EK-DEUNA-UG-001

DEUNA USER'S GUIDE

digital

EK-DEUNA-UG-001

DEUNA USER'S GUIDE

Prepared by Educational Services
of
Digital Equipment Corporation

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

Printed in U.S.A.

The manuscript for this book was created on a DIGITAL Word Processing System and, via a translation program, was automatically typeset on DIGITAL's DECset Integrated Publishing System. Book production was done by Educational Services Development and Publishing in Nashua, NH.

CONTENTS

Page

CHAPTER 1 INTRODUCTION

CHAPTER 2 INSTALLATION

CHAPTER 3 SERVICE

CHAPTER 4 PROGRAMMING

APPENDIX A FLOATING DEVICE ADDRESSES AND VECTORS

APPENDIX B REMOTE BOOTING AND DOWN-LINE LOADING

APPENDIX C NETWORK INTERCONNECT EXERCISER

APPENDIX D VECTOR ADDRESS (REVB)

APPENDIX E DEUNA MICROCODE ECO PROCESS

FIGURES

TABLES

Table No. Title Page

EXAMPLES

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This chapter introduces the DIGITAL Equipment UNIBUS Network Adapter (DEUNA), including its operation and specifications, and overviews the ETHERNET local area network. Additional documents are listed for the reader who wishes more information about the ETHERNET, the DEUNA, or local area networks.

1.2 ETHERNET OVERVIEW

ETHERNET is a local area network that provides a communications facility for high-speed data exchange among computers and other digital devices located within a moderately sized geographic area. It is intended primarily for use in such areas as office automation, distributed data processing, terminal access, and other situations requiring economical connection to a local communication medium carrying traffic at high-peak data rates.

The primary characteristics of ETHERNET include:

- Topology – Branching bus
- Medium – Shielded coaxial cable, Manchester encoded digital base-band signaling
- Data Rate (Physical Channel) – 10 million bits per second (maximum)
- Maximum Separation of Nodes – 2.8 kilometers (1.74 miles)
- Maximum Number of Nodes – 1,024
- Network Control – Multiaccess – fair distribution to all nodes
- Access Control – Carrier Sense, Multiple Access with Collision Detect (CSMA/CD)
- Packet Length – 64 to 1518 bytes (includes variable data field of from 46 to 1500 bytes less 8-byte preamble).

The ETHERNET falls into a middle ground between long-distance, low-speed networks that carry data for hundreds or thousands of kilometers and specialized high-speed interconnections generally limited to tens of meters. Using a branching bus topoloy, ETHERNET provides a local area communications network allowing a 10M bits/s data rate over a coaxial cable at a distance of up to 2.8 km (1.74 mi).

A single ETHERNET can connect up to 1,024 nodes for a local point-to-point/multipoint network. An example of a typical large-scale ETHERNET configuration is shown in Figure 1-1.

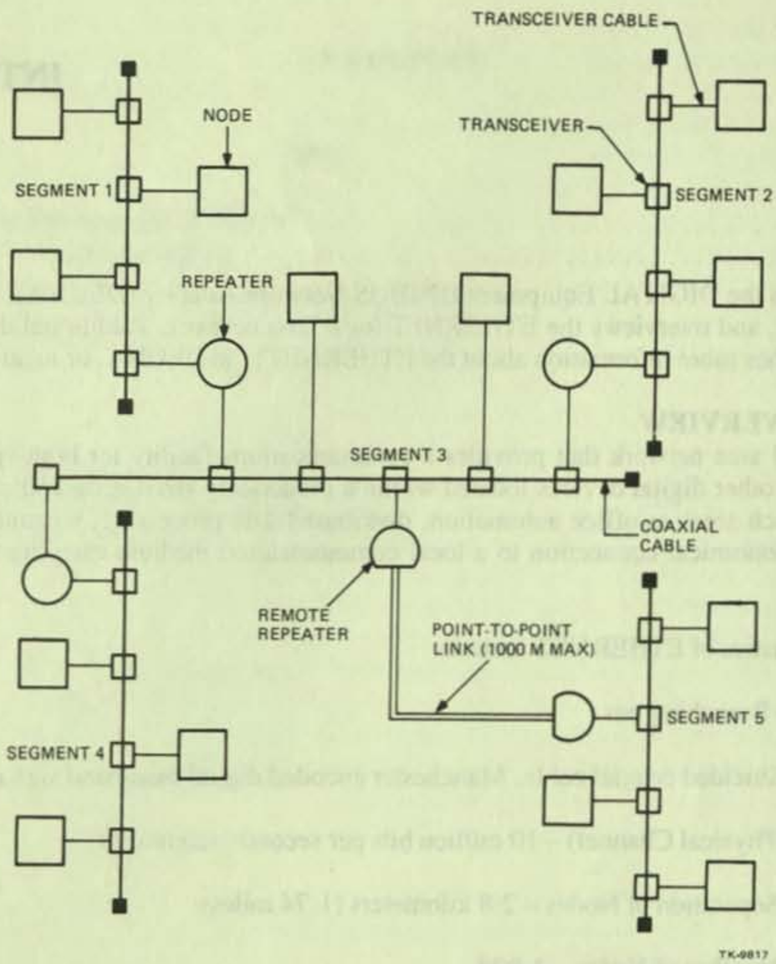


Figure 1-1 Typical Large-Scale ETHERNET Configuration

To configure ETHERNET, certain limits are imposed on the physical channel to ensure the optimal performance of the network. The maximum configuration for an ETHERNET is as follows:

- A segment of coaxial cable can be a maximum of 500 meters (1640.5 feet) in length. Each segment must be terminated at both ends in its characteristic impedance.
- Up to 100 nodes can be connected to any segment of the cable. Nodes on a cable segment must be spaced at least 2.5 meters (8.2 feet) apart.

- The maximum length of coaxial cable between any two nodes is 1,500 meters (4921.5 feet).
- The maximum length of the transceiver cable between a transceiver and a controller is 50 meters (164.05 feet).

NOTE

In addition to internal cabling between the DEUNA and its bulkhead assembly, the DEUNA will support an additional 40 meters of transceiver cable.

- A maximum of 1,000 meters (3281 feet) of point-to-point link is allowed for extending the network.
- Repeaters can be used to continue signals from one cable segment of the ETHERNET to another. A maximum of two repeaters can be placed in the path between any two nodes.

1.3 DEUNA GENERAL DESCRIPTION

The DEUNA is a data communications controller used to interface VAX-11 and PDP-11 family computers to the ETHERNET local area network. Features of the DEUNA include:

- High speed transmission and reception
- 10M bit data rate
- Transmit and receive data link and buffer management
- Data encapsulation and decapsulation
- Data encoding and decoding
- Collision detection and automatic retransmission
- 32-bit Cyclic Redundancy Check (CRC) error detection
- 32 KB (16 KW) buffer for datagram reception transmission, and maintenance requirements
- Down-line loading and remote load detect capabilities
- Internal ROM based microdiagnostics to facilitate diagnosis and maintenance of both the DEUNA-AA and the DIGITAL H4000 transceiver
- Unique 48-bit Default Physical Address (reprogrammable)

The DEUNA has two hex-height modules, a bulkhead interconnect panel, and associated cables. It physically and electrically connects to the ETHERNET cable via the DIGITAL H4000 transceiver and the appropriate transceiver cable as shown in Figure 1-2.

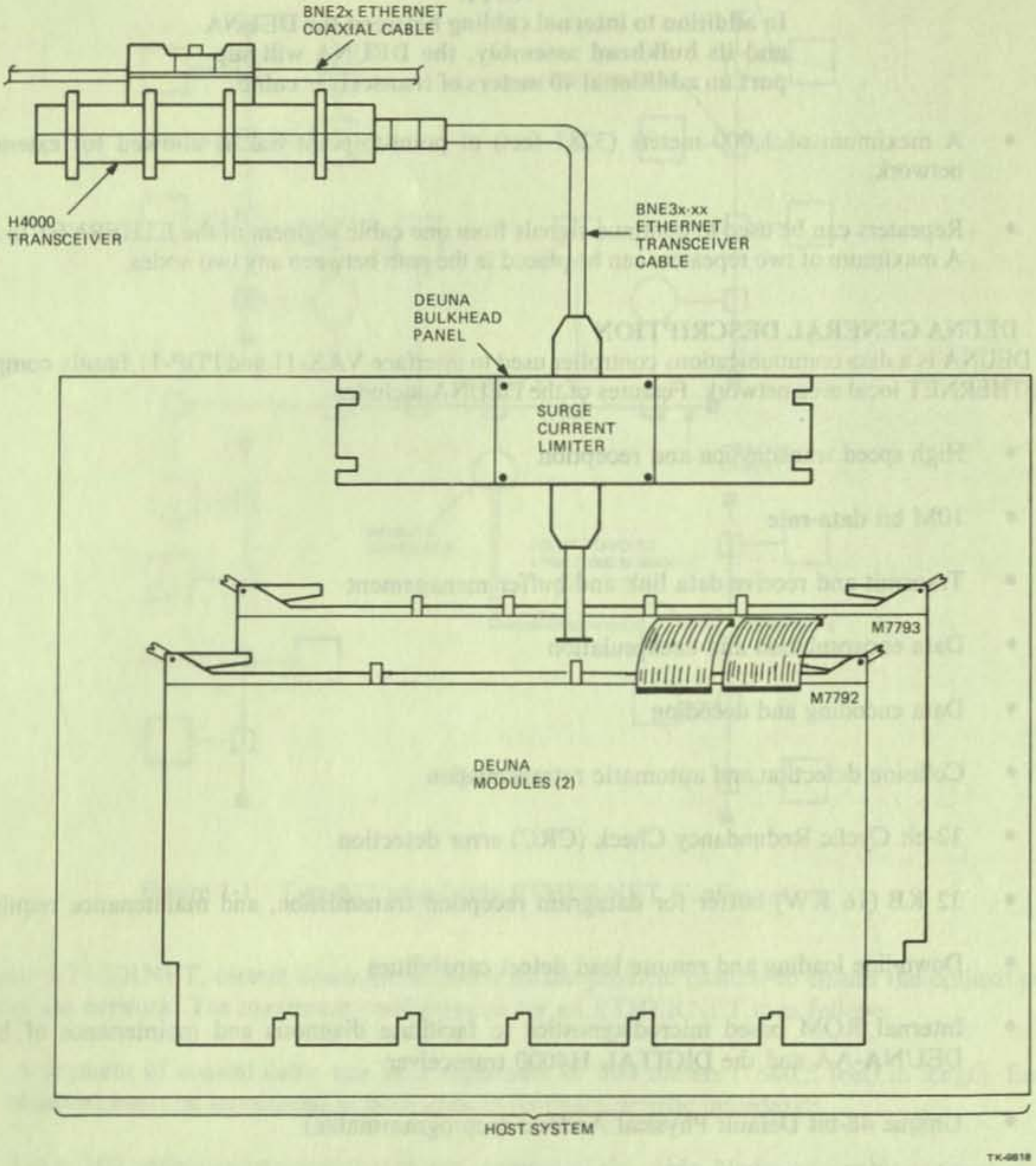


Figure 1-2 DEUNA to ETHERNET Connection

1.4 DEUNA SYSTEM OPERATION

The DEUNA controller performs both the ETHERNET data link layer functions and a portion of the physical channel functions. It also provides the following network maintainability features.

- Loopbacks maintenance messages from other stations.
- Periodically transmits system identification.
- Loads and remotely boots UNIBUS systems from other stations on the network.

The DEUNA is a microprocessor-based device that, when connected to the DIGITAL H4000 ETHERNET transceiver, provides all the logic necessary to connect VAX-11 and UNIBUS PDP-11 family minicomputers to an ETHERNET local area network (Figures 1-3 and 1-4). The controller performs data encapsulation and decapsulation, data link management, and all channel access functions to ensure maximum throughput with minimum host processor intervention.

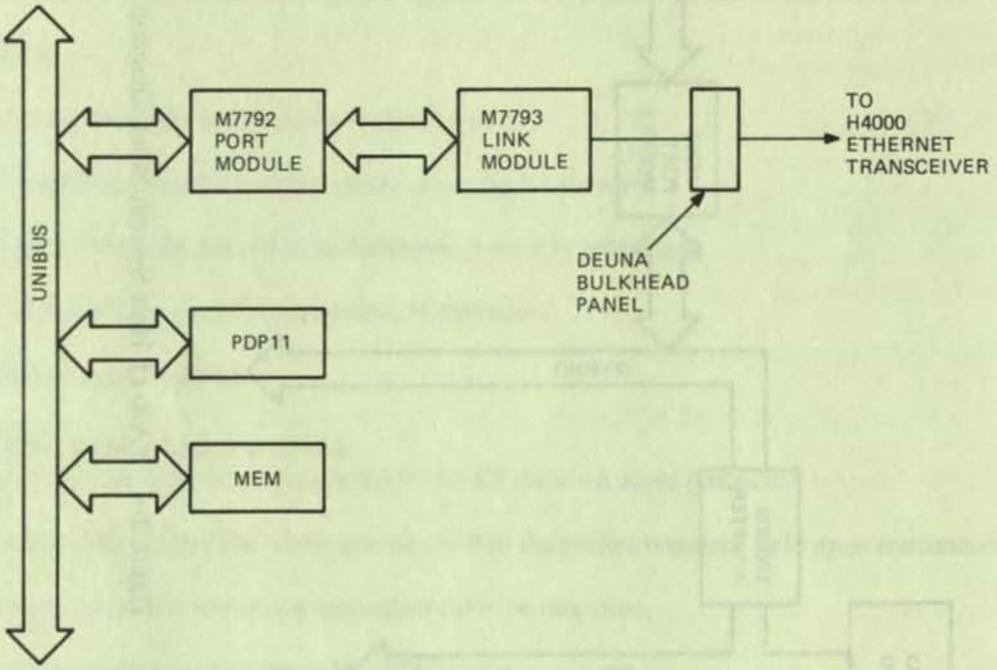


Figure 1-3 PDP-11 Host System Block Diagram

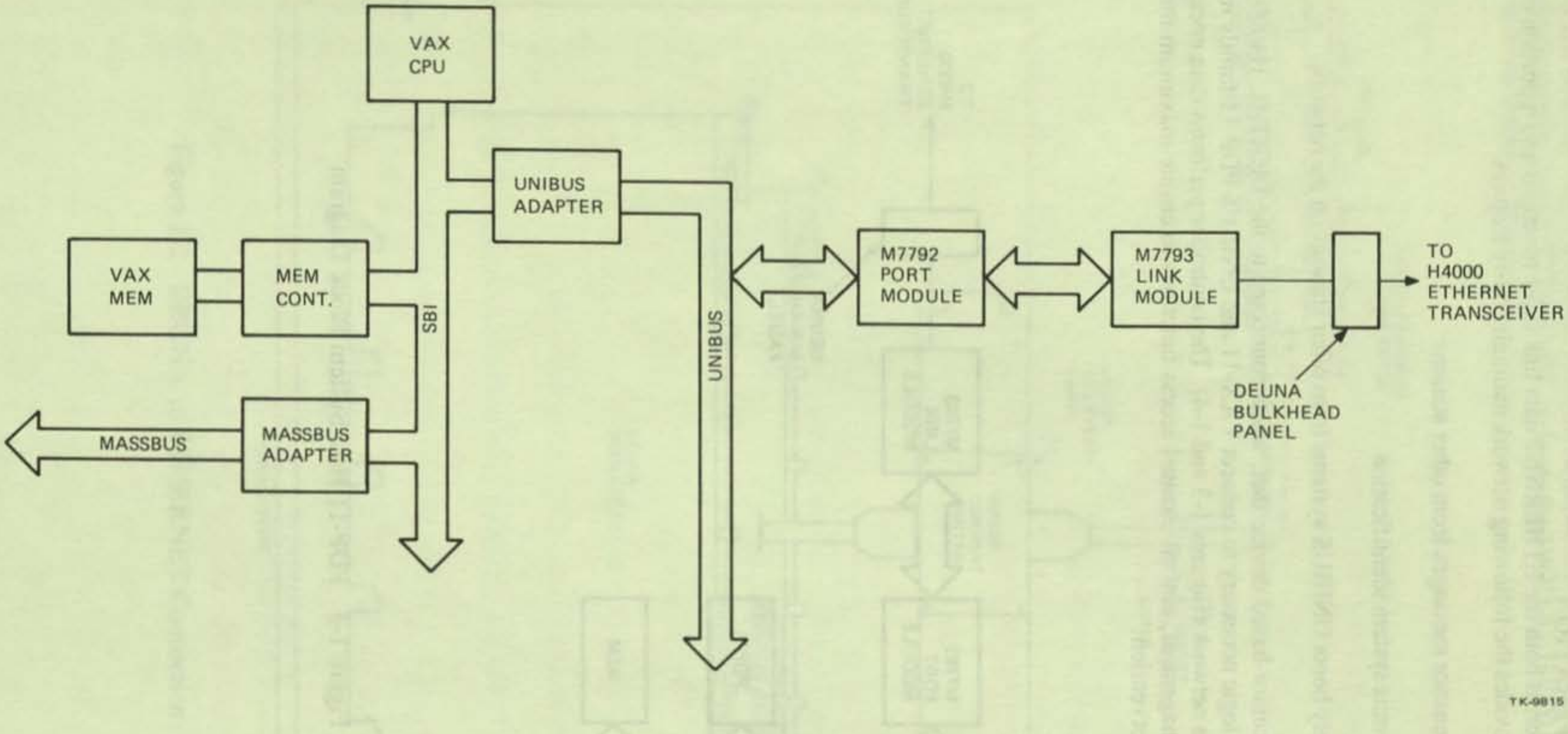


Figure 1-4 VAX-11 Host System Block Diagram

1.4.1 ETHERNET Physical Channel Functions

The DEUNA provides the following specific ETHERNET physical channel functions necessary to interface to the DIGITAL H4000 ETHERNET transceiver:

During Transmission

- Generates the 64-bit preamble for synchronization.
- Provides parallel-to-serial conversion of the frame.
- Generates the Manchester encoding of data.
- Ensures proper channel access by monitoring and sensing the carrier from any stations' transmission.
- Monitors the self-test collision detect signal from the DIGITAL H4000 transceiver.

During Reception

- Senses carrier from any station's transmission.
- Performs Manchester decoding of the incoming bit streams.
- Synchronizes to the preamble and removes it prior to processing.
- Provides serial-to-parallel conversion of the frame.
- Buffers received packets.

1.4.2 ETHERNET Data Link Functions

The DEUNA provides the following specific ETHERNET data link layer functions:

- Calculates the 32-bit CRC value and places it in the packet sequence field upon transmission.
- Attempts automatic retransmission upon collision detection.
- Checks incoming packets for proper CRC value.
- Performs address filtration.

1.4.3 Data Encapsulation

The ETHERNET packet format is shown in Figure 1-5. Each packet begins with a 64-bit preamble used for synchronization by the receiving station, and ends with a 32-bit packet check sequence. Packets are separated by a specified minimum spacing period of 9.6 μs.

The destination address field contains the address(es) of the station(s) where the packet is sent. The address may represent: the unique or physical addresses of a particular station; a multicast, or group address, associated with a set of stations; and a broadcast address for all stations on the network.

The source address field specifies the physical address of the transmitting station. Each DEUNA has a unique 48-bit address value determined during manufacture. This value is called the default physical address. The system software can override this value and assign a different physical address.

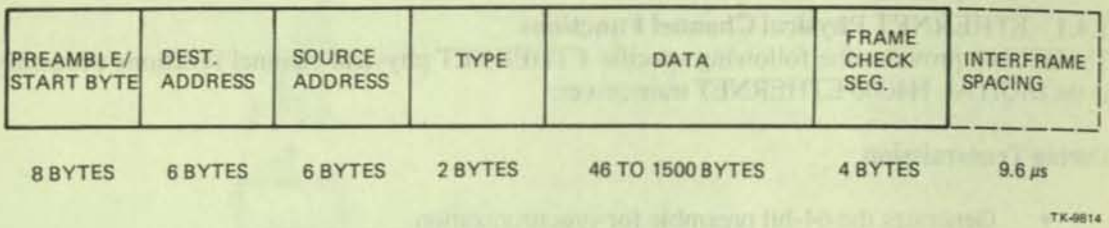


Figure 1-5 Format of an ETHERNET Data Packet

The type field is specified for use by high-level network protocols. It indicates how the content of the data field is to be interpreted. The type field is used by higher-level architecture to further decapsulate the data.

The data field may have between 46 and 1500 bytes of data. The DEUNA can be initialized to automatically insert null characters if the amount of data is less than the minimum 46-byte data size.

The packet check sequence contains a 32-bit Cyclic Redundancy Check (CRC) value determined and inserted by the DEUNA during transmission.

1.4.4 Data Decapsulation

The DEUNA continuously monitors the signals transmitted by the DIGITAL H4000 transceiver. After sensing a carrier, the preamble sequence of the received packet is used by the controller for synchronization. It then processes the destination address field through a hardware comparator to determine whether or not the incoming packet is intended for its station. The DEUNA accepts only packets with a destination address that matches one of the following types of address:

1. The physical address of the station
2. The broadcast address for all stations
3. One of the ten multicast group addresses the user may assign to the DEUNA, when desired
4. Any multicast address, when desired
5. All addresses, when desired

The DEUNA performs a hardware comparison of the 6-byte destination address to determine if there is a match with the station's physical address or with one of the ten user-designated multicast addresses. If necessary, all multicast addresses may be passed to higher-level software for decoding when more than ten multicast address groups are required by the user.

To assist in network management functions and fault diagnosis, the DEUNA can operate in a mode that effectively disregards the internal address filter logic. This allows all packets received from the network to be accepted. The DEUNA verifies the integrity of the received data by performing a 32-bit CRC check on the received packet.

1.4.5 Link Management

The method by the ETHERNET for channel access is called carrier sense, multiple access with collision, detect (CSMA/CD). The DEUNA controls all of the link management functions necessary to successfully place or remove a packet of data on the ETHERNET network. These functions include:

- **Carrier Deference** – The DEUNA monitors the physical channel for traffic. When the channel is busy, it defers to the passing packet by delaying any transmission of its own.
- **Collision Detection** – Collisions occur when two or more stations attempt to transmit data simultaneously on the channel. The DEUNA monitors the collision sense signal generated by the DIGITAL H4000 transceiver. When a collision is detected, the DEUNA continues to transmit long enough to ensure that all network stations detect the collision.
- **Collision Backoff and Retransmission** – When a controller attempts transmission and encounters a collision on the channel, it attempts a retransmission a short time later. The schedule for retransmission is determined by a controlled randomization process. The DEUNA attempts to transmit a total of sixteen times and reports an error if it is not successful.

1.4.6 Functional Overview

The DEUNA is a microprocessor-controlled interface between the UNIBUS (host memory) and the ETHERNET. It has two basic functions: Receive and Transmit.

1.4.6.1 Receive Function – Figure 1-6 shows the data path through the DEUNA for the receive function. The data travels through the DEUNA as follows:

1. Data from the ETHERNET is received by the LINK which:
   - Performs Manchester decoding of data
   - Decapsulates data
   - Filters address
   - Converts serial to parallel data
   - Checks CRC
   - Moves data to local memory and notifies T11 that there is a message to be sent to host memory
2. When the message is in local memory, the T11 microprocessor gets the starting address of where the message is to go in host memory and sets up the DMA engine.
3. The DMA engine moves the message to host memory.
4. After data is moved the T11 informs the host of the message.

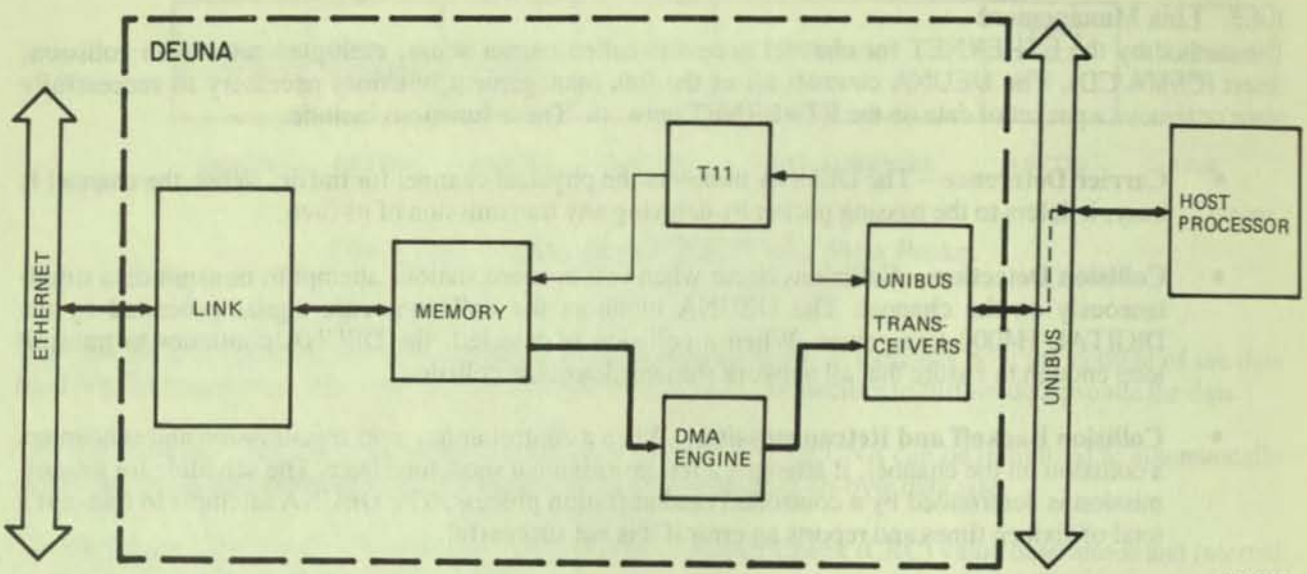


Figure 1-6 DEUNA Receive Data Path

1.4.6.2 Transmit Function – Figure 1-7 shows the data path through the DEUNA for the transmit function. The data travels through the DEUNA as follows:

1. The host processor notifies the T11 that there is a message the host wants transmitted on the ETHERNET.

2. The T11 moves the message from host memory to local memory via DMA and tells the link there is a message to be transmitted.

3. The link transmits the message by doing the following:

   - Generates the preamble.

   - Performs parallel-to-serial conversion.

   - Generates CRC.

   - Performs manchester encoding of data.

   - Transmits the message.

   - If there is a collision on the ETHERNET, attempts to re-transmit the message up to 15 times.

4. The T11 notifies the host when the message has been transmitted.

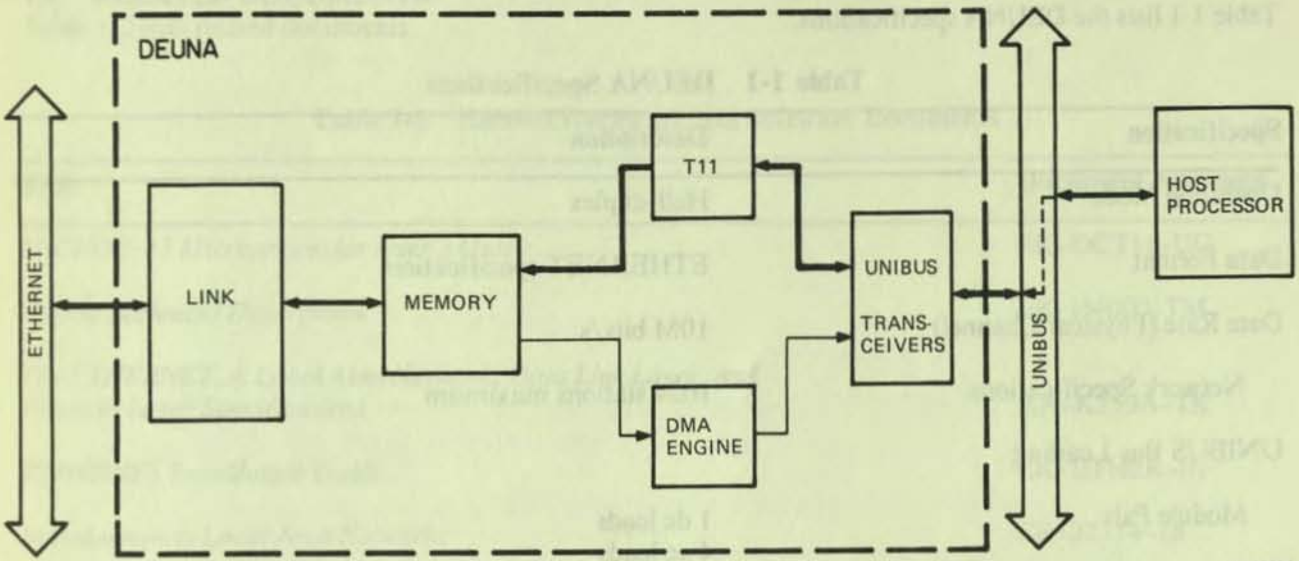


Figure 1-7 DEUNA Transmit Data Path

1.4.7 Diagnostics and Maintenance

The DEUNA utilizes both microdiagnostics and extensive system and network diagnostics to greatly minimize the time to isolate and diagnose a network communication fault. On-board self-test microdiagnostics automatically test the major DEUNA component logic both on powerup and at the user's discretion. Light-emitting diodes on the edge of the port module (M7792) indicate a specific module problem.

The DEUNA does not transmit longer than the maximum ETHERNET packet transmit period. It contains an automatic control to prevent monopolizing the ETHERNET channel. A built-in Time Domain Reflectometry circuit is provided to help find the location and nature of cable faults.

The controller continuously monitors the power applied to the DIGITAL H4000 transceiver to ensure compliance with the tranceiver requirements. In addition, the H4000 provides a positive functional verification (heartbeat) after every attempted transmission which indicates its proper operation, including the collision sense circuitry.

Comprehensive system diagnostics provide loopback capability through the DEUNA, transceiver, or the ETHERNET network itself. The DEUNA allows remote stations to loopback once it has successfully passed the on-board self-test microdiagnostic. This provides both a local and remote station diagnostic capability. Network error conditions are detected and statistics tabulated for use by higher level network management applications.

1.5 DEUNA SPECIFICATIONS

Table 1-1 lists the DEUNA specifications.

Table 1-1 DEUNA Specifications

Specification	Description
Operating Mode	Half-duplex
Data Format	ETHERNET specification
Date Rate (Physical Channel)	10M bits/s
Network Specifications	1024 stations maximum
UNIBUS Bus Loading	
Module Pair	1 dc loads 4 ac loads
DC Power Requirements	
Port Module Link Module	+5 V, 7.0 A +5 V, 9.0 A −15 V, 1.0 A (for H4000 transceiver)
Physical Size	
Port and Link Modules	Height (hex): 21.4 cm (8.4 in) Length: 39.8 cm (15.7 in)
Cable Interface Panel	Height: 10.6 cm (4.0 in) Length: 10.6 cm (4.0 in)
Operating Environment	
Temperature	10°C to 40°C (50°F to 104°F)
Relative Humidity	10 to 90% (noncondensing)
Wet Bulb Temperature	28°C (82°F) maximum
Dew Point	2°C (36°F) minimum
Altitude	Sea level to 2.4 km (8,000 ft)
Shipping Environment	
Temperature	−40°C to 66°C (−40°F to 151°F)
Relative Humidity	0 to 90% (noncondensing)
Altitude	Sea level to 9 km (30,000 ft)

1.6 RELATED DOCUMENTS

Table 1-2 lists related documents.

Table 1-2 Related Hardware and Software Documents

Title	Document Numbers
MICRO T-11 Microprocessor User's Guide	EK–DCT11–UG
H4000 Technical Description	EK–H4000–TM
The ETHERNET, A Local Area Network, Data Link Layer, and Physical Layer Specifications	AA–K759A–TK
ETHERNET Installation Guide	EK–ETHER–IN
Introduction to Local Area Networks	EB–22714–18
DEUNA Maintenance Print Set	MP–10378

DIGITAL personnel may order hardcopy documents from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532

Attn: Publishing and Circulation Services (NRO3/W3)
Order Processing Section

Customers may order hardcopy documents from:

Digital Equipment Corporation
Accessories and Supplies Group
Cotton Road
Nashua, New Hampshire 03060

For information call: 1-800-257-1710

Information concerning microfiche libraries may be obtained from:

Digital Equipment Corporation
Micropublishing Group (BUO/E46)
12 Crosby Drive
Bedford, MA 01730

CHAPTER 2
INSTALLATION

2.1 SCOPE

This chapter provides the necessary information and procedure for installing a DEUNA in a PDP-11 or VAX-11 host system. The chapter is divided into the following sections.

- Unpacking and Inspection – Verify that shipment is complete and undamaged.
- Preinstallation Considerations – Verify that the host system meets the installation requirements of the DEUNA.
- Preinstallation Preparation – Prepare the host system and the DEUNA subsystem for proper operation.
- Installation and Cabling – Install and cable the DEUNA in the host system.
- Testing – Verify that the DEUNA and the host system are operating correctly.

2.2 UNPACKING AND INSPECTION

Unpacking a DEUNA subsystem consists of removing the equipment from its shipping containers, verifying that there are no missing parts, and inspecting the equipment for damage. Report any damages or shortages to the shipper and notify the DIGITAL representative.

1. Before opening the shipping containers, check them for external damage such as dents, holes, or crushed corners.
2. Open and unpack each container. Inventory the contents against the shipping list. Table 2-1 lists the parts supplied with each DEUNA subsystem.

NOTE

Shipping containers and packing materials should be retained if reshipment is contemplated.

3. Inspect each DEUNA part for shipping damage. Check the modules carefully for cracks, breaks, or loose components such as socketed chips.

