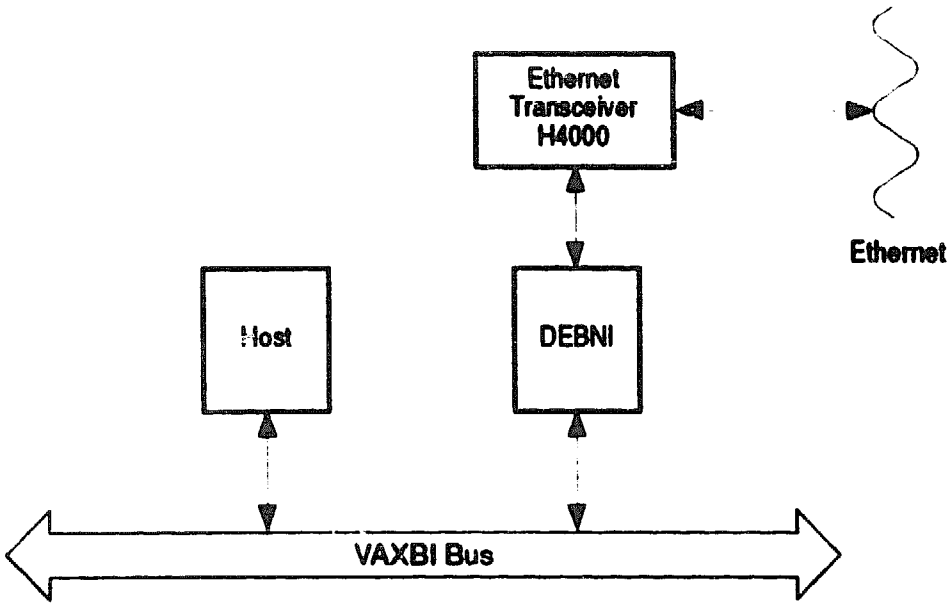
1.1 BASIC FUNCTIONS

The DEBNI 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 MicroVAX processor, the DEBNI can control operations independently of the host processor. The details of Ethernet transactions, including data transfer over the VAXBI bus, are thus transparent to the host processor (see Figure 1–1).

[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 DEBNI Module in a VAXBI System



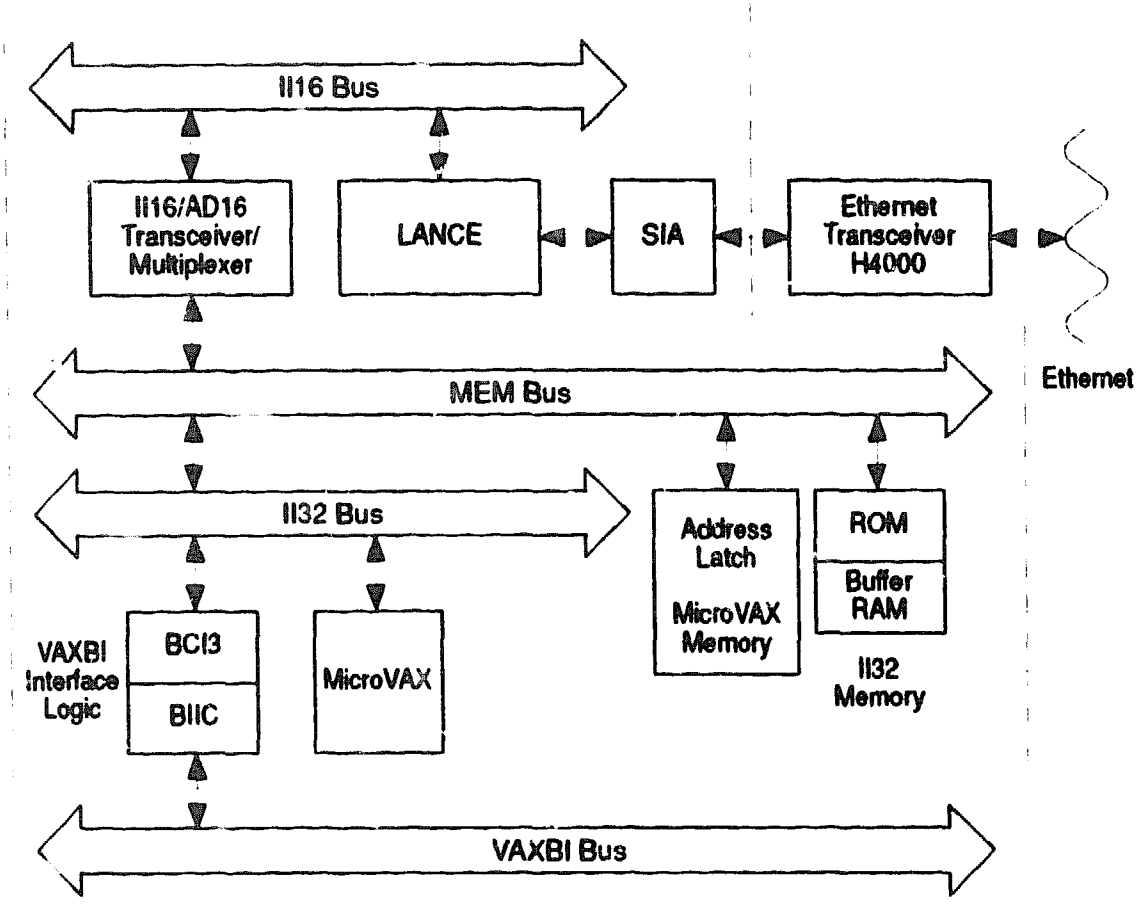
The DEBNI lets the host processor communicate with other nodes in an Ethernet/802 local area network. The DEBNI implements the complete Ethernet protocol and complies with the IEEE 802 specifications.

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

The DEBNI firmware includes a console monitor program that allows a user at any terminal on the network to monitor the DEBNI operation and the network utilization. The console monitor program is accessible only if the DEBNI firmware console-enable jumper is installed (see Section 2.4).

Figure 1–2 is a block diagram of the DEBNI module.

Figure 1–2 DEBNI Block Diagram



The MicroVAX is a 32-bit, single-chip processor that serves as the DEBNI CPU. On the DEBNI module, the MicroVAX is dedicated to running firmware; it cannot be used directly by application programs running on the host processor or by a user at the system console.

The DEBNI Ethernet interface consists of the following chipset:

- A Local Area Network Controller for Ethernet (LANCE chip)
- A Serial Interface Adapter (SIA chip)

The LANCE and SIA chips implement the physical layer and portions of the data link layer of the Ethernet/802 interface.

The DEBNI VAXBI interface is implemented by the BIIC and BCI3 chips. The BIIC has its own power-up self-test.

1.2 PHYSICAL DESCRIPTION

The DEBNI option consists of a single board. The DEBNI has one cable that connects the module with an Ethernet transceiver. This cable is not part of the DEBNI option but is included in the cabinet kits for the DEBNI. (The cabinet kits are described below.)

The Ethernet transceiver cable is a 9-conductor, shielded cable with a male connector. The cable has three main connectors: P1, P2, and P3. The P1 connector connects to the VAXBI backplane. The P2 connector is an industry-standard Ethernet connector that connects to the Ethernet bus. The P3 connector is a +15V direct-current power connection.

Table 1–1 lists the items on the DEBNI packing list. These are the items included with the DEBNI option.

Table 1–1 Packing List for DEBNI Option

Part Number	Quantity	Description
T1034-YA	1	DEBNI module
EK-DEBNI-IN	1	Installation Guide
EK-DEBNI-RN	1	Release Note
17-00684-02	1	Power interface cable needed by some VAX 82xx/83xx Configuration 1 systems. See Section 2.1 for more information.

Table 1–2 lists DEBNI cabinet kits, which must be ordered separately from the DEBNI option. For new systems not included in this table, please see the *Systems and Options Catalog*.

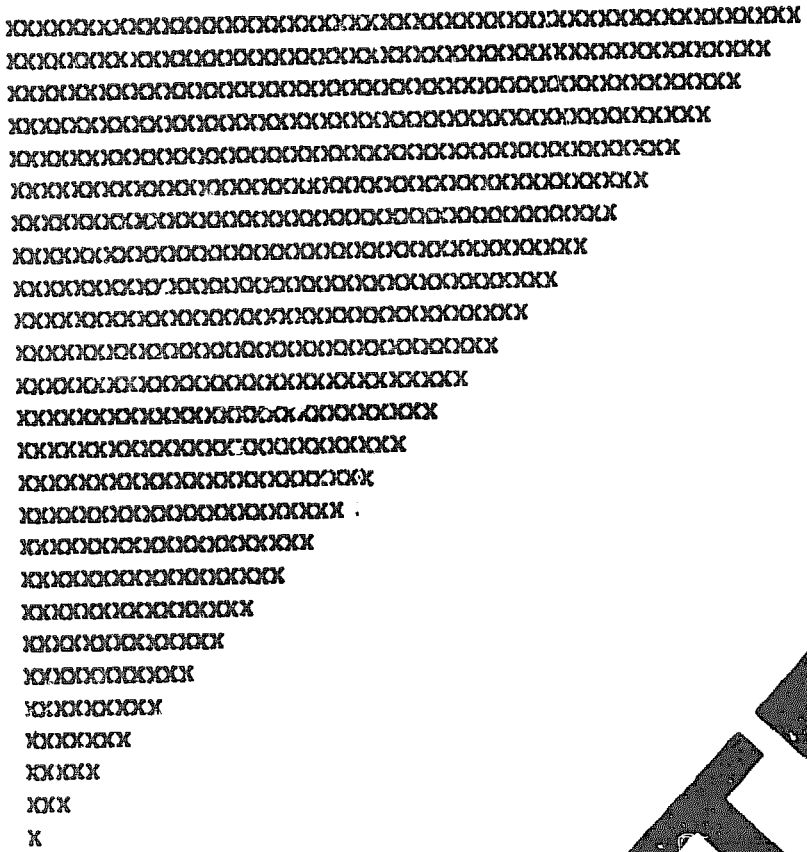
Table 1–2 Cabinet Kits for DEBNI Options

System/Enclosure	Kit Number	Contents
VAX 6xxx system cabinets	CK-DEBNA-LD	Ethernet I/O connector panel (74-26407-41) 3' internal Ethernet cable (17-01496-01) Firmware console-enable jumper (17-01149-01) Ethernet loopback connector (12-22196-02)
VAX 8800, 8810, 8550, and 8530 system cabinets; VAX 8840, 8830, and 8820 system and expansion cabinets; and VAX 6xxx VAXBI expansion cabinets (new VAXBI expansion cabinet (H9657))	CK-DEBNA-LJ	Ethernet I/O connector panel (74-26407-41) 5' internal Ethernet cable (17-01601-03) Firmware console-enable jumper (17-01149-01) Ethernet loopback connector (12-22196-02)
VAX 8350, 8250	CK-DEBNA-LM	Ethernet I/O connector panel (70-18799-00) 8' internal Ethernet cable (17-01601-02) Firmware console-enable jumper (17-01149-01) Ethernet loopback connector (12-22196-02)
VAX 8810, 8800, 8700, 8550, and 8530 expansion cabinets (old VAXBI expansion cabinet (H9652))	CK-DEBNA-LN	Ethernet I/O connector panel (70-18799-00) 15' internal Ethernet cable (17-01601-04) Firmware console-enable jumper (17-01149-01) Ethernet loopback connector (12-22196-02)