Table 2-1 DEUNA Parts List

Description	Part Number
DEUNA Port Module	M7792
DEUNA Link Module	M7793
Module Interconnect Cable	BCO8R-1 (2)
Bulkhead Cable Assembly	70-18798-08
Bulkhead Interconnect Panel Assembly	70-18799-00
DEUNA User's Guide	EK-DEUNA-UG

2.3 PREINSTALLATION CONSIDERATIONS

The following factors should be considered before installing a DEUNA to verify that the host system can accept the DEUNA and that it can be installed correctly.

2.3.1 Backplane Requirements

The DEUNA requires two hex-height, Small Peripheral Controller (SPC) slots that can be configured for Nonprocessor Request (NPR) operation. Two adjacent slots are preferred, but not necessary. Any SPC backplane (DD11-B(REV E) or later) can accept the DEUNA modules. The DEUNA can be placed anywhere on the UNIBUS before all UNIBUS repeaters.

2.3.2 Bus Latency Constraints

Bus latency is an important factor in determining where to place the DEUNA in the backplane. On systems with many high-speed Direct Memory Access (DMA) devices, the bus latency may adversely affect the DEUNA's performance.

To obtain optimum performance, select a backplane location that places DEUNA on the UNIBUS bus before all devices with a lower NPR rate and before all UNIBUS repeaters. The closer the physical placement of the DEUNA to the processor, the higher its DMA device priority. If optimum performance is not a factor, the DEUNA can be installed anywhere on the UNIBUS (before all repeaters) that meets the requirements of the system. Reconfigure the system as necessary to provide the DEUNA with backplane slots at the selected UNIBUS location for the desired performance.

2.3.3 Loading Requirements

Make sure that system loading capacities are not exceeded as a result of installing the DEUNA subsystem. Tables 2-2 and 2-3 list the UNIBUS loading and power supply current requirements of the DEUNA, respectively.

NOTE

Check power supply voltages before and after installation to verify that no overvoltage or overloading conditions exist.

Table 2-2 DEUNA UNIBUS Loading

Modules	UNIBUS DC Loads	UNIBUS AC Loads
M7792 & M7793 (combined)	1	4

Table 2-3 DEUNA Power Chart

Module	Voltage Rating	Maximum Voltage	Minimum Voltage	Backplane Pin
M7792	+ 5 Volts @ 7.0 A	+ 5.25 Volts	+ 4.75 Volts	CA2
M7793	+ 5 Volts @ 9.0 A	+ 5.25 Volts	+ 4.75 Volts	CA2
	-15 Volts @ 1.0 A (for H4000 Transceiver)	-15.75 Volts	-14.25 Volts	FB2

2.4 PREINSTALLATION PREPARATION

Prepare the host system and DEUNA subsystem for proper operation using the following procedure.

2.4.1 Backplane Power Checks and Preparation

Perform the following operations on the backplane slots previously selected for DEUNA module installation.

1. With system power OFF, conduct resistance checks on the backplane voltage sources to ground to be sure that no short circuit conditions exist.

2. Turn system power ON. Verify that backplane voltages are within specified tolerances. Refer to Table 2-3 for the voltage ranges and backplane pin assignments. Turn system power OFF.

3. If present, remove the grant continuity modules.

4. If present, remove the Nonprocessor Grant (NPG) jumper wire that runs between backplane pins CA1 and CB1 on the slot selected for installation of the M7792 port module.

NOTE

If the M7792 port module is removed from the system, be sure to either replace the NPG jumper wire and install a G727 single-height grant module, or install a G7273 dual-height grant module.

2.4.2 Device Address Assignment

Assign the DEUNA a device address from the Input/Output (I/O) page of memory address space. The first DEUNA being installed in a system must be assigned the address 774510. For the second, and any subsequent DEUNA being installed in the same system, the device address must be selected from the floating address space of the I/O page. The device address is assigned by configuring switch pack E40 on the M7792 port module to the desired address.

2.4.2.1 First DEUNA Device Address (774510) – Assign device address 774510 to the first DEUNA being installed in a system by configuring switch pack E40 on the M7792 port module as shown below. Note that this address could overlap the twenty-third (23rd) DP11 if present in the system. Refer to Figure 2-1 for the location of E40 on the M7792 module.

M7792 - E40									
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
OFF	ON	ON	OFF	ON	OFF	ON	ON	OFF	OFF

NOTE

An OFF (open) switch responds to a logical one (1) on the bus.

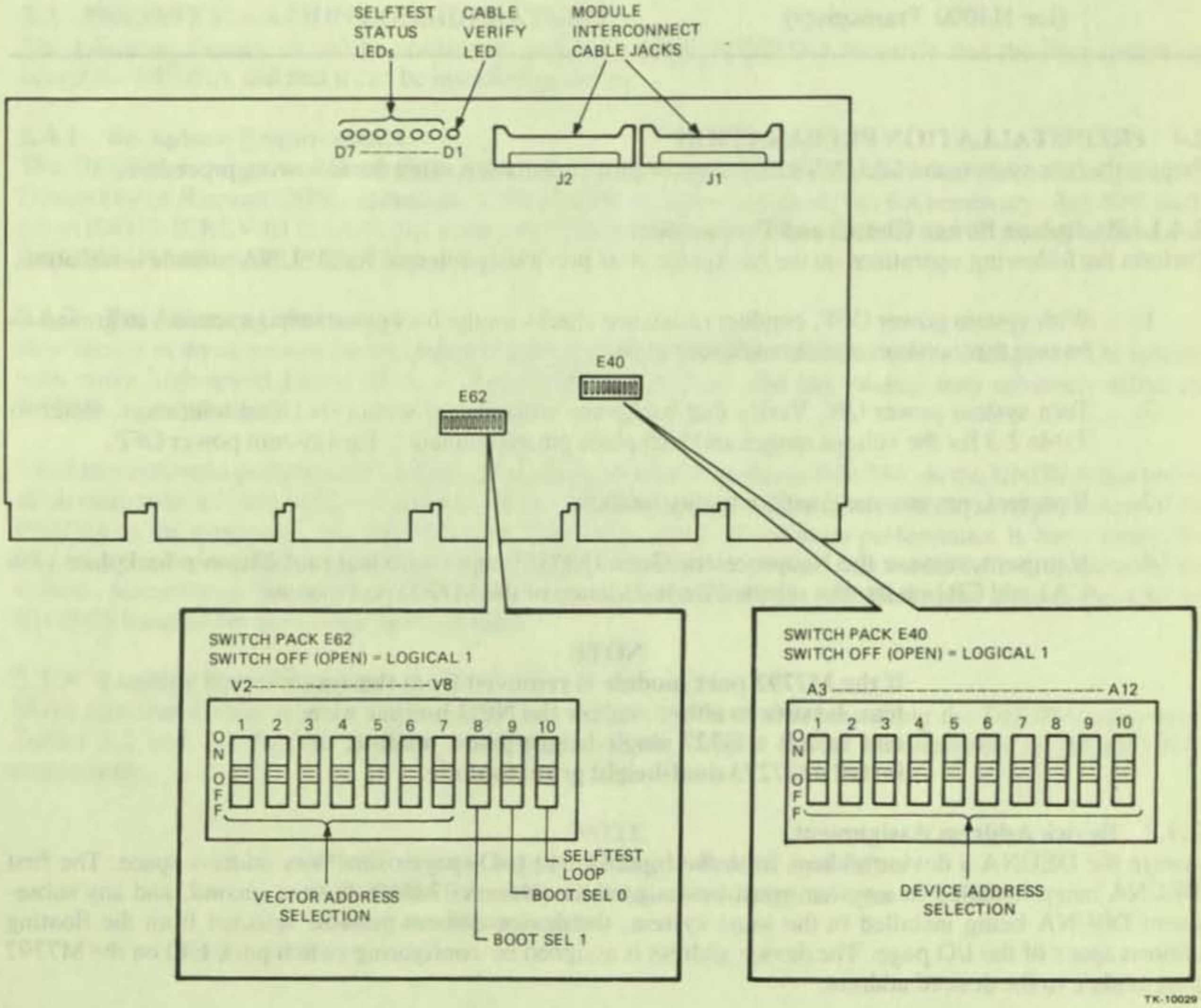


Figure 2-1 M7792 Port Module Physical Layout

2.4.2.2 Second DEUNA Device Address (Floating Address) – Assign a device address to the second (or subsequent) DEUNA being installed in a system by configuring switch pack E40 on the M7792 port module to the desired address determined from the floating address allocation. Refer to Table 2-4 for the correlation between switch number and address bit. The ranking device address assignment of the DEUNA is twenty-five (25). Refer to Appendix A for more information on floating address allocation.

Table 2-4 Floating Address Assignment

MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	SWITCH PACK E40										0	0	0

SWITCH NUMBER	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	FLOATING ADDRESS
										OFF	760010
									OFF		760020
									OFF	OFF	760030
								OFF			760040
								OFF		OFF	760050
								OFF	OFF		760060
								OFF	OFF	OFF	760070
							OFF				760100

						OFF					760200

						OFF	OFF				760300

					OFF						760400

					OFF		OFF				760500

					OFF	OFF					760600

					OFF	OFF	OFF				760700

				OFF							761000

			OFF								762000

			OFF	OFF							763000

		OFF									764000

NOTE: SWITCH OFF (OPEN) RESPONDS TO LOGICAL ONE ON THE UNIBUS.

TK-9760

2.4.3 Vector Address Assignment

NOTE
For M7792 Revision B modules, refer to Appendix D of this guide.

Assign the DEUNA a vector address from the reserved vector area of memory address space. The first DEUNA being installed in a system must be assigned the vector 120. The second (and any subsequent) DEUNA being installed in the same system must select the vector address from the floating vector area of reserved vector address space. The vector address is assigned by configuring switch pack E62 on the M7792 port module to the desired vector.

2.4.3.1 First DEUNA Vector Address (120) – Assign vector address 120 to the first DEUNA in the system by configuring S1-S7 of switch pack E62, on the M7792 port module, as shown below. Note that this vector is also used by the XY11. Refer to Figure 2-1 for the location of E62 on the M7792 module.

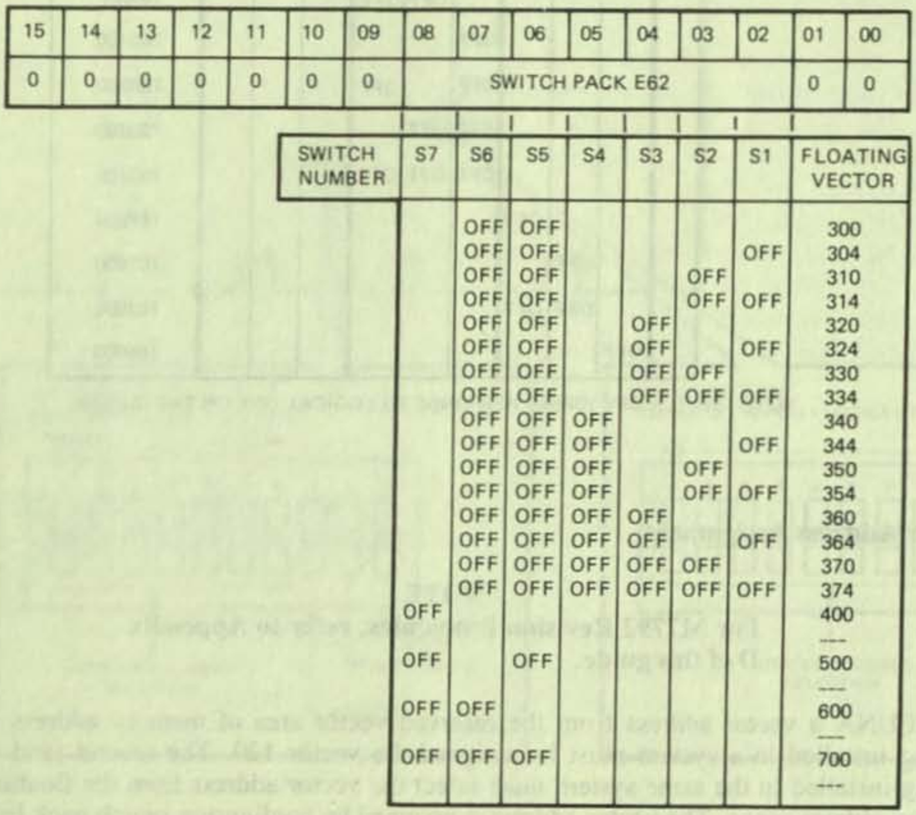
M7792 - E62

S1	S2	S3	S4	S5	S6	S7
ON	ON	OFF	ON	OFF	ON	ON

NOTE
An OFF (open) switch produces a logical one (1) on the bus.

2.4.3.2 Second DEUNA Vector Address (Floating Vector) – To assign a vector address to the second (or subsequent) DEUNA, configure S1-S7 of switchpack E62 on the M7792 port module to the desired vector determined from the floating vector allocation. Refer to Table 2-5 for the correlation between switch number and address bit. The ranking vector address assignment of the DEUNA is forty-seven (47). Refer to Appendix A for more information on floating vector allocation.

Table 2-5 Floating Vector Assignment



15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	SWITCH PACK E62							0	0

SWITCH NUMBER	S7	S6	S5	S4	S3	S2	S1	FLOATING VECTOR
		OFF	OFF					300
		OFF	OFF				OFF	304
		OFF	OFF			OFF		310
		OFF	OFF			OFF	OFF	314
		OFF	OFF		OFF			320
		OFF	OFF		OFF		OFF	324
		OFF	OFF		OFF	OFF		330
		OFF	OFF		OFF	OFF	OFF	334
		OFF	OFF	OFF				340
		OFF	OFF	OFF			OFF	344
		OFF	OFF	OFF		OFF		350
		OFF	OFF	OFF		OFF	OFF	354
		OFF	OFF	OFF	OFF			360
		OFF	OFF	OFF	OFF		OFF	364
		OFF	OFF	OFF	OFF	OFF		370
		OFF	OFF	OFF	OFF	OFF	OFF	374
	OFF							400

	OFF		OFF					500

	OFF	OFF						600

	OFF	OFF	OFF					700

TK-10019

2.4.4 Boot Option Selection (PDP-11 Host Systems Only)

The DEUNA provides for remote booting and down-line loading of PDP-11 family host systems. These functions are switch selectable via two boot option select switches located on switch pack E62 on the M7792 port module.

NOTE
Refer to Appendix B for additional information on DEUNA remote booting and down-line loading.

When installing a DEUNA in a PDP-11 family host system, configure switches S8 and S9 on switch pack E62 (M7792 module) for the boot function desired. Table 2-6 lists the switch settings and corresponding boot option functions. Refer to Figure 2-1 for the location of E62 on the M7792 module.

When installing a DEUNA in a VAX-11 family host system, set both S8 and S9 on E62 (M7792 module) to the ON (disabled) position of the switch.

NOTE
An OFF (open) switch produces a logical one (1). This is the ENABLED state of the switch function.

Table 2-6 Boot Option Selection (M7792 E62 - S8 & S9)

BOOT SEL 1	BOOT SEL 0	Function
ON*	ON*	Remote boot disabled
OFF	ON	Remote boot with system load
ON	OFF	Remote boot with ROM
OFF	OFF	Remote boot with power up boot and system load

* Switch settings for a DEUNA installed in a VAX-11 system.

2.4.5 Self-Test Loop (For Manufacturing Use)

NOTE
For M7792 Revision B modules, refer to Appendix D of this guide.

The self-test loop is provided on the DEUNA for manufacturing testing. This is a switch-selectable feature that allows the on-board self-test diagnostic program, once it is initiated, to continuously loop on itself. This feature is controlled by S10 on switch pack E62 on the M7792 port module and should be disabled during installation.

When installing a DEUNA, disable the self-test loop feature by setting S10 on switch pack E62 (M7792 module) to the ON (closed) position, as indicated in Table 2-7. Refer to Figure 2-1 for the location of E62 on the M7792 module.

Table 2-7 Self-Test Loop Switch (M7792 E62 - S10)

Position	Function
ON* (closed)	DISABLED
OFF (open)	ENABLED

* Switch setting for normal operation

2.5 INSTALLATION AND CABLING

Install and cable the DEUNA component parts in the host system using the following procedure.

2.5.1 M7792 Port Module Installation

1. Locate the two BC08R-1 module interconnect cables supplied.

2. Plug one end of one of the cables into J1 on the M7792 module. Plug one end of the second cable into J2. Refer to Figure 2-2 for the physical layout of the M7792 port module and Figure 2-3 for cable connection details.

NOTE

BC08S-1 cables may be substituted for BC08R-1 cables. No restrictions exist regarding alignment of the BC08R-1 (or BC08S-1) cables with J1 and J2. Neither the cables nor the jacks are keyed.

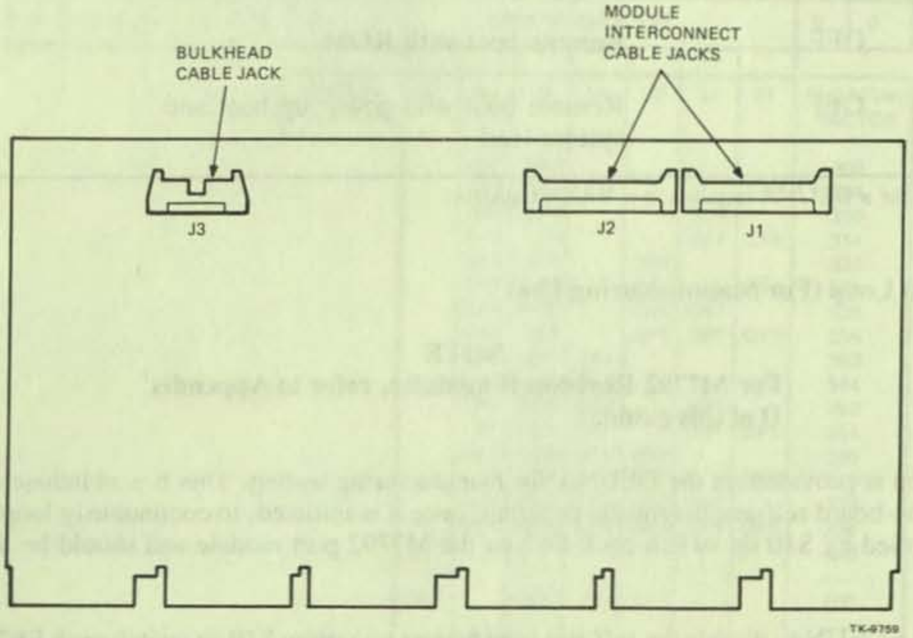


Figure 2-2 M7793 Link Module Physical Layout

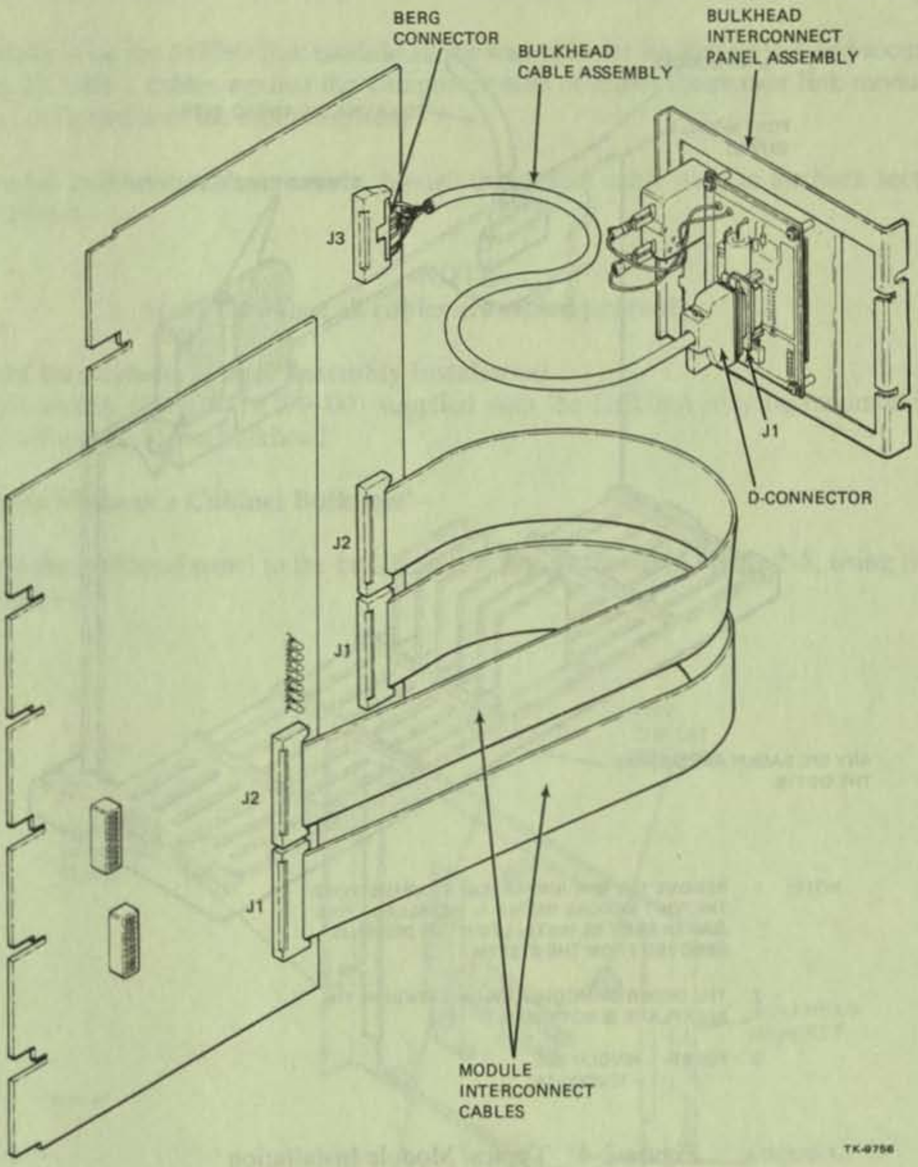
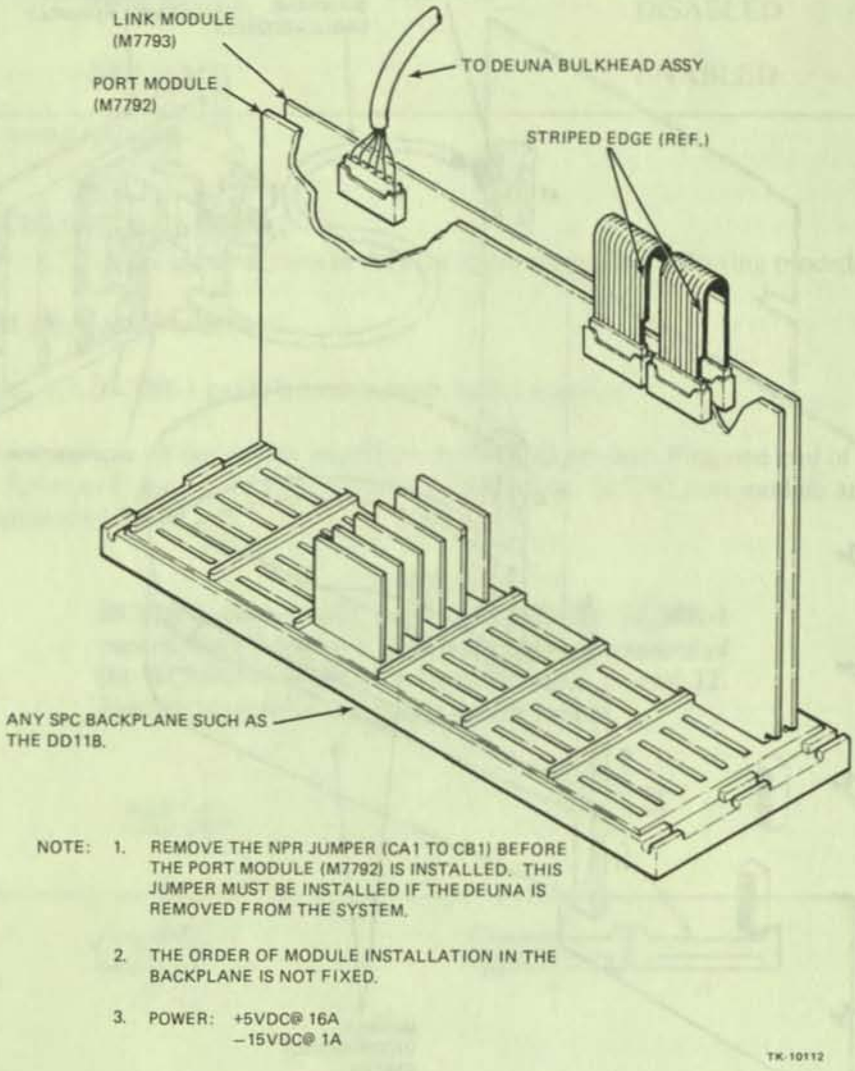


Figure 2-3 DEUNA Cable Connection Details

3. Carefully insert and secure the M7792 port module into the SPC backplane slot previously selected (Figure 2-4).



NOTE: 1. REMOVE THE NPR JUMPER (CA1 TO CB1) BEFORE THE PORT MODULE (M7792) IS INSTALLED. THIS JUMPER MUST BE INSTALLED IF THE DEUNA IS REMOVED FROM THE SYSTEM.

2. THE ORDER OF MODULE INSTALLATION IN THE BACKPLANE IS NOT FIXED.

3. POWER: +5VDC@ 16A
−15VDC@ 1A

Figure 2-4 Typical Module Installation

2.5.2 M7793 Link Module Installation

1. Slide the M7793 link module into the module guides of a slot adjacent to the M7792 port module. DO NOT insert or secure the module all the way into the slot at this time (Figure 2-4).

2. Connect the two BC08R-1 bus cables from J1 and J2 on the port module to J1 and J2 on the link module. There should be NO TWISTS in these cables. Refer to Figure 2-2 for the physical layout of the M7793 module and Figure 2-3 for cable connection details.

3. Locate the bulkhead cable assembly supplied (P/N 70-18798-00) and carefully plug the BERG™ connector end into J3 on the link module. Do NOT force the connector into the jack. Both the connector and jack are keyed and may be connected together only when aligned correctly.

4. Carefully slide the M7793 link module all the way in to the backplane slot and secure it. Fold each of the BC08R-1 cables against the component side of either the port or link module to allow the cables to fit inside of the mounting box.

5. Route the bulkhead cable assembly through the cabinet cable ways to the back section of the system cabinet.

NOTE
Make sure that all cables are seated properly.

2.5.3 Bulkhead Interconnect Panel Assembly Installation

The Bulkhead Assembly (P/N 70–18799–00) supplied with the DEUNA may be mounted in host system cabinets with or without a cabinet bulkhead.

2.5.3.1 Cabinets Without a Cabinet Bulkhead –

1. Secure the bulkhead panel to the bulkhead bracket, as shown in Figure 2-5, using the four (4) captive screws.

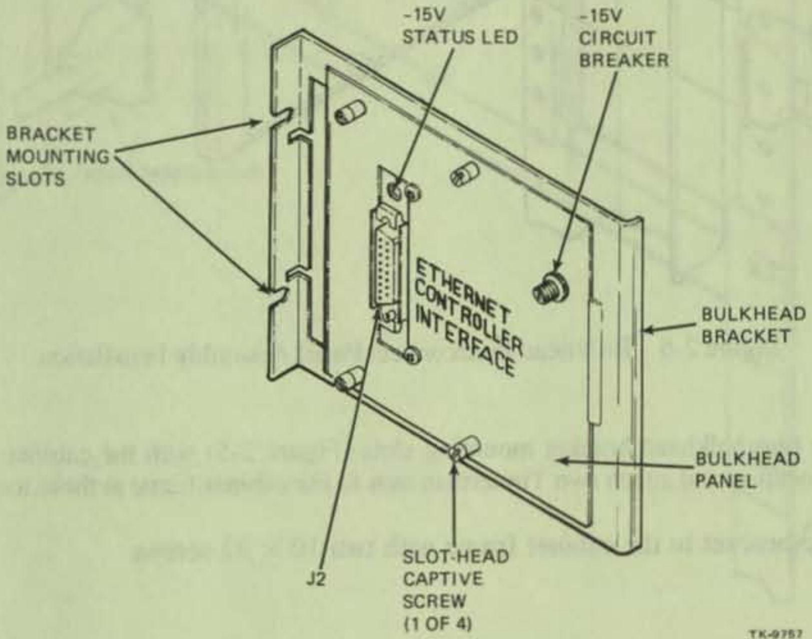


Figure 2-5 Bulkhead Interconnect Panel Assembly

BERG™ is a trademark of Berg Electronics.

2. Select a mounting location at the back of the system cabinet with no obstructions. The entire bulkhead assembly should be mounted on the cabinet frame (Figure 2-6).

CAUTION

The back of the bulkhead panel contains a circuit board which carries − 15 V. Be sure this circuitry will not be touching anything that could cause a short circuit on power-up.

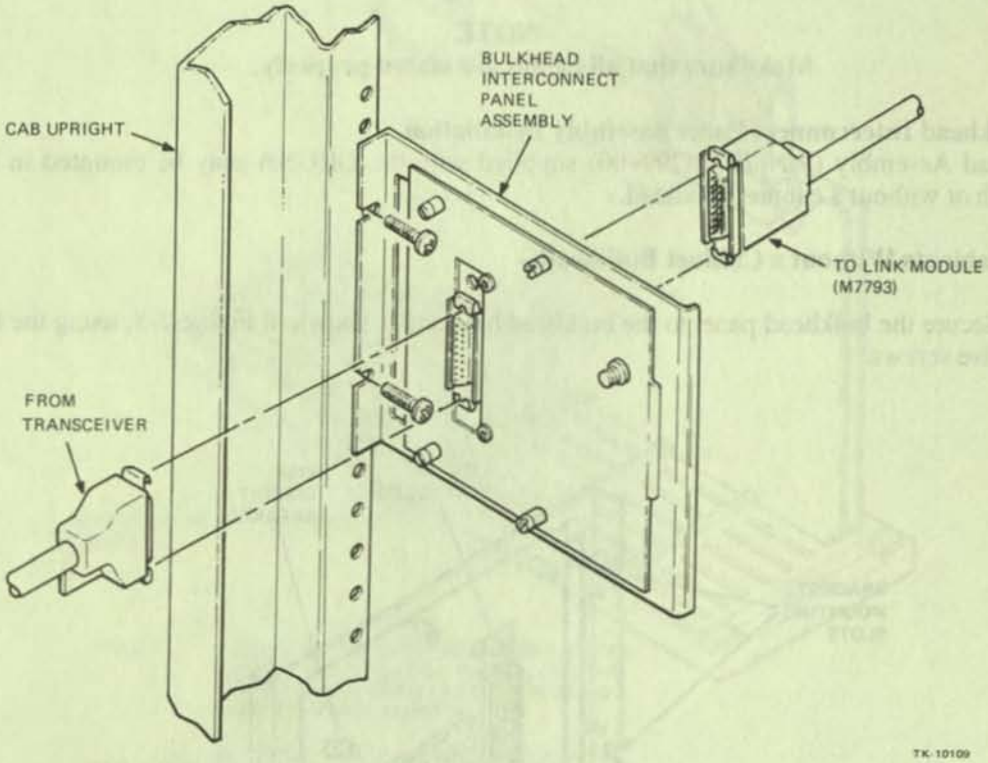


Figure 2-6 Bulkhead Interconnect Panel Assembly Installation

3. Align the two bulkhead bracket mounting slots (Figure 2-5) with the cabinet frame holes at the selected location and attach two Tinnerman nuts to the cabinet frame at these locations.

4. Secure the bracket to the cabinet frame with two 10 × 32 screws.

2.5.3.2 Cabinets With a Cabinet Bulkhead –

1. Mount the bulkhead panel to an available I/O cutout on the cabinet bulkhead (Figure 2-7).

2. Secure the bulkhead panel to the cabinet bulkhead with the four captive screws.

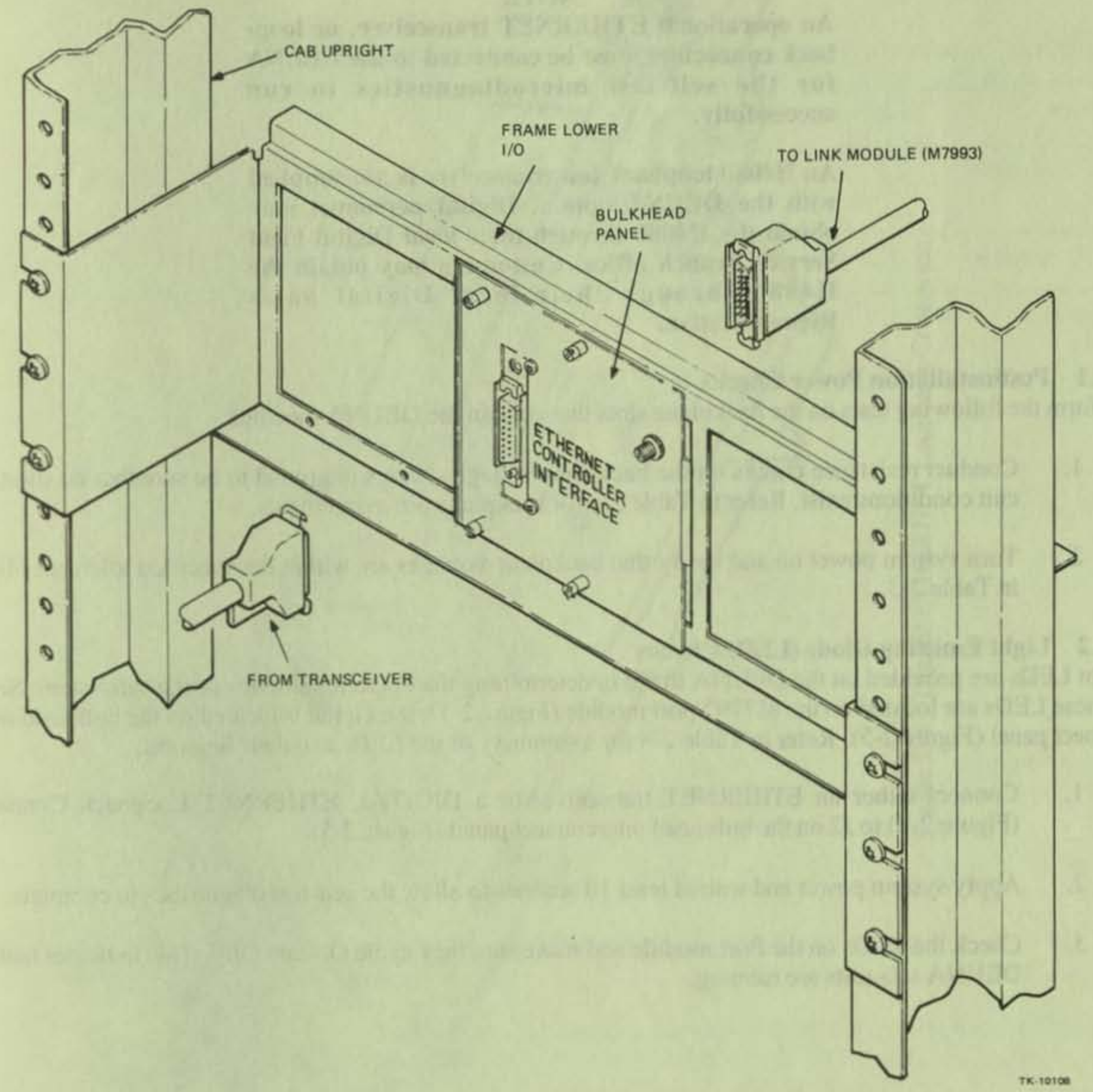


Figure 2-7 Typical System Cabinet Bulkhead Installation

2.5.3.3 Connect the D-Connector – Connect the D-connector on the bulkhead cable assembly to J1 on the back (component side) of the bulkhead panel circuit board as shown in Figure 2-3. Secure the connector and J1 together by sliding the latch assembly located on J1 to the lock position.

2.6 TESTING

Perform the following system tests to verify that the DEUNA and the host system are operating correctly.

> **NOTE**
>
> **An operational ETHERNET transceiver, or loopback connector, must be connected to the DEUNA for the self-test microdiagnostics to run successfully.**
>
> **An H4080 loopback test transceiver is not supplied with the DEUNA option. Digital personnel may obtain the H4080 through their local Digital Field Service Branch office. Customers may obtain the H4080 through their local Digital Sales Representative.**

2.6.1 Postinstallation Power Checks

Perform the following tests on the backplane slots that contain the DEUNA modules.

1. Conduct resistance checks on the backplane voltage sources to ground to be sure that no short circuit conditions exist. Refer to Table 2-3 for backplane pin assignments.

2. Turn system power on and verify that backplane voltages are within the specified tolerances listed in Table 2-3.

2.6.2 Light Emitting Diode (LED) Checks

Eight LEDs are provided on the DEUNA to aid in determining the operational status of the subsystem. Seven of these LEDs are located on the M7792 port module (Figure 2-1); the eighth is located on the bulkhead interconnect panel (Figure 2-5). Refer to Table 2-8 for a summary of the LEDs and their function.

1. Connect either an ETHERNET transceiver or a DIGITAL ETHERNET Loopback Connector (Figure 2-8) to J2 on the bulkhead interconnect panel (Figure 2-5).

2. Apply system power and wait at least 10 seconds to allow the self-test diagnostics to complete.

3. Check the LEDs on the Port module and make sure they cycle ON and OFF. This indicates that the DEUNA sub-tests are running.

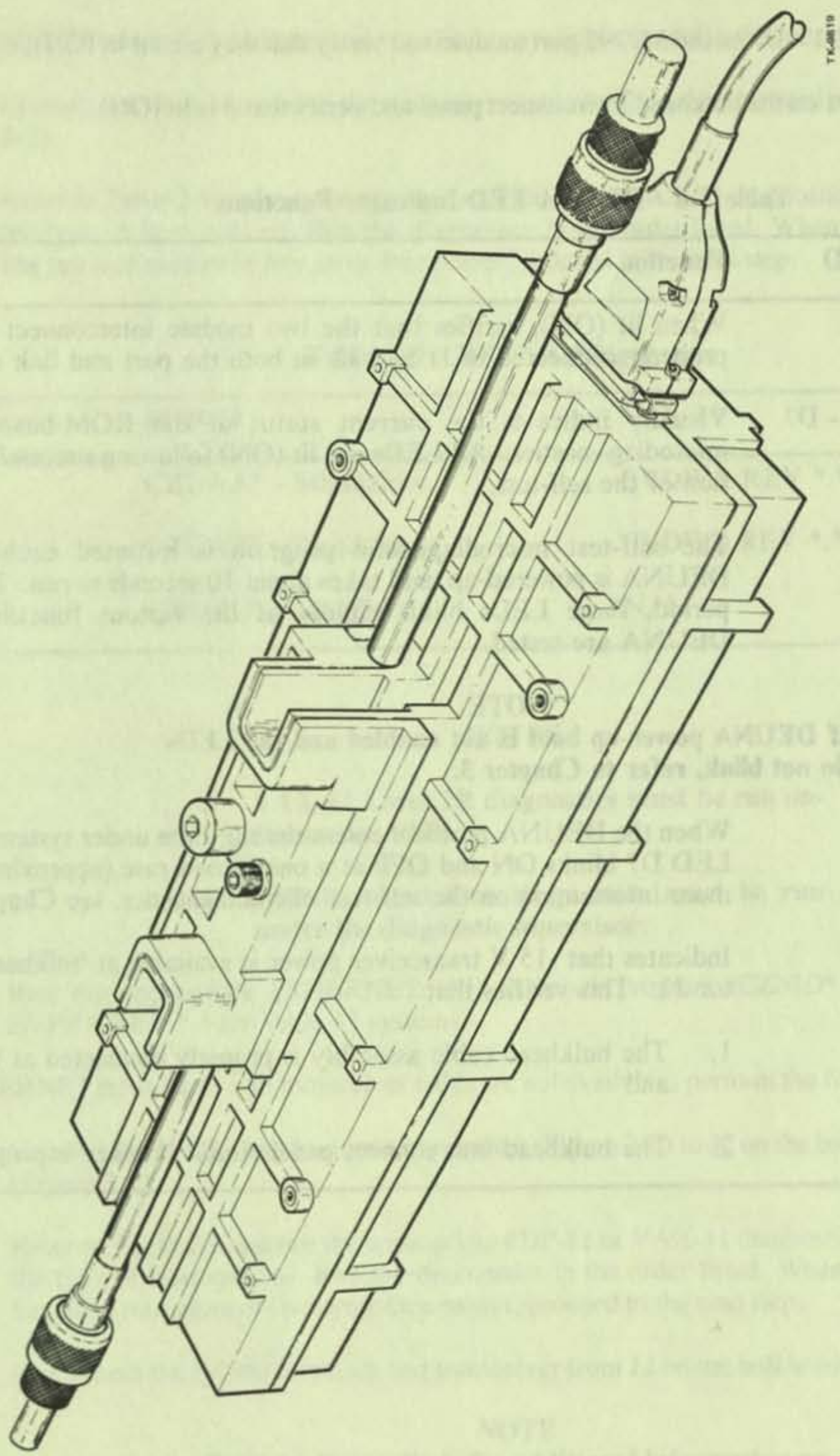


Figure 2-8 Digital ETHERNET Loopback Connector

NOTE

If power-up boot is enabled, self-test will not run and all LEDs will be ON.

4. Check LEDs D1 – D7 on the M7792 port module and verify that they are all lit (ON).
5. Check LED D1 on the bulkhead interconnect panel and verify that it is lit (ON).

Table 2-8 DEUNA LED Indicator Functions

Location	LED	Function
M7792 Module	D1	When lit (ON), verifies that the two module interconnect cables are properly connected to J1 and J2 on both the port and link modules.
M7792 Module	D2 - D7	Visually indicates the current status of the ROM-based self-test microdiag- nostics. All LEDs are lit (ON) following successful completion of the self-test. The self-test microdiagnostic program is initiated each time the DEUNA is powered-up, and takes about 10 seconds to run. During this period, these LEDs blink rapidly as the various functions of the DEUNA are tested. **NOTE** **If DEUNA power-up boot is not enabled and the LEDs do not blink, refer to Chapter 3.** When the DEUNA protocol enters the run state under system software, LED D7 blinks ON and OFF at a one second rate (approximate). For more information on the self-test microdiagnostics, see Chapter 3.
Bulkhead Panel	D1	Indicates that -15 V transceiver power is available at bulkhead connector J2. This verifies that 1. The bulkhead cable assembly is properly connected at both ends, and 2. The bulkhead interconnect panel circuit breaker is properly set.

2.6.3 Diagnostic Acceptance Procedure

The final step in the DEUNA installation procedure is to exercise the M7792 Port module and the M7793 Link module as one complete unit on the UNIBUS bus and on the ETHERNET cable (if possible).

If an ETHERNET transceiver and transceiver cable are available, perform the following steps:

1. Connect and lock one end of the transceiver cable to J2 on the bulkhead interconnect panel (Figure 2-5).

2. Refer to Table 2-9 and run the appropriate PDP-11 or VAX-11 diagnostic programs (depending on the type of host system). Run the diagnostics in the order listed. When each diagnostic program has run a minimum of five error-free passes, proceed to the next step.

Table 2-9 DEUNA Diagnostics

Diagnostic	PDP-11	VAX-11
Repair	CZUAA* - Standalone	EVDWA REV *.* - Level 3 - Standalone
Functional	CZUAB* - Standalone	EVDWB REV *.* - Level 2R - On-Line
DEC/X-11	CXUAC* - Standalone	N/A

NOTES

1. **VAX-11 Level 2R diagnostics must be run on-line under VMS.**

2. **PDP-11 standalone diagnostics must be run under the diagnostic supervisor.**

3. Run the appropriate ETHERNET (NI) exerciser program (CZNID* for PDP-11 systems or EVPBA REV *.* for VAX-11 systems).

If an ETHERNET transceiver and transceiver cable are not available, perform the following steps:

1. Connect the H4080 loopback test transceiver (Figure 2-8) to J2 on the bulkhead interconnect panel (Figure 2-5).

2. Refer to Table 2-9 and run the appropriate PDP-11 or VAX-11 diagnostic programs (depending on the type of host system). Run the diagnostics in the order listed. When each diagnostic program has run a minimum of five error-free passes, proceed to the next step.

3. Disconnect the H4080 loopback test transceiver from J2 on the bulkhead interconnect panel.

NOTE

Refer to Appendix C for additional information on the NI exerciser programs.

[illegible] Diagnostic Acceptance Procedure

The first step in the [illegible] is to [illegible] the [illegible] module and the [illegible] module [illegible] on the [illegible] (if possible).

If an ETHERNET transceiver and transceiver cable is not available, perform the following steps:

1. Connect the loopback test connector [illegible] on the bulkhead interconnect panel (Figure 2-5).

2. Refer to Table 2-9 and run the appropriate [illegible] diagnostic program (depending on the type of host system). Run the diagnostics in the order listed. When each diagnostic program has run a minimum of two error-free passes, proceed to the next step.

CZUAA? - Standalone	EVDWA? REV *.* - Level 3 - Standalone
CZUAB? - Standalone	EVDWB REV *.* - Level 2R - On-Line
CXUAC? - Standalone	[illegible]

NOTE

VAX-11 Level 2R diagnostics must be run on [illegible] under the diagnostic supervisor.

Run the appropriate ETHERNET [illegible] EVPBA REV *.* for VAX-11 systems.

If an ETHERNET transceiver and transceiver cable is not available, perform the following steps:

1. Connect the H4080 loopback test connector [illegible] on the bulkhead interconnect panel (Figure 2-5).

2. Refer to Table 2-9 and run the appropriate [illegible] diagnostic program (depending on the type of host system). Run the diagnostics in the order listed. When each diagnostic program has run a minimum of two error-free passes, proceed to the next step.

3. Disconnect the H4080 loopback test connector from [illegible] on the bulkhead interconnect panel.

NOTE

Refer to Appendix D for additional information on the NI exerciser programs.

CHAPTER 3
SERVICE

3.1 SCOPE

This chapter provides information for servicing the DEUNA. It is divided into the following sections:

- Maintenance Philosophy – Defines the DEUNA Field Replaceable Unit (FRU).
- Diagnostic Description – Describes all VAX-11 and PDP-11 diagnostics for the DEUNA.
- Corrective Maintenance – Describes both VAX-11 and PDP-11 corrective maintenance procedures for the DEUNA using troubleshooting flow charts.

A description of the DEUNA Network Interconnect (NI) Exerciser can be found in Appendix C.

3.2 MAINTENANCE PHILOSOPHY

The Maintenance Philosophy for the DEUNA is isolation of the Field Replaceable Unit (FRU). The FRUs for the DEUNA are faulty modules, cables, or the bulkhead assembly.

It is possible for some apparent failures in the DEUNA to be caused by faults in the ETHERNET physical channel; that is, transceiver cable, transceiver, or ETHERNET cable. Faults that can be isolated to the ETHERNET physical channel should be referred to Network support.

3.3 DIAGNOSTIC DESCRIPTION

This section describes the diagnostics available for the DEUNA on both VAX-11 and PDP-11 systems. Section 3.4 describes the proper order for running these diagnostics. The individual diagnostic abstracts provide specific instructions for running each diagnostic.

3.3.1 Self-Test

The DEUNA Self-Test verifies the DEUNA microprocessor, internal memory, the UNIBUS interface, and the link module through various loopback levels. The path from the DEUNA to the transceiver and ETHERNET coaxial cable is also verified during Self-Test.

The ROM-based Self-Test is initiated in two ways: each time the DEUNA is powered up and when a Self-Test Port Command is issued. The Self-Test Port Command is issued by writing a 3 to PCSR0. Refer to Section 4.3 of this document for a description of PCSR0 and the Self-Test Port Command.

The results of the Self-Test are available on LEDs (D2 through D7) on the Port module (M7792). The execution time of the Self-Test is seven to ten seconds. During execution, the Self-Test LEDs should turn ON and OFF indicating the various tests being performed. If the Self-Test LEDs remain ON and do not cycle ON and OFF, this is considered a DEUNA failure, probably the M7792 module. Refer to Figure 3-1 and Table 3-1 for a description of the Port module LEDs.

In addition to the Self-Test LEDs, one LED on the Port module D1, verifies the cable connection between the Port and Link modules.

NOTE
When the DEUNA is in the Running State, LEDs D1 through D6 are constantly on; D7 blinks on and off at a rate of about once per second.

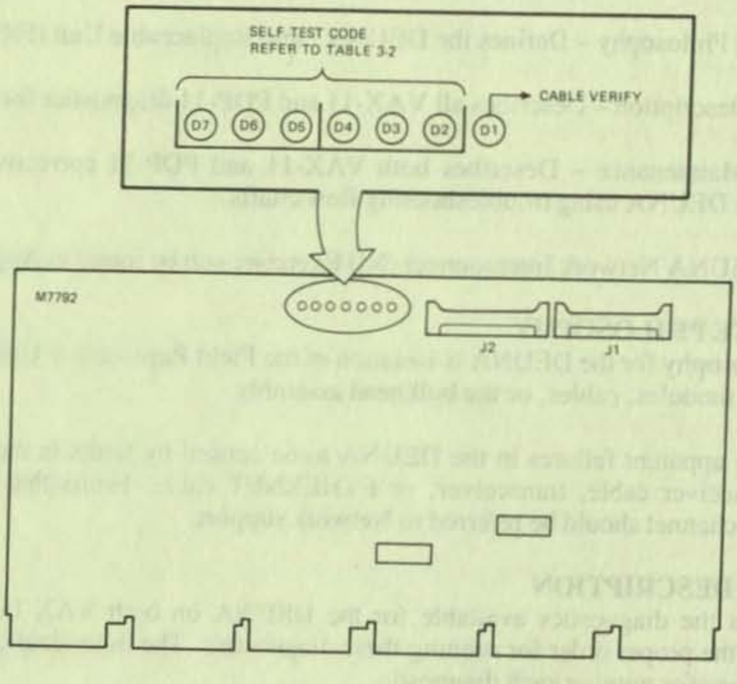


Figure 3-1 DEUNA Port Module Self-Test LEDs

Table 3-1 DEUNA Self-Test LED Codes

Code	D7	D6	D5	D4	D3	D2	Test Name	(Module)
77	ON	ON	ON	ON	ON	ON	Never Got Started	M7792/M7793
1						ON	CPU Instruction	M7792
2					ON		ROM	M7792
3					ON	ON	Writeable Control Store	M7792
4				ON			T11 UNIBUS Address Register	M7792
5				ON		ON	Receiver UNIBUS DMA	M7792
6				ON	ON		PCSR1 Lower Byte & T11 DMA Read	M7792/UNIBUS
7				ON	ON	ON	PCSR0 Upper Byte & T11 DMA Write	M7792
10			ON				PCSR0 Lower Byte & Link Mem. DMA	M7792
11			ON			ON	PCSR2 & PCSR3	M7792
12			ON		ON		Timer	M7792
13			ON		ON	ON	Physical Address ROM	M7792
20		ON					Link Memory	M7792/M7793
							Local Loopback	
26		ON		ON	ON		Bugcheck (NI & UNIBUS in HALTED STATE) – Internal	M7792/M7793
							Transmit Buffer Resource Allocation Error on Boot	
30		ON	ON				Transmitter Timeout	M7792/M7793
31		ON	ON			ON	Receiver Timeout	M7792/M7793
32		ON	ON		ON		Buffer Comparison	M7792/M7793
33		ON	ON		ON	ON	Byte Count	M7792/M7793
34		ON	ON	ON			Receiver Status	M7792/M7793
35		ON	ON	ON		ON	CRC Error	M7792/M7793
36		ON	ON	ON	ON		Match Bit Error	M7792/M7793
37		ON	ON	ON	ON	ON	TDR Error	M7792/M7793
							Transmitter Buffer Address	
40	ON						Transmitter Timeout	M7793
41	ON					ON	Receiver Timeout	M7793
42	ON				ON		Buffer Comparison	M7793
43	ON				ON	ON	Byte Count	M7793
44	ON			ON			Receiver Status	M7793
45	ON			ON		ON	CRC Error	M7793
							Receiver Buffer Address	
50	ON		ON				Transmitter Timeout	M7793
51	ON		ON			ON	Receiver Timeout	M7793
52	ON		ON		ON		Buffer Comparison	M7793
53	ON		ON		ON	ON	Byte Count	M7793
54	ON		ON	ON			Receiver Status	M7793
55	ON		ON	ON		ON	CRC Error	M7793
60	ON	ON					Runt Packet	M7793
61	ON	ON				ON	Minimum Packet Size	M7793
62	ON	ON			ON		Maximum Packet Size	M7793
63	ON	ON			ON	ON	Oversize Packet	M7793
64	ON	ON		ON			CRC	M7793
65	ON	ON		ON		ON	Collision	M7793
66	ON	ON		ON	ON		Heartbeat	M7793
67	ON	ON		ON	ON	ON	Half Duplex	M7793
70	ON	ON	ON				Multicast	M7793
71	ON	ON	ON			ON	Address Recognition	M7793

Table 3-1 DEUNA Self-Test LED Codes (Cont)

Code	D7	D6	D5	D4	D3	D2	Test Name	(Module)
72	ON	ON	ON		ON		External Loopback	M7793/H4000
73	ON	ON	ON		ON	ON	Internal Transmit Buffer Resource Allocation	M7792/M7793
74	ON	ON	ON	ON			Link Memory Parity Error	M7792/M7793
75	ON	ON	ON	ON		ON	Internal Unexpected Interrupt	M7792/M7793
76	ON	ON	ON	ON	ON		Internal Register Error	M7792/M7793
77	ON	ON	ON	ON	ON	ON	Self Test Done, No Errors (State = 2, DNI set)	

NOTE
ON represents a logical ONE (1); OFF represents a logical ZERO (0). For this table, all LEDs are assumed to be OFF unless noted otherwise.

3.3.2 DEUNA VAX-11 Functional Diagnostic (EVDWB REV *.*)

EVDWB allows the user to verify the DEUNA functional operation. It tests all DEUNA hardware functions the VMS driver is capable of using. It is a VAX/VMS Level 2R (on-line only) running under the VAX-11 Diagnostic Supervisor (VDS).

EVDWB is compatible with VAX/VMS Version 3.0 or later and the VAX-11 Diagnostic Supervisor Version 6.5 or later.

A summary of the tests performed by the DEUNA VAX-11 Functional Diagnostic (EVDWB) is contained in Table 3-2.

Table 3-2 DEUNA VAX-11 Functional Diagnostic Summary (EVDWB REV *.*)

Test #	Name	Verifies
1	Read Internal ROM	The internal 16K byte ROM can be read; there are no CRC errors.
2	Read/Write Internal WCS	Data patterns can be written and read from WCS memory.
3	Internal Link ADDRESS	All Link Memory locations can be accessed.
4	Read/Write Internal Link Memory	Data patterns can be written and read from all Link Memory locations.
5	Transmit CRC	The Transmit CRC logic functions properly.

Table 3-2 DEUNA VAX-11 Functional Diagnostic Summary (EVDWB) (Cont)

Test #	Name	Verifies
6	Receive CRC	The Receive CRC logic functions properly.
7	Promiscuous Address	The DEUNA in the Promiscuous Mode will accept all datagrams regardless of destination address.
8	Enable All Multicast	The DEUNA in the Enable All Multicast Mode will accept all datagrams with multicast destination addresses.
9	Station Address	The Link Module recognizes the physical, multicast, and broadcast addresses of the node and discards datagrams with non-enabled addresses.
10	Pad Runt Packets	The DEUNA can pad, transmit, receive, and store in host memory loopback datagrams that are less than 64 bytes long.
11	No Receive Buffer	The appropriate error will be flagged if a loopback is attempted and there are no receive buffers owned by the DEUNA.
12	DEUNA Stress	The DEUNA can function properly under heavy traffic loading conditions.

3.3.3 DEUNA PDP-11 Functional Diagnostic (CZUAB*)

CZUAB verifies the functional operation of up to eight DEUNAs on a PDP-11 processor. It runs under the Diagnostic Runtime Services (DRS PDP-11 Diagnostic Supervisor) and only in standalone, off-line environment. The DRS provides APT compatibility for this diagnostic.

A summary of the tests performed by the CZUAB Diagnostic is contained in Table 3-3.

Table 3-3 DEUNA PDP-11 Functional Diagnostic Summary (CZUAB*)

Test #	Name	Verifies
1-4	PCSR Read Access	A device is present at the PCSR addresses specified for the DEUNA under test.
5	PCSR2 Static Bit	All bits in the PCSR2 register can be set and cleared as specified.
6	PCSR3 Static Bit	All bits in the PCSR3 register can be set and cleared as specified.

Table 3-3 DEUNA PDP-11 Functional Diagnostic Summary (CZUAB*) (Cont)

Test #	Name	Verifies
7	Self-Test	The ROM-based Self-Test can be run successfully when invoked via SELF TEST Port Command.
8	Port Command	No errors occur when a Port Command is issued.
9	Interrupt Logic	A DEUNA interrupt can be generated.
10	Read Internal ROM	Reads and verifies internal ROM.
11	Read/Write Internal WCS	The internal WCS memory can be written and read.
12	Read/Write Mode Function	The Read/Write Mode Port Function is operational.
13	Read/Write Link Memory	Exercises the internal link memory by reading and writing patterns throughout the memory.
14	Internal Loopback	No errors occur when a datagram is transmitted and received in the Internal Loopback Mode.
15	CRC Checking	The CRC Checking Logic is operational.
16	Force CRC Error	CRC Error Detection is operational.
17	Disable Receive Chaining	The Disable Data Chaining Mode is operational.
18	Transmit Chaining Error	The Buffer Length Error (BUFL) bit can be set in the Transmit Descriptor Ring.
19	No Receive Buffer	A Receive Buffer Error (RBUF) can be generated.
20	Data Chaining	Transmit and receive data chaining in either internal or external loopback mode.
21	Physical Address	The Physical Address detection is operational by attempting loopbacks with currently enabled and disabled Physical Address.
22	Multicast Address	The Multicast Address detection is operational by attempting loopbacks with currently enabled and disabled Multicast Addresses.
23	Promiscuous Mode	The DEUNA in the Promiscuous Mode will accept all packets regardless of the destination address.

Table 3-3 DEUNA PDP-11 Functional Diagnostic Summary (CZUAB*) (Cont)

Test #	Name	Verifies
24	Enable All Multicast	The DEUNA in the Enable All Multicast Mode will accept all packets with Multicast destination addresses.
25	Pad Runt Packets	The DEUNA can pad, transmit, receive, and store in host memory loopback datagrams that are less than 64 bytes long.
26	Half Duplex	The Half-Duplex Mode is operational.
27	Simultaneous Operations	The DEUNA can perform several operations at the same time.
28	Print Device Parameters	Prints the Default Physical Address, the microcode revision, and the switch pack settings.

3.3.4 DEUNA VAX-11 Repair Level Diagnostic (EVDWA REV *.*)

EVDWA is a VAX-11 LEVEL 3 diagnostic that runs in the off-line, stand-alone mode only. It runs under the VAX-11 Diagnostic Supervisor. It detects and isolates errors to the functional unit or the FRU. It tests all DEUNA hardware functions that can be tested using diagnostic DCT-11 microprocessor microcode. They use both the internal and external loopback mode.

Table 3-4 summarizes the DEUNA VAX-11 Repair Level Diagnostic.

Table 3-4 DEUNA VAX-11 Repair Level Diagnostic Summary (EVDWA REV *.*)

Test #	Name	Verifies
1-4	PCSR Read Access	PCSRs 0 through 3 can be read by the host; predetermined bits appear in the expected bit positions.
5	Reset	The Reset State for all UNIBUS registers.
6	RCSR2 Read/Write	PCSR2 can be read as well as written by the host.
7	PCSR3 Read/Write	PCSR3 can be read as well as written by the host.
8	NOP	The DEUNA processor (T11) can respond to Port Commands.
9	Self Test	The DEUNA can execute the ROM-based Self-Test and report results.

Table 3-4 DEUNA VAX-11 Repair Level Diagnostic Summary (EVDWA REV *.*) (Cont)

Test #	Name	Verifies
10	DEUNA ROM Dump	The data path from the DEUNA processor to the UNIBUS interface is able to transfer data reliably.
11	Data Dump/Load	The data path to the WCS using the DUMP/LOAD INTERNAL MEMORY Port Command.
12	Load and Start Function	The Load and Start Microaddress Port Function is operational.
13	Comprehensive WCS	The WCS memory is error free by performing functional and dynamic tests.
14	Interrupt	The DEUNA will generate an interrupt when enabled, can generate an interrupt vector, and can arbitrate for control of the UNIBUS.
15	Microcode Partition 3	
	Interrupt Bit	Each of the interrupt bits in PCSR0 can cause an interrupt.
	Timer	The internal timer is operating within normal limits.
	Comprehensive Link Memory	The Link Memory is error free by performing functional and dynamic tests.
	DMA "TO" Address Register	The DMA "TO" Address Register is checked by writing and reading it.
	DMA "FROM" Address Register	The DMA "FROM" Address Register is checked by writing and reading it.
	DMA Block Transfer	The DMA Engine can transfer a data block of maximum size to host memory.
	DMA Ripple	The counting function of the DMA "TO" Address Register is checked.

Table 3-4 DEUNA VAX-11 Repair Level Diagnostic Summary (EVDWA REV *.*) (Cont)

Test #	Name	Verifies
16	Microcode Partition 4	
	XMIT Done	The XMIT State Machine will generate a ''Transmit Done'' interrupt after completing a diagram transmission.
	Receiver Done	The Receive State Machine is operational and an interrupt occurs when a datagram is received.
	Data Byte Framing	Data is being framed on byte boundaries.
	Data Word Framing	Data is being framed on word boundaries.
	Data Path Pattern	The Link Module data path has no stuck-at-one/stuck-at-zero (SA0/SA1) errors.
	Status Mux Verification	The Status Mux is operational.
	Link (Even) Byte Counter	The byte counters are functioning properly for datagrams with even number of bytes.
	Link (Odd) Byte Counter	The byte counters are functioning properly for datagrams with odd number of bytes.
	Link Byte Counter Maximum	The byte counter will not wrap around if the maximum count is exceeded.
	Link FIFO Addressing	The address paths through the link address FIFOs are functioning properly.
	Link Memory Arbitration	The Link Memory Arbitration logic is operational.
17	Microcode Partition 5	
	Station Address Pattern	The Link RAM has no SA0/SA1 errors.

Table 3-4 DEUNA VAX-11 Repair Level Diagnostic Summary (EVDWA REV *.*) (Cont)

Test #	Name	Verifies
	Station Address Rejection	The Link will not recognize a datagram with a destination address that is not contained in the Station Address RAM.
	Station Address RAM Position	The Link will recognize the physical address regardless of where it is located in Station Address RAM.
	Multicast Address	Verifies that the Multicast Address detection is operational by attempting loopbacks with currently enabled and disabled Multicast Addresses.
18	Microcode Partition 6	
	CRC Data Pattern	The CRC circuitry generates the correct CRC residual under various datagram conditions.
	CRC Error	The CRC circuitry can detect an incorrect CRC in a received datagram.
	CRC Pattern Length	The receive CRC circuitry can detect incorrect CRCs in datagrams of different lengths.
	Runt	The Receive State Machine can detect and discard datagrams of less than 64 bytes.
	Half-Duplex	The DEUNA functions as specified in the Half-Duplex Mode.
19	Microcode Partition 7	
	Collision	The Receive State Machine responds to a collision.
	TDR Counter	The TDR counter is capable of counting.
	Retry Logic	The Retry logic is functioning properly.
	Print Device Parameters	Prints the Default Physical Address, the microcode revision, and switchpack settings.

3.3.5 DEUNA PDP-11 Repair Level Diagnostic (CZUAA*)

CZUAA runs in the off-line, standalone mode under Diagnostic Run-time Services (DRS). It detects and isolates errors to the functional unit or the FRU. It tests all DEUNA hardware functions that can be tested using diagnostic DCT-11 microcode. It uses both the internal and external loopback mode.

Refer to Table 3-5 for a summary of the DEUNA PDP-11 Repair Level Diagnostic (CZUAA*).

Table 3-5 DEUNA PDP-11 Repair Level Diagnostic Summary (CZUAA*)

Test #	Name	Verifies
1-4	PCSR Read Access	PCSRs 0 through 3 can be read by the host; predetermined bits appear in the expected bit positions.
5	Reset	The Reset State for all UNIBUS registers.
6	PCSR2 Read/Write	PCSR2 can be read as well as written by the host.
7	PCSR3 Read/Write	PCSR3 can be read as well as written by the host.
8	NOP	The DEUNA processor (T11) can respond to Port Commands.
9	Self-Test	The DEUNA can execute the ROM-based Self-Test and report results.
10	DEUNA ROM Dump	The data path from the DEUNA processor to the UNIBUS interface is able to transfer data reliably.
11	WCS Load/Dump	The data path to the WCS using the DUMP/LOAD INTERNAL MEMORY Port Command.
12	Load and Start Function	The Load and Start Microaddress Port Function is operational.
13	Comprehensive WCS	The WCS memory is error free by performing functional and dynamic tests.
14	Interrupt	The DEUNA will generate an interrupt when enabled, can generate an interrupt vector, and can arbitrate for control of the UNIBUS.
15	PCSR0 Interrupt Bit	Each of the interrupt bits in PCSR0 can cause an interrupt.
16	Timer	The internal timer is operating within limits.
17	Link Memory	The Link Memory is error free by performing functional and dynamic tests.

Table 3-5 DEUNA PDP-11 Repair Level Diagnostic Summary (CZUAA*) (Cont)

Test #	Name	Verifies
18	DMA "TO" Address Register	The DMA "TO" Address Register is checked by writing and reading it.
19	DMA "FROM" Address Register	The DMA "FROM" Address Register is checked by writing and reading it.
20	DMA Block Transfer	The DMA Engine can transfer a data block of maximum size to host memory.
21	Transmit Done	The Transmit State Machine will generate a "Transmit Done" interrupt after completing a datagram transmission.
22	Receiver Done	The Receive State Machine is operational and an interrupt occurs when a datagram is received.
23	Data Byte Framing	Data is being framed on byte boundaries.
24	Data Word Framing	Data is being framed on word boundaries.
25	Data Path Pattern	The Link Module data path has no struck-at-one/stuck-at-zero (SA0/SA1) errors.
26	Status Mux	The Status Mux is operational.
27	Link (Even) Byte Counter	The byte counters are functioning properly for datagrams with even number of bytes.
28	Link (Odd) Byte Counter	The byte counters are functioning properly for datagrams with odd number of bytes.
29	Link Byte Counter Maximum	The byte counter will not wrap around if the maximum count is exceeded.
30	Link FIFO Addressing	The address paths through the link address FIFOs are functioning properly.
31	Receive Link Memory Address	The Receive Link Memory Address logic can access all Link Memory locations.