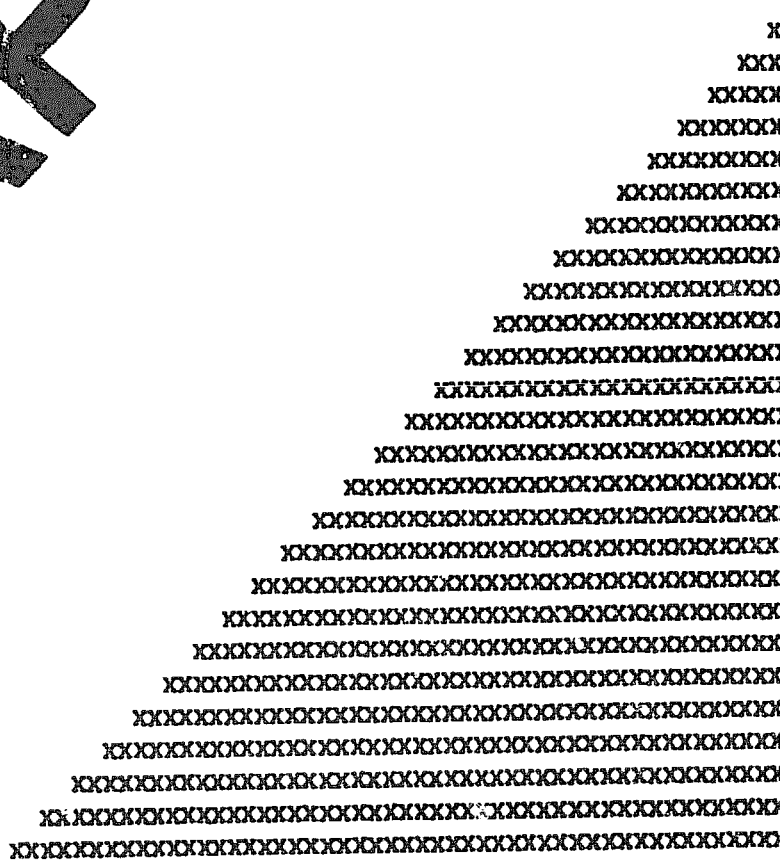
The DEBNI module has three status-indicator lights:

- Two DEBNI OK LEDs (yellow)
- One External Loopback LED (green)

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



CHAPTER 2



2 Installation

This chapter explains how to install the DEBNI option into a VAXBI computer system.

For complete instructions for installing an Ethernet transceiver connected to a DEBNI, see the installation guides for the host computer system.

WARNINGS

POWER OFF—Shut off system power and disconnect the main system power cord before performing any procedure in this chapter.

STABILIZE THE CHASSIS—For 82xx/83xx Configuration 1 systems: make sure to extend the stabilizer leg(s) before you pull out the processor drawer.

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 a VAXBI card cage, place it in a conductive container.

2.1 INSTALLATION

The following steps describe the installation process:

1. Power down the host computer system by:
   a. Turning the POWER switch to the OFF position
   b. Setting the system circuit breaker in the rear to OFF
2. For 82xx/83xx Configuration 1 systems, extend the cabinet stabilizer leg(s).
3. Open the cabinet.
4. Make sure you are wearing the ESD wrist strap that is attached to the system chassis.

5 For 82xx/83xx Configuration 1 systems, slide out the card cage and rotate it until it locks into the vertical position.

6 Remove the card cage cover.

7 On some systems, it may be necessary to install a transition header (a cable connector for the backplane) on the VAXBI backplane opposite the slot for the DEBNI. (The VAXBI transition header is part number 12-22246-01.) Use only the torque screwdriver (P/N 29-17381-00). Torque both screws to 6 +0/–1 inch-pounds.

8 Make sure there is a node ID plug in place. Note the number on the node ID plug; this number is now the installed module's VAXBI node ID. The node ID number must be unique with respect to the other node numbers on the VAXBI bus.

9 Run the internal Ethernet cable from backplane segment E2 to the Ethernet I/O connector panel. See Figure 2–1 and Section 2.3.

10 Connect the pigtail connector (P3) from the internal Ethernet cable to a +15V two-prong connector from the power supply. [1] Table 2–1 describes the power connection for various VAX systems.

Table 2–1 Power Connection for Internal Ethernet Cable

VAXBI System	Power Connection
VAX 6xxx	Any H7214 regulator plug (J2) located in the rear of the cabinet
VAX 82xx, 83xx Configuration 1 (12-slot VAXBI card cage) and VAXBI Expansion Box BA32-B	2-pin Mate-N-Lok pigtail connector located in the bottom of the box
VAX 82xx, 83xx Configuration 2 (24-slot VAXBI card cage)	2-pin Mate-N-Lok pigtail connector located in the rear of the cabinet
VAX 85xx, 87xx, 88xx, and VAXBI Expansion Cabinet H9657	2-pin Mate-N-Lok pigtail connector labeled P3 or P4 and located in the rear of the cabinet

NOTE

All the pigtail connectors from the power supply may already be used. (Some systems have one or two.) In

[1] For VAX 82xx/83xx Configuration 1 systems and for the VAXBI expander box (BA32-B), you may first have to install the +15V connector to the power supply. The connector's part number is 17-00684-02.

this case, the external Ethernet cable from the I/O connector panel must not be connected directly to an H4000 transceiver, a DESTA (a thin-wire box), or a DECOM broadband transceiver—none of which have their own power supplies. The cable may, however, be connected to any of the following devices, which do have their own power supplies and which may, in turn, be connected to an H4000:

- **A DELNI**
- **A DEMPR (a thin-wire version of the DELNI)**
- **A DEBET (a bridge)**

11 If you wish to have access to the DEBNI's console monitor program, install the firmware console-enable jumper on backplane segment E1. (See Figure 2–2 and Section 2.4.)

12 On the module-insertion side of the card cage, locate the card cage slot to which the DEBNI cables and jumper are connected on the connector side of the card cage.

13 Lift the locking lever to open the slot.

14 Slide the DEBNI module into the slot until it stops: this is a zero insertion force card cage.

15 Close the locking lever.

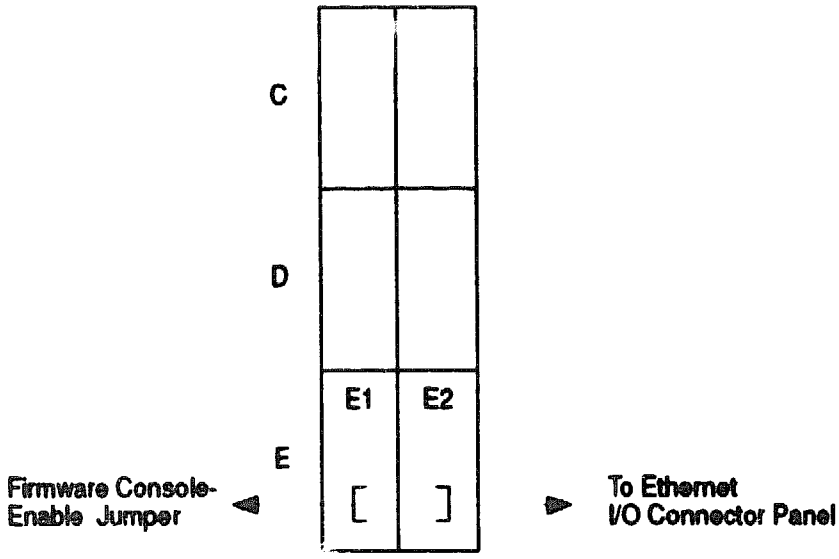
16 Replace the card cage cover.

17 For 82xx/83xx Configuration 1 systems, rotate the card cage to the horizontal position and retract the stabilizer leg(s).

18 Close the cabinet.

19 Verify the installation as described in the following section.

Figure 2–1 Cabling at the DEBNI VAXBI Connector



2.2 INSTALLATION VERIFICATION PROCEDURE

Follow these steps to verify that the DEBNI is properly installed:

1. Turn the system power on and verify that the DEBNI passes both self-test (the two yellow DEBNI OK LEDs light) and the LANCE external loopback test (the green External Loopback LED lights). (See Chapter 3.) If the self-test and/or external loopback test fails, check to see that the module is properly seated in the card cage and that the cables are properly installed. If the module continues to fail self-test, swap in a different DEBNI module if one is available. You can also try installing the module in a different slot.
2. After the module passes both self-test and the LANCE external loopback test, boot the operating system.
3. Configure the network database and start DECnet.
4. If the system is unable to communicate over the Ethernet, verify that the network software is installed and configured properly.

5 If the network software is properly installed and configured, shut down the system and check the hardware to isolate the faulty field-replaceable unit (FRU) as follows:

a. Disconnect the external Ethernet transceiver cable (BNE3) at the transceiver end.

b. Install a loopback connector (part number 12-22196-02) on the cable.

c. Run self-test and observe one of the following:

- If the External Loopback LED 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 does not light, one of the following is bad: transceiver cable, internal Ethernet cable, backplane, or DEBNI module. Go to the next step.

d. Disconnect the external transceiver cable at the I/O connector panel and install a loopback connector in its place.

e. Rerun the self-test and observe one of the following:

- If the External Loopback LED lights, the transceiver cable is bad. Replace the cable and rerun the self-test to verify proper operation. No further action is required.
- If the External Loopback LED does not light, one of the following is bad: internal Ethernet cable, backplane, or DEBNI module. Replace the internal Ethernet cable and install the loopback connector on the new cable. Go to the next step.

f. Rerun the self-test and observe one of the following:

- If the External Loopback LED lights, the removed cable is bad. Rerun the self-test to verify proper operation. No further action is required.
- If the External Loopback LED does not light, either the DEBNI 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 the self-test and observe one of the following:

- If the External Loopback LED lights, the removed DEBNI module is bad. Rerun the self-test to verify proper operation. No further action is required.
- If the External Loopback LED does not light, the backplane is bad. Install the DEBNI module in a different slot. Rerun the self-test to verify proper operation. Consider replacing the card cage.

2.3 INTERNAL ETHERNET CABLE

The internal Ethernet cable connects the VAXBI backplane to the I/O connector panel (see Table 1–2 for part numbers). The external transceiver cable is ordered separately; it runs from the outside of the connector panel to an Ethernet transceiver, such as an H4000 baseband transceiver, DECOM broadband transceiver, or DELNI local network interconnect.

On the backplane, the internal cable plugs into segment E2 opposite the DEBNI slot. (Viewed with the node ID plugs at the top, E2 is the right side of the segment farthest from the node ID plugs. See Figure 2–1.) The +15V pigtail connector supplies power to the H4000 transceiver; however, you should plug it in (if possible) regardless of the type of transceiver.

Table 2–2 lists pinouts of the internal Ethernet cable connector at the VAXBI card cage. Table 2–3 lists the connector pinouts of the transceiver cable.

Table 2–2 Internal Ethernet Cable Pinouts (On VAXBI Backplane)

Pin	Signal	Description
E01–E04	Unconnected	
E05–E09	Logic Ground	
E10	Ethernet Collision L	Differential collision detect
E11	Ethernet Collision H	signals from the Ethernet bus.
E12	Ethernet Receive L	Differential receive signals from
E13	Ethernet Receive H	the Ethernet bus.
E14	Ethernet Transmit L	Differential transmit signals to
E15	Ethernet Transmit H	the Ethernet bus.

Table 2–2 (Cont.) Internal Ethernet Cable Pinouts (On VAXBI Backplane)

Pin	Signal	Description
E16 E46	Firmware Console-Enable	Enables access to the DEBNI console monitor program. The signal is asserted if there is a jumper connection between pins E16 and E46 in the DEBNI section of the VAXBI backplane. The signal is sent to the Status Register, which is a DEBNI-internal register.

Table 2–3 Transceiver Cable Connector Pinouts

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	
14	Reserved	
15	Reserved	

2.4 FIRMWARE CONSOLE-ENABLE JUMPER

Install the firmware console-enable jumper if you want to have access to the DEBNI console monitor program. This program provides visibility into the DEBNI operation and the network utilization. The DEBNI console is accessible from any terminal on the network. (See the *DEBNI VAXBI Network Adapter Technical Manual*, EK-DEBNI-TM, for further information.)