Table 3-5 DEUNA PDP-11 Repair Level Diagnostic Summary (CZUAA*) (Cont)

Test #	Name	Verifies
32	Transmit Link Memory Address	The Transmit Link Memory Address logic can access all Link Memory locations.
33	Link Memory Arbitration	The Link Memory Arbitration logic is operational.
34	Station Address Pattern	The Link RAM has no SA0/SA1 errors.
35	Station Rejection	The link will not recognize a datagram with a destination address that is not contained in the Station Address RAM.
36	Physical Address RAM Position	The link will recognize the physical address regardless of where it is located in Station Address RAM.
37	Multicast Address	The Multicast Address detection is operational by attempting loopbacks with currently enabled and disabled Multicast Addresses.
38	CRC Data Pattern	The CRC circuitry generates the correct CRC residual under various datagram conditions.
39	CRC Error	The CRC circuitry can detect an incorrect CRC in a received datagram.
40	CRC Pattern Length	The receive CRC circuitry can detect incorrect CRCs in datagrams of different lengths.
41	Receive Buffer Recover (Runt)	The Receive State Machine can detect and discard datagrams of less than 64 bytes.
42	Half-Duplex	The DEUNA functions as specified in the Half-Duplex Mode.
43	Collision	The Receive State Machine responds to a collision.
44	TDR Counter	The TDR counter is capable of counting.
45	Retry Logic	The Retry logic is functioning properly.
46	Print Device Parameters	Prints the Default Physical Address, the microcode revision, and the switchpack settings.

3.3.6 NI Exerciser (CZUAD*/EVDWC REV *.*)

The NI Exerciser determines the ability of nodes on the NI (ETHERNET) to communicate with each other. It includes analysis of errors obtained while running the Exerciser to provide the operator with meaningful error messages.

Refer to Appendix C for a general description of the NI Exerciser. Refer to the individual diagnostic abstract for specific information on running each diagnostic.

3.3.7 DEC/X11 DEUNA Module (CXUAC*)

The DEC/X11 DEUNA Module obtains maximum bus activity for a sustained period of time by transmitting multiple packets on each pass. At the start of each pass, the program allocates a number of transmit buffers (three to ten depending on a random number). It then calculates varying size buffers for a total byte count of approximately 1000 bytes. A table is generated for each packet including starting address, byte count, and expected CRC. Receive buffers are then allocated to align with the transmit buffers, allowing for header and CRC verification.

The default loopback made for the test is external. However, internal loopback may be selected by setting a software switch when configuring the DEUNA DEC/X11 module.

3.4 CORRECTIVE MAINTENANCE

Corrective maintenance of the DEUNA is accomplished by using the ROM-based Self-Test and the diagnostics to isolate the faulty FRU. The FRUs for the DEUNA are:

- M7792 DEUNA Port Module
- M7793 DEUNA Link Module
- BC08R-1 (2) Internal Cables
- 70–18798–00 DEUNA Bulkhead Cable Assembly
- 70–18799–00 DEUNA Bulkhead Assembly

Figure 3-2 describes the DEUNA troubleshooting procedure for both the VAX-11 and the PDP-11 systems.

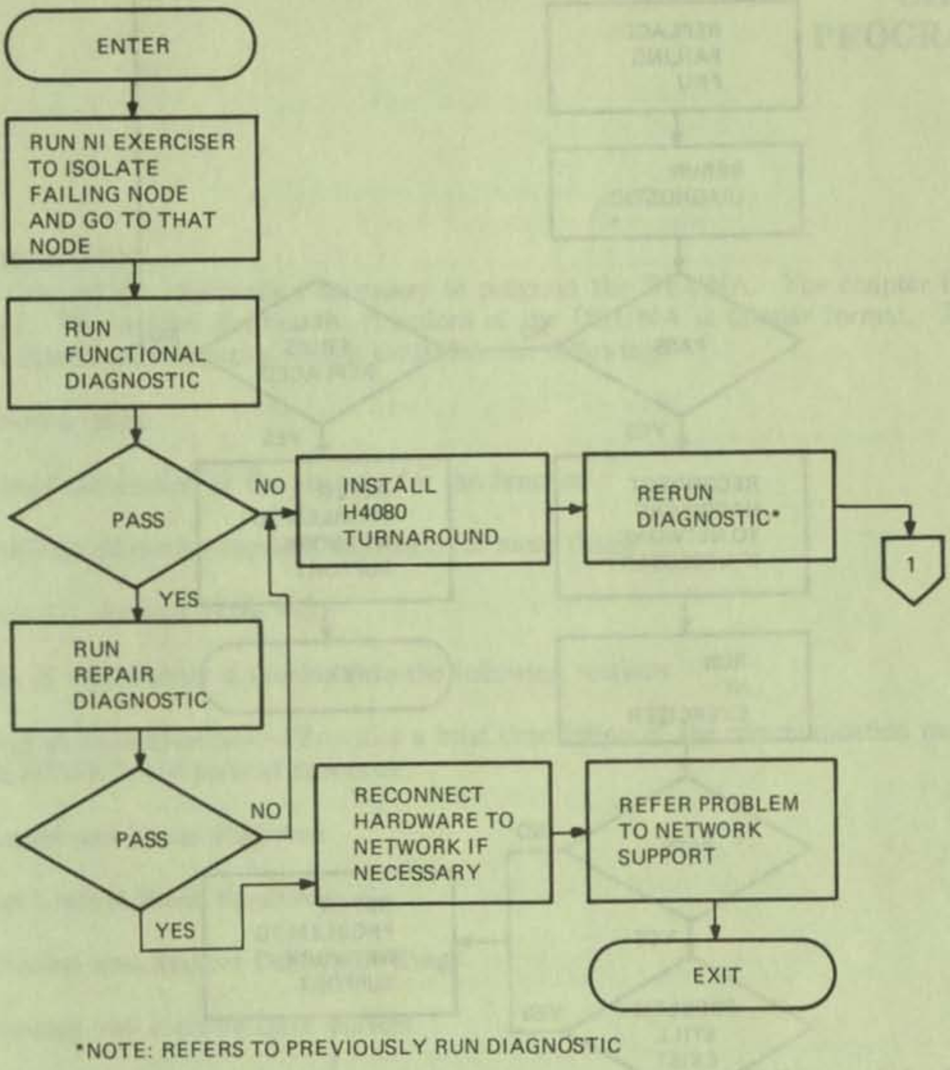

ENTER
RUN NI EXERCISER TO ISOLATE FAILING NODE AND GO TO THAT NODE
RUN FUNCTIONAL DIAGNOSTIC
PASS
NO
YES
INSTALL H4080 TURNAROUND
RERUN DIAGNOSTIC*
1
RUN REPAIR DIAGNOSTIC
PASS
NO
YES
RECONNECT HARDWARE TO NETWORK IF NECESSARY
REFER PROBLEM TO NETWORK SUPPORT
EXIT
*NOTE: REFERS TO PREVIOUSLY RUN DIAGNOSTIC
TK-9720

Figure 3-2 DEUNA Troubleshooting Procedure (Sheet 1 of 2)

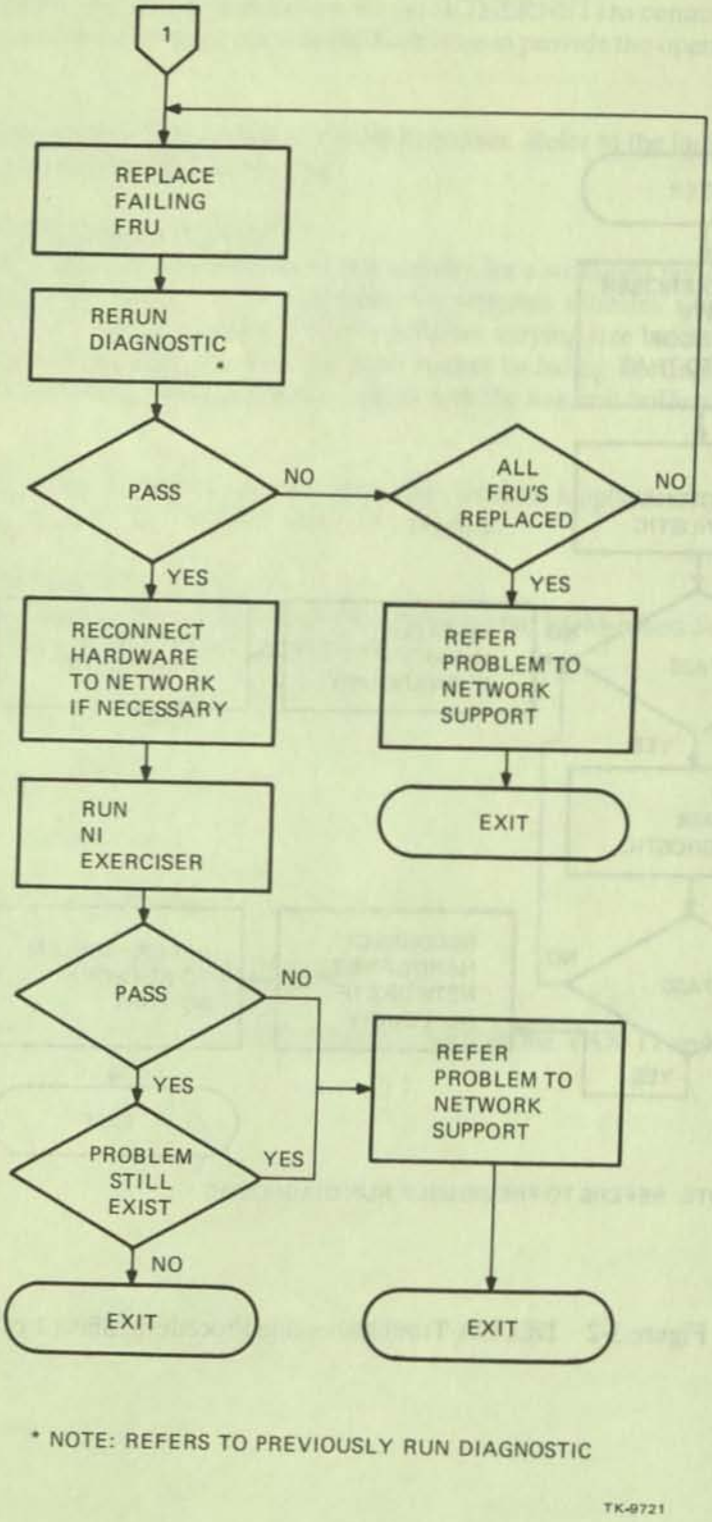

1
REPLACE FAILING FRU
RERUN DIAGNOSTIC *
PASS
NO
ALL FRU'S REPLACED
NO
YES
YES
RECONNECT HARDWARE TO NETWORK IF NECESSARY
REFER PROBLEM TO NETWORK SUPPORT
EXIT
RUN NI EXERCISER
PASS
NO
YES
REFER PROBLEM TO NETWORK SUPPORT
PROBLEM STILL EXIST
YES
NO
EXIT
EXIT
* NOTE: REFERS TO PREVIOUSLY RUN DIAGNOSTIC
TK-9721

Figure 3-2 DEUNA Troubleshooting Procedure (Sheet 2 of 2)

CHAPTER 4
PROGRAMMING

4.1 INTRODUCTION

This chapter contains the information necessary to program the DEUNA. The chapter is divided into several sections. This section defines the functions of the DEUNA in tabular format. The tables are separated into functional categories. Each table lists the following:

- Function name
- A brief description of the purpose for the function
- A pointer to the appropriate section(s) for more detail

Refer to Tables 4-1 through Table 4-5.

The remainder of this chapter is divided into the following sections:

- Programming Overview – Provides a brief description of the communication method between the DEUNA and its host processor.
- Control and Status Registers
- Port Control Block Functions
- Transmit and Receive Descriptor Rings
- Transmit and Receive Data Buffers
- DEUNA Operation – Describes the interaction between the DEUNA and the Port-Driver.
- Exceptional Operations – Describes the DEUNA operations other than the normal transmit and receive datagram service. These functions include Loopback Message and Remote Console Operations.

Table 4-1 DEUNA Control Functions

Function	Purpose	Reference Section
Interrupt System Processor	Allows the DEUNA to get the attention of the port-driver	4.3
Read/Write Interrupt Enable	Allows port-driver to determine if interrupts are enabled	4.3
Driver Reset	Used by the port-driver to place the DEUNA in the reset state	4.3

Table 4-2 DEUNA Port Commands

Function	Purpose	Reference Section
Poll	Informs DEUNA of datagram ready for transmission or receive buffer is available	4.3
Stop	Used by the port-driver to stop transmit and receive datagram service	4.3
Get Port Control Block Base Address	Informs the DEUNA that the Port Control Base Address has been supplied to it	4.3
Execute Command	Informs the DEUNA that a command is in the Port Control Block and should be read and executed	4.3
Start	Used by the port-driver to start the DEUNA processing datagrams	4.3

Table 4-3 DEUNA Data Functions

Function	Purpose	Reference Section
Transmit Packet	Transmits data packets over the Ethernet	4.9.6
Receive Packet	Receives data packets from the Ethernet	4.9.5

Table 4-4 DEUNA Ancillary Commands

Function	Purpose	Reference Section
Read Default Physical Address	Provides port-driver with the unique physical address of the DEUNA	4.4.3
Read/Write Physical Address	Allows the port-driver to read or change the physical address currently being used by the DEUNA	4.4.4
Read/Write Multicast Address Table	Allows the port-driver to read or write the Multicast address table currently being used by the DEUNA	4.4.5
Read/Write Descriptor Ring Format	Allows the port-driver to read or write the current base address and lengths of the transmit and receive descriptor rings	4.4.6
Read/Write Mode	Allows the port-driver to read or write the current mode of operation of the DEUNA	4.4.8
Read Counters	Used by the port-driver to read internal maintenance counters	4.4.7
Read and Clear Counters	The port-driver reads then clears the maintenance counters	4.4.7

Table 4-4 DEUNA Ancillary Commands (Cont)

Function	Purpose	Reference Section
Read/Read and Clear Status	Allows the port-driver to retrieve the internal status of the DEUNA	4.4.9
Read/Write Internal Memory	Allows the port-driver to read and write the internal memory of the DEUNA	4.4.10
Load and Start Microaddress	Allows the port-driver to start execution of WCS-loaded microcode	4.4.2
Write System /D Parameters	Used by the port-driver to build the system-dependent parameter list	4.4.11
Write Load Server Address	Provides DEUNA with the destination address for Request Program Load message	4.4.12

Table 4-5 Maintenance Functions

Function	Purpose	Reference Section
Self-Test	Executed by the DEUNA in the reset state	4.3
TDR	Aid in locating network cable faults	4.7
Maintenance Counters	List counters used for network maintenance	4.4.7
Loop	Used by a remote DEUNA to loop a message through the local DEUNA	4.10.1
Identification	DEUNA response to a Request ID message	4.10.2
Down-line Load	Supports down-line load of system or communications processor	4.10.2.1
Boot	Remote or local initiation of boot	4.10.2.1

4.2 PROGRAMMING OVERVIEW

The operation of the DEUNA is controlled by a program in host memory called the port driver. Communication between the DEUNA and the host processor is accomplished in two ways: by Port Commands between the host and the DEUNA's Control and Status Registers (CSRs) and by Ancillary Commands through shared data structures in host memory via Port Control Block (PCB) Functions.

The host processor issues a Port Command by writing bits ⟨03:00⟩ of the Port Control and Status Register 0 (PCSR0). The DEUNA responds by executing the Port Command and setting the Done Interrupt (DNI) or Port Command Error Interrupt (PCEI) bits.

Refer to Sections 4.3 and 4.9.2 for more information on Port Commands.

The host processor issues an ancillary command by writing to a data structure in host memory rather than directly to the DEUNA PCSRs. Port Functions are used by the port driver program to set up operational and maintenance parameters for the DEUNA. Refer to Section 4.4 for more information on Port functions.

The data structure used for Port Functions is called the Port Control Block (PCB). It consists of four 16-bit words in host memory. The Function Code is written to the low byte of the first word of the PCB. The rest of the PCB is written with Port Function specific information depending on the Port Function to be executed. This information can be pointers to other data structures in host memory or data to be used in executing the Port Function. Refer to Figure 4-1.

The following sequence is an example of communication between the host's port-driver program and the DEUNA using Port Commands and Port Functions.

1. The port-driver loads the DEUNA with the starting address of the PCB (Get PCB Port Command).
2. The port-driver loads the PCB with the appropriate Port Function Code and, if necessary, sets up other memory data structures.
3. The port-driver instructs the DEUNA to fetch the Port Function located in the PCB (Get Command Port Command).
4. The DEUNA reads the PCB via DMA and executes the Port Function.
5. The DEUNA notifies the host of completion of the Port Command via interrupt; either Done Interrupt (DNI) for successful completion or Port Command Error Interrupt (PCEI) for failure.

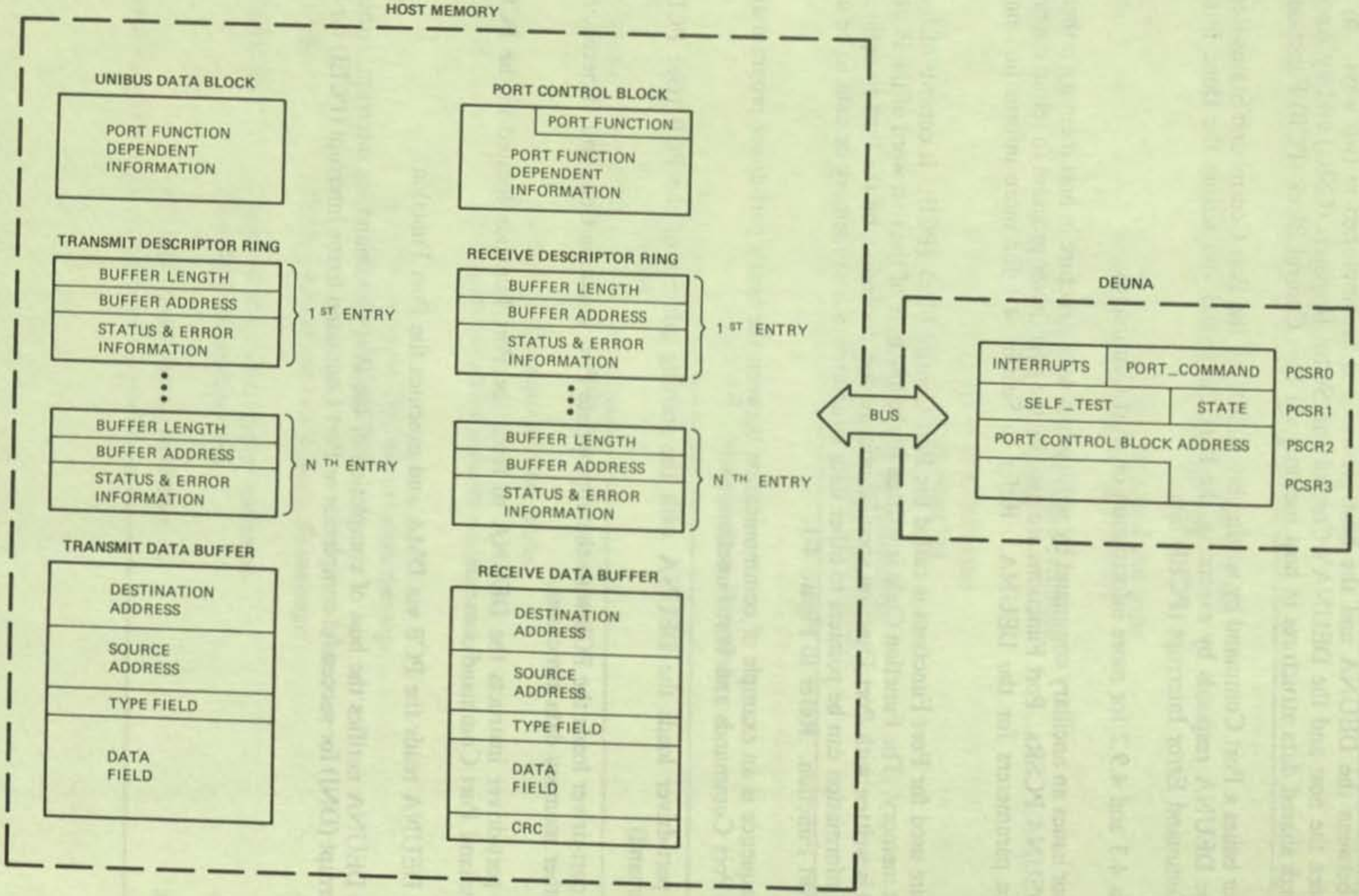


Figure 4-1 DEUNA CSRs and Host Memory Data Structures

Several other data structures may be used by the port-driver when issuing Port Functions to the DEUNA. These structures are Port Function dependent and include the following:

- UNIBUS Data Block (UDB) – The UDB is a data structure in host memory that is of variable size and content depending on the Port Function being executed. It contains supporting information for the Port Function such as pointers to other data structures (see Figure 4-1). Refer to Section 4.4 for more information on specific UDB formats.

- Descriptor Rings – There are two descriptor ring structures: one for transmit and one for receive. They are variable in length and composed of the address of the data buffer, the length of the buffer, and status information associated with the buffer. Refer to Sections 4.5 and 4.6 for a more detailed description of the descriptor rings.

- Data Buffers – The data buffers are contiguous portions of host memory used for packet buffering. Refer to Sections 4.7 and 4.8 for data buffer descriptions and formats.

4.3 PORT CONTROL AND STATUS REGISTERS

There are four control and status registers associated with the DEUNA. They reside at addresses in the UNIBUS I/O page and can be accessed by word or byte operations. The DEUNA accesses PCSR2 and PCSR3 over the UNIBUS. Tables 4-6 through 4-10 and Figures 4-2 through 4-5 describe the Port Control and Status registers bit format and bit descriptions.

Table 4-6 PCSR 0 Bit Descriptions

Bits	Name	Description
⟨15⟩	SERI	Status Error Interrupt – Indicates the presence of an error condition flagged in status register accessible by the port command function. Set by the DEUNA; cleared by the port-driver.
⟨14⟩	PCEI	Port Command Error Interrupt – Indicates the occurrence of either a function error or a UNIBUS timeout during the execution of a port command. Bit 7 of PCSR1 distinguishes between the two error conditions. Set by the DEUNA; cleared by the port-driver.
⟨13⟩	RXI	Receive Ring Interrupt – Attention bit for ring updates. Set by the DEUNA; cleared by the port-driver. When set, indicates that the DEUNA has placed a message on the ring.
⟨12⟩	TXI	Transmit Ring Interrupt – Attention bit for ring updates. Set by the DEUNA; cleared by the port-driver. When set, indicates that transmission has been suspended, all messages found on the transmit ring have been sent, or an error was encountered during a transmission.
⟨11⟩	DNI	Done Interrupt – Interrupts when the DEUNA completes a port command.

Table 4-6 PCSR 0 Bit Descriptions (Cont)

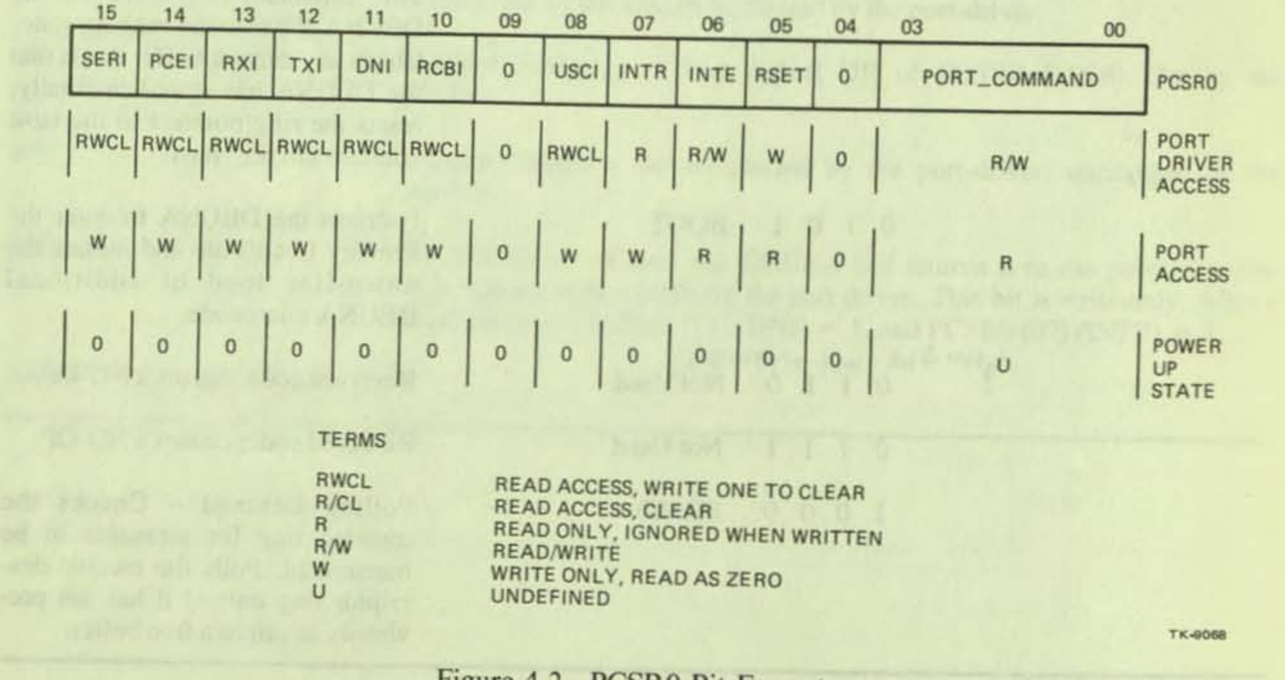
Bits	Name	Description
⟨10⟩	RCBI	Receive Buffer Unavailable Interrupt – Interrupts when the DEUNA discards an incoming message due to receive ring buffers being unavailable. Once set by the DEUNA, RCBI will not be set again until after the DEUNA has received a PDMD port command and discarded a subsequent message. Set by the DEUNA; cleared by the port-driver.
⟨09⟩	ZERO	
⟨08⟩	USCI	Unsolicited State Change Interrupt – Interrupts when the DEUNA performs the following actions: Fatal Error – A transition into the NI AND UNIBUS HALTED state from the READY, RUNNING, UNIBUS HALTED, or NI HALTED states. This state change is caused by the DEUNA detecting an internal fatal error (for example, internal parity error). Communication Processor Boot – A transition into the PRIMARY LOAD state caused by the reception of a remote boot request of the communication processor (DEUNA microcode). Communication Processor Boot – A transition into the READY state from the PRIMARY LOAD state following the reception of the memory load with transfer address message, as part of a remote boot request. The three conditions are distinguished by examining the State field of PCSR1. Set by the DEUNA; cleared by the port-driver.
⟨07⟩	INTR	Interrupt Summary – The logical OR of PCSR0 ⟨15:08⟩. Set by the DEUNA.
⟨06⟩	INTE	Interrupt Enable – Set or cleared by the port-driver; unchanged by the DEUNA.
⟨05⟩	RSET	DEUNA Reset – Clears the DEUNA and returns it to the power-up state when written with a ONE by the port driver. This bit is write-only. After a successful reset, PCSR0 ⟨11⟩ (DNI) = 1, and PCSR0 ⟨07⟩ (INTR) = 1. (Doesn't run self-test?)
⟨04⟩	ZERO	

Table 4-6 PCSR 0 Bit Descriptions (Cont)

Bits	Name	Description		
⟨03:00⟩	PORT_COMMAND			
		0 0 0 0	NO-OP	DNI bit not set (see Section 4.3.1).
		0 0 0 1	GET PCBB	Instructs the DEUNA to fetch the address of the Port Control Block from PCSRs 2 and 3. The DEUNA accesses PCSRs over the UNIBUS, and retains a copy of the address internally. If the address of the Port Control block is changed, this command must be repeated to inform the DEUNA.
		0 0 1 0	GET CMD	Instructs the DEUNA to fetch and execute a command found in the first word of the Port Control Block. The address of the Port Control Block was obtained through the Get PCBB command.
		0 0 1 1	SELF-TEST	Instructs the DEUNA to enter the RESET state and execute self-test.
		0 1 0 0	START	Enables transmission and reception of packets from the port-driver. This command is ignored by the DEUNA if it is in the running state. Clears any current buffer status that the DEUNA has stored internally; resets the ring pointers to the base addresses of the rings.
		0 1 0 1	BOOT	Instructs the DEUNA to enter the Primary Load state and initiate the down-line load of additional DEUNA microcode.
		0 1 1 0	Not Used	Reserved code; causes a NO-OP.
		0 1 1 1	Not Used	Reserved code; causes a NO-OP.
		1 0 0 0	PDMD	Polling Demand – Checks the transmit ring for messages to be transmitted. Polls the receive descriptor ring only if it has not previously acquired a free buffer.

Table 4-6 PCSR 0 Bit Descriptions (Cont)

Bits	Name	Description		
		1 0 0 1	Not Used	Reserved code; causes a NO-OP, sets DNI.
		1 0 1 0	Not Used	Reserved code; causes a NO-OP, sets DNI.
		1 0 1 1	Not Used	Reserved code; causes a NO-OP, sets DNI.
		1 1 0 0	Not Used	Reserved code; causes a NO-OP, sets DNI.
		1 1 0 1	Not Used	Reserved code; causes a NO-OP, sets DNI.
		1 1 1 0	Not Used	Reserved code; causes a NO-OP, sets DNI.
		1 1 1 1	STOP	Suspends operation of the DEUNA and causes a transition to the Ready state. Causes no action if the DEUNA is not in the Running state.



	15	14	13	12	11	10	09	08	07	06	05	04	03–00
PCSR0	SERI	PCEI	RXI	TXI	DNI	RCBI	0	USCI	INTR	INTE	RSET	0	PORT_COMMAND
PORT DRIVER ACCESS	RWCL	RWCL	RWCL	RWCL	RWCL	RWCL	0	RWCL	R	R/W	W	0	R/W
PORT ACCESS	W	W	W	W	W	W	0	W	W	R	R	0	R
POWER UP STATE	0	0	0	0	0	0	0	0	0	0	0	0	U

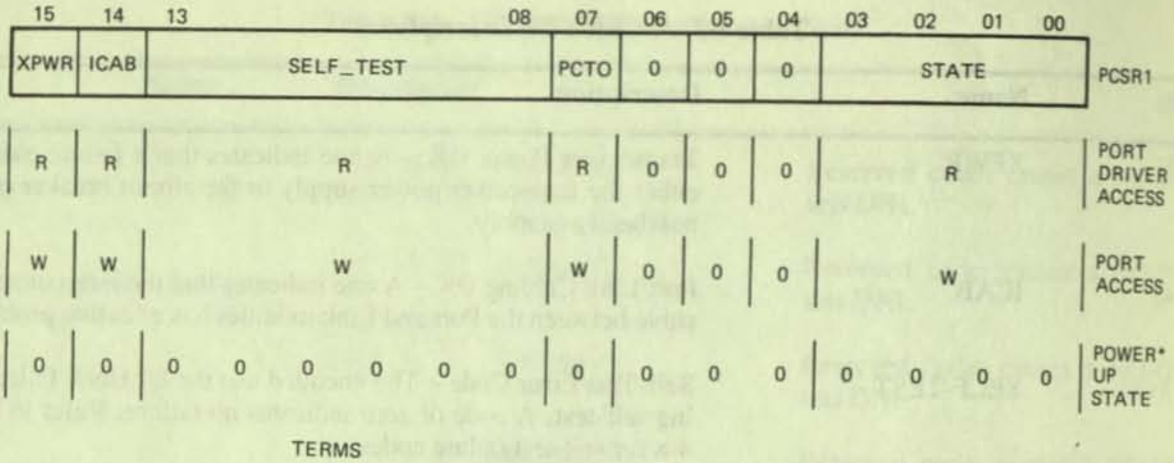
TERMS

RWCL	READ ACCESS, WRITE ONE TO CLEAR
R/CL	READ ACCESS, CLEAR
R	READ ONLY, IGNORED WHEN WRITTEN
R/W	READ/WRITE
W	WRITE ONLY, READ AS ZERO
U	UNDEFINED

Figure 4-2 PCSR0 Bit Format

Table 4-7 PCSR 1 Bit Descriptions

Bits	Name	Description
⟨15⟩	XPWR	Transceiver Power OK – A one indicates that a failure exists in either the transceiver power supply or the circuit breaker on the bulkhead assembly.
⟨14⟩	ICAB	Port/Link Cabling OK – A one indicates that the interconnecting cable between the Port and Link modules has a seating problem.
⟨13:08⟩	SELF-TEST	Self-Test Error Code – The encoded test the DEUNA failed during self-test. A code of zero indicates no failure. Refer to Table 4-8 for self-test failure codes.
⟨07⟩	PCTO	Port Command Timeout – A UNIBUS timeout was encountered while executing a port command (refer to Section 4.9). Valid only after the PCEI bit of PCSR0 is set by the DEUNA. This bit is used to distinguish between a DEUNA failure to complete a port command due to a UNIBUS timeout and a function error.
⟨06:04⟩	Zeros	
⟨03:00⟩	STATE	0 0 0 0 RESET 0 0 0 1 PRIMARY LOAD 0 0 1 0 READY 0 0 1 1 RUNNING 0 1 0 0 Not Used 0 1 0 1 UNIBUS HALTED 0 1 1 0 NI HALTED 0 1 1 1 NI AND UNIBUS HALTED Fatal internal error (for example, parity error). An interrupt condition. When the DEUNA is in this state, the USCI bit of PCSR0 is also set. Cleared by the port-driver setting the RSET bit. 1 X X X Not used



TERMS

RWCL READ ACCESS, WRITE ONE TO CLEAR
R/CL READ ACCESS, CLEAR
R READ ONLY, IGNORED WHEN WRITTEN
R/W READ/WRITE
W WRITE ONLY, READ AS ZERO
U UNDEFINED

*NOTE: THE RESET STATE IS A TRANSITORY STATE. AFTER SUCCESSFUL RESET, PCSR1<03:00>=2.