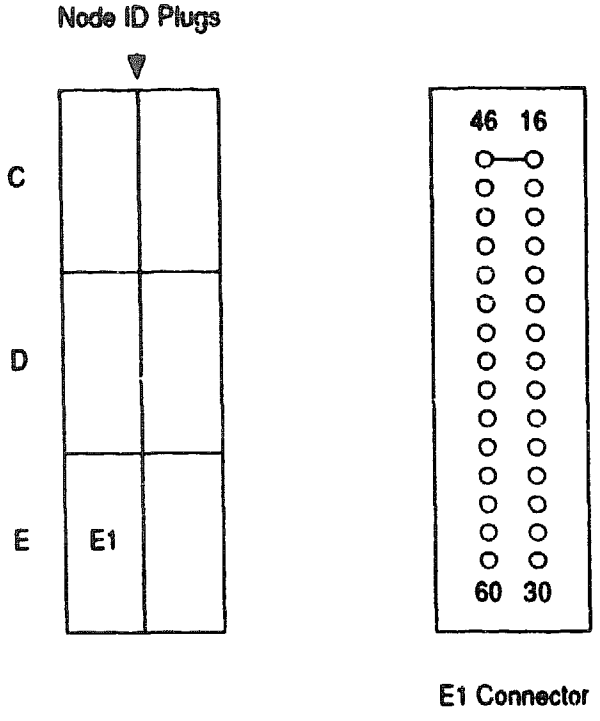
If the firmware console-enable jumper is not installed, you will not be able to access the DEBNI console.

The jumper (P/N 17-01149-01) is a 30-pin connector that is installed on segment E1 of the VAXBI backplane, opposite the DEBNI slot. (Viewed with the node ID plugs at the top, E1 is the left side of the segment farthest from the node ID plugs.) The jumper connects pins E16 and E46, as shown in Figure 2–2.

NOTE

On the DEBNA Ethernet controller and the DEBNK Ethernet/tape controller, this jumper was used to enable remote booting. The firmware console-enable jumper for the DEBNI has the same part number and is installed in the same location as the boot-enable jumper for the DEBNA and DEBNK. The only difference is the function controlled by the jumper.

Figure 2–2 Firmware Console-Enable Jumper



2.5 REMOVAL

To remove a DEBNI module, follow these steps:

1 Power down the host computer system by:

   a. Turning the POWER switch to the OFF position

   b. Setting the system circuit breaker to OFF

2 For 82xx/83xx Configuration 1 systems, extend the cabinet stabilizer leg(s).

3 Open the cabinet.

4 Make sure you are wearing an ESD wrist strap that is attached to the system chassis.

5 On the module-insertion side of the open cabinet:

   a. For 82xx/83xx Configuration 1 systems, rotate the card cage until it locks in the vertical position.

b. Remove the card cage cover.

c. Locate the desired card cage slot.

d. Lift the locking lever to open the slot.

e. Slide the circuit board out of the card cage slot.

f. Put the board into a conductive container.

g. Close the locking lever.

h. Replace the plastic card cage cover if another module will not be installed.

6 On the cabling side of the open cabinet:

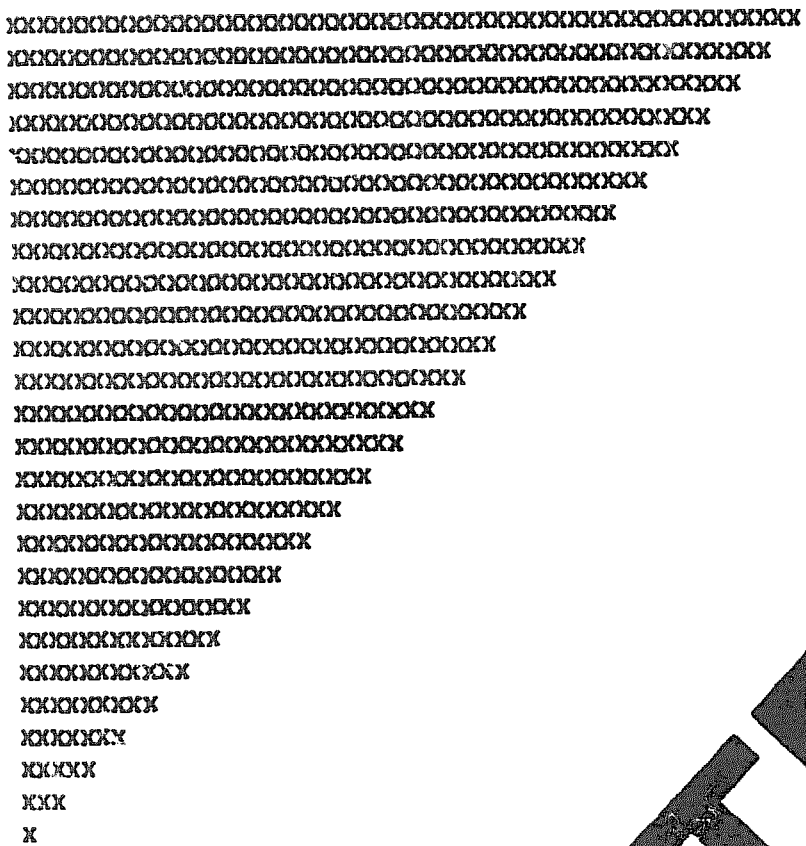
a. Locate the same card cage slot.

b. If another module will be installed, leave the cables and jumpers in place; otherwise; remove the cables and jumper.

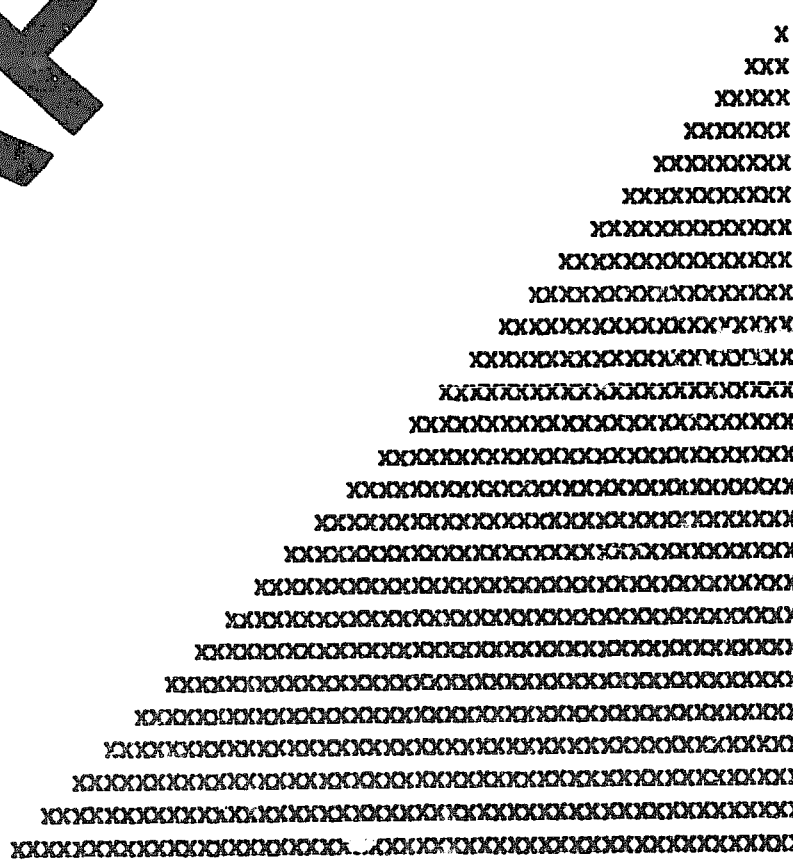
7 If another module will not be installed:

a. For 82xx/83xx Configuration 1 systems, rotate the card cage to the horizontal position and retract the stabilizer leg(s).

b. Close the cabinet.



CHAPTER 3



3 Power-Up Self-Test

The DEBNI's 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 DEBNI 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.

3.1 HOW TO RUN SELF-TEST

There are three ways of running self-test for the DEBNI:

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

2 On processor reset—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 VAXBI node to run its own self-test.

3 From the console—A field service engineer can invoke the DEBNI's D0 tests from the system console of a VAX 6xxx, VAX 82xx, or VAX 83xx system. (These are the same tests that run during power-up or reset self-test.)

The following example shows how to use VAX console commands on a VAX 6xxx system to run the self-test on a DEBNI located at VAXBI node 2. (The XMI-to-VAXBI adapter is at XMI node E.) For a description of the Z command used in this example, see the system *Owner's Manual*.

```
>[CTRL/P]
>>> Z/BI:2 E [RETURN]
?33 Z connection successfully started
T/R [RETURN]
RBD2> ST 0 [RETURN]
;        P          2      0118 00000001
;00000000 00000000 00000000 00000000 00000000 00000000 00000000
;   PUDR: FFFF0002
RBD2>[CTRL/Z][CTRL/P]
?31 Z connection terminated by ^P
>>>
```

The following example shows how to use VAX console commands on a VAX 82xx or 83xx system to run the self-test on a DEBNI at node 2. For a description of the Z command used in this example, see the system *Owner's Manual*.

```
>[CTRL/P]
>>>Z 2 [RETURN]
T/R [RETURN]
RBD2>DO [RETURN]
;        P          2      0118 00000001
;00000000 00000000 00000000 00000000 00000000 00000000 00000000
;   PUDR: FFFF0002
RBD2>[CTRL/Z][CTRL/P]
?31 Z connection terminated by ^P
>>>
```

If you are on a VAX 6xxx console and do not know which VAXBI node the DEBNI is at, use the SHOW CONFIGURATION command at the console prompt to locate the DEBNI.

Another way of locating the DEBNI nodes is to use the EXAMINE command to read the Device (DTYPE) Register at each node until you find the DEBNI, which has a device type of 0118 (hex). A module's DTYPE Register is always at the module's base address. (See Appendix D for a formula for determining the base address of a VAXBI node.)

3.2 REPORTING SELF-TEST RESULTS

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

3.2.1 Self-Test Results in LEDs

There are three status-indicator lights on the module:

- 2 yellow DEBNI OK LEDs
- 1 green External Loopback LED

The location of the LEDs is shown in Figure 3–1.

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

At power-up or reset, all the LEDs are off. If the DEBNI passes all the executed tests (excluding the LANCE external loopback test), the DEBNI firmware lights the two yellow DEBNI OK LEDs; otherwise, these LEDs remain off. If the LANCE external loopback self-test passes, the firmware lights the green External Loopback LED; otherwise this LED remains off.

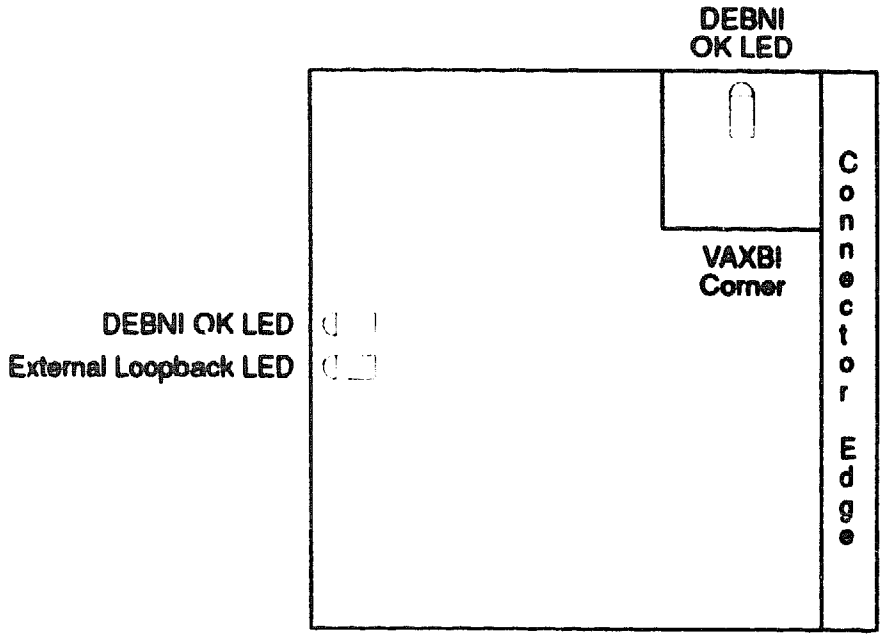
3.2.2 Self-Test Results in the Power-Up Diagnostic Register

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 Appendix B for a detailed description of the XPUD Register.

3.3 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 DEBNI 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 DEBNI is disconnected from the transceiver. In any case, such an error condition prevents the DEBNI from transmitting or receiving Ethernet packets.

Figure 3–1 LED Locations



If the XPUD Register indicates that all of the self-test routines failed, the problem is probably the BIIC, MicroVAX, or MicroVAX ROM. If one of these components fails, the self-test routine stops, and the MicroVAX enters a wait state.

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

3.4 TESTED COMPONENTS

Self-test tests the following components and functions on the DEBNI module:

- MicroVAX
- MicroVAX ROM
- MicroVAX RAM
- MicroVAX Interrupt Request (IRQ) lines

- BIIC and BCI3 chips
- LANCE chip
- Interval timer

3.5 UNTESTED COMPONENTS AND FUNCTIONS

The following components and functions are not tested:

MicroVAX:

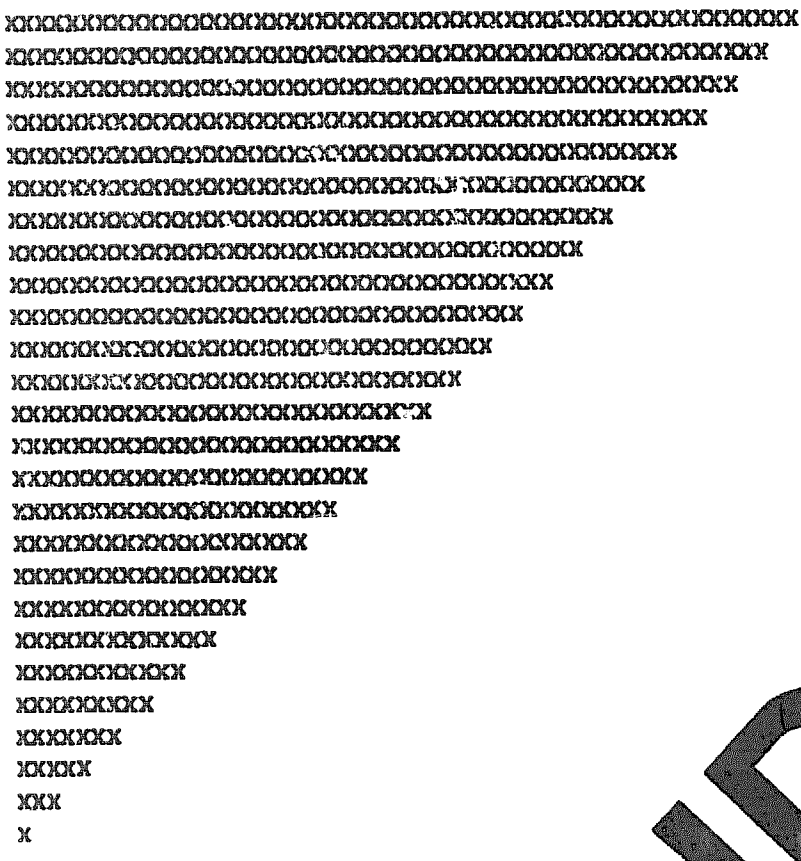
- A complete instruction set

Ethernet interface logic functions:

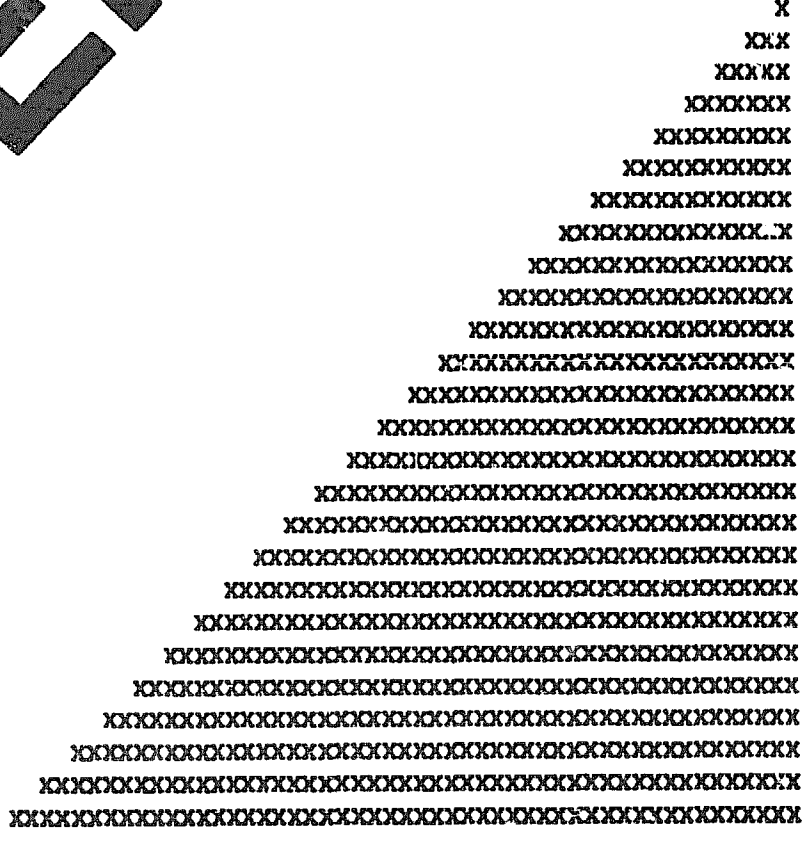
- More (multiple retries of packet transmission)
- One (one retry of a packet transmission)
- Babble error
- Time domain reflectometry
- Late collision
- Loss of carrier
- Memory error

In the BCI3/BIIC logic:

- BIIC nodespace locking
- BIIC nodespace Write Sense bits
- BCI3 datamove operations with quadwords and octawords
- BCI3 Nonlocal Memory Reference (NLMR) line



APPENDIX A



A 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 Registers

This appendix describes the DEBNI registers that are useful for troubleshooting problems that may occur during installation. Table B–1 summarizes these registers. Table B–2 explains the access types for these registers, as well as the bit types for the registers. Each register is then described in detail.

Table B–1 Registers Useful During Installation—Summary

Name	Mnemonic	VAXBI Address	Type[1]	Description
Device	DTYPE	bb + 0	RO	Indicates the VAXBI node's device type and device firmware revision level.
VAXBI Control and Status	VAXBICSR	bb + 4	R/W	Indicates whether the BIIC chip passed its internal self-test and whether the DEBNI as a whole passed its self-test.
Bus Error	BER	bb + 8	R/W	Records VAXBI bus errors and loopback errors.
Power-Up Diagnostic	XPUD	bb + FC	R0	Indicates whether self-test has completed and which tests in the self-test passed.

[1] With respect to the port driver

Table B–2 Codes for Bit Types

Code	Description
DCLOC	Cleared by the BIIC following a successful BIIC self-test; initiated on the deassertion of BCI DC LO L from the BIIC.
DCLOL	Loaded by the BIIC on the last cycle in which BCI DC LO L is asserted. If the BCI signal lines are not driven during this cycle, these bits are set.

Table B–2 (Cont.) Codes for Bit Types

Code	Description
DCLOS	Set by the BIIC following a successful BIIC self-test; initiated on the deassertion of BCI DC LO L from the BIIC.
DMW	BIIC diagnostic mode writeable; reserved for use by DIGITAL.
RO	Read-only. Write-type transactions must not change the value of this register or bit.
R/W	Normal read/write register or bit.
SC	Special case: operation defined in detailed description.
STOPC	Cleared by the BIIC on receipt of a STOP command to the node.
STOPS	Set by the BIIC on receipt of a STOP command to the node.
W1C	Write-1-to-clear bit. Write-type transactions cannot set this bit.

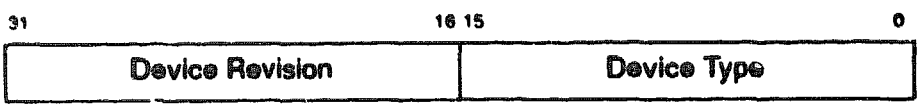
Device Register (DTYPE)

The Device Register identifies the DEBNI device type and firmware revision. The device type for the DEBNI is 0118 (hexadecimal).

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

ADDRESS

Nodespace base address + 0



BITS<31:16>

Name: Device Revision

Mnemonic: DREV

Type: R/W, DMW, DCLOC

Indicates the device revision of the DEBNI. The high-order byte of th field is the major revision number and the low-order byte of the field is the minor revision number.

The major revision field is decoded as follows for the first 10 major revisions of the DEBNI:

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

Code (hex)	Module Revision Level
08	H
09	I
0A	J

The minor revision field is decoded as follows for the first 10 minor revisions of the DEBNI:

Code (hex)	Minor Revision Level
00	00
01	01
02	02
03	03
04	04
05	05
06	06
07	07
08	08
09	09
0A	10

BITS<15:0>

Name: Device Type

Mnemonic: DTYPE

Type: R/W, DMW, DCLOC

A value of 0118 (hex) indicates that the adapter is a DEBNI module.

VAXBI Control and Status Register (VAXBICSR)

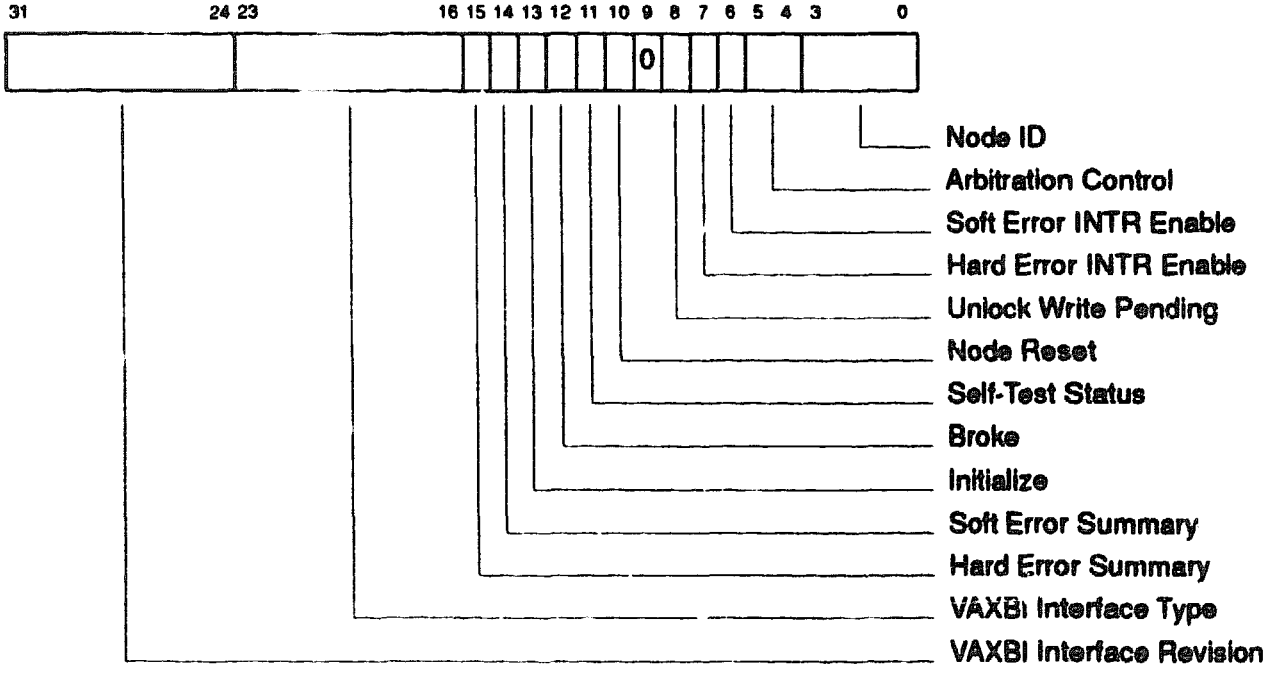
The VAXBICSR controls the DEBNI's VAXBI interface and provides status information about this interface. The register is written by the port and, with one exception, is read-only to the port driver: the port driver may issue a node reset to the DEBNI by writing the register with C00 (hex) to set the Node Reset bit and the Self-Test Status bit. Otherwise, the port driver may not write the register.