TK-9069

Figure 4-3 PCSR1 Bit Format

Table 4-8 PCSR 1 ⟨13:08⟩ Self-Test Codes

PCSR1 ⟨13:08⟩						
13	12	11	10	09	08	Test
0	0	0	0	0	0	Completed – No Errors (state=2, DNI set)
1	1	1	1	1	0	CPU Instruction
1	1	1	1	0	1	ROM
1	1	1	1	0	0	Writeable Control Store
1	1	1	0	1	1	T11 UNIBUS Address Register
1	1	1	0	1	0	Receiver UNIBUS DMA
1	1	1	0	0	1	PCSR1 Lower Byte and T11 DMA Read
1	1	1	0	0	0	PCSR0 Upper Byte and T11 DMA Write
1	1	0	1	1	1	PCSR0 Lower Byte and Link Memory DMA
1	1	0	1	1	0	PCSR2 and PCSR3
1	1	0	1	0	1	Timer
1	1	0	1	0	0	Physical Address ROM
1	0	1	1	1	1	Link Memory
						Local Loopback
1	0	0	1	1	1	Transmitter Timeout
1	0	0	1	1	0	Receiver Timeout
1	0	0	1	0	1	Buffer Comparison
1	0	0	1	0	0	Byte Count
1	0	0	0	1	1	Receiver Status
1	0	0	0	1	0	CRC Error
1	0	0	0	0	1	Match Bit Error
1	0	0	0	0	0	TDR Error

Table 4-8 PCSR 1 ⟨13:08⟩ Self-Test Codes (Cont)

	PCSR1 ⟨13:08⟩					
13	12	11	10	09	08	Test
						Transmitter Buffer Address
0	1	1	1	1	1	Transmitter Timeout
0	1	1	1	1	0	Receiver Timeout
0	1	1	1	0	1	Buffer Comparison
0	1	1	1	0	0	Byte Count
0	1	1	0	1	1	Receiver Status
0	1	1	0	1	0	CRC Error
						Receive Buffer Addressing
0	1	0	1	1	1	Transmitter Timeout
0	1	0	1	1	0	Receiver Timeout
0	1	0	1	0	1	Buffer Comparison
0	1	0	1	0	0	Byte Count
0	1	0	0	1	1	Receiver Status
0	1	0	0	1	0	CRC Error
0	0	1	1	1	1	Runt Packet
0	0	1	1	1	0	Minimum Packet Size
0	0	1	1	0	1	Maximum Packet Size
0	0	1	1	0	0	Oversize Packet
0	0	1	0	1	1	CRC
0	0	1	0	1	0	Collision
0	0	1	0	0	1	Heartbeat
0	0	1	0	0	0	Half-Duplex
0	0	0	1	1	1	Multicast
0	0	0	1	1	0	Address Recognition
0	0	0	1	0	1	External Loopback
0	0	0	0	0	0	Never Got Started
						Bug Check (NI & UNIBUS in HALTED State)
0	0	0	0	0	1	Internal Restart Error
0	0	0	0	1	0	Internal Unexpected Interrupt
0	0	0	0	1	1	Link Memory Parity Error
0	0	0	1	0	0	Internal Transmit Buffer Resource Allocation Error
1	0	1	0	0	1	Internal Transmit Buffer Resource Allocation Error on Boot

NOTE

If the LEDs display an alternating pattern after the time required for self-test to run, an unexpected interrupt was received during self-test.

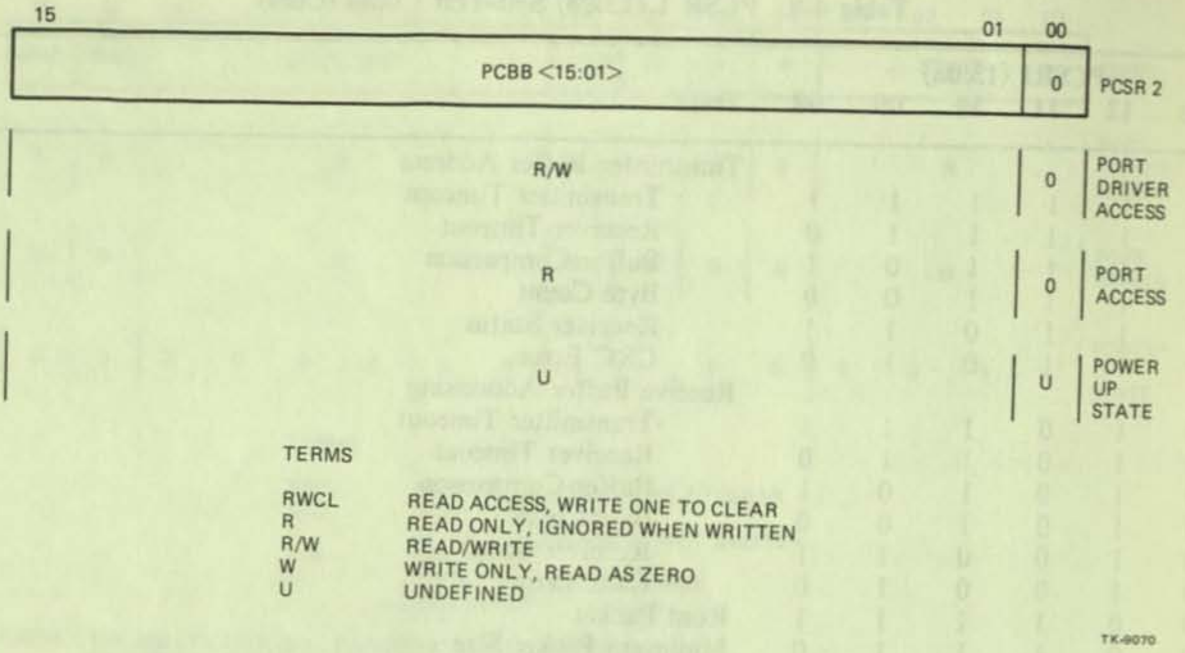
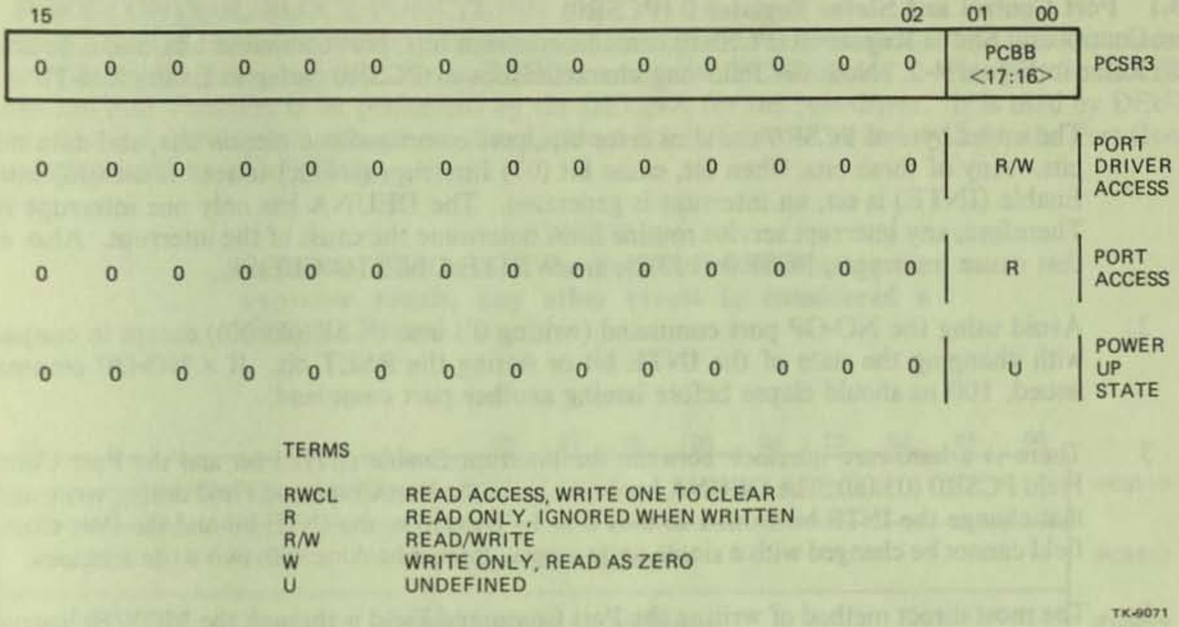


Figure 4-4 PCSR2 Bit Format

Table 4-9 PCSR 2 Bit Description

Bits	Name	Description
⟨15:00⟩	PCBB	The low order 16 bits of the address of the Port Control Block Base. The PCBB is read by the Port as an even number.



15													02	01 00	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	PCBB <17:16>	PCSR3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	PORT DRIVER ACCESS
0	0	0	0	0	0	0	0	0	0	0	0	0	0	R	PORT ACCESS
0	0	0	0	0	0	0	0	0	0	0	0	0	0	U	POWER UP STATE

TERMS

RWCL	READ ACCESS, WRITE ONE TO CLEAR
R	READ ONLY, IGNORED WHEN WRITTEN
R/W	READ/WRITE
W	WRITE ONLY, READ AS ZERO
U	UNDEFINED

Figure 4-5 PCSR3 Bit Format

Table 4-10 PCSR 3 Bit Description

Bits	Name	Description
⟨15:02⟩		Zeros
⟨01:00⟩	PCBB	The high order two bits of the address of the Port Control Block Base.

4.3.1 Port Control and Status Register 0 (PCSR0)

Port Control and Status Register 0 (PCSR0) contains interrupt bits, port command bits, and a device reset bit. Refer to Figure 4-2. Note the following characteristics of PCSR0 (refer to Example 4-1).

1. The upper byte of PCSR0 contains error bits, port command completion bits, and data transfer bits. Any of these bits, when set, cause bit ⟨07⟩ Interrupt (INTR) to set. If bit ⟨06⟩ Interrupt Enable (INTE) is set, an interrupt is generated. The DEUNA has only one interrupt vector. Therefore, any interrupt service routine must determine the cause of the interrupt. Also, all bits that cause interrupts, PCSR0 ⟨15:08⟩, are WRITE ONE TO CLEAR.

2. Avoid using the NO-OP port command (writing 0's into PCSR ⟨03:00⟩) except in conjunction with changing the state of the INTE bit or setting the RSET bit. If a NO-OP command is issued, 100 μs should elapse before issuing another port command.

3. There is a hardware interlock between the Interrupt Enable (INTE) bit and the Port Command Field PCSR0 ⟨03:00⟩. The DEUNA hardware locks the Port Command Field during write accesses that change the INTE bit from 1 to 0 or 0 to 1. Therefore, the INTE bit and the Port Command field cannot be changed with a single write access. It must be done with two write accesses.

4. The most direct method of writing the Port Command Field is through the MOV(B) instruction. However, the INTE bit must be overwritten (not changed) to successfully write the Port Command Field.

5. The high byte of PCSR0 should be cleared using a byte command.

6. For all Port Commands, except a NO-OP, command execution begins with the getting of the Port Command bits in PCSR0. Command execution ends with the setting of either the DN1 or PCEI bits in PCSR0. Only one Port Command can be executing at any time.

COMMAND	PCSR0 BEFORE	PCSR0 AFTER	COMMENT
mov #100, @PCSR0	000002	000102	;INTE bit changed ;so Port Command ;field does not ;change
mov #101, @PCSR0	000102	000101	;INTE bit unchanged ;so Port Command ;field does change ;causes GET PCBB ;command
mov #102, @PCSR0	000101	000102	;INTE bit unchanged ;GET CMD Issued

Example 4-1 Writing Interrupt Bit in PCSR0

4.4 PORT CONTROL BLOCK FUNCTIONS

The Port Control Block is four words of contiguous data located in host memory. The DEUNA accesses the Port Control Block through the address (PCBB) contained in PCSRs 2 and 3. The Port Control Block contains the Port Function to be performed by the DEUNA for the port-driver. It is used by DEUNA initialization and maintenance operations. See Figure 4-6 and Tables 4-11 and 4-12 for the Port Control Block formats and bit descriptions.

NOTE

In Tables 4-13 to 4-30 the Port Driver checks are the expected result, any other result is considered a Function Error.

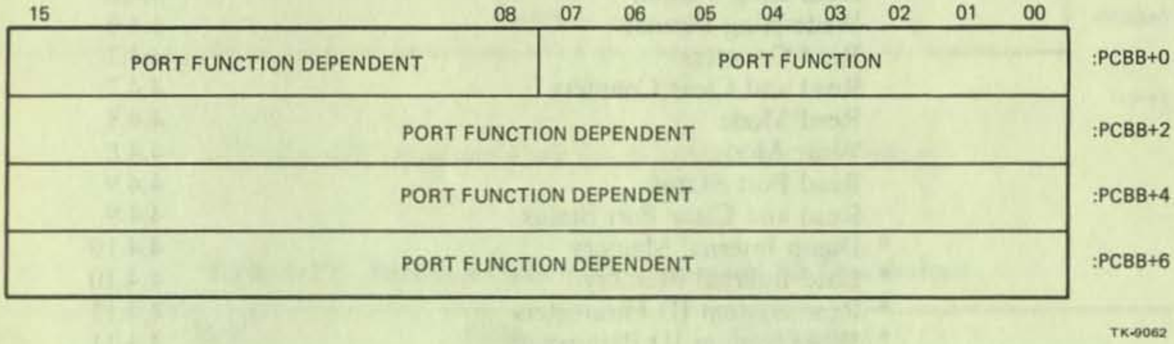


Figure 4-6 Port Control Block Diagram

Table 4-11 Port Control Block Bit Descriptions

Word	Bits	Description
PCBB+0	⟨15:08⟩	Interpreting these bits depends upon the Port Function field.
PCBB+0	⟨07:00⟩	Port Function – Used to pass the DEUNA a function. Written by the port-driver; unchanged by the DEUNA.
PCBB+2	⟨15:00⟩	Interpreting these bits depends upon the Port Function field.
PCBB+4	⟨15:00⟩	Interpreting these bits depends upon the Port Function field.
PCBB+6	⟨15:00⟩	Interpreting these bits depends upon the Port Function field.

Table 4-12 Port Control Functions

Function Code	Reference Function Name	Section
0	No-Operation	4.4.1
1	* Load and Start Microaddress	4.4.2
2	Read Default Physical Address	4.4.3
3	No-Operation	4.4.3
4	Read Physical Address	4.4.4
5	Write Physical Address	4.4.4
6	Read Multicast Address List	4.4.5
7	Write Multicast Address List	4.4.5
10	Read Ring Format	4.4.6
11	Write Ring Format	4.4.6
12	Read Counters	4.4.7
13	Read and Clear Counters	4.4.7
14	Read Mode	4.4.8
15	Write Mode	4.4.8
16	Read Port Status	4.4.9
17	Read and Clear Port Status	4.4.9
20	* Dump Internal Memory	4.4.10
21	* Load Internal Memory	4.4.10
22	* Read System ID Parameters	4.4.11
23	* Write System ID Parameters	4.4.11
24	* Read Load Server Address	4.4.12
25	* Write Load Server Address	4.4.12

* These Port Control Functions are intended for maintenance purposes.

4.4.1 Function Code 0 – No-Operation

See Figure 4-7 and Table 4-13 for the bit formats and bit descriptions of the No-Operation function. For more detail refer to Section 4.3.1.

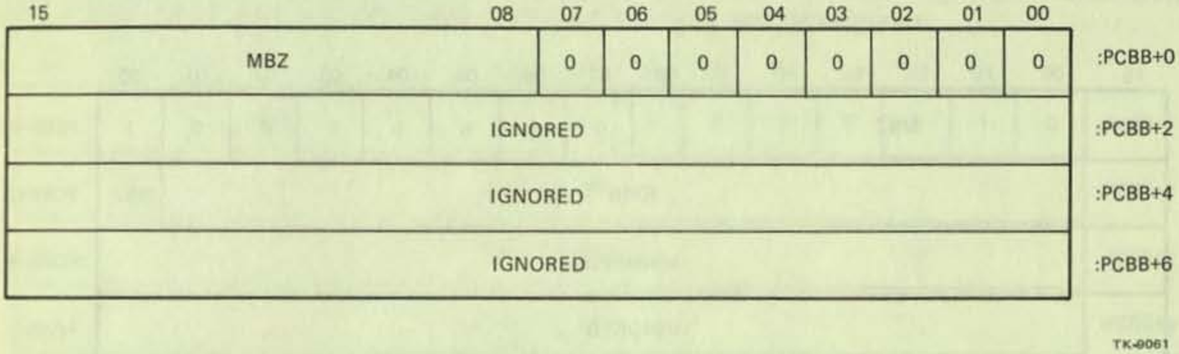


Figure 4-7 Function Code 0 – No Operation Bit Format

Table 4-13 Function Code 0–No–Operation Bit Descriptions

Word	Bits	Field	Description
PCBB+0	⟨15:00⟩	OPCODE	Opcode=0- NO-OP

4.4.2 Function Code 1 – Load and Start Microaddress

This function code is used by the port-driver to instruct the DEUNA to start execution of WCS loaded microcode. The microcode is loaded via Function Code 21 - Load Internal Memory (refer to Section 4.4.10). Both functions are intended for maintenance purposes such as diagnostic testing. See Figure 4-8 and Table 4-14 for bit format and descriptions.

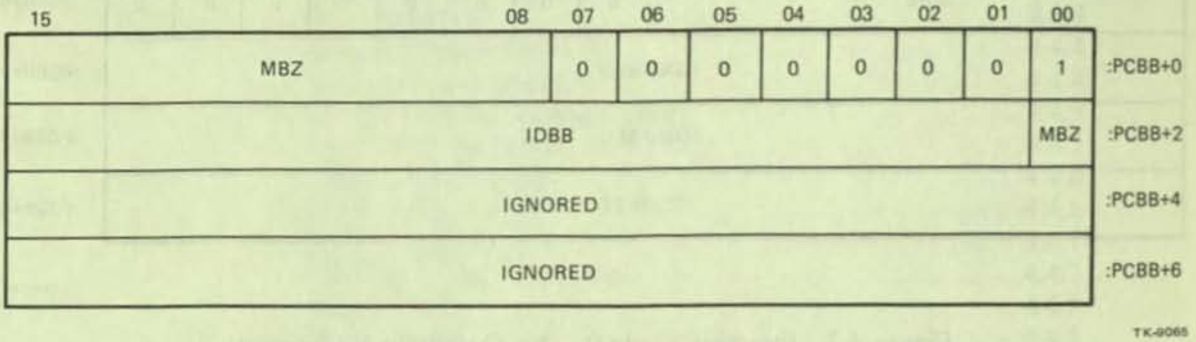


Figure 4-8 Function Code 1 – Load and Start Microaddress Bit Format

Table 4-14 Function Code 1 - Load and Start Microaddress Bit Descriptions

Word	Bits	Field	Description
PCBB+0	⟨15:08⟩	MBZ	Must be zero.
PCBB+0	⟨07:00⟩	OPCODE	OPCODE=1 – Load and Start Microadddress. Instructs the DEUNA to start executing from the microaddress supplied to it. Written by the port-driver, unchanged by the DEUNA.
PCBB+2	⟨15:01⟩	IDBB	The word address of the internal data block the DEUNA is to start executing from.
PCBB+2	⟨00⟩	MBZ	Must be zero.

Port Driver Checks	Resultant Error
(PCBB+0)⟨15:08⟩=0	Function Error
(PCBB+2)⟨00⟩=0	Function Error – Write Function Check Only
State≠RUNNING	Function Error

4.4.3 Function Codes 2 – Read Default Physical Address

Function Code 2 allows you to read the Default Physical Address from the DEUNA. The DEUNA Default Physical Address is the address value residing in the Physical Address ROM on the DEUNA Port module. The physical address is the unique address value associated with a given station on the network. The ETHERNET physical address is distinct from all other physical addresses on all ETHERNETs. The physical address used may be changed by using Function Code 5 – Write Physical Address (refer to Section 4.4.5). See Figure 4-9 and Table 4-15 for bit format and descriptions.

15	08	07	06	05	04	03	02	01	00	
MBZ		0	0	0	0	0	0	1	0	:PCBB+0
DPA <15:00>										:PCBB+2
DPA <31:16>										:PCBB+4
DPA <47:32>										:PCBB+6

TK-9066

Figure 4-9 Function Code 2 – Read Default Physical Address Bit Format

Table 4-15 Function Code 2 – Read Default Physical Address

Word	Bits	Field	Description
PCBB+0	⟨15:08⟩	MBZ	Must be zero
PCBB+0	⟨07:00⟩	OPCODE	OPCODE=2 – Read default physical address out of the DEUNA. OPCODE=3 – No operation. Written by the port-driver, unchanged by the DEUNA.
PCBB+2	⟨15:01⟩	DPA⟨15:01⟩	Address bits ⟨15:01⟩ of the Default Physical Address. Written by the DEUNA for a read function.
PCBB+2	⟨00⟩	DPA⟨00⟩	Must be written a zero for physical addresses.
PCBB+4	⟨15:00⟩	DPA⟨31:16⟩	The middle order 16 address bits of the Default Physical Address. Written by the DEUNA for a read function.
PCBB+6	⟨15:00⟩	DPA⟨47:32⟩	The high order 16 address bits of the Default Physical Address. Written by the DEUNA for a read function.

Port Driver Checks	Resultant Error
(PCBB+0)⟨15:08⟩=0	Function Error

4.4.4 Function Code 3 – No-Operation

This function code causes a NO-OP to be executed by the DEUNA (refer to Section 4.4.1).

4.4.5 Function Codes 4/5 – Read/Write Physical Address

Function Codes 4 and 5 read or change the Physical Address the DEUNA is currently using for address comparison. The DEUNA returns the powerup default Physical Address when read if an address has not been previously written into it. The DEUNA maintains only one Physical Address. The last write of the Physical Address replaces all previous writes. Bit ⟨00⟩ of any physical address must always be a value of zero. See Figure 4-10 and Table 4-16 for bit format and bit descriptions.

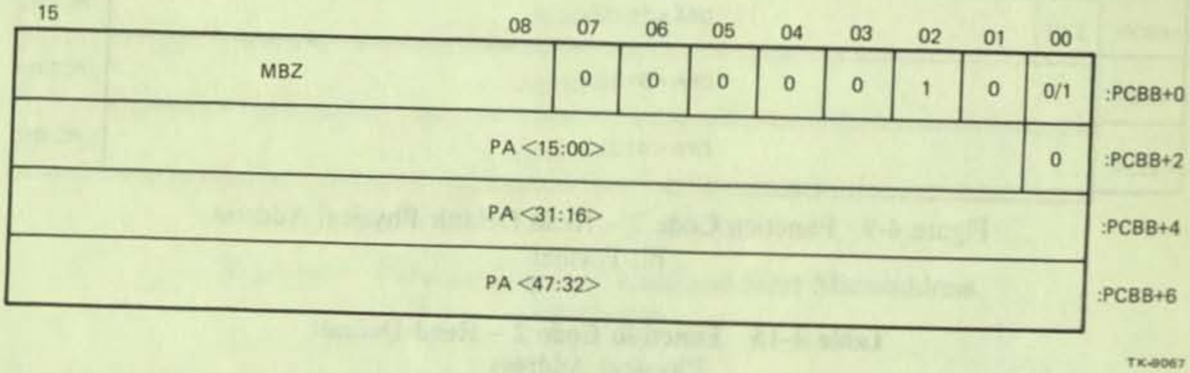


Figure 4-10 Function Codes 4/5 – Read/Write Physical Address Physical Address Bit Format

Table 4-16 Function Codes 4/5 – Read/Write Physical Address Bit Descriptions

Word	Bits	Field	Description
PCBB+0	⟨15:08⟩	MBZ	Must be zero. Written by the port-driver; unchanged by the DEUNA.
PCBB+0	⟨07:00⟩	OPCODE	OPCODE=4 – Read physical address out of the DEUNA. OPCODE=5 – Write physical address into the DEUNA. Written by the port-driver; unchanged by the DEUNA.
PCBB+2	⟨15:01⟩	PA⟨15:01⟩	Address bits ⟨15:01⟩ of the Physical Address. Written by the port-driver for a write function, written by the DEUNA for a read function.
PCBB+2	⟨00⟩	PA⟨00⟩	Must be written zero for physical addresses.
PCBB+2	⟨15:00⟩	PA⟨31:16⟩	The middle order 16 address bits of the Physical Address. Written by the port-driver for a write function, written by the DEUNA for a read function.
PCBB+6	⟨15:00⟩	PA⟨47:32⟩	The high order 16 address bits of the Physical Address. Written by the port-driver for a write function, written by the DEUNA for a read function.

Port Driver Checks	Resultant Error
(PCBB+0)⟨15:08⟩=0	Function Error
(PCBB+2)⟨00⟩=0	Function Error-Write Function Check Only

4.4.6 Function Codes 6/7 – Read/Write Multicast Address List

These two Function Codes enable reading and writing of Multicast addresses. A Multicast Address is an address value that a group of logically related stations respond to. The DEUNA can store a maximum of ten Multicast addresses. The Read Multicast Address List function provides the port-driver with the Multicast address table the DEUNA is currently using for address compare. If no previous Write Multicast address has been done, the UDBB will be unchanged, indicating no Multicast address comparison. The Write Multicast address function is used to enable or change the Multicast address comparison. See Figure 4-11 and Table 4-17 for bit format and bit descriptions.

Each Multicast Address Entry in the Multicast Address Table must have a one in the least significant bit, LA⟨00⟩=1. The UNIBUS Data Block is written by the port-driver and read by the DEUNA for a write function. The UNIBUS Data Block is read by the port-driver and written by the DEUNA for a read function (see Figure 4-12).

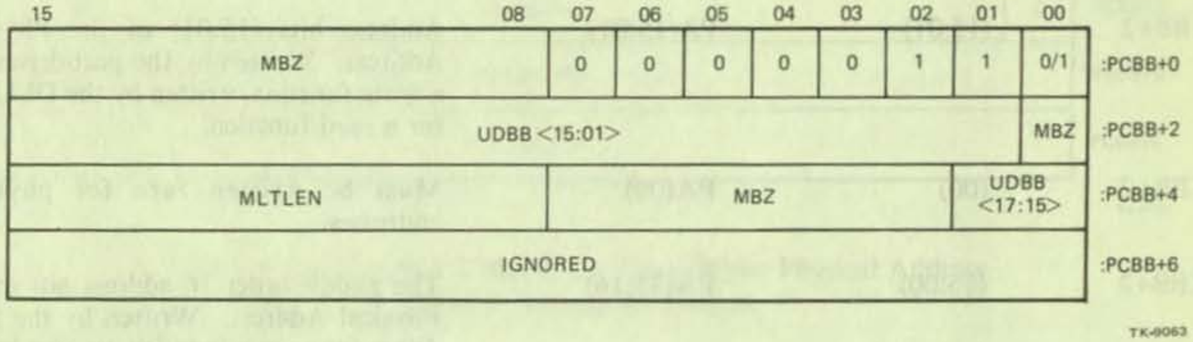


Figure 4-11 Function Codes 6/7 – Read/Write Multicast Address List PCBB Bit Format

Table 4-17 Function Codes 6/7 – Read/Write Multicast Address List PCBB Bit Descriptions

Word	Bits	Field	Description
PCBB+0	⟨15:08⟩	MBZ	Must be zero.
PCBB+0	⟨07:00⟩	OPCODE	OPCODE=6 – Read Multicast address table out of the DEUNA. OPCODE=7 – Write Multicast address table into the DEUNA. Written by the port-driver; unchanged by the DEUNA.
PCBB+2	⟨15:01⟩	UDBB ⟨15:01⟩	The low order 15 address bits of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA.
PCBB+2	⟨00⟩	MBZ	Must be zero.

Table 4-17 Function Codes 6/7 – Read/Write Multicast Address List PCBB Bit Descriptions (Cont)

Word	Bits	Field	Description
PCBB+4	⟨15:08⟩	MLTLEN	Multicast Address Table length. The number of Multicast Addresses read from or written to the UNIBUS Data Block expressed as an unsigned integer. The length in words of the UNIBUS Data Block is three times MLTLEN. Written by the port-driver; unchanged by the DEUNA. When reading, the Multicast Address table and the MLTLEN field is less than the number of Multicast Addresses in the DEUNA, the DEUNA will return, without error, a truncated list equal to the number asked for, starting with the first address in the list. When reading or writing the Multicast Address Table and the MLTLEN field is greater than the maximum number of allowable Multicast Addresses, the DEUNA will abort the command and set the appropriate error status.
PCBB+4	⟨07:02⟩	MBZ	Must be zero.
PCBB+4	⟨01:00⟩	UDBB ⟨17:16⟩	The high order two address bits of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA.

Port Driver Checks	Resultant Error
(PCBB+0)⟨15:08⟩=0	Function Error
(PCBB+2)⟨00⟩=0	Function Error
MLTLEN < MAXMLT	Function Error
MLTLN≠0	Function Error – Read Function Check Only
LA⟨00⟩=1	Function Error – Write Function Check Only

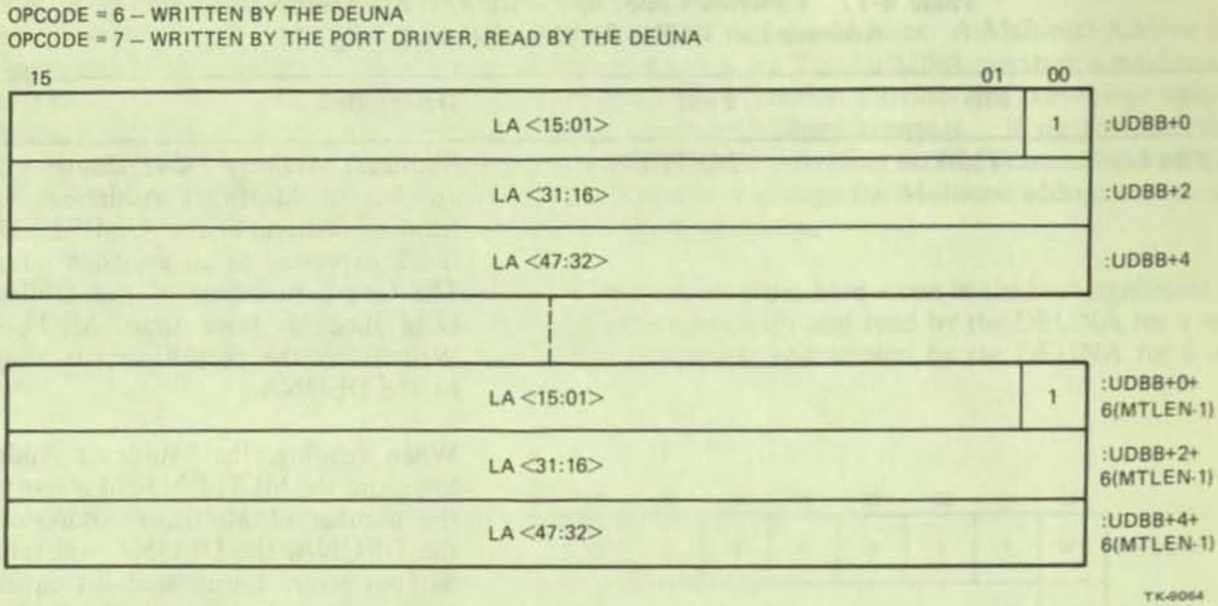


Figure 4-12 Function Codes 6/7 – Read/Write Multicast Address List UDBB Bit Format

4.4.7 Function Codes 10/11 – Read/Write Ring Format

This function provides the port-driver with the current base addresses and lengths of the transmit and receive descriptor rings. If no previous Write Descriptor Ring Format function has been done, the DEUNA responds with zeros in all address and length fields. The Write Descriptor Ring Format function is used to initialize the DEUNA.

Refer to Figure 4-13 and Table 4-18 for PCBB bit format and bit descriptions. For UDBB bit format and bit descriptions, refer to Figure 4-14 and Table 4-19.

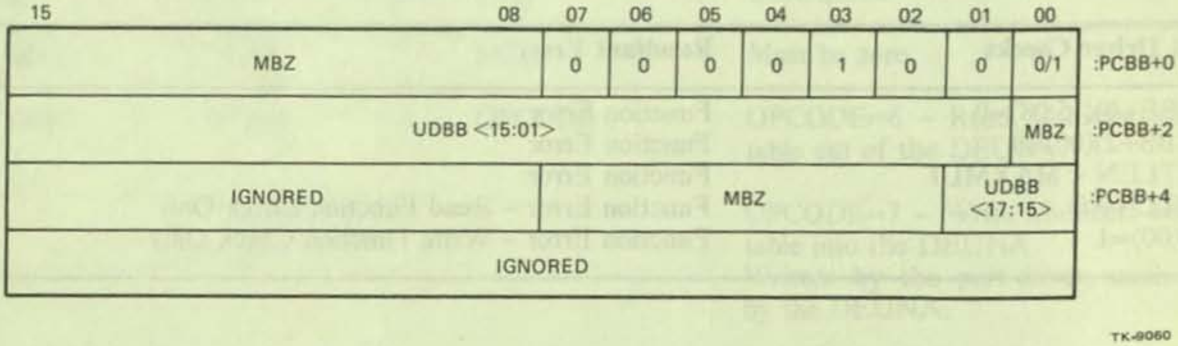


Figure 4-13 Function Codes 10/11 – Read/Write Ring Format PCBB Bit Format

Table 4-18 Function Code 10/11 – Read/Write Ring Format PCBB Bit Descriptions

Word	Bits	Field	Description
PCBB+0	⟨15:08⟩	MBZ	Must be zero.
PCBB+0	⟨07:00⟩	OPCODE	OPCODE=10 – Read descriptor ring specification out of the DEUNA. OPCODE=11 – Write descriptor ring specification into the DEUNA. Written by the port-driver; unchanged by the DEUNA.
PCBB+2	⟨15:01⟩	UDBB ⟨15:01⟩	The low order 15 address bits of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA.
PCBB+2	⟨00⟩	MBZ	Must be zero.
PCBB+4	⟨07:02⟩	MBZ	Must be zero.
PCBB+4	⟨01:00⟩	UDBB ⟨17:16⟩	The high order two address bits of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA.