NOTE

Both the Node Reset bit and the Self-Test Status bit should be set to cause a node reset. If the Self-Test Status bit is not set, the DEBNI's BIIC temporarily disables its VAXBI drivers to isolate the node from the VAXBI bus, thereby causing the transaction that initiated the node reset to fail.

ADDRESS

Nodespace base address + 4



BITS<31:24>

Name: VAXBI Interface Revision

Mnemonic: IREV

Type: RO

Indicates the revision of the device that provides the primary interface from the node to the VAXBI bus.

BITS<23:16>

Name: VAXBI Interface Type

Mnemonic: ITYPE

Type: RO

Indicates the type of device that provides the primary interface from the node to the VAXBI bus. For the DEBNI, this device is the BIIC, which has an ITYPE code of 0000 0001 (hex).

BIT<15>

Name: Hard Error Summary

Mnemonic: HES

Type: RO

When set, indicates that one or more hard error bits in the Bus Error Register (BER) is set.

BIT<14>

Name: Soft Error Summary

Mnemonic: SES

Type: RO

When set, indicates that one or more soft error bits in the Bus Error Register (BER) is set.

BIT<13>

Name: Initialize

Mnemonic: INIT

Type: W1C, DCLOS, STOPS

Not used by the DEBNI.

BIT<12>

Name: Broke

Mnemonic: None

Type: W1C, DCLOS

When set, indicates that the BIIC passed its power-up self-test. The DEBNI clears this bit after the DEBNI node passes self-test.

BIT<11>

Name: Self-Test Status

Mnemonic: STS

Type: R/W, DCLOS

When set, indicates that the BIIC has passed its internal self-test. When clear, indicates that the BIIC has not yet passed this self-test. If the BIIC fails its self-test, it disables its VAXBI drivers to isolate the node from the VAXBI bus.

BIT<10>

Name: Node Reset

Mnemonic: NRST

Type: SC

The port driver sets this bit to selectively reset the DEBNI. Setting the bit forces the BIIC to assert BCI DC LO L, which causes the DEBNI to execute its power-down and power-up sequences, including self-test.

BIT<9>

Name: Reserved

Mnemonic: None

Type: RO

Reserved; must be zero.

BIT<8>

Name: Unlock Write Pending

Mnemonic: UWP

Type: W1C, DCLOC

When set, indicates that the DEBNI has executed a successful Interlock Read with Cache Intent (IRCI) transaction and has not yet executed the corresponding Unlock Write Mask with Cache Intent (UWMCI) transaction. If the DEBNI issues a UWMCI transaction when UWP is cleared, the Interlock Sequence Error (ISE) bit sets in the Bus Error Register (BER). The out-of-sequence UWMCI command is not aborted.

BIT<7>

Name: Hard Error INTR Enable

Mnemonic: HEIE

Type: R/W, DCLOC, STOPS

The DEBNI clears this bit to disable BIIC hard error interrupts to the host. When a hard error occurs, the BIIC will not automatically interrupt the host.

BIT<6>

Name: Soft Error Interrupt Enable

Mnemonic: SEIE

Type: R/W, DCLOC, STOPC

The DEBNI clears this bit to disable BIIC soft error interrupts to the host. When a soft error occurs, the BIIC will not automatically interrupt the host.

BITS<5:4>

Name: Arbitration Control

Mnemonic: ARB

Type: R/W, DCLOC

Indicates the VAXBI arbitration mode to be used by the DEBNI as follows:

ARB		Arbitration Mode
5	4	
0	0	Dual round robin
0	1	Fixed-high priority (Reserved)
1	0	Fixed-low priority (Reserved)
1	1	Disable Arbitration (Reserved)

BITS<3:0>

Name: Node ID

Mnemonic: None

Type: RO, DMW, DCLOL

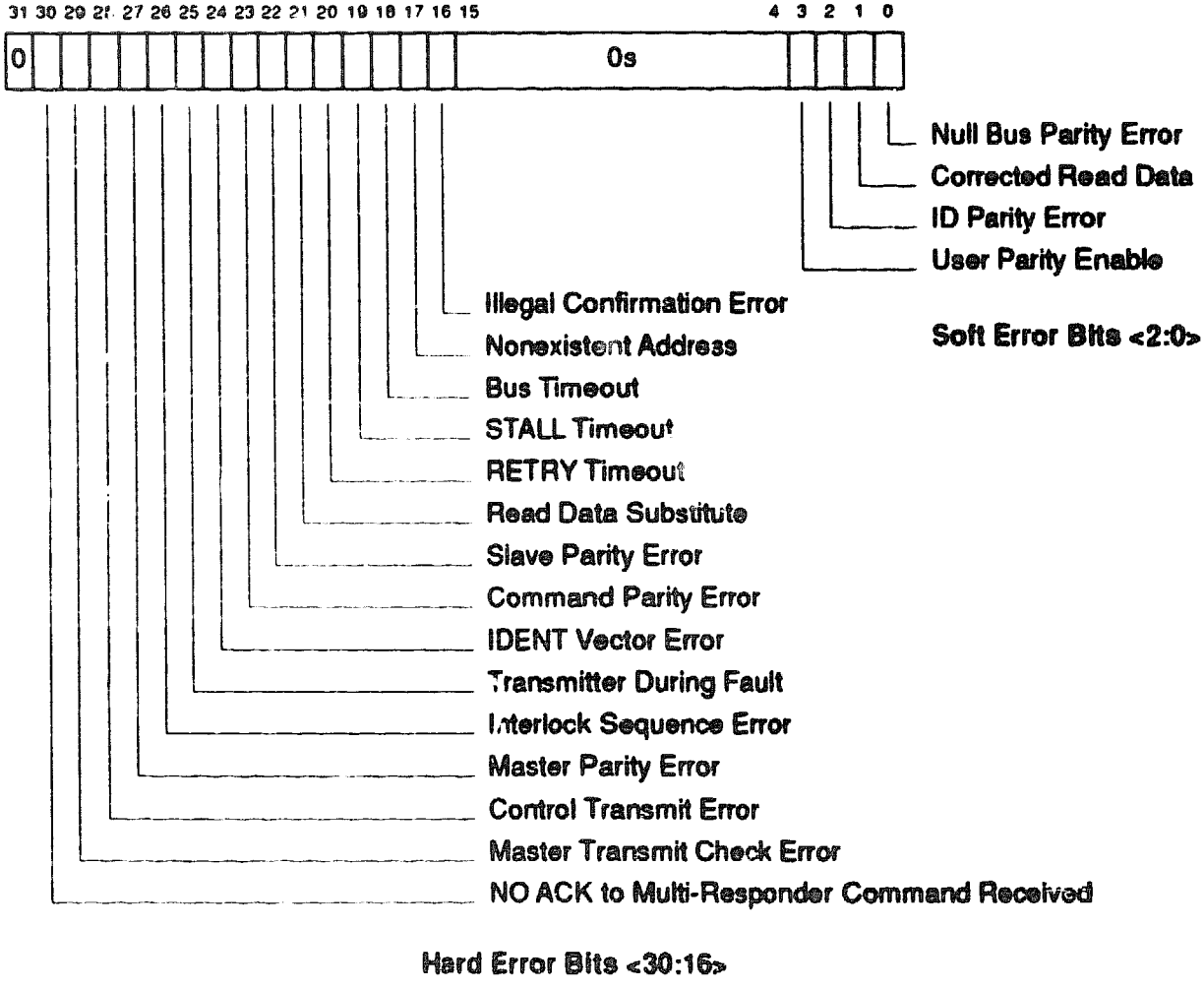
Indicates the node ID, which is determined by the DEBNI node ID plug. The Node ID field is loaded by the DEBNI hardware on power-up or reset.

Bus Error Register (BER)

The BER records errors during VAXBI and loopback transactions. An error sets the corresponding bit in the register and is also reflected in the EV (event) code that the BIIC passes to the BCI3.

ADDRESS

Nodespace base address + 8



BIT<31>

Name: Reserved

Mnemonic: None

Type: RO

Reserved; must be zero.

BIT<30>

Name: NO ACK to Multi-Responder Command Received

Mnemonic: NMR

Type: W1C, DCLOC

Sets if the BIIC receives a NO ACK in response to a VAXBI interrupt that it has issued.

BIT<29>

Name: Master Transmit Check Error

Mnemonic: MTCE

Type: W1C, DCLOC

The BIIC reads all VAXBI data that it transmits to verify transmission accuracy. The MTCE bit sets when received data does not match transmitted data. The BIIC does not perform this check on the assertion of the encoded ID on the I lines during an embedded arbitration cycle.

BIT<28>

Name: Control Transmit Error

Mnemonic: CTE

Type: W1C, DCLOC

Sets if the BIIC detects that BI NO ARB L, BI BSY L, or BI CNF<2:0> L is deasserted while the BIIC is trying to assert the signal. The BIIC does not check the state of BI NO ARB L during burst mode transactions.

BIT<27>

Name: Master Parity Error

Mnemonic: MPE

Type: W1C, DCLOC

Sets if the BIIC, as VAXBI bus master, detects a parity error on the VAXBI bus during a read-type or vector ACK data cycle.

BIT<26>

Name: Interlock Sequence Error

Mnemonic: ISE

Type: W1C, DCLOC

Sets if the DEBNI successfully issues an Unlock Write Mask with Cache Intent (UWMCI) command without having previously issued a corresponding Interlock Read with Cache Intent (IRCI) command. The BIIC detects this sequence error by noticing that the Unlock Write Pending (UWP) bit in the VAXBICSR was cleared when the UWMCI command was issued.

BIT<25>

Name: Transmitter During Fault

Mnemonic: TDF

Type: W1C, DCLOC

Sets if the BIIC, as VAXBI bus master or VAXBI slave, detects a parity error during a cycle in which it was responsible for transmitting proper parity on the VAXBI bus. The responsibility for transmitting parity during a VAXBI cycle is as follows:

VAXBI Cycle	Node Responsible for Transmitting Parity
Command/address	Master
Read-type ACK data	Slave
Write-type data	Master

VAXBI Cycle	Node Responsible for Transmitting Parity
Broadcast data	Master
Vector ACK data	Slave
Embedded ARB (encoded master ID)	Master
IDENT (decoded master ID)	Master

The TDF bit is not set for parity errors that occur during loopback cycles.

BIT<24>

Name: IDENT Vector Error

Mnemonic: IVE

Type: W1C, DCLOC

Sets if the BIIC issues an interrupt vector and does not receive an acknowledgment (ACK) from the receiving node.

BIT<23>

Name: Command Parity Error

Mnemonic: CPE

Type: W1C, DCLOC

Sets if the BIIC detects an error in a command/address cycle. The transaction can be either a VAXBI or a loopback transaction.

BIT<22>

Name: Slave Parity Error

Mnemonic: SPE

Type: W1C, DCLOC

Sets if the BIIC, as a VAXBI slave, detects a parity error during a write-type ACK, write-type STALL, or Broadcast ACK data cycle.

BIT<21>

Name: Read Data Substitute

Mnemonic: RDS

Type: W1C, DCLOC

Sets if the BIIC receives a Read Data Substitute (RDS) or RESERVED status code during a read-type or IDENT transaction (for vector status). For this bit to be set, the BIIC must have received good parity from the data cycle that contained the RDS or RESERVED code. This bit is set even if the transaction is aborted some time after the receipt of the RDS or RESERVED code.

BIT<20>

Name: RETRY Timeout

Mnemonic: RTO

Type: W1C, DCLOC

Sets if the BIIC, as VAXBI bus master, receives 4096 consecutive RETRY responses from a node.

BIT<19>

Name: STALL Timeout

Mnemonic: STO

Type: W1C, DCLOC

Sets if the BIIC's slave port asserts the STALL code on the BCI RS<1:0> lines for 128 consecutive cycles.

BIT<18>

Name: Bus Timeout

Mnemonic: BTO

Type: W1C, DCLOC

Sets if the BIIC, as VAXBI bus master, is unable to start at least one transaction (out of possibly several that are pending) before 4096 cycles have elapsed.

BIT<17>

Name: Nonexistent Address

Mnemonic: NEX

Type: W1C, DCLOC

Sets when the BIIC receives a NO ACK for a read- or write-type command that it has issued. NEX sets only if the BIIC's parity check and master transmit check of the command/address data were successful (that is, CPE and MTCE were not set for this command/address cycle). NEX is not set for NO ACK responses to other commands.

BIT<16>

Name: Illegal Confirmation Error

Mnemonic: ICE

Type: W1C, DCLOC

Sets if the BIIC receives a reserved or illegal confirmation code. This bit can be set during either master or slave transactions. Note that a NO ACK command confirmation is not an illegal response.

BITS<15:4>

Name: Reserved

Mnemonic: None

Type: RO

Reserved; must be zeros.

BIT<3>

Name: User Parity Enable

Mnemonic: UPEN

Type: W1C, DCLOC

Indicates the BIIC parity mode. The DEBNI initializes this bit to zero to select BIIC-generated parity for VAXBI transactions initiated by the DEBNI.

BIT<2>

Name: ID Parity Error

Mnemonic: IPE

Type: W1C, DCLOC

Sets if the BIIC detects a parity error on BI I<3:0> when the master's encoded ID is asserted during embedded arbitration cycles. All nodes perform this parity check.

BIT<1>

Name: Corrected Read Data

Mnemonic: CRD

Type: W1C, DCLOC

Sets if the master receives a Corrected Read Data (CRD) status code. For this bit to be set, the BIIC logic also requires the receipt of good parity for the data cycle that contains the CRD code This bit is set even if the transaction aborts after the BIIC has received the CRD status code.

BIT<0>

Name: Null Bus Parity Error

Mnemonic: NPE

Type: W1C, DCLOC

Sets if the BIIC detects odd parity on the VAXBI bus during the second cycle of a 2-cycle sequence during which BI NO ARB L and BI BSY L were unasserted.

Power-Up Diagnostic Register (XPUD)

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

The XPUD is treated as follows:

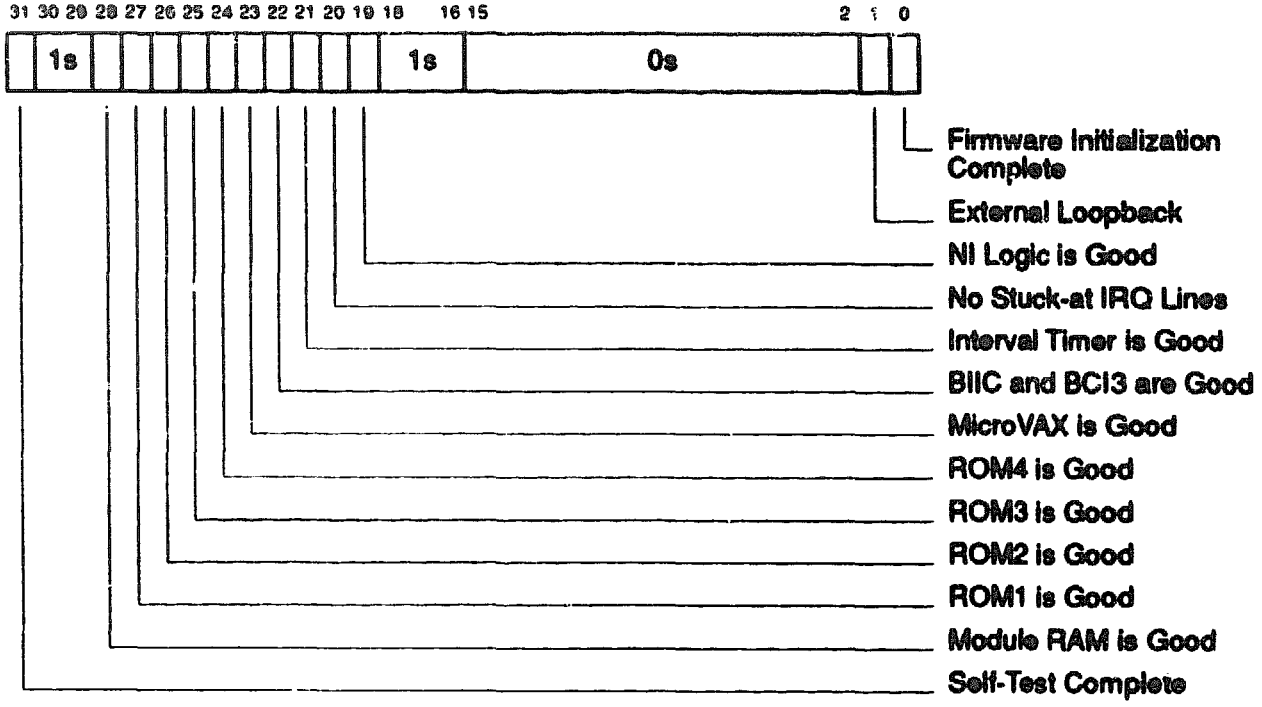
- The DEBNI initializes the XPUD Register to all zeros on power-up or reset.
- When a subtest in the self-test routine passes, its corresponding bit in the XPUD Register sets.
- If a subtest fails, the corresponding bit remains cleared.

The XPUD Register of a DEBNI that passes self-test has a value of FFFFXXXX (hex), where X = don't care.

Power-Up Diagnostic Register (XPUD)

ADDRESS

Nodespace base address + FC



BIT<31>

Name: Self-Test Complete

Mnemonic: STC

Type: R/W, DCLOC

Sets when the DEBNI self-test completes.

BITS<30:29>

Name: Reserved

Mnemonic: None

Type: RO

Reserved; must be ones.

BIT<28>

Name: Module RAM is Good
Mnemonic: RAM
Type: R/W, DCLOC
Sets when the DEBNI RAM passes self-test.

BIT<27>

Name: ROM 1 is Good
Mnemonic: ROM1
Type: R/W, DCLOC
Sets when the checksum validation for ROM chip 1 succeeds.

BIT<26>

Name: ROM 2 is Good
Mnemonic: ROM2
Type: R/W, DCLOC
Sets when the checksum validation for ROM chip 2 succeeds.

BIT<25>

Name: ROM 3 is Good
Mnemonic: ROM3
Type: R/W, DCLOC
Sets when the checksum validation for ROM chip 3 succeeds.

BIT<24>

Name: ROM 4 is Good
Mnemonic: ROM4
Type: R/W, DCLOC
Sets when the checksum validation for ROM chip 4 succeeds.

BIT<23>

Name: MicroVAX is Good
Mnemonic: UVAX
Type: R/W, DCLOC
Sets when the MicroVAX passes self-test.

BIT<22>

Name: BIIC and BCI3 are Good
Mnemonic: BI
Type: R/W, DCLOC
Sets when the BIIC and BCI3 chips pass self-test.

BIT<21>

Name: Interval Timer is Good
Mnemonic: TMR
Type: R/W, DCLOC
Sets when the interval timer passes self-test.

BIT<20>

Name: No Stuck-at IRQ Lines
Mnemonic: IRQ
Type: R/W
Sets when no IRQ (interrupt request) line is stuck asserted.

BIT<19>

Name: NI Logic is Good
Mnemonic: NI
Type: R/W, DCLOC
Sets when the Ethernet interface logic, including the LANCE chip, is good.

BITS<18:16>

Name: Reserved

Mnemonic: None

Type: RO

Reserved; must be ones.

BITS<15:2>

Name: Reserved

Mnemonic: None

Type: RO

Reserved; must be zeros.

BIT<1>

Name: External Loopback

Mnemonic: None

Type: R/W, DCLOC

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

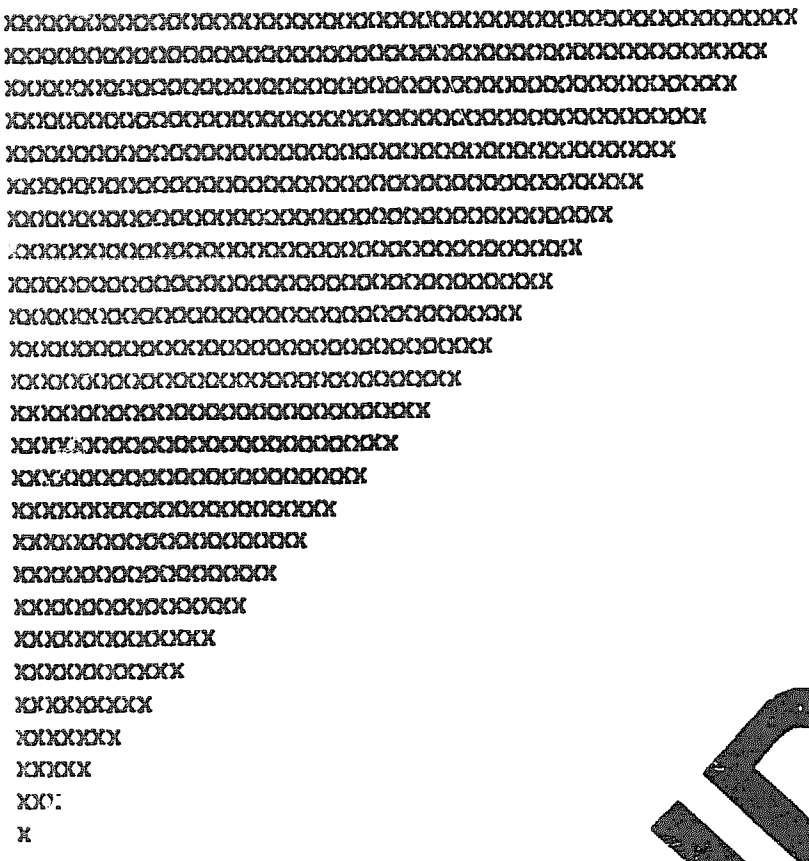
BIT<0>

Name: Firmware Initialize Complete

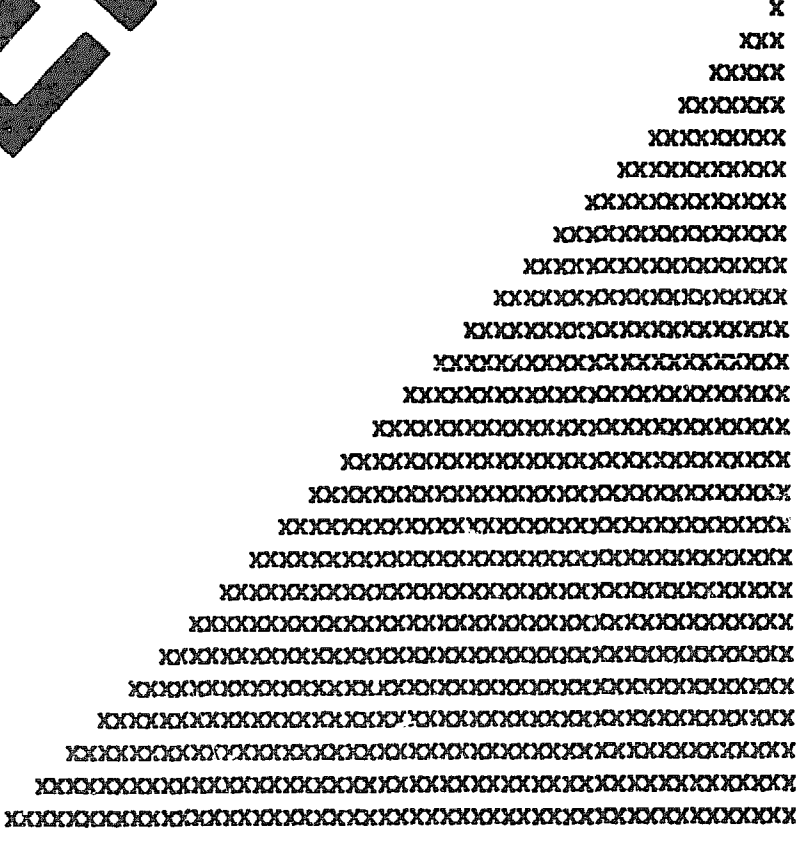
Mnemonic: None

Type: R/W, DCLOC

Sets when the DEBNI firmware completes initialization.