Port Driver Checks	Resultant Error
(PCBB+0)⟨15:08⟩=0	Function Error
(PCBB+2)⟨00⟩=0	Function Error
(PCBB+4)⟨07:02⟩=0	Function Error

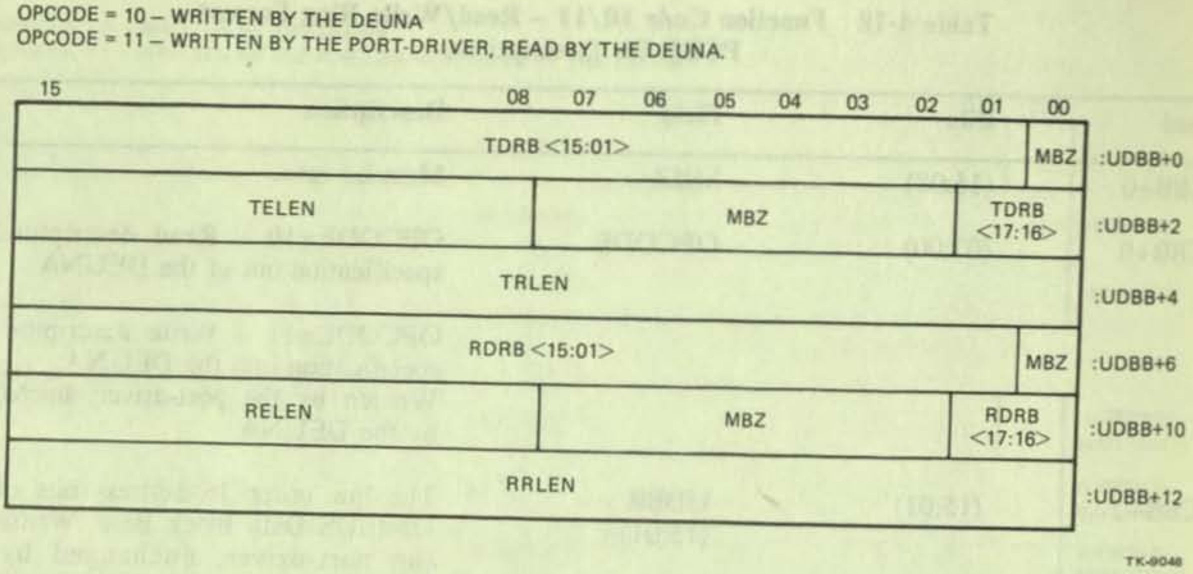


Figure 4-14 Function Codes 10/11 – Read/Write Ring Format UDBB Bit Format

Table 4-19 Function Code 10/11 – Read/Write Ring Format UDBB Bit Descriptions

Word	Bits	Field	Description
UDBB+0	⟨15:01⟩	TDRB ⟨15:01⟩	Address bits ⟨15:01⟩ of the Transmit Descriptor Ring Base.
UDBB+0	⟨00⟩	MBZ	Must be zero.
UDBB+2	⟨15:08⟩	TELEN	Number of words in each entry in the Transmit Descriptor Ring. TELEN must be greater than 4. Expressed as an 8-bit unsigned integer.
UDBB+2	⟨07:02⟩	MBZ	Must be zero.
UDBB+2	⟨01:00⟩	TDRB ⟨17:16⟩	The high order two address bits of the Transmit Descriptor Ring Base.
UDBB+4	⟨15:00⟩	TRLEN	Number of entries in the Transmit Descriptor Ring. Expressed as a 16-bit unsigned integer.
UDBB+6	⟨15:01⟩	RDRB ⟨15:01⟩	Address bits ⟨15:01⟩ of the Receive Descriptor Ring Base. Written by the port-driver; unchanged by the DEUNA.
UDBB+6	⟨00⟩	MBZ	Must be zero.

Table 4-19 Function Code 10/11 – Read/Write Ring Format UDBB Bit Descriptions (Cont)

Word	Bits	Field	Description
UDBB+10	⟨15:08⟩	RELEN	Number of words in each entry in the Transmit Descriptor Ring. TELEN must be greater than 4. Expressed as an 8-bit unsigned integer.
UDBB+10	⟨07:02⟩	MBZ	Must be zero.
UDBB+10	⟨01:00⟩	RDRB ⟨17:16⟩	The high order two address bits of the Receive Descriptor Ring Base.
UDBB+12	⟨15:00⟩	RRLEN	Number of entries in the Receive Descriptor Ring. Expressed as a 16-bit unsigned integer. An RRLEN value of 1 is illegal.

Port Driver Checks	Resultant Error
(UDBB+0)⟨00⟩=0	Function Error
(UDBB+2)⟨07:02⟩=0	Function Error
(UDBB+6)⟨00⟩=0	Function Error
(UDBB+10)⟨07:02⟩=0	Function Error
(UDBB+12)⟨15:00⟩≠1	Function Error

4.4.8 Function Codes 12/13 – Read/Read and Clear Counters

This function is used by the port-driver to read the counters held by the DEUNA.

Refer to Figure 4-15 and Table 4-20 for the PCBB bit format and bit descriptions. The counter values are unsigned integers. Counters latch at their maximum values to indicate overflow. Refer to Figure 4-16 and Table 4-21 for UDBB counter format and counter descriptions.

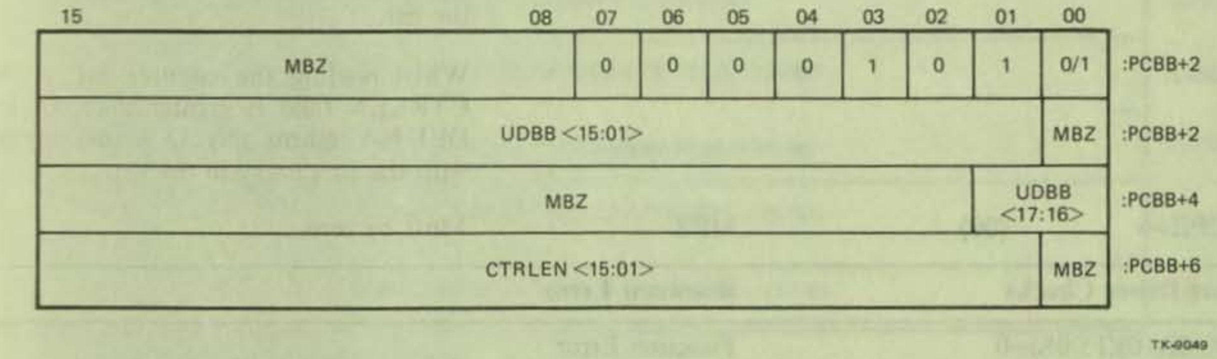


Figure 4-15 Function Codes 12/13 – Read/Write and Clear Counters PCBB Bit Format

Table 4-20 Function Code 12/13 – Read/Read and Clear Counters PCBB Bit Descriptions

Word	Bits	Field	Description
PCBB+0	⟨15:08⟩	MBZ	Must be zero.
PCBB+0	⟨07:00⟩	OPCODE	OPCODE=12 – Read counters out of the DEUNA. OPCODE=13 – Read counters out of the DEUNA and clear counters.
PCBB+2	⟨15:01⟩	UDBB ⟨15:01⟩	Address bits ⟨15:01⟩ of the UNIBUS Block Base. Written by the port-driver; unchanged by the DEUNA.
PCBB+2	⟨00⟩	MBZ	Must be zero.
PCBB+4	⟨15:02⟩	MBZ	Must be zero.
PCBB+4	⟨01:00⟩	UDBB ⟨17:16⟩	The high order two address bits of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA.
PCBB+6	⟨15:01⟩	CTRLEN	Counter List Length – The number of words allocated in the UNIBUS Data Block to accomplish the function. Written by the port-driver; unchanged by the DEUNA. In the DEUNA, CTRLEN has a maximum value of 32 decimal. When reading the counter list, if the CTRLEN field is less than 32, the DEUNA returns the number of words asked for, starting with the first entry in the list. When reading the counter list, if the CTRLEN field is greater than 32, the DEUNA returns only 32 words, starting with the first entry in the list.
PCBB+6	⟨00⟩	MBZ	Must be zero.

Port Driver Checks	Resultant Error
(PCBB+0)⟨15:08⟩=0	Function Error
(PCBB+2)⟨00⟩=0	Function Error
(PCBB+4)⟨15:02⟩=0	Function Error
(PCBB+6)⟨00⟩=0	Function Error

OPCODE=12– READ COUNTERS
OPCODE=13– READ AND CLEAR COUNTERS

15			00	
UNIBUS DATA BLOCK LENGTH				:UDBB+0
SECONDS SINCE LAST ZEROED <15:00>				:UDBB+2
PACKETS RECEIVED <15:00>				:UDBB+4
PACKETS RECEIVED <31:16>				:UDBB+6
MULTICAST PACKETS RECEIVED <15:00>				:UDBB+10
MULTICAST PACKETS RECEIVED <31:16>				:UDBB+12
<15:03> = 0	MLEN	FRAM	CRC	:UDBB+14*
PACKETS RECEIVED WITH ERROR <15:00>				:UDBB+16
DATA BYTES RECEIVED <15:00>				:UDBB+20
DATA BYTES RECEIVED <31:16>				:UDBB+22
MULTICAST DATA BYTES RECEIVED <15:00>				:UDBB+24
MULTICAST DATA BYTES RECEIVED <31:16>				:UDBB+26
RECEIVE PACKETS LOST – INTERNAL BUFFER ERROR <15:00>				:UDBB+30
RECEIVE PACKETS LOST – LOCAL BUFFER ERROR <15:00>				:UDBB+32
PACKETS TRANSMITTED <15:00>				:UDBB+34
PACKETS TRANSMITTED <31:16>				:UDBB+36
MULTICAST PACKETS TRANSMITTED <15:00>				:UDBB+40
MULTICAST PACKETS TRANSMITTED <31:16>				:UDBB+42
PACKETS TRANSMITTED/3+ ATTEMPTS <15:00>				:UDBB+44
PACKETS TRANSMITTED/3+ ATTEMPTS <31:16>				:UDBB+46

*PACKETS RECEIVED WITH ERROR BIT MAP

TK-9047

Figure 4-16 UNIBUS Data Block Format for Counter List
(Sheet 1 of 2)

<table>
<tr><td colspan="7">PACKETS TRANSMITTED 2 ATTEMPTS <15:00></td><td>:UDBB+50</td></tr>
<tr><td colspan="7">PACKETS TRANSMITTED 2 ATTEMPTS <31:16></td><td>:UDBB+52</td></tr>
<tr><td colspan="7">PACKETS TRANSMITTED – DEFERRED <15:00></td><td>:UDBB+54</td></tr>
<tr><td colspan="7">PACKETS TRANSMITTED – DEFFERED <31:16></td><td>:UDBB+56</td></tr>
<tr><td colspan="7">DATA BYTES TRANSMITTED <15:00></td><td>:UDBB+60</td></tr>
<tr><td colspan="7">DATA BYTES TRANSMITTED <31:16></td><td>:UDBB+62</td></tr>
<tr><td colspan="7">MULTICAST DATA BYTES TRANSMITTED <15:00></td><td>:UDBB+64</td></tr>
<tr><td colspan="7">MULTICAST DATA BYTES TRANSMITTED <31:16></td><td>:UDBB+66</td></tr>
<tr><td><15:06> = 0</td><td>LCOL</td><td>MLEN</td><td>0</td><td>0</td><td>LCAR</td><td>RTRY</td><td>:UDBB+70*</td></tr>
<tr><td colspan="7">TRANSMIT PACKETS ABORTED <15:00></td><td>:UDBB+72</td></tr>
<tr><td colspan="7">TRANSMIT COLLISION CHECK FAILURE <15:00></td><td>:UDBB+74</td></tr>
<tr><td colspan="7"><15:00> = 0</td><td>:UDBB+76</td></tr>
</table>

*TRANSMIT PACKET ABORTED BIT MAP

TK-9059

Figure 4-16 UNIBUS Data Block Format for Counter List (Sheet 2 of 2)

Table 4-21 Function Code 12/13 – Read/Read and Clear Counters UDBB Descriptions

Word	Name	Description		
UDBB+0	UNIBUS Data Block Length	The number of words written into the UNIBUS Data Block by the DEUNA to accomplish the read counter function.		
UDBB+2	Seconds Since Last Zeroed	16 bits for the number of seconds since the counters were last zeroed.		
UDBB+4 UDBB+6	Packets Received	32 bits for the total number of error-free datagrams received.		
UDBB+10 UDBB+12	Multicast Packets Received	32 bits for the total number of error-free multicast datagrams received.		
UDBB+14	Packets Received with Error	Bitmap		
		Bit	**Name**	**Description**
		⟨00⟩	CRC	Block Check Error – A datagram failed only the CRC check.
		⟨01⟩	FRAM	Framing Error – A datagram failed the CRC check and did not contain an integral multiple of 8 bits.
		⟨02⟩	MLEN	Message Length Error – A datagram was larger than 1518 bytes.
		⟨15:03⟩		0
UDBB+16	Packets Received	16 bits for the total number of datagrams received with one or more errors logged in the bitmap. Includes only datagrams that passed destination address comparison.		
UDBB+20 UDBB+22	Data Bytes Received	32 bits for the total number of data bytes received error free, exclusive of data link protocol overhead.		
UDBB+24 UDBB+26	Multicast Bytes Received	32 bits for the total number of multicast data bytes received error free, exclusive of data link protocol overhead.		
UDBB+30	Receive Packets Lost–Internal Buffer Error	16 bits for the total number of discards of an incoming packet due to lack of internal buffer space. Incoming packets must be error-free to be counted.		

Table 4-21 Function Code 12/13 – Read/Read and Clear Counters UDBB Descriptions (Cont)

Word	Name	Description
UDBB+32	Received Packets Lost – Local Buffer Error	16 bits for the total number of problems with a receive ring data buffer. This counter is incremented for the following reasons: • Buffer Unavailable – Datagram lost because there was no available buffer on the receive ring. • Buffer Too Small – Datagram truncated because it was larger than the available buffer space on the receive ring.
UDBB+34 UDBB+36	Packets Transmitted	32 bits for the total number of datagrams successfully transmitted, including transmissions in which the collision test signal failed to assert.
UDBB+40 UDBB+42	Multicast Packets Transmitted	32 bits for the total number of multicast datagrams successfully transmitted, including transmissions in which the collision test signal failed to assert.
UDBB+44 UDBB+46	Packets Transmitted 3+ Attempts	32 bits for the total number of datagrams successfully transmitted on three or more attempts, including transmissions in which the collision test signal failed to assert.
UDBB+50 UDBB+52	Packets Transmitted 2 Attempts	32 bits for the total number of datagrams successfully transmitted on two attempts, including transmissions in which the collision test signal failed to assert.
UDBB+54 UDBB+56	Packets Transmitted Deferred	32 bits for the total number of datagrams successfully transmitted on the first attempt after deferring, including transmissions in which the collision test signal failed to assert.
UDBB+60 UDBB+62	Data Bytes Transmitted	32 bits for the total number of data bytes successfully transmitted.
UDBB+64 UDBB+66	Multicast Data Bytes Transmitted	32 bits for the total number of multicast data bytes successfully transmitted.

Note: The counter values dealing with the Collision Test Signal are only valid when the DEUNA is connected to an H4000 or similar tranceiver with a collision test feature and the Enable Collision Test (ECT) bit is set in the DEUNA Mode Register (refer to Section 4.4.8).

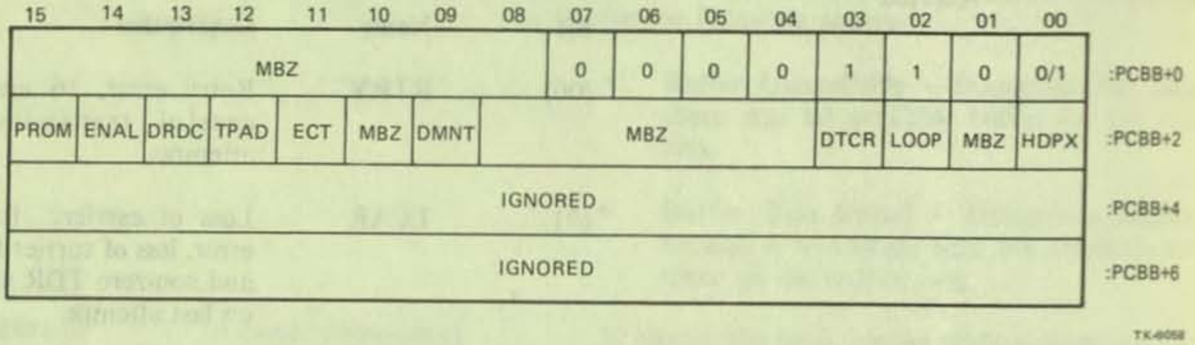
Table 4-21 Function Code 12/13 – Read/Read and Clear Counters UDBB Descriptions (Cont)

Word	Name	Description		
UDBB+70	Transmit Packets Aborted	Bitmap		
		Bit	**Name**	**Description**
		⟨00⟩	RTRY	Retry error, 16 unsuccessful transmission attempts.
		⟨01⟩	LCAR	Loss of carrier. Retry error, loss of carrier flag, and non-zero TDR value on last attempt.
		⟨02⟩	0	Always = 0
		⟨03⟩	0	Always = 0
		⟨04⟩	MLEN	Data Block too long. The DEUNA aborted the transmission because the datagram exceeded the maximum packet length.
		⟨05⟩	LCOL	Late collision on the last transmission attempt.
		⟨15:06⟩	0	Always = 0.
UDBB+72	Transmit Packets Aborted	16 bits for the total number of datagrams aborted during transmission for one of the bitmapped errors.		
UDBB+74	Transmit Collision Detect Failure	16 bits for the total number of times the collision test signal failed to assert following an apparently successful transmission.		
UDBB+76	ZEROS			

Note: The counter values dealing with the Collision Test Signal are only valid when the DEUNA is connected to an H4000 or similar transceiver with a collision test feature and the Enable Collision Test (ECT) bit is set in the DEUNA Mode Register (refer to Section 4.4.8).

4.4.9 Function Codes 14/15 – Read/Write Mode

This function is used by the port-driver to read or write the mode register of the DEUNA. The mode register is used to program the operation of the DEUNA when it is in the RUNNING state. Refer to Figure 4-17 and Table 4-22 for the PCBB bit formats and bit descriptions.



	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
	MBZ								0	0	0	0	1	1	0	0/1	:PCBB+0
	PROM	ENAL	DRDC	TPAD	ECT	MBZ	DMNT	MBZ					DTCR	LOOP	MBZ	HDPX	:PCBB+2
	IGNORED																:PCBB+4
	IGNORED																:PCBB+6

Figure 4-17 Function Codes 14/15 – Read/Write Mode PCBB Bit Format

Table 4-22 Function Code 14/15 – Read/Write Mode PCBB Bit Descriptions

Word	Bits	Name	Description
PCBB + 0	⟨15:08⟩	MBZ	Must be zero.
PCBB + 0	⟨07:00⟩	OPCODE	OPCODE = 14 – Read the mode out of the DEUNA. OPCODE = 15 – Write the mode into the DEUNA. Written by the port-driver; unchanged by the DEUNA.
PCBB + 2	⟨15⟩	PROM	Promiscuous Mode – Instructs the DEUNA to accept all incoming packets regardless of the destination address field. Written by the DEUNA for a read. Written by the port-driver for a write. Cleared internally upon power up.
PCBB + 2	⟨14⟩	ENAL	Enable All Multicast – Instructs the DEUNA to accept all incoming packets with Multicast destinations. Written by the DEUNA for a read. Written by the port-driver for a write. Cleared internally upon power up.

Table 4-22 Function Code 14/15 – Read/Write Mode PCBB Bit Descriptions (Cont)

Word	Bits	Name	Description
PCBB + 2	⟨13⟩	DRDC	Disable data chaining mode on received messages. When the DEUNA is in the mode, it truncates messages that do not fit in a single buffer. Status information remains intact. Written by the port-driver for a write. Written by the DEUNA for a read. Cleared internally upon power up.
PCBB + 2	⟨12⟩	TPAD	Transmit Message Pad Enable – Instructs the DEUNA to pad messages shorter than 64 bytes long, not including the CRC, for transmission. The DEUNA pads the data field only. Written by the port-driver for a write. Written by the DEUNA for a read. Cleared internally upon power up.
PCBB + 2	⟨11⟩	ECT	Enable Collision Test – Instructs the DEUNA to check for collision test after each transmission. This bit should only be used with tranceivers that have the collision test feature, for example H4000.
PCBB + 2	⟨10⟩	MBZ	Must be zero.
PCBB + 2	⟨09⟩	DMNT	Disable maintenance message. Instructs the DEUNA not to transmit a response to all incoming loop, boot, request ID, and memory load with transfer address messages. In addition, the DEUNA will not issue the system ID message. This bit is an aid in running on-line diagnostics. Written by the DEUNA for a read. Written by the port-drive for a write. Cleared internally upon power up.
PCBB + 2	⟨08:04⟩	MBZ	Must be zero.

Table 4-22 Function Code 14/15 – Read/Write Mode PCBB Bit Descriptions (Cont)

Word	Bits	Name	Description
PCBB + 2	⟨03⟩	DTCR	Disable Transmit CRC – Instructs the DEUNA not to append 4 bytes of link generated CRC to the transmitted packet, not to transmit a response to all incoming loop, boot, request ID, and memory load with transfer address messages. In addition, the DEUNA will not issue the system ID message. Written by the DEUNA for a read. Written by the port-driver for a write. Cleared internally upon power up.
PCBB + 2	⟨02⟩	LOOP	Internal Loopback Mode – Disables the DEUNA from the transceiver, and loops the output of the DEUNA transmitter logic to the input of the receiver logic. The collision test fails if enabled during transmissions with LOOP set. Written by the DEUNA for a read. Written by the port-driver for a write. Cleared internally upon power up.
PCBB + 2	⟨01⟩	MBZ	Must be zero.
PCBB + 2	⟨00⟩	HDPX	Half-Duplex Mode – When clear, indicates that the DEUNA will receive messages transmitted to itself over the wire. Messages received in this manner will not undergo CRC check use; CRC error status will be returned with them. When set, indicates that the DEUNA will not receive messages transmitted to itself. However, the DEUNA recognizes the transmitted message as being addressed to itself and sets the MTCH bit in the transmit ring following the tranmission attempt. Cleared internally upon power up.

Port Driver Checks	Resultant Error
(PCBB + 0)⟨15:08⟩ = 0	Function Error
(PCBB + 2)⟨10,08:04,01⟩ = 0	Function Error – Write Function Check Only

4.4.10 Function Codes 16/17 – Read/Read and Clear Port Status

This function is used by the port-driver to read and clear status from the DEUNA. Function code 17 will clear the high byte of PCBB+2. Refer to Figure 4-18 and Table 4-23 for PCBB bit format and bit descriptions. ˙

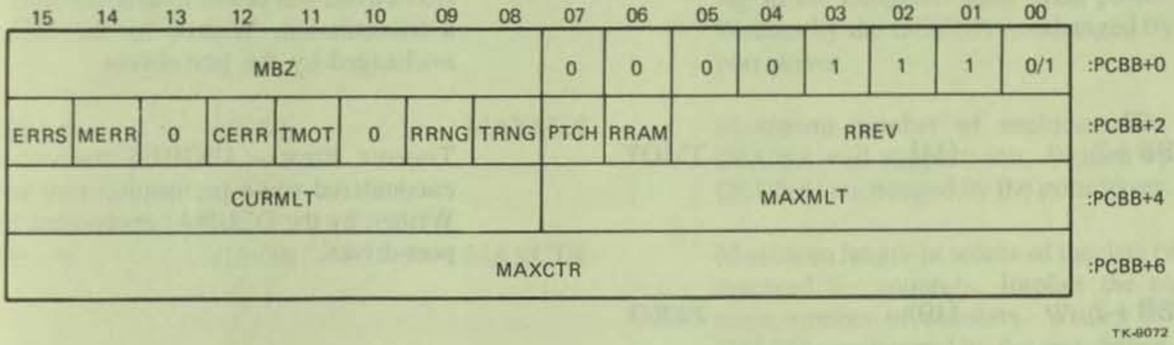


Figure 4-18 Function Codes 16/17 – Read/Read and Clear Port Status PCBB Bit Format

Table 4-23 Function Code 16/17 – Read/Read and Clear Port Status

Word	Bits	Field	Description
PCBB + 0	⟨15:08⟩	MBZ	Must be zero.
PCBB + 0	⟨07:00⟩	OPCODE	OPCODE = 16 – Read status from the DEUNA. OPCODE = 17 – Read status from the DEUNA and clear status in the DEUNA. Written by the port-driver; unchanged by the DEUNA.
PCBB + 2	⟨15⟩	ERRS	Error Summary – Logical OR of MERR, RBUF, TMOT, FNER, RRNG, TRNG, and LEN. Written by the DEUNA; unchanged by the port-driver.
PCBB + 2	⟨14⟩	MERR	Multiple Errors – Multiple ring access errors encountered while handling buffer access errors. Written by the DEUNA; unchanged by the port-driver.
PCBB + 2	⟨13⟩	ZERO	

Table 4-23 Function Code 16/17 – Read/Read and Clear Port Status (Cont)

Word	Bits	Field	Description
PCBB + 2	⟨12⟩	CERR	Collision Test Error – The transceiver collision circuit has failed to activate following a transmission. Written by the DEUNA; unchanged by the port-driver.
PCBB + 2	⟨11⟩	TMOT	Timeout Error – UNIBUS timeout error encountered while performing ring access. Written by the DEUNA; unchanged by the port-driver.
PCBB + 2	⟨10⟩	ZERO	
PCBB + 2	⟨09⟩	RRNG	Receiver Ring Error – DEUNA encountered a ring parsing error while accessing the receive descriptor ring. Written by the DEUNA; unchanged by the port-driver.
PCBB + 2	⟨08⟩	TRNG	Transmit Ring Error – DEUNA encountered a ring parsing error while accessing the transmit descriptor ring. Written by the DEUNA; unchanged by the port-driver.
PCBB + 2	⟨07⟩	PTCH	ROM Patch – DEUNA WCS contains a patch for the ROM based operational micrcode. Set by the DEUNA; unchanged by the port driver.
PCBB + 2	⟨06⟩	RRAM	RAM Microcode Operational – DEUNA is executing from RAM rather than ROM microcode. Written by the DEUNA; unchanged by the port-driver.
PCBB + 2	⟨05:00⟩	RREV	ROM revision – The revision number of the DEUNA on-board microcode. Written by the DEUNA; unchanged by the port-driver.

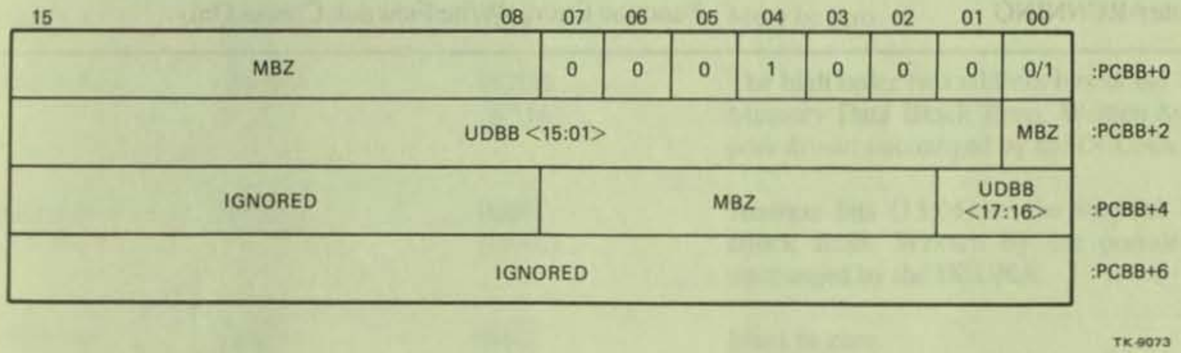
Table 4-23 Function Code 16/17 – Read/Read and Clear Port Status (Cont)

Word	Bits	Field	Description
PCBB + 4	⟨15:08⟩	CURMLT	The current number of multicast IDs residing in the DEUNA. Zero upon power up. Written by the DEUNA; unchanged by the port-driver.
PCBB + 4	⟨07:00⟩	MAXMLT	Maximum number of multicast IDs the DEUNA will support: ten. Written by the DEUNA; unchanged by the port-driver.
PCBB + 6	⟨15:00⟩	MAXCTR	Maximum length in words of the data block reserved for counters. Implies the maximum number of counters. Written by the DEUNA; unchanged by the port-driver.

Port Driver Checks	Resultant Error
(PCBB + 0)⟨15:08⟩ = 0	Function Error
(PCBB + 2)⟨13,10⟩ = 0	Function Error – Write Function Check Only

4.4.11 Function Codes 20/21 - Dump/Load Internal Memory

These functions are used to block move data or microcode between the host memory and the internal memory (WCS) of the DEUNA. It is used for maintenance purposes such as diagnostics. The data move is done by the DEUNA. Refer to Figure 4-19 and Table 4-24 for the PCBB bit format and bit descriptions. Refer to Figure 4-20 and Table 4-25 for UDBB format and bit descriptions.



15 – 08	07	06	05	04	03	02	01	00	
MBZ	0	0	0	1	0	0	0	0/1	:PCBB+0
UDBB <15:01>								MBZ	:PCBB+2
IGNORED	MBZ						UDBB <17:16>		:PCBB+4
IGNORED									:PCBB+6

Figure 4-19 Function Codes 20/21 – Load/Dump Internal Memory PCBB Bit Format

Table 4-24 Function Code 20/21 – Dump/Load Internal Memory PCBB Bit Descriptions

Word	Bits	Field	Description
PCBB+0	⟨15:08⟩	MBZ	Must be zero.

PCBB+0	⟨07:00⟩	OPCODE	OPCODE=20 – Dump internal RAM of DEUNA.
			OPCODE=21 – Load internal RAM of DEUNA.

PCBB+2	⟨15:01⟩	UDBB ⟨15:01⟩	Address bits ⟨15:01⟩ of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA.
PCBB+2	⟨00⟩	MBZ	Must be zero.
PCBB+4	⟨15:08⟩	IGNORED	Ignored by the DEUNA.
PCBB+4	⟨07:02⟩	MBZ	Must be zero.
PCBB+4	⟨01:00⟩	UDBB ⟨17:16⟩	The high order two address bits of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA.
PCBB+6	⟨15:00⟩	IGNORED	Ignored by the DEUNA.

Port Driver Checks	Resultant Errors
(PCBB+0)⟨15:08⟩=0	Function Error
(PCBB+2)⟨00⟩=0	Function Error
(PCBB+4)⟨07:02⟩=0	Function Error
State≠RUNNING	Function Error – Write Function Checks Only

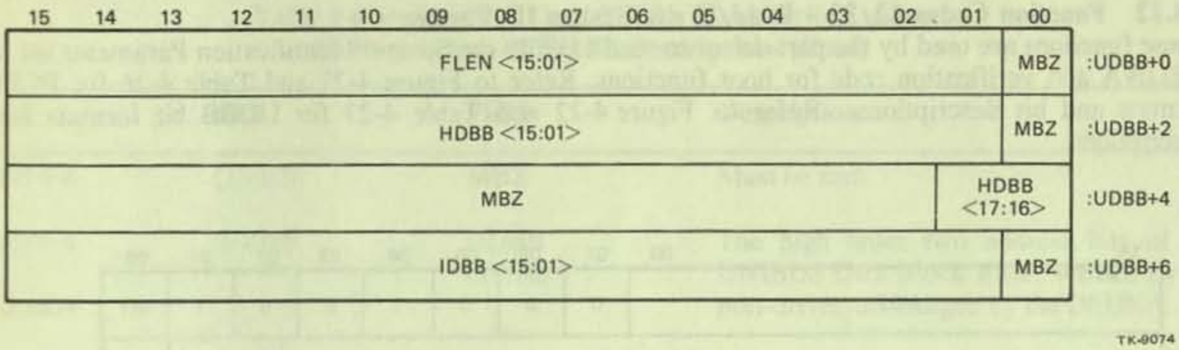


Figure 4-20 Function Codes 20/21 – Load/Dump Internal Memory UDBB Bit Format

Table 4-25 Function Code 20/21 – Load/Dump Internal Memory UDBB Bit Descriptions

Word	Bits	Field	Description
UDBB + 0	⟨15:01⟩	FLEN	Function length – An unsigned integer indicating the number of words to be transferred between UDBB and IDBB. Set by the port-driver; unchanged by the DEUNA.
UDBB + 0	⟨00⟩	MBZ	Must be zero.
UDBB + 2	⟨15:01⟩	HDBB ⟨15:01⟩	Address bits ⟨15:01⟩ of the Host Memory Data Block Base. Written by the port-driver; unchanged by the DEUNA.
UDBB + 2	⟨00⟩	MBZ	Must be zero.
UDBB + 4	⟨15:02⟩	MBZ	Must be zero.
UDBB + 4	⟨01:00⟩	HDBB ⟨17:16⟩	The high order two address bits of the Host Memory Data Block Base. Written by the port-driver; unchanged by the DEUNA.
UDBB + 6	⟨15:01⟩	IDBB ⟨15:01⟩	Address bits ⟨15:01⟩ of the Internal Data Block Base. Written by the port-driver; unchanged by the DEUNA.
UDBB + 6	⟨00⟩	MBZ	Must be zero.