APPENDIX C



C Bootstrapping with the DEBNI

Most host systems in which the DEBNI resides can bootstrap their operating systems either locally (from disk or tape) or remotely (from an Ethernet network).

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 hardware/software then reads this parameter.

The DEBNI plays a role in the booting whenever (a) the specified device from which to boot is the DEBNI 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 DEBNI) instead of from a disk.

2 A remote system sends the DEBNI a command to boot involuntarily—that the DEBNI's host system should reboot itself.

C.1 Network Booting

The DEBNI 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 MOP message to the load assistant multicast address. This message 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 has volunteered 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 spot 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 DEBNI'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 two VAX systems.

On a VAX 82xx/83xx:

```
>>> B ET50
```

On a VAX 6xxx:

```
>>> B /XMI:D/BI:6 ET0
```

where:

D is the XMI node number of the XMI-to-VAXBI adapter

6 is the VAXBI node number of the DEBNI

C.2 Involuntary Booting

In an involuntary boot, a remote node sends the DEBNI a Maintenance Operations Protocol (MOP) Boot message. This operation is typically used for booting a system that is remotely located from an operator.

If the DEBNI port is in the uninitialized state, it responds to the Boot message if the following two conditions are met:

- The packet containing the Boot message is addressed to the DEBNI default physical address (DPA), which is the MAC address from the DEBNI's MAC Address ROM.
- The boot verification code in the Boot message matches the DEBNI default boot verification code of 424E49424F415244 (hex).

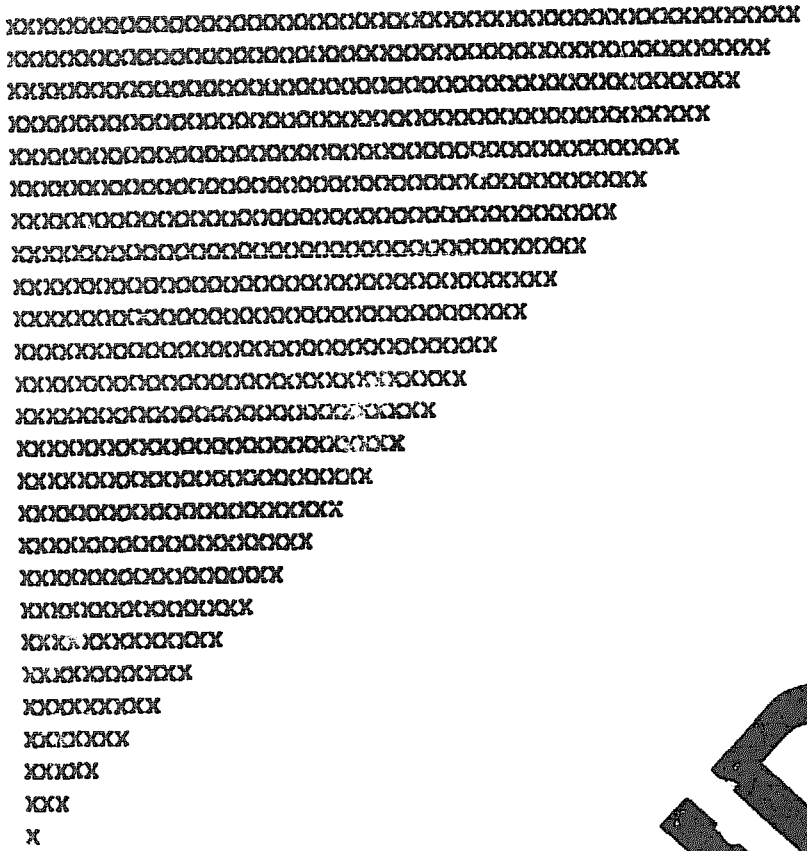
If the port is in the initialized state (which is the normal case), it responds to the Boot message if the following three conditions are met:

1 The packet containing the Boot message is addressed to the DEBNI default physical address (DPA). In this case, the DPA is either an address assigned by the host or, if no such address has been assigned, the MAC address from the DEBNI's MAC Address ROM.

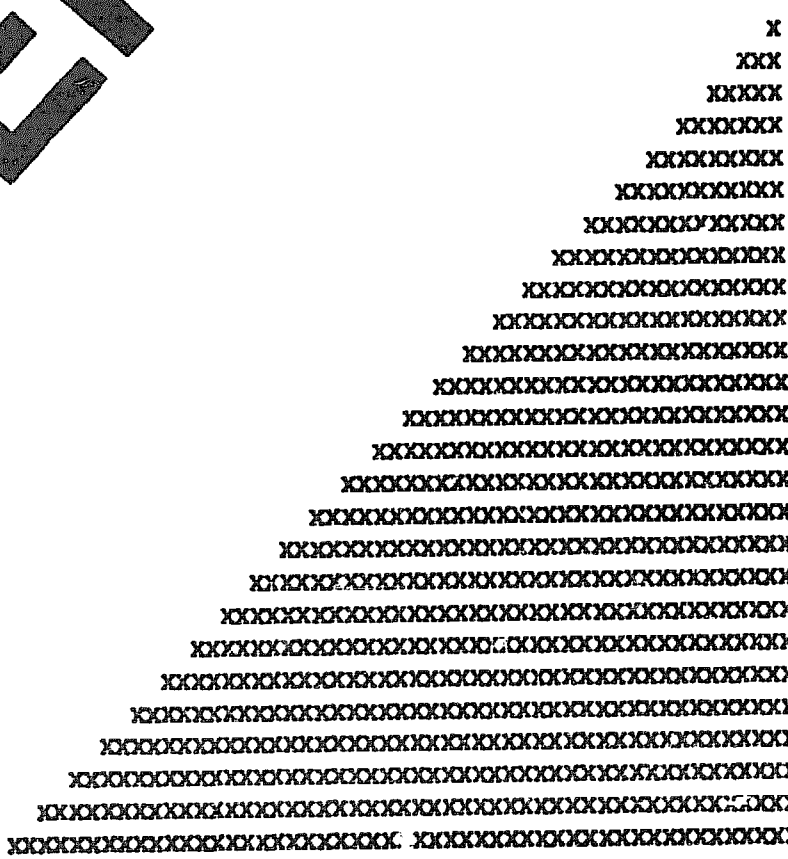
2 The boot verification code in the Boot message matches the DEBNI boot verification code. If the port driver did not assign the DEBNI a boot verification code, the DEBNI uses its default boot verification code.

3 The DEBNI's Boot Message Flag was set by the port driver to enable involuntary booting over the network.

In response to a valid Boot message, the DEBNI asserts BI RESET L on the VAXBI bus, which causes a VAXBI 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 Read the DEBNI Ethernet Address

The DEBNI's default Ethernet address, which is also called the default physical address (DPA), is stored in the DEBNI MAC address ROM. The DEBNI uses the DPA as its Ethernet address until the operating system assigns it a DECnet address during the operating system boot.

D.1 System with DECnet

If DECnet is running on your system, invoke the Network Control Program (NCP) and use the SHOW KNOWN LINE CHARACTERISTICS command to display the DEBNI's DPA as follows:

```
$ MC NCP

NCP>SHOW KNOWN LINE CHARACTERISTICS

Known Line Volatile Characteristics as of 26-APR-1989 16:06:41

Line = BNA-0

Receive buffers           = 21
Controller                = normal
Protocol                  = Ethernet
Service timer             = 4000
Hardware address          = 08-00-2B-05-F9-A7
Device buffer size        = 1498

NCP>EXIT
```

The hardware address is the DEBNI's DPA.

D.2 VAX 6xxx System

If you are on a VAX 6xxx system, use the SHOW ETHERNET console command to display the DEBNI DPA as follows:

```
>>>SHOW ETHERNET

XMI:E B:5 08-00-2B-08-CD-F3
```

The command displays the XMI node number of the XMI-to-VAXBI adapter, the VAXBI node number of the DEBNI, and the DEBNI DPA.

If the SHOW ETHERNET command cannot find the DEBNI, use the EXAMINE console command to read the DEBNI DPA. The address is stored in six bytes at the following VAXBI addresses:

bb + 218

bb + 219

bb + 21A

bb + 21B

bb + 21C

bb + 21D

where address (bb) is the base address of the DEBNI nodespace computed (in hex) as follows:

20000000 + (2000000 * XMI node ID of XMI-to-VAXBI adapter) + (2000 * DEBNI VAXBI node ID)

The following example shows how to examine the DPA of a DEBNI at VAXBI node D through an XMI-to-VAXBI adapter at XMI node C. The DTYPE register is examined first to confirm that the module being examined is a DEBNI (device type = 0118).

```
>>>E 3801A000                    ! Examine DTYPE register
   P 3801A000   00000118

>>>E/N:5/B 3801A218              ! Examine address
   P 3801A218   08
   P 3801A219   00
   P 3801A21A   2B
   P 3801A21B   08
   P 3801A21C   CD
   P 3801A21D   F3
```

The address bytes are displayed in the order that they are transmitted over Ethernet (in this example, 08-00-2B-08-CD-F3).

D.3 VAX 82xx/83xx Systems

If you are on a VAX 82xx or 83xx system, use the EXAMINE console command to read the DEBNI DPA. The address is stored in six bytes at the following addresses:

bb + 218

bb + 219

bb + 21A

bb + 21B

bb + 21C

bb + 21D

where address (bb) is the base address of the DEBNI nodespace computed (in hex) as follows:

20000000 + (2000 * DEBNI VAXBI node number)

The following example shows how to examine the DPA of a DEBNI at VAXBI node 5. The DTYPE register is examined first to confirm that the module being examined is a DEBNI (device type = 0118).

```
>>>E 2000A000                    ! Examine DTYPE Register
   P 2000A000 01000118
>>>E/B 2000A218                  ! Examine first byte
   P 2000A218 08
>>>E                             ! Examine second byte
   P 2000A219 00
>>>E                             ! Examine third byte
   P 2000A21A 2B
>>>E                             ! Examine fourth byte
   P 2000A21B 0B
>>>E                             ! Examine fifth byte
   P 2000A21C CE
>>>E                             ! Examine sixth byte
   P 2000A21D AB
```

The address bytes are displayed in the order that they are transmitted over Ethernet (in this example, 08-00-2B-0B-CE-AB).