Port Driver Checks	Resultant Error
(UDBB + 0)⟨00⟩ = 0	Function Error
(UDBB + 2)⟨00⟩ = 0	Function Error
(UDBB + 4)⟨15:02⟩ = 0	Function Error
(UDBB + 6)⟨00⟩ = 0	Function Error

4.4.12 Function Codes 22/23 – Read/Write System ID Parameters

These functions are used by the port-driver to read or write the System Identification Parameter list of the DEUNA and verification code for boot functions. Refer to Figure 4-21 and Table 4-26 for PCBB bit formats and bit descriptions. Refer to Figure 4-22 and Table 4-27 for UDBB bit formats and bit descriptions.

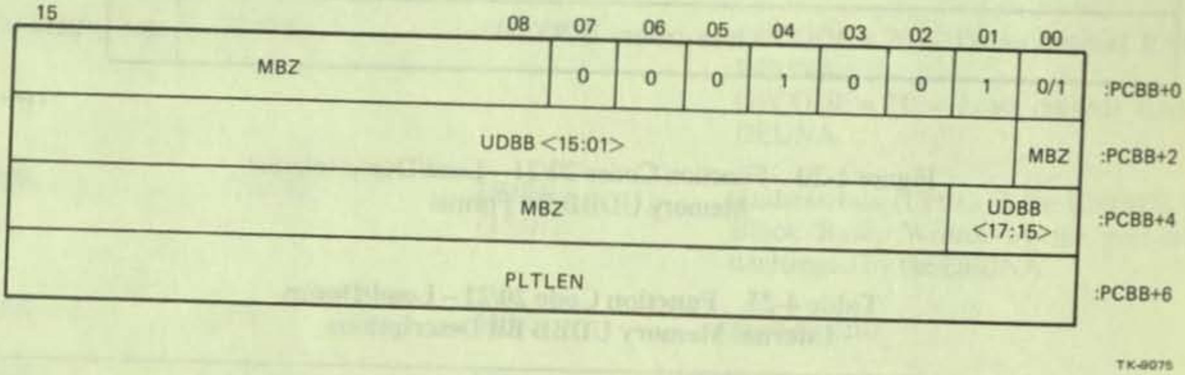


Figure 4-21 Function Codes 22/23 – Read/Write System ID Parameters PCBB Bit Format

Table 4-26 Function Code 22/23 – Read/Write System ID Parameters PCBB Bit Descriptions

Word	Bits	Field	Description
PCBB + 0	⟨15:08⟩	MBZ	Must be zero.
PCBB + 0	⟨07:00⟩	OPCODE	OPCODE = 22 – Read system ID parameter list out of the DEUNA. OPCODE = 23 – Write system ID parameter list into the DEUNA. Written by the port-driver; unchanged by the DEUNA.
PCBB + 2	⟨15:01⟩	UDBB ⟨15:01⟩	The low order 15 address bits of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA.
PCBB + 2	⟨00⟩	MBZ	Must be zero.

Table 4-26 Function Code 22/23 – Read/Write System ID Parameters PCBB Bit Descriptions (Cont)

Word	Bits	Field	Description
PCBB + 4	⟨15:02⟩	MBZ	Must be zero.
PCBB + 4	⟨01:00⟩	UDBB ⟨17:16⟩	The high order two address bits of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA.
PCBB + 6	⟨15:01⟩	PLTLEN	System ID Parameter list length. The length in words of the UNIBUS Data Block Base. Written by the port-driver; unchanged by the DEUNA. The maximum value of PLTLEN is 100 decimal. When reading the System ID Parameter list, if the PLTLEN field is less than 100 words, the DEUNA will return, without error, a truncated list equal to the number asked for, starting with the first entry in the list. When reading or writing the System ID Parameter list, if the PLTLEN field is greater than 100 words, the DEUNA will abort the command and set the appropriate error status.

Port Driver Checks	Resultant Error
(PCBB + 0)⟨15:08⟩ = 0	Function Error
(PCBB + 2)⟨00⟩ = 0	Function Error
(PCBB + 4)⟨15:02⟩ = 0	Function Error
(PCBB + 6)⟨00⟩ = 0	Function Error
27<PLTEN≤100 decimal	Function Error

OPCODE = 22 READ SYSTEM ID PARAMETER LIST.
OPCODE = 23 WRITE SYSTEM ID PARAMETER LIST.

15 08	07 00	
VC <15:00>		:UDBB+0
VC <31:16>		:UDBB+2
VC <47:32>		:UDBB+4
VC <63:48>		:UDBB+6
MBZ	SOFTID	:UDBB+10
UNDEFINED		:UDBB+12
UNDEFINED		:UDBB+14
UNDEFINED		:UDBB+16
UNDEFINED		:UDBB+20
UNDEFINED		:UDBB+22
UNDEFINED		:UDBB+24
TYPE		:UDBB+26
CCOUNT		:UDBB+30
MBZ	CODE	:UDBB+32
RECNUM		:UDBB+34
MVTYPE		:UDBB+36
MVVER	MVLEN	:UDBB+40
MVUECO	MVECO	:UDBB+42
FTYPE		:UDBB+44
FVAL1	FLEN	:UDBB+46
HATYPE <07:00>	FVAL2	:UDBB+50
HALEN	HATYPE <15:08>	:UDBB+52
HA <15:00>		:UDBB+54
HA <31:16>		:UDBB+56
HA <47:32>		:UDBB+60
DTYPE		:UDBB+62
DVALUE	DLEN	:UDBB+64
PARAM		:UDBB+66
PARAM		:UDBB+70
PARAM		:UDBB+72
PARAM		

TK-8057

Figure 4-22 Function Codes 22/23 – Read/Write System ID Parameters UDBB Bit Format

Table 4-27 Function Code 22/23 – Read/Write System ID Parameters UDBB Bit Descriptions

Word	Bits	Field	Description
UDBB+0	⟨15:00⟩	VC⟨15:00⟩	Word0 of the Boot verification code.
UDBB+2	⟨15:00⟩	VC⟨31:16⟩	Word1 of the Boot verification code.
UDBB+4	⟨15:00⟩	VC⟨47:32⟩	Word2 of the Boot verification code.
UDBB+6	⟨15:00⟩	VC⟨63:48⟩	Word3 of the Boot verification code.
			Written by the DEUNA for read function; written by the port-driver for a write function. The DEUNA default value of the verification code is VC⟨63:00⟩ = 0.
UDBB+10	⟨15:08⟩	MBZ	Zeros.
UDBB+10	⟨07:00⟩	SOFTID	Software Identification – Written by the DEUNA for read function; written by the port-driver during a write function. The DEUNA default value is SOFTID = 0.
UDBB+12	⟨15:00⟩		Undefined
UDBB+14	⟨15:00⟩		Undefined
UDBB+16	⟨15:00⟩		Undefined
UDBB+20	⟨15:00⟩		Undefined
UDBB+22	⟨15:00⟩		Undefined
UDBB+24	⟨15:00⟩		Undefined
UDBB+26	⟨15:00⟩	TYPE	ETHERNET Type – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is 260 hex.
UDBB+30	⟨15:00⟩	CCOUNT	Character Count – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is COUNT = 28 decimal.
UDBB+32	⟨15:00⟩	MBZ	Zeros

Table 4-27 Function Code 22/23 – Read/Write System ID Parameters UDBB Bit Descriptions (Cont)

Word	Bits	Field	Description
UDBB + 32	⟨07:00⟩	CODE	Code – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is CODE = 7.
UDBB + 34	⟨15:00⟩	RECNUM	Receipt number – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is RECNUM = 0.
UDBB + 36	⟨15:00⟩	MVTYPE	MOP Version Type – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is MVTYPE = 1.
UDBB + 40	⟨15:08⟩	MVVER	MOP Version/Version – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is MVVER = 3.
UDBB + 40	⟨07:00⟩	MVLEN	MOP Version Length – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is MVLEN = 3.
UDBB + 42	⟨15:08⟩	MVUECO	MOP Version User ECO – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is MVUECO = 0.
UDBB + 42	⟨07:00⟩	MVECO	MOP Version ECO – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is MVECO = 0.
UDBB + 44	⟨15:00⟩	FTYPE	Function Type – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is FTYPE = 2.

Table 4-27 Function Code 22/23 – Read/Write System ID Parameters UDBB Bit Descriptions (Cont)

Word	Bits	Field	Description
UDBB+46	⟨15:08⟩	FVAL1	Function value 1 – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is FVAL1 = 5.
UDBB+46	⟨07:00⟩	FLEN	Function Length – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is FLEN = 2.
UDBB+50	⟨15:08⟩	HATYPE ⟨07:00⟩	Byte 0 of the Hardware Address Type – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is HATYPE = 7.
UDBB+50	⟨07:00⟩	FVAL2	Function Value 2 – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is FVAL2 = 0.
UDBB+52	⟨15:08⟩	HALEN	Hardware Address Length – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is HALEN = 6.
UDBB+52	⟨07:00⟩	HATYPE ⟨15:08⟩	Byte 1 of the Hardware Address Type – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is HATYPE = 0.
UDBB+54	⟨15:00⟩	HA⟨15:00⟩	Word0 of the Hardware Address
UDBB+56	⟨15:00⟩	HA⟨31:16⟩	Word1 of the Hardware Address
UDBB+60	⟨15:00⟩	HA⟨47:32⟩	Word2 of the Hardware Address
			Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is the default physical address.

Table 4-27 Function Code 22/23 – Read/Write System ID Parameters UDBB Bit Descriptions (Cont)

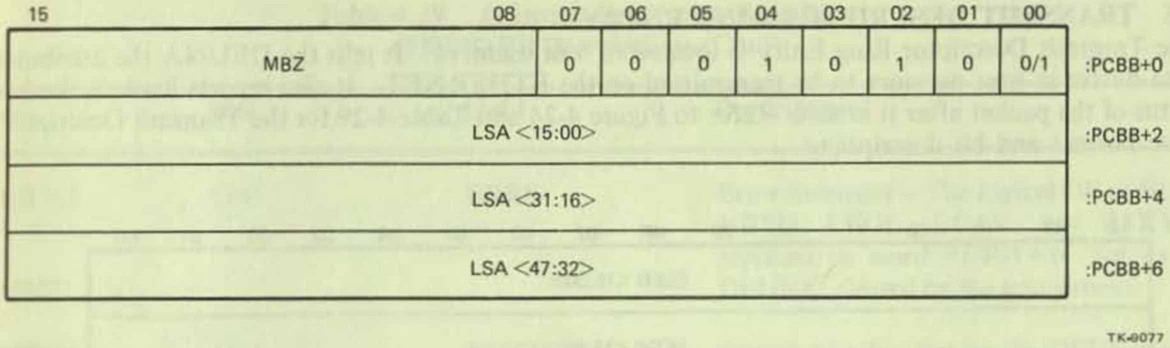
Word	Bits	Field	Description
UDBB + 62	⟨15:00⟩	DTYPE	Device Type – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is DTYPE = 64 hex.
UDBB + 64	⟨15:08⟩	DVALUE	Device Value – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is DVALUE = 1.
UDBB + 64	⟨08:00⟩	DLEN	Device Length – Written by the DEUNA for a read function; written by the port-driver for a write function. The DEUNA default value is DLEN = 1.
UDBB + 66	⟨15:00⟩	PARAM	Additional Parameters – Written by the DEUNA for a read function; written by the port-driver for a write function.
Port Driver Checks			
None			

4.4.13 Function Codes 24/25 – Read/Write Load Server Address

Function codes 24 and 25 read or change the Load Server Address used by the DEUNA when in the Primary Load State (refer to Section 4.10.2.4). If no write function occurs prior to being issued a Read function, the DEUNA will return the Load Server Multicast address (AB-00-00-01-00-00 hex). Refer to Figure 4-23 and Table 4-28 for PCBB bit format and bit descriptions.

NOTE

In this Chapter the hex value of the multi-byte fields will be shown in parentheses (0123) and then the order of transmission is shown following 23-01 hex with 23 being the least significant byte. The least significant bit of the least significant byte (23) is transmitted first.



15 … 08	07	06	05	04	03	02	01	00	
MBZ	0	0	0	1	0	1	0	0/1	:PCBB+0
LSA <15:00>									:PCBB+2
LSA <31:16>									:PCBB+4
LSA <47:32>									:PCBB+6

Figure 4-23 Function Codes 24/25 – Read/Write Load Server Address PCBB Bit Format

Table 4-28 Function Code 24/25 – Read/Write Load Server Address PCBB Bit Descriptions

Word	Bits	Field	Description
PCBB + 0	⟨15:08⟩	MBZ	Must be zero. Written by the port-driver; unchanged by the DEUNA.

PCBB + 0	⟨07:00⟩	OPCODE	OPCODE = 24 – Read Load Server Address.
			OPCODE = 25 – Write Load Server Address.
			Written by the port-driver; unchanged by the DEUNA.

PCBB + 2	⟨15:00⟩	LSA ⟨15:00⟩	The low order 16 address bits of the load server address.
PCBB + 4	⟨15:00⟩	LSA ⟨31:16⟩	The middle order 16 address bits of the load server address.
PCBB + 6	⟨15:00⟩	LSA ⟨47:32⟩	The high order 16 address bits of the load server address. Written by the DEUNA for a read function. Written by the port-driver for a write function.

Port Driver Checks	Resultant Error
(PCBB + 0)⟨15:08⟩ = 0	Function Error

4.5 TRANSMIT DESCRIPTOR RING ENTRY

The Transmit Descriptor Ring Entry is located in host memory. It tells the DEUNA the attributes of a data buffer in host memory to be transmitted on the ETHERNET. It also reports back to the host the status of the packet after it is sent. Refer to Figure 4-24 and Table 4-29 for the Transmit Descriptor Ring Base Format and bit descriptions.

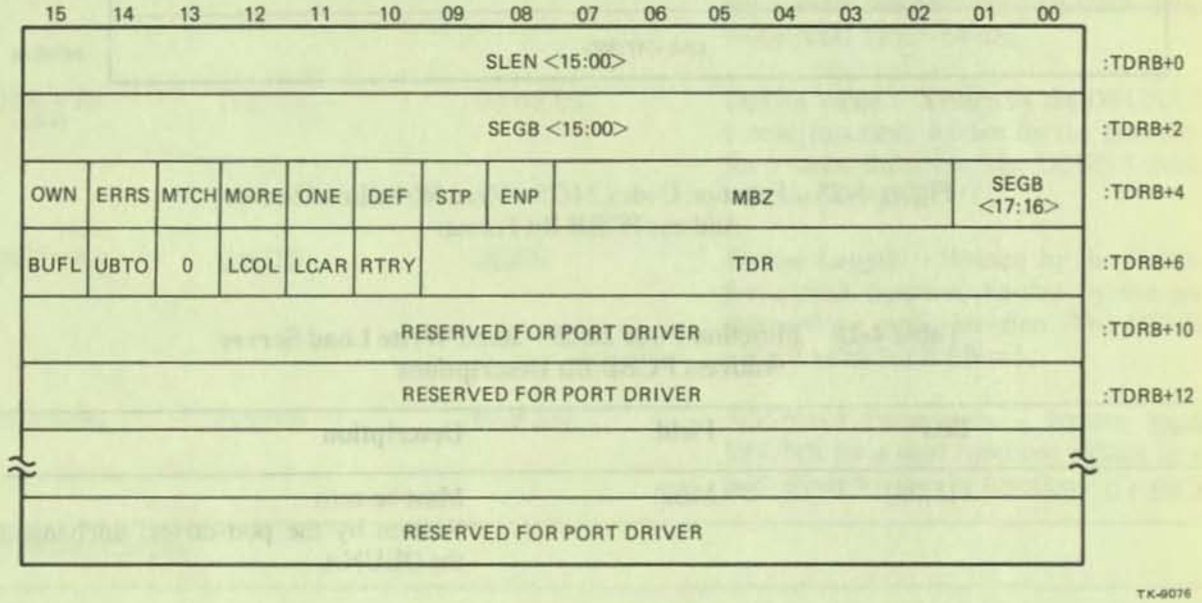


Figure 4-24 Transmit Descriptor Ring Entry Format

Table 4-29 Transmit Descriptor Ring Base (TDRB) Bit Descriptions

Word	Bits	Field	Description
TDRB + 0	⟨15:00⟩	SLEN	Segment Length – Number of bytes in a segment. Illegal if the number of bytes in the transmitted data field is less than 64 or greater than 1518 unless TPAD is enabled for a message less than 64 bytes (refer to Table 4-22). Set by the port-driver; unchanged by the DEUNA.
TDRB + 2	⟨15:00⟩	SEGB	The low order 16 address bits of the segment pointed to by the descriptor. Written by the port-driver; unchanged by the DEUNA.
TDRB + 4	⟨15⟩	OWN	Port ownership – Indicates that the descriptor entry is owned by the port-driver (= 0) or by the DEUNA (= 1). Set by the port-driver; cleared by the DEUNA.

Table 4-29 Transmit Descriptor Ring Base (TDRB) Bit Descriptions (Cont)

Word	Bits	Field	Description
TDRB+4	⟨14⟩	ERRS	Error Summary – The logical OR of BUFL, UBTO, LCOL, LCAR, and RTRY as reported in word TDRB+6. Set by the DEUNA; cleared by the port-driver.
TRDB+4	⟨13⟩	MTCH	Station Match – Set by the DEUNA when the destination address of the transmit message matches one of the addresses of the DEUNA.
TDRB+4	⟨12⟩	MORE	Multiple retries needed. Set by the DEUNA when more than one retry was needed to successfully transmit a packet; cleared by the port-driver.
TDRB+4	⟨11⟩	ONE	One Collision – Set by the DEUNA when exactly one retry was needed to transmit a packet; cleared by the port-driver.
TDRB+4	⟨10⟩	DEF	Deferred – Set when the DEUNA experienced no collisions but had to defer while trying to transmit a packet; cleared by the port-driver.
TDRB+4	⟨9⟩	STP	Start of packet – Set by the port-driver; unchanged by the DEUNA. Used for intra-packet data chaining.
TDRB+4	⟨8⟩	ENP	End of packet – Set by the port-driver; unchanged by the DEUNA. Used for intra-packet data chaining.
TDRB+4	⟨07:02⟩	MBZ	Must be zero.
TDRB+4	⟨01:00⟩	SEGB	The high order two address bits of the segment pointed to by the descriptor. Written by the port-driver; unchanged by the DEUNA.

Table 4-29 Transmit Descriptor Ring Base (TDRB) Bit Descriptions (Cont)

Word	Bits	Field	Description
TDRB+6	⟨15⟩	BUFL	Buffer length error – One or more of the following conditions: 1. The total length of the packet, including chained buffers, is less than the length of the minimum allowable packet length. This is 14 bytes if the DEUNA is in the data padding mode (TPAD=1). If the DEUNA is not in the data padding mode, the minimum length is 64 bytes if the DEUNA is not in the disable transmit CRC mode, or 60 bytes if the DEUNA is in the disable transmit CRC mode (DTCR=1). The BUFL bit is set in the transmit descriptor ring entry in which the packet length overflowed. 2. THe total length of the packet, including chained buffers, exceeds the length of the maximum allowable packet length; 1514 bytes if the DEUNA is not in the disable transmit CRC mode, or 1518 bytes if the DEUNA is in the disable transmit CRC mode (DTCR=1). The BUFL bit is set in the transmit descriptor ring entry in which the packet length overflowed. 3. While searching the ring to find the beginning of a packet to be transmitted, the BUFL bit is set in each transmit descriptor ring entry it owns but does not have the STP bit set while DEUNA is searching for an STP flag.

Table 4-29 Transmit Descriptor Ring Base (TDRB) Bit Descriptions (Cont)

Word	Bits	Field	Description
			4. While in the data chaining mode, if the DEUNA found an entry it owned with the STP bit set, or encountered a buffer it did not own while searching for an entry in which the ENP bit was set. The BUFL bit is set in the transmit descriptor ring entry before the entry DEUNA does not own; BUFL is set in the last entry the DEUNA owns. 5. While in the data chaining mode, if the DEUNA found an entry it owned with the STP bit set, or encountered an entry with the STP bit set while searching for an entry in which the ENP flag was set. The BUFL bit is set in the transmit descriptor ring entry before the entry containing the asserted STP flag. Packet transmission does not occur if BUFL is set for one or more of the buffers that make up the packet. Set by the DEUNA; cleared by the port-driver.
TDRB + 6	⟨14⟩	UBTO	UNIBUS timeout – A UNIBUS timeout was encountered while accessing the buffer pointed to by the descriptor ring entry. (Refer to Section 4.9.8.) Set by the DEUNA; cleared by the port-driver.
TDRB + 6	⟨13⟩	Zero	
TDRB + 6	⟨12⟩	LCOL	Late collision – A collision has occurred after the slot time of the channel has elapsed. Set by the DEUNA; cleared by the port-driver.

Table 4-29 Transmit Descriptor Ring Base (TDRB) Bit Descriptions (Cont)

Word	Bits	Field	Description
TDRB + 6	⟨11⟩	LCAR	Loss of carrier – Carrier was either not present on the channel during transmission (indicating a shorted cable) or the carrier was lost during transmission of a broken carrier detect circuit. Set by the DEUNA; cleared by the port-driver.
TDRB + 6	⟨10⟩	RTRY	Retry – Transmitter has failed in 16 attempts to transmit the packet due to collisions on the medium. Set by the DEUNA; cleared by the port-driver.
TDRB + 6	⟨9:0⟩	TDR	Time domain reflectometry value – Valid only when RTRY is set.

Port Driver Checks	Resultant Error
(TRDB + 4)⟨07:02⟩ = 0	Ring Error
TPAD = 1, DTCR = 0 14 ≤ packet length ≤ 1514	Ring Error
TPAD = 0, DTCR = 0 60 ≤ packet length ≤ 1514	Ring Error
TPAD = 0, DTCR = 1 64 ≤ packet length ≤ 1518	Ring Error

4.6 RECEIVE DESCRIPTOR RING ENTRY

The Receive Descriptor Ring is located in host memory. It tells the DEUNA where to put received messages and reports the status of those messages to the port-driver. Refer to Figure 4-25 and Table 4-30 for bit format and bit descriptions.

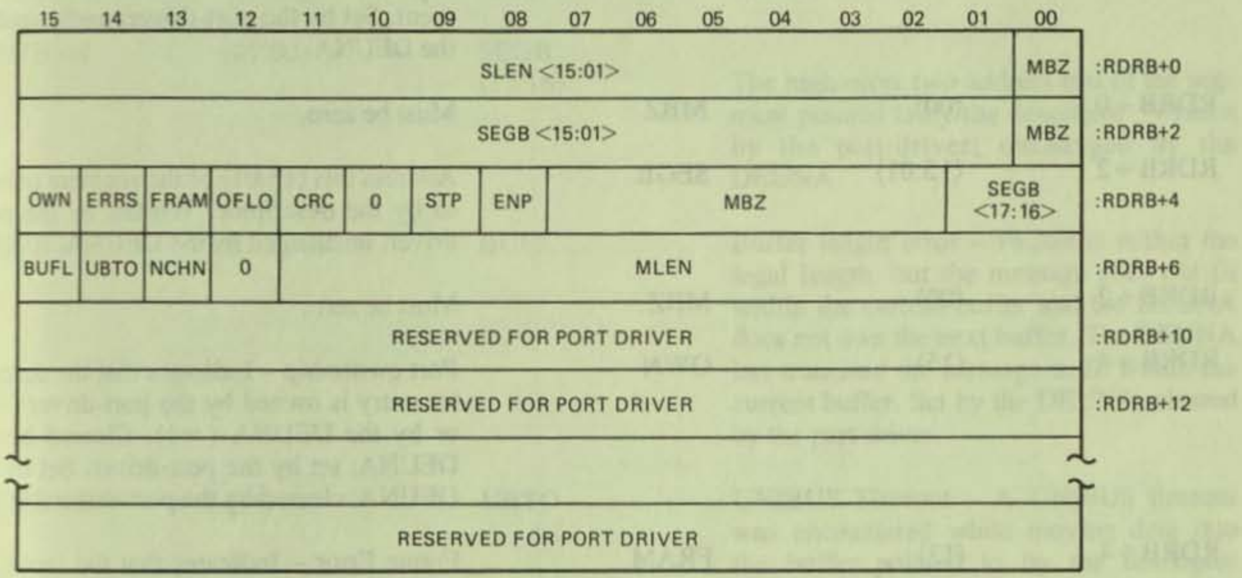


Figure 4-25 Receive Descriptor Ring Entry Format

Table 4-30 Receive Descriptor Ring Entry Bit Descriptions

Word	Bits	Field	Description
RDRB+0	⟨15:01⟩	SLEN	Segment length – Number of bytes in a segment. Set by the port-driver; unchanged by the DEUNA.
RDRB+0	⟨00⟩	MBZ	Must be zero.
RDRB+2	⟨15:01⟩	SEGB	Address bits ⟨15:01⟩ of the segment pointed to by the descriptor. Written by the port-driver; unchanged by the DEUNA.
RDRB+2	⟨00⟩	MBZ	Must be zero.
RDRB+4	⟨15⟩	OWN	Port ownership – Indicates that the descriptor entry is owned by the port-driver (=0) or by the DEUNA (=1). Cleared by the DEUNA; set by the port-driver. Set by the DEUNA; cleared by the port-driver.
RDRB+4	⟨13⟩	FRAM	Frame Error – Indicates that the incoming packet contains a non-integer multiple of eight bits. Set by the DEUNA; cleared by the port-driver.
RDRB+4	⟨12⟩	OFLO	Message Overflow – The message in the buffer is longer than the maximum allowable ETHERNET packet. Data chaining was not attempted; the message was truncated to fit in the buffer. Set by the DEUNA; cleared by the port-driver.
RDRB+4	⟨11⟩	CRC	Cyclical Redundancy Check – Frame check error, data is not valid. This bit is not valid for maintenance operations with the DTCR not set and loopback set, because the CRC value is not checked during receive. Set by the DEUNA, cleared by the port-driver.
RDRB+4	⟨10⟩	Zero	
RDRB+4	⟨9⟩	STP	Start of packet – Set by the DEUNA; unchanged by the port-driver. Used for intra-packet data chaining.
RDRB+4	⟨8⟩	ENP	End of Packet – Set by the DEUNA; unchanged by the port-driver. Used for intra-packet data chaining.

Table 4-30 Receive Descriptor Ring Entry Bit Descriptions (Cont)

Word	Bits	Field	Description
RDRB+4	⟨07:02⟩	MBZ	Must be zero.
RDRB+4	⟨01:00⟩	SEGB ⟨17:16⟩	The high order two address bits of the segment pointed to by the descriptor. Written by the port-driver; unchanged by the DEUNA.
RDRB+6	⟨15⟩	BUFL	Buffer length error – Packet is within the legal length, but the message does not fit within the current buffer and the DEUNA does not own the next buffer. The DEUNA has truncated the message to fit within the current buffer. Set by the DEUNA; cleared by the port-driver.
RDRB+6	⟨14⟩	UBTO	UNIBUS Timeout – A UNIBUS timeout was encountered while moving data into the buffer pointed to by the descriptor entry. (Refer to Section 4.9.8.) Set by the DEUNA; cleared by the port-driver.
RDRB+6	⟨13⟩	NCHN	No Data Chaining – When set, indicates when set that the DEUNA was in the non-data chaining mode at the time the buffer was written. The message may be truncated to fit in the single buffer. Other status information is valid. STP and ENP are also set. Written by the DEUNA; cleared by the port-driver.
RDRB+6	⟨12⟩	Zero	
RDRB+6	⟨11:00⟩	MLEN	Message length – The length in bytes of packet placed in the buffer(s). This field is valid only in the descriptor entry the ENP flag is set in. Written by the DEUNA; cleared by the port-driver.

Port Driver Checks	Resultant Error
(RDRB+0)⟨00⟩=0	Ring error
(RDRB+2)⟨00⟩=0	Ring error
(RDRB+0)⟨07:02⟩=0	Ring error

4.7 TRANSMIT DATA BUFFER FORMAT

Transmit Data Buffers may begin on arbitrary byte boundaries. Refer to Figure 4-26 for the format of a Transmit Data Buffer starting on an even byte boundary and Figure 4-27 for a Transmit Data Buffer starting on an odd byte boundary.

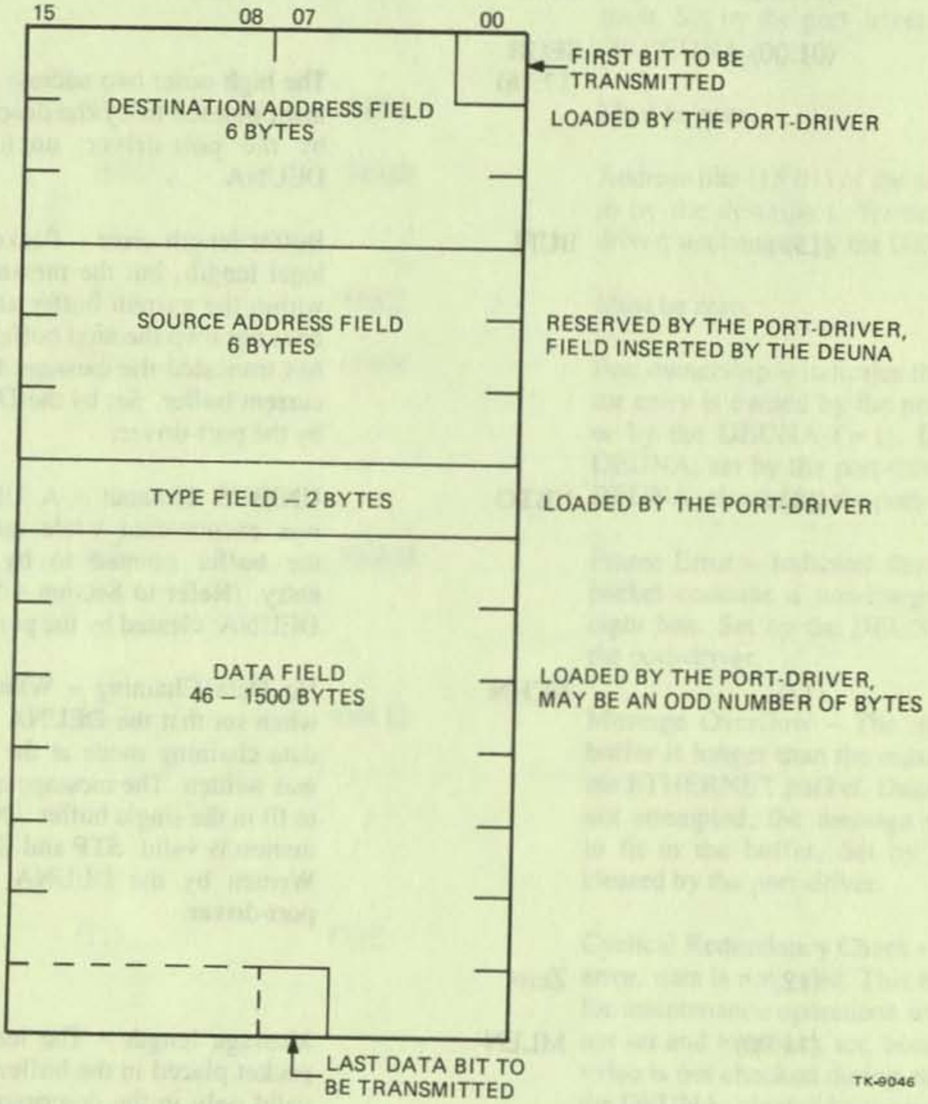

15
08
07
00
FIRST BIT TO BE TRANSMITTED
DESTINATION ADDRESS FIELD 6 BYTES
LOADED BY THE PORT-DRIVER
SOURCE ADDRESS FIELD 6 BYTES
RESERVED BY THE PORT-DRIVER, FIELD INSERTED BY THE DEUNA
TYPE FIELD – 2 BYTES
LOADED BY THE PORT-DRIVER
DATA FIELD 46 – 1500 BYTES
LOADED BY THE PORT-DRIVER, MAY BE AN ODD NUMBER OF BYTES
LAST DATA BIT TO BE TRANSMITTED
TK-8046

Figure 4-26 Transmit Data Buffer Starting on an Even Byte Boundary

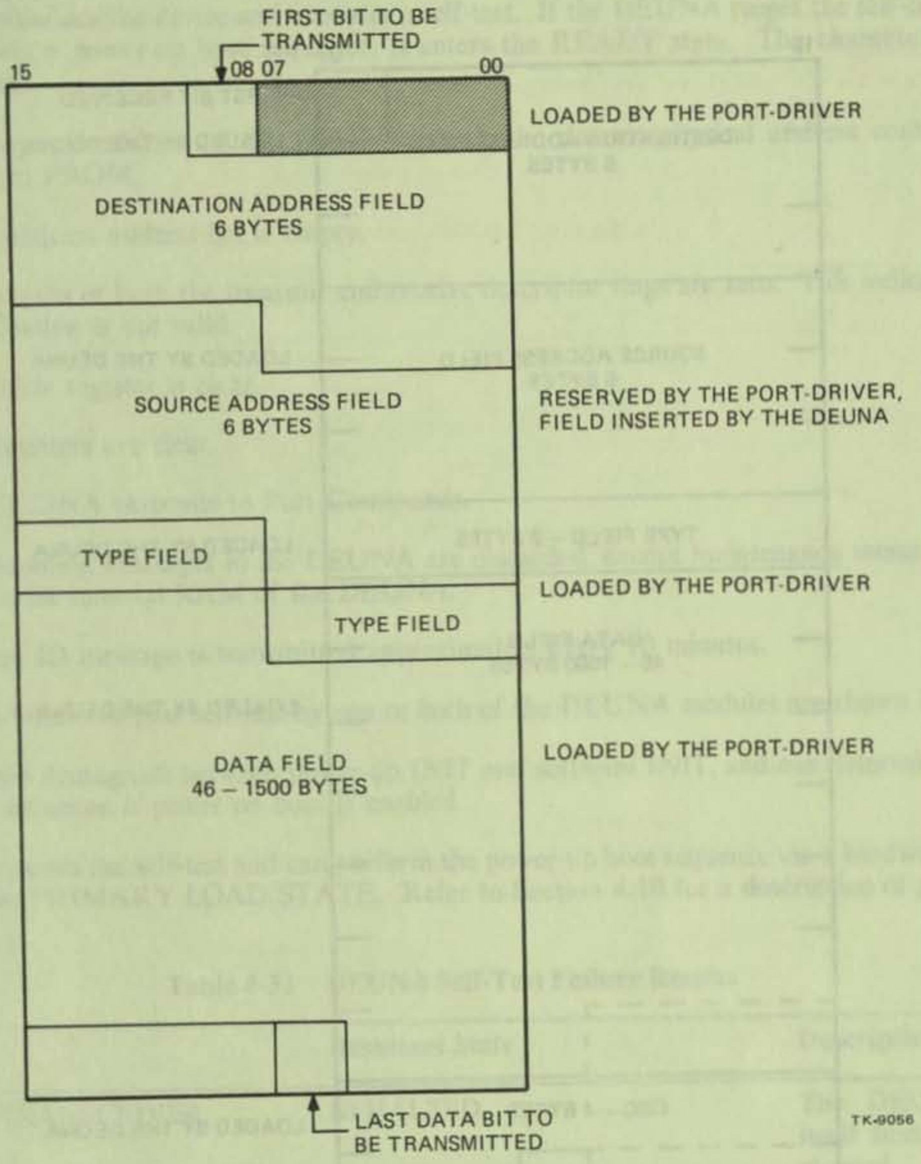

FIRST BIT TO BE TRANSMITTED
15
08 07
00
LOADED BY THE PORT-DRIVER
DESTINATION ADDRESS FIELD
6 BYTES
SOURCE ADDRESS FIELD
6 BYTES
RESERVED BY THE PORT-DRIVER, FIELD INSERTED BY THE DEUNA
TYPE FIELD
LOADED BY THE PORT-DRIVER
TYPE FIELD
LOADED BY THE PORT-DRIVER
DATA FIELD
46 – 1500 BYTES
LAST DATA BIT TO BE TRANSMITTED
TK-9056

Figure 4-27 Transmit Data Buffer Starting on an Odd Byte Boundary

4.8 RECEIVE DATA BUFFER FORMAT

Receive Data Buffers must begin on an even byte boundary. Refer to Figure 4-28 for the Receive Data Buffer Format.