D.4 VAX 85xx/87xx/88xx Systems

If you are on a VAX 85xx, 87xx, or 88xx system, use the EXAMINE console command to read the DEBNI DPA. The address is stored in six bytes at the following VAXBI addresses:

bb + 218

bb + 219

bb + 21A

bb + 21B

bb + 21C

bb + 21D

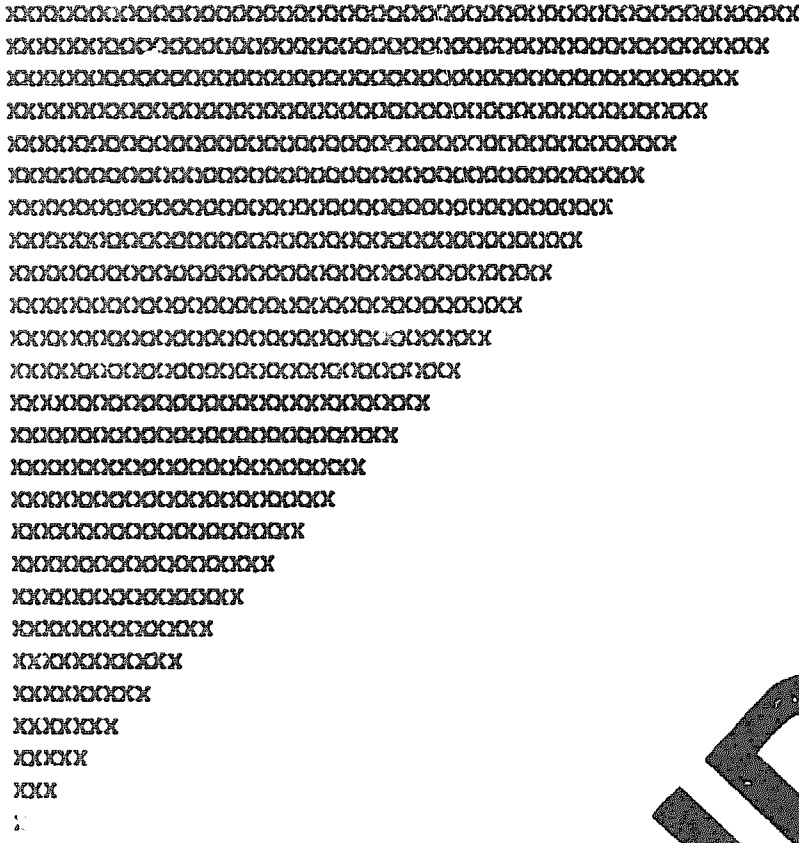
where address (bb) is the base address of the DEBNI nodespace computed (in hex) as follows:

20000000 + (2000000 * NBIA adapter number) + (2000 * DEBNI VAXBI node ID)

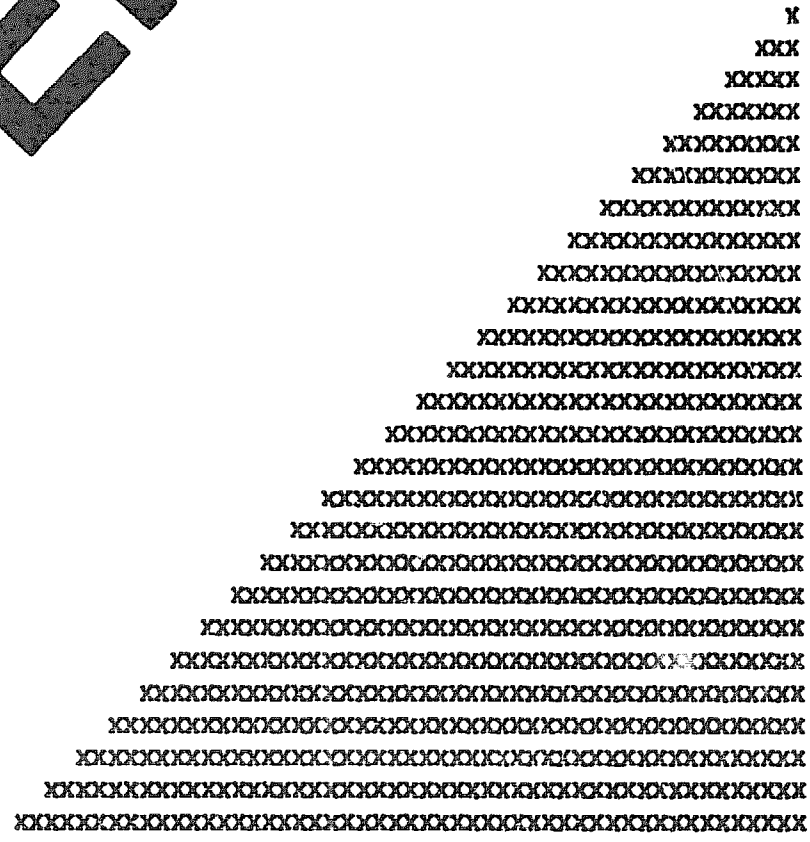
The following example shows how to examine the DFA of a DEBNI at VAXBI node 1. The NBIA adapter number is 1. The DTYPE register is examined first to confirm that the module being examined is a DEBNI (device type = 0118).

```
>>>E 22002000                    ! Examine DTYPE register
   P 22002000   000D0118
>>>E/N:5/B 22002000              ! Examine address
   P 22002218   08
   P 22002219   00
   P 2200221A   2B
   P 2200221B   08
   P 2200221C   CD
   P 2200221D   3F
```

The address bytes are displayed in the order that they are transmitted over Ethernet (in this example, 08-00-2B-08-CD-3F).



APPENDIX E



E How to Upgrade a DEBNA Module to a DEBNI Module

A DEBNA module can be upgraded to a DEBNI module in the field by replacing the four DEBNA ROMs with four DEBNI ROMs. In addition, a new module identification (ID) label and a new module revision label must be pressed into place over the corresponding old DEBNA labels. The upgrade changes the boot-enable jumper on the DEBNA module to the DEBNI firmware console-enable jumper (Section 2.4).

CAUTION

As indicated in the following table, the TBK50 option, which is a controller for the TK50 tape drive, has the same module number (T1034-00) as does the DEBNA. DO NOT attempt to upgrade a TBK50 to a DEBNI, since the upgrade would disable the TBK50's ability to control the TK50 tape drive.

Option	Module No.	Controller Type	Device Type
TBK50	T1034-00	Tape Drive	410E
DEBNA	T1034-00	Ethernet	410F
DEBNI	T1034-YA	Ethernet	0118

Figure E–1 shows the ROM locations on the DEBNA module, as well as the location of the module labels. The insert sheet in the update kit also provides pertinent information about the upgrade.

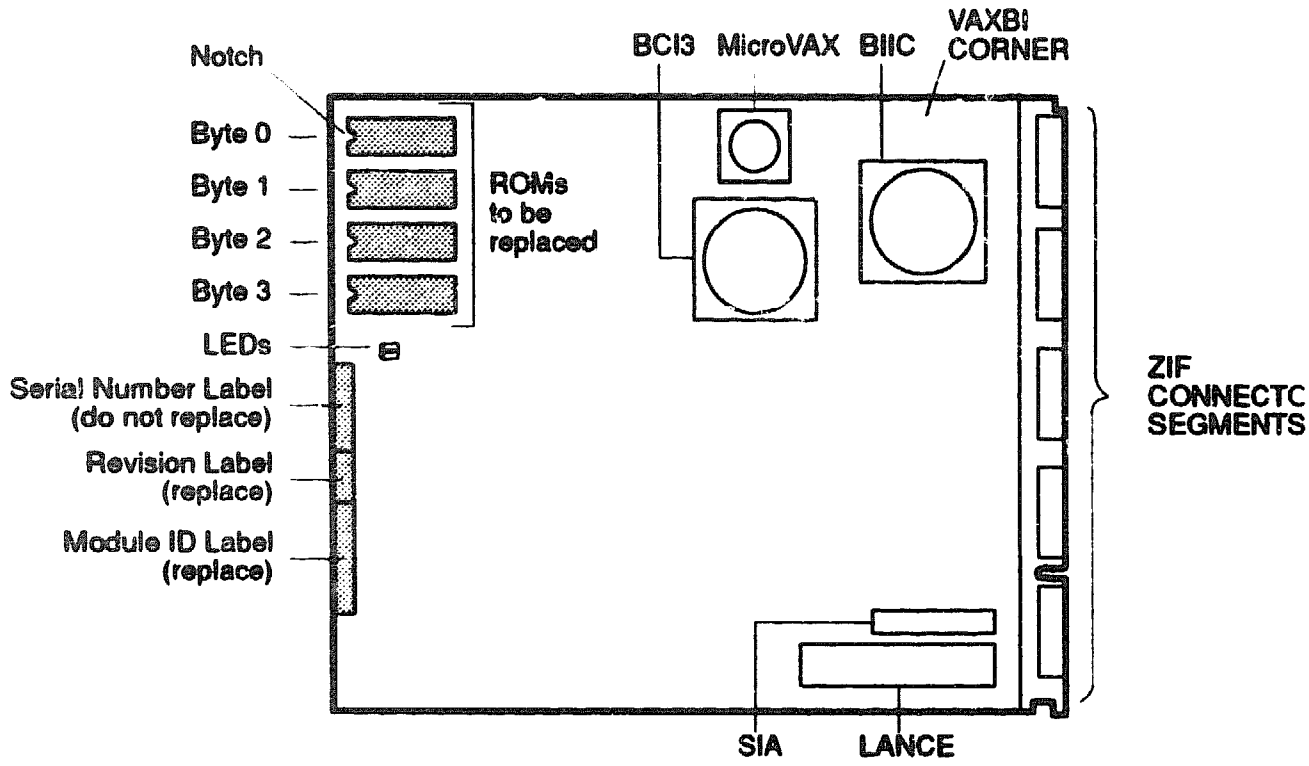
The following is a step-by-step procedure for upgrading a DEBNA to a DEBNI:

WARNINGS

POWER OFF—Shut off system power and disconnect the main system power cord before performing any procedure in this chapter.

STABILIZE THE CHASSIS—For 82xx/83xx Configuration 1 systems: make sure to extend the stabilizer leg(s) before you pull out the processor drawer.

Figure E–1 ROM and Label Locations



WEAR ESD WRIST STRAP—You must wear an antistatic wrist strap that is connected to the processor cabinet whenever you work inside the chassis.

USE CONDUCTIVE CONTAINERS—Whenever you remove a circuit board from a VAXBI card cage, place it in a conductive container.

1 Power down the host computer system by:

   a. Turning the POWER switch to the OFF position

   b. Setting the system circuit breaker in the rear to OFF

2 For 82xx/83xx Configuration 1 systems, extend the cabinet stabilizer leg(s)

3 Open the cabinet.

4 Make sure you are wearing the ESD wrist strap that is attached to the system's chassis.

5 For 82xx/83xx Configuration 1 systems, slide out the card cage and rotate it until it locks into the vertical position.

6 Remove the card cage cover.

7 Locate the slot that contains the DEBNA module.

8 Lift the locking lever to open the slot.

9 Slide the DEBNA out of the slot.

CAUTION

Use extreme care when removing and installing the ROMs. Failure to do so could damage the chip leads.

10 Carefully remove the four ROMs from the DEBNA module.

11 If necessary, gently bend inward the two rows of pins on each ROM so that the ROM will fit into the socket:

a. Place the ROM on its side on a table top covered with an anti-static sheet.

b. Press down very carefully on the ROM so that the row of pins touching the table top bends inward slightly.

c. Turn the ROM over on its other side and repeat the procedure for the other row of pins.

d. Make small adjustments in this way until the ROM fits into the socket.

12 Install the new ROMs into the four ROM sockets. Normally, the lowest-numbered ROM should be installed in the socket for byte 0 (see Figure E–1), the next higher-numbered ROM into the socket for byte 1, and so forth. However, see the insert sheet in the upgrade kit for an illustration showing where each ROM should be installed. Align the chip leads carefully when inserting the ROMs into the sockets. Make sure that the notched end of each ROM is near the board's edge (see Figure E–1).

13 Press the module ID label into place over the old module ID label (see Figure E–1).

14 Choose the appropriate module revision label according to the instructions on the insert sheet in the upgrade kit. Then press this label into place over the old module revision label (see Figure E–1).

NOTE

The serial number label on the module is *not* replaced.

15 Slide the converted module back into its card cage slot until the module stops: this is a zero insertion force card cage.

16 Close the locking lever.

17 Replace the card cage cover.

18 For 82xx/83xx Configuration 1 systems, rotate the card cage to the horizontal position and retract the stabilizer leg(s).

19 Close the cabinet.

20 Verify the operation of the converted module as described in Section 2.2.

21 If there are any verification problems, make sure that the ROMs are installed in the correct order and that each ROM is seated and oriented properly.


INDEX

Index