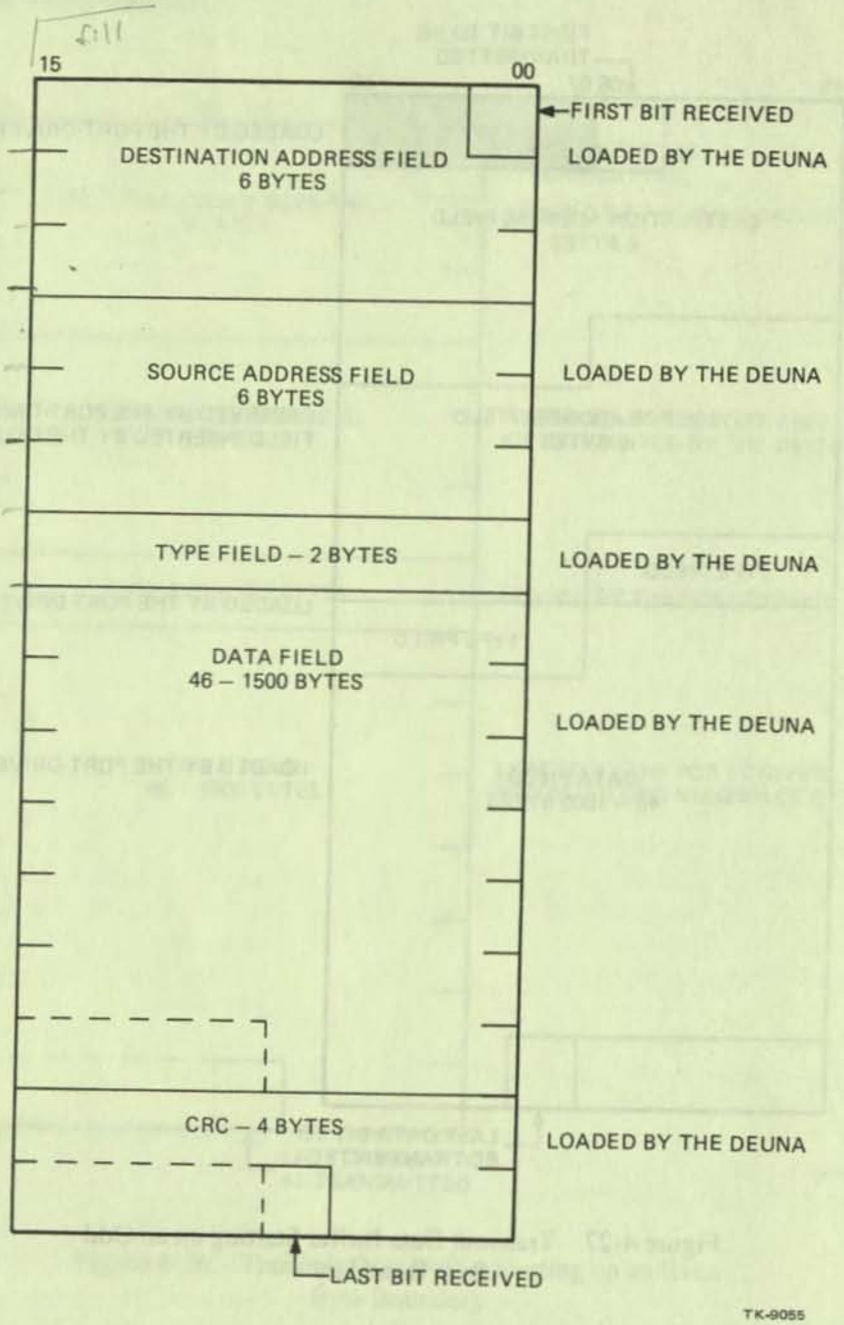


Figure 4-28 Receive Data Buffer Format

4.9 DEUNA OPERATION

4.9.1 Power On

When power is applied to the DEUNA, the device enters the RESET state. In this state, the DEUNA microprocessor initializes the device and executes a self-test. If the DEUNA passes the self-test and is not enabled to perform a power-up boot sequence, it enters the READY state. The characteristics of the READY state are:

- The physical address of the DEUNA equals the default physical address contained in the on-board PROM.
- The multicast address list is empty.
- The lengths of both the transmit and receive descriptor rings are zero. This indicates that ring specification is not valid.
- The mode register is clear.
- The counters are clear.
- The DEUNA responds to Port Commands.
- All incoming messages to the DEUNA are discarded, except maintenance messages processed within the internal RAM of the DEUNA.
- System ID message is transmitted approximately every 10 minutes.

The results of a failure to pass self-test by one or both of the DEUNA modules are shown in Table 4-31.

The DEUNA can distinguish between power-up INIT and software INIT, and can determine if it should execute a boot sequence if power on boot is enabled.

If the DEUNA passes the self-test and can perform the power-up boot sequence via a hardware switch, the device enters the PRIMARY LOAD STATE. Refer to Section 4.10 for a description of power-up boot operation.

Table 4-31 DEUNA Self-Test Failure Results

Failing Unit	Resultant State	Description
LINK/CABLE/TRANSCEIVER	NI HALTED	The DEUNA isolates itself from the physical channel.
PORT/UNIBUS	UNIBUS HALTED	The DEUNA does not become UNIBUS master.
LINK/PORT	NI and UNIBUS HALTED	The DEUNA does not access the channel or become UNIBUS master.

NOTE

If the system UNIBUS arbitor is off, self-test will fail.

4.9.2 Port Command Capability

The primary means of communication between the DEUNA and the Host processor is through the Port Command facility. Table 4-32 summarizes the DEUNA Port Commands. A more detailed description of the DEUNA Port Commands can be obtained by referring to Section 4.3.

The Port Command Operation uses three fields in PCSR0: The Done Interrupt (DNI) bit ⟨11⟩, PCEI Error Interrupt bit ⟨14⟩, and the Port Command Field bits ⟨03:00⟩. Refer to Figure 4-29 for a description of the Port Command sequence.

Table 4-32 DEUNA Port Commands

Command	Description
GET PCBB	Fetch the base address of the Port Control Block.
GET COMMAND	Fetch and execute the Port Function specified in the Port Control Block.
SELF-TEST	Enter RESET State and execute Self Test.
START	Start the Reception and Transmission Processes.
STOP	Stop the Reception and Transmission Processes.
BOOT	Boot DEUNA microcode via down-line load.
POLLING DEMAND	Poll the transmit and receive rings for a new message to transmit or a new free receive buffer.

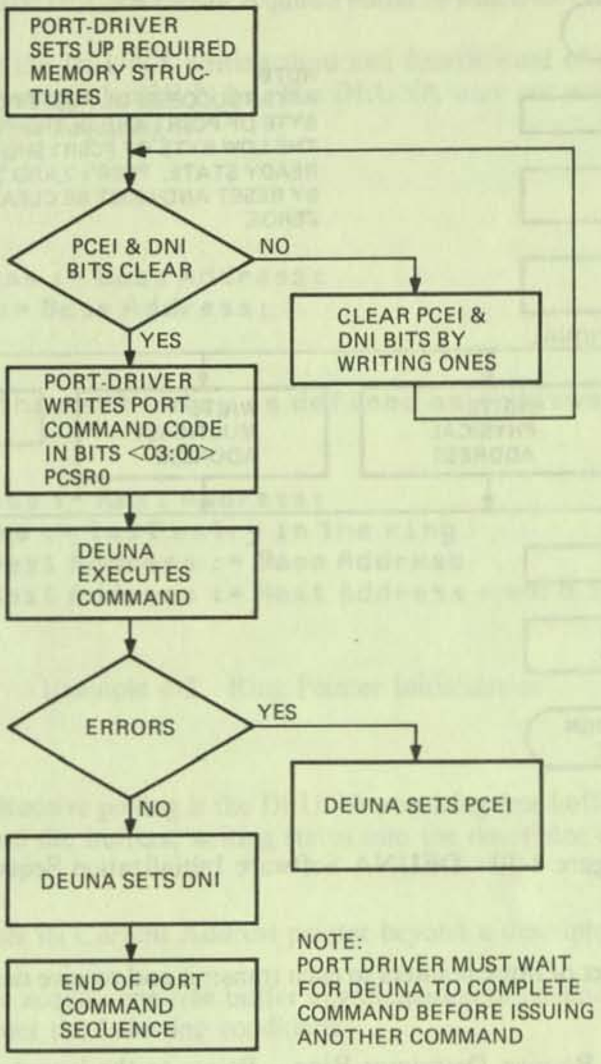

PORT-DRIVER SETS UP REQUIRED MEMORY STRUCTURES
PCEI & DNI BITS CLEAR
NO
YES
CLEAR PCEI & DNI BITS BY WRITING ONES
PORT-DRIVER WRITES PORT COMMAND CODE IN BITS <03:00> PCSR0
DEUNA EXECUTES COMMAND
ERRORS
YES
NO
DEUNA SETS PCEI
DEUNA SETS DNI
END OF PORT COMMAND SEQUENCE
NOTE:
PORT DRIVER MUST WAIT FOR DEUNA TO COMPLETE COMMAND BEFORE ISSUING ANOTHER COMMAND
TK-9518

Figure 4-29 Port Command Sequence

4.9.3 Software Initialization

A sequence of Port Commands must be issued by the port-driver to prepare the DEUNA for datagram service. See Figure 4-30 for a description of the DEUNA Initialization Sequence.

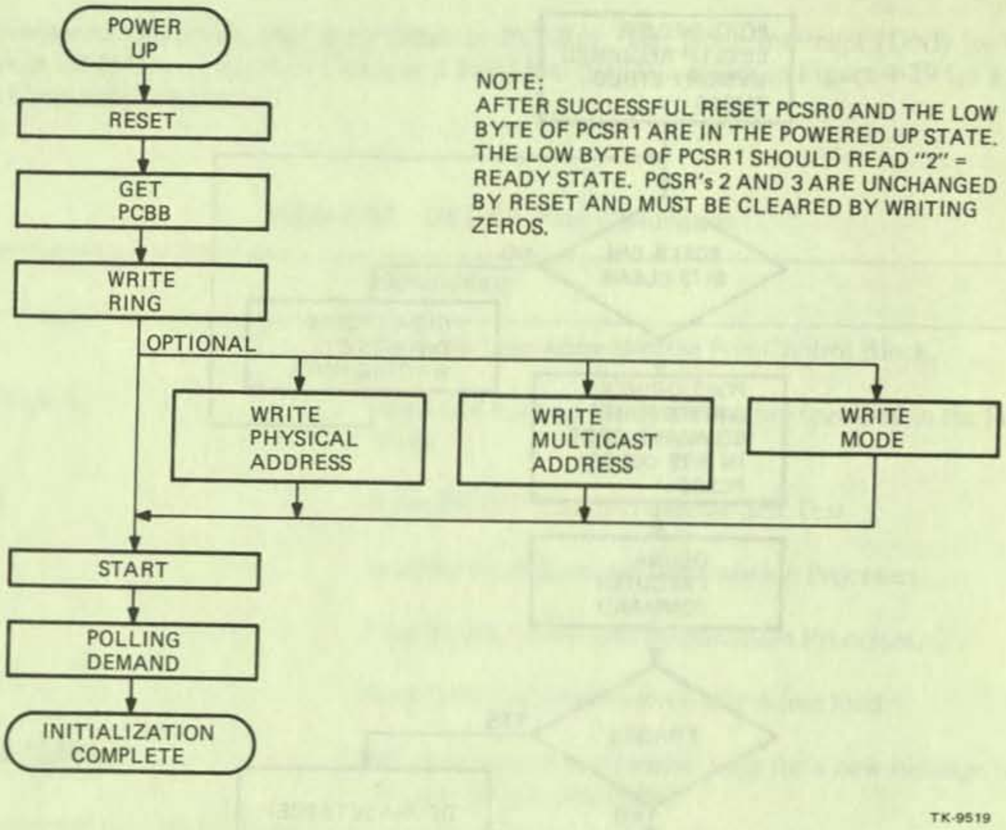

POWER UP
RESET
GET PCBB
WRITE RING
OPTIONAL
WRITE PHYSICAL ADDRESS
WRITE MULTICAST ADDRESS
WRITE MODE
START
POLLING DEMAND
INITIALIZATION COMPLETE
NOTE:
AFTER SUCCESSFUL RESET PCSR0 AND THE LOW BYTE OF PCSR1 ARE IN THE POWERED UP STATE. THE LOW BYTE OF PCSR1 SHOULD READ "2" = READY STATE. PCSR's 2 AND 3 ARE UNCHANGED BY RESET AND MUST BE CLEARED BY WRITING ZEROS.
TK-9519

Figure 4-30 DEUNA Software Initialization Sequence

4.9.4 Polling

The DEUNA maintains a set of three pointers to each transmit and receive descriptor ring. They are base, current, and next address.

- Base Address of Receive Descriptor Ring – Points to the lowest addressed receive descriptor ring entry. This pointer is a constant.
- Base Address of Transmit Descriptor Ring – Points to the lowest addressed transmit descriptor ring entry. This pointer is a constant.
- Current Address of Receive Descriptor Ring – Points to the current position in the receive ring. This pointer is a variable.
- Current Address of Transmit Descriptor Ring – Points to the current position in the transmit ring. This pointer is a variable.
- Next Address of Receive Descriptor Ring – Points to the receive descriptor entry following the Current Address pointer. This pointer is a variable.
- Next Address of Transmit Descriptor Ring – Points to the transmit descriptor entry following the Current Address pointer. This pointer is a variable.

Upon entering the RUNNING state, the current and next pointers are initialized as in Example 4-2.

Buffer acquisition is defined as the DEUNA reading the first three words of the descriptor entry: Status, Buffer Length, and Buffer Address (refer to Sections 4.5 and 4.6). If the OWN bit is set the DEUNA is said to have acquired the buffer. The DEUNA cannot acquire a buffer in which the OWN bit is clear.

Buffer release is defined as the DEUNA writing third and fourth word of a descriptor entry (refer to Sections 4.5 and 4.6) and clearing the OWN bit. The DEUNA may not write a descriptor entry or the buffer it points to without first acquiring it .

```
BEGIN
Current Address := Base Address;
Next Address := Base Address;
END;

Advancing to the next entry is defined as follows:

BEGIN
Current Address := Next Address;
IF Next Address := last entry in the ring
        THEN Next Address := Base Address
        ELSE Next Address := Next Address + word length of entry
END;
```

Example 4-2 Ring Pointer Initialization

4.9.4.1 Receive Polling – Receive polling is the DEUNA acquiring free buffers on the receive descriptor ring, writing packet data into the buffers, writing status into the descriptor entry, and advancing to the next entry on the ring.

The DEUNA never advances its Current Address pointer beyond a descriptor entry it has not acquired.

The DEUNA always tries to acquire one free buffer in anticipation of incoming messages. The DEUNA performs receive polling under the following conditions.

- Immediately after being placed in the RUNNING state.
- In response to a Polling Demand port command in the RUNNING state when the DEUNA has not acquired a free buffer.
- The DEUNA has received a message and has not acquired a free buffer.
- The DEUNA is writing a buffer pointed to by the current descriptor ring entry and has not acquired the next buffer.
- The DEUNA has written a complete message to the receive descriptor ring and has not acquired a new buffer.

If the message to be written to the receive ring is larger than the buffer the DEUNA has acquired for it, the DEUNA attempts to chain that buffer and sequential buffers together to build the message. (Buffer chaining must be enabled by writing to the mode register; see Section 4.4.8.) The STP flag is set by the DEUNA in the first descriptor entry; the ENP flag is set in the last descriptor entry to delimit the message.

While in data chaining mode, the DEUNA tries to acquire the next buffer before releasing the current buffer. In doing so, the DEUNA is guaranteed an entry in which to report status should the DEUNA run out of buffers. The DEUNA always sets the ENP flag in the last buffer it releases for a message. The DEUNA only writes status into the entry in which the ENP flag is set. (Status is only valid in the entry in which the DEUNA sets the ENP flag.) The DEUNA writes a maximum of one packet in any one buffer.

4.9.4.2 Transmit Polling – Transmit polling is the DEUNA searching the transmit ring, finding and building messages from it, and reporting the status of the attempted transmission. The DEUNA must be in the RUNNING state for it to poll. The port driver directs the DEUNA to do transmit polling by issuing the PDMD port command only. Once the DEUNA starts polling the transmit ring, it continues in sequential order until it finds an entry in which the OWN bit is clear. At that time, transmit polling is suspended until it is reinitiated by the port driver issuing the PDMD port command.

The transmit polling sequence is as follows:

1. The DEUNA is in the RUNNING state and the port driver issues the PDMD port command.

2. After a conditional poll of the receive descriptor ring, the DEUNA reads the current entry of the transmit descriptor ring. If the DEUNA is performing the first poll after entering the RUNNING state, it starts at the base address of the transmit ring.

3. If the OWN bit is not set, indicating that the DEUNA does not own the descriptor entry, the DEUNA suspends transmit polling.

4. If the OWN bit is set, but the STP bit is not set (indicating that the DEUNA owns the entry, but the entry is not the beginning of a message) the DEUNA reads data in, steps to the next descriptor entry, and tests the OWN bit.

5. If the OWN bit is set and the STP bit is set, indicating that the DEUNA has found the beginning of a message, the DEUNA reads the buffer into its internal buffer.

   - If the ENP bit is also set in the entry in which the STP bit was set, indicating that the entire message is contained in the buffer, the DEUNA attempts transmission, writes the status into the entry, and clears the ownership bit.

   - Data chaining occurs if the ENP bit is not set in the entry in which the STP bit was set. Before clearing the ownership bit of the current entry, the DEUNA looks ahead to the next entry. If the DEUNA owns the next entry, it clears the current entry. The next entry now becomes the current entry, the data in the buffer is appended to the internal DEUNA buffer, and the test for ENP is repeated.

   This procedure is repeated until the ENP flag is found or an error is encountered. Transmission does not begin until the entire message is resident in the DEUNA internal memory. After transmission is attempted, the DEUNA writes the appropriate status into the last entry it has acquired and clears the ownership bit.

- If, while in the transmit data chaining mode, the DEUNA encounters a situation that prevents acquisition of the entire message, or the message is found to be too large, the DEUNA writes status into the current entry it owns and clears the OWN bit.

6. The DEUNA repeats this procedure until it finds an entry it does not own, which causes transmit polling to cease.

4.9.5 Datagram Reception

Messages arrive at the DEUNA asynchronously. Upon receipt, the DEUNA strips the preamble as it searches for the start bit. After finding the start bit, the DEUNA compares the next six bytes against its table of addresses. If the address comparison is not successful, the DEUNA ignores the message. If the address comparison is successful, the DEUNA stores the message in internal memory. If the message is shorter than 64 bytes, the DEUNA purges internal memory and retains no status of the message. Messages longer than 64 bytes are reported to the ring descriptors.

4.9.6 Datagram Transmission

After acquiring and building a transmit packet in link memory, the DEUNA attempts transmission. The DEUNA transmits only after an interpacket gap has elapsed (during which it sees no activity on the wire). The format of the outbound data stream is given in Table 4-33.

If a collision occurs during transmission, the DEUNA aborts the transmission, performs a "collision jam," reschedules based upon the truncated binary backoff algorithm, and retransmits. The DEUNA will attempt up to 16 transmissions per message.

Table 4-33 Format of an ETHERNET Data Packet

Message Part	Length (bytes)	Source
Preamble/Start bit	8	DEUNA
Destination Address	6	Data Buffer
Source Address	6	DEUNA
Type	2	Data Buffer
Data	46-1500	Data Buffer
CRC	4	DEUNA (optional)

4.9.7 Parameter Alteration

The DEUNA responds to all Port Functions in the READY state. The ability of the DEUNA to respond to Port Functions while in the RUNNING state varies with the specific function. Table 4-34 summarizes the impact of the DEUNA executing Port Functions while in the RUNNING state.

Table 4-34 RUNNING State Parameter Alteration Impact Summary

Function Code	Function Description	Impact*
0	No Operation	None
1	Load and Start Microaddress	Port
2	Read Default Physical Address	None
3	No Operation	None
4	Read Physical Address	None
5	Write Physical Address	Link
6	Read Multicast Address List	None
7	Write Multicast Address List	Link
10	Read Ring Format	None
11	Write Ring Format	Port
12	Read Counters	None
13	Read and Clear Counters	None
14	Read Mode	None
15	Write Mode	Link
16	Read Port Status	None
17	Clear Port Status	None
20	Dump Internal Memory	None
21	Load Internal Memory	Port
22	Read Load Server Address	None
23	Write Load Server Address	None
24	Read System ID Parameters	None
25	Write System ID Parameters	None

* Impact:

1. None – There is no disturbance to the receive or transmit packet throughput. Response to this Port Function is solely a matter of microprocessor workload.

2. Link – Response to these Port Functions require the DEUNA to temporarily disengage from the NI while the Link is being modified.

   - Reception – All message activity during the Link modification time is ignored. Messages that completed prior to Link modification time and resident in the DEUNA internal packet buffers are not discarded during Link modification.

   - Transmission – Any message being currently transmitted completes before the DEUNA disengages the link.

3. Port – The DEUNA should not be issued these Port Functions while in the RUNNING state. The DEUNA will execute a No Operation if issued one of these functions in the RUNNING state and set the PCEI bit of PCSR0.

4.9.8 Suspension of Operation – Port Command

Suspension of DEUNA operation occurs when the DEUNA is issued a STOP Port Command while in the RUNNING state. When the DEUNA receives a STOP Port Function:

1. Any single transmission scheduled from the descriptor ring in the process of transmission is allowed to complete.
2. All descriptor ring and buffer reads and writes stop.
3. All incoming packets are discarded, except for maintenance messages.

> NOTE
>
> **While the DEUNA is in the running state, datagram service is suspended for the following UNIBUS error conditions.**
>
> - **Port Command UNIBUS Timeout**
> - **Transmit Ring Error UNIBUS Timeout**
> - **Receive Ring Error UNIBUS Timeout**
>
> **Before restarting datagram service, the port driver must remove the DEUNA from the running state by issuing a STOP port command.**

4.9.9 Restart of Operation

The DEUNA operation restarts when the DEUNA is issued a START Port Command following a STOP Port Command. If no Port Functions have been executed which alter the internal state of the DEUNA, the following parameters remain intact from suspension to restart.

- Physical Address
- Multicast Address List
- Ring Format
- Mode
- Counters
- Status Register

The DEUNA retains no state information about descriptor ring entries; it owns no buffers after a restart. The Current Address pointers are set to the Base Addresses of the rings after a restart.

4.9.10 DEUNA States

4.9.10.1 DEUNA State Related Functions – The DEUNA functions may be summarized as follows.

- **Command Response** – The ability of the DEUNA to receive and execute Port Commands from the UNIBUS conductor.

- **Datagram Service** – The ability of the DEUNA to transmit and receive packets between the NI and the buffers in UNIBUS memory using ring structures for communication between the DEUNA and the Host CPU.

- **Counters** – The ability of the DEUNA to maintain counter information relating to the activity on the NI.

- **Loop Service** – The ability of the DEUNA to receive and transmit special Loop packets independent of the port-driver.

- **Remote Console** – The ability of the DEUNA to recognize the Request ID and Boot message and to generate the System ID message independent of the port-driver. The Boot message is honored only if the DEUNA is Remote Boot Enabled.

- **Down-Line Load Service** – The ability of the DEUNA to generate the Program Request message and recognize the Memory Load with Transfer Address message independent of the port-driver.

Table 4-35 summarizes the functions enabled in DEUNA states.

Table 4-35 DEUNA State Function Summary

STATE	Command Response	Datagram Service	Counters	Loop Service	Remote Console	Down-Line Load Service
RESET	D	D	D	D	D	D
PRIMARY LOAD	D	D	E	E	E	E
READY	E	D	E	E	E	E
RUNNING	E	E	E	E	E	E
UNIBUS HALTED	D	D	E	E	E	E
NI HALTED	E	D	D	D	D	D
NI AND UNIBUS HALTED	D	D	D	D	D	D

D = FUNCTION DISABLED
E = FUNCTION ENABLED

4.9.10.2 DEUNA State Transition – Table 4-36 summarizes the events that cause the DEUNA to make a transition from one state to another.

Table 4-36 DEUNA State Transition

From State	Transition Event	To State
	Power up	Reset State
Reset State	Self-Test Successful	Ready State
	Power Up Flag set and Remote Boot Enable Switch set	Primary Load State
	Self-Test Failure – Link Module	NI Halted State
	Self-Test Failure – Port or Port/Link Module	NI and UNIBUS Halted State
	Bus Init, Port Driver Reset	Reset State
Primary Load State	Successful Boot	State determine by down-line loaded microcode
	Unsuccessful System Boot	Primary Load State
	Fatal UNIBUS Error	UNIBUS Halted State
	Fatal NI Error	NI Halted State
	Fatal Internal Error	NI and UNIBUS Halted State
	Bus Init, Port Driver Reset	Reset State
	Successful Communications Processor Boot	READY State
	Memory Load Timeout on Communications Processor Boot	READY State

Table 4-36 DEUNA State Transition (Cont)

From State	Transition Event	To State
Ready State	Start Command	Running State
	Boot Command, Boot Message, and Remote Boot Enable Switch Set	Primary Load State
	Fatal UNIBUS Error	UNIBUS Halted State
	Fatal NI Error	NI Halted State
	Fatal Internal Error	NI and UNIBUS Halted State
	Bus Init, Port Driver Reset	Reset State
Running State	Stop Command	Ready State
	Boot Command, Boot Message, and Remote Boot Enable Switch set	Primary Load State
	Fatal UNIBUS Error	UNIBUS Halted State
	Fatal NI Error	NI Halted State
	Fatal Internal Error	NI and UNIBUS Halted State
	Bus Init, Port Driver Reset	Reset State
UNIBUS Halted State	Boot Message and Remote Boot Enable Switch set	Primary Load State
	Fatal NI Error, Fatal Internal Error	NI and UNIBUS Halted State
	Bus Init, Port Driver Reset	Reset State

Table 4-36 DEUNA State Transition (Cont)

From State	Transition Event	To State
NI Halted State	Fatal UNIBUS Error, Fatal Internal Error	NI and UNIBUS Halted State
	Bus Init, Port Driver Reset	Reset State
NI and UNIBUS Halted State	Bus Init, Port Driver Reset	Reset State

4.9.10.3 DEUNA State Information Retention – Table 4-37 summarizes the state of the internal information retained or reset by the DEUNA when making a transition from one state to another.

Table 4-37 State Information Retention Summary

From State	To State(s)	Status of Internal State
Reset State	Primary Load State, Ready State, UNIBUS Halted State, NI Halted State	Reset: Ring Formats Counters Physical Address Multicast Address List Mode Register Status Register Ring pointers Internal Memory Load Server Address System ID PCSR(s)
Primary Load State		
State information retained is a function of the down-line loaded microcode that is executing.		
Ready State	Primary Load State, Running State, UNIBUS Halted State, NI Halted State	Retained: Ring Formats Counters Physical Address Multicast Address List Mode Register Status Register Ring pointers Internal Memory Load Server Address System ID PCSRs Reset: Ring Pointers

Table 4-37 State Information Retention Summary (Cont)

From State	To State(s)	Status of Internal State
Running State	Primary Load State, Ready State, UNIBUS Halted State, NI Halted State	Retained: Ring Formats Counters Physical Address Multicast Address List Mode Register Status Register Ring pointers Internal Memory Load Server Address System ID PCSRs
UNIBUS Halted State	Primary Load State, NI and UNIBUS Halted State	Retained: Ring Formats Counters Physical Address Multicast Address List Mode Register Status Register Internal Memory Load Server Address System ID PCSRs
NI Halted State	NI and UNIBUS Halted State	Retained: Ring Formats Counters Physical Address Multicast Address List Mode Register Status Register Internal Memory Load Server Address System ID PCSRs

4.10 EXCEPTIONAL OPERATIONS

4.10.1 Channel Loopback

The ROM-based microcode of the DEUNA supports Channel Loopback independent of the port-driver. Loopback messages are recognized by the DEUNA as having the unique Loopback value in the type field and either the physical address of the DEUNA or the broadcast address in the destination address field. Refer to Figure 4-31 and Table 4-38 for the Loopback Message format and description. Messages with multicast addresses other than broadcast are not checked by the DEUNA for the Loopback type. They are treated as normal datagrams in the Running State only.

There are two types of Loopback messages: Forward and Reply. The Loopback type is determined by the Function field within the message header.

- **Forward** – Forward messages are transmitted by the DEUNA, but are not placed on the receive descriptor ring.

- **Reply** – Reply messages are placed on the receive descriptor ring, but are not transmitted.

Refer to Figure 4-32 for a detailed description of DEUNA Loopback processing.

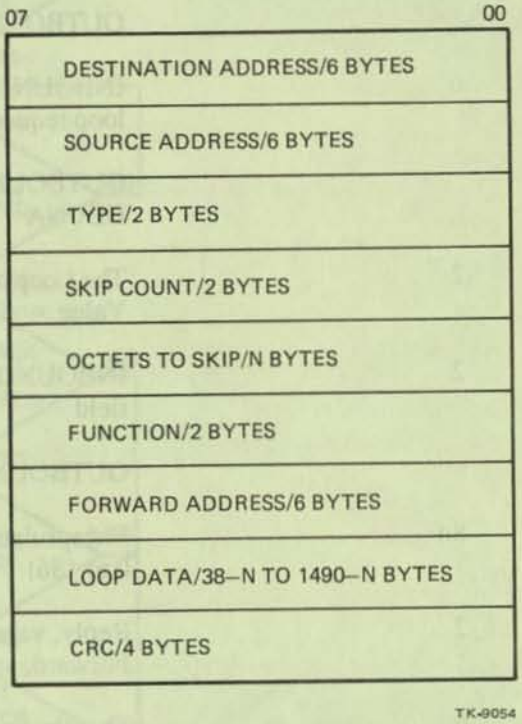


Figure 4-31 Loop Message Format

Table 4-38 Loopback Message Field Descriptions

Field	Length (Bytes)	Description
DESTINATION ADDRESS	6	INBOUND – The physical address of the DEUNA, or the broadcast address OUTBOUND – The forward address
SOURCE ADDRESS	6	INBOUND – The physical address of the loop requesting station OUTBOUND – The physical address of the DEUNA
TYPE	2	The Loop test message type Value = (0060) 60-00 hex
SKIP COUNT	2	INBOUND – The offset of the Function field OUTBOUND – The offset plus 8
OCTETS TO SKIP	8n	Encapsulated loop header information (n = 0 to 186)
FUNCTION	2	Reply, value = (0001) 01-00 hex Forward, value = (0002) 02-00 hex
FORWARD ADDRESS	6	The physical address the inbound message is to be sent to (This field does not exist for a reply message.)
LOOP DATA	36 to 1490-8n	The Loop test data
CRC	4	INBOUND – Block check character OUTBOUND – DEUNA appended block check character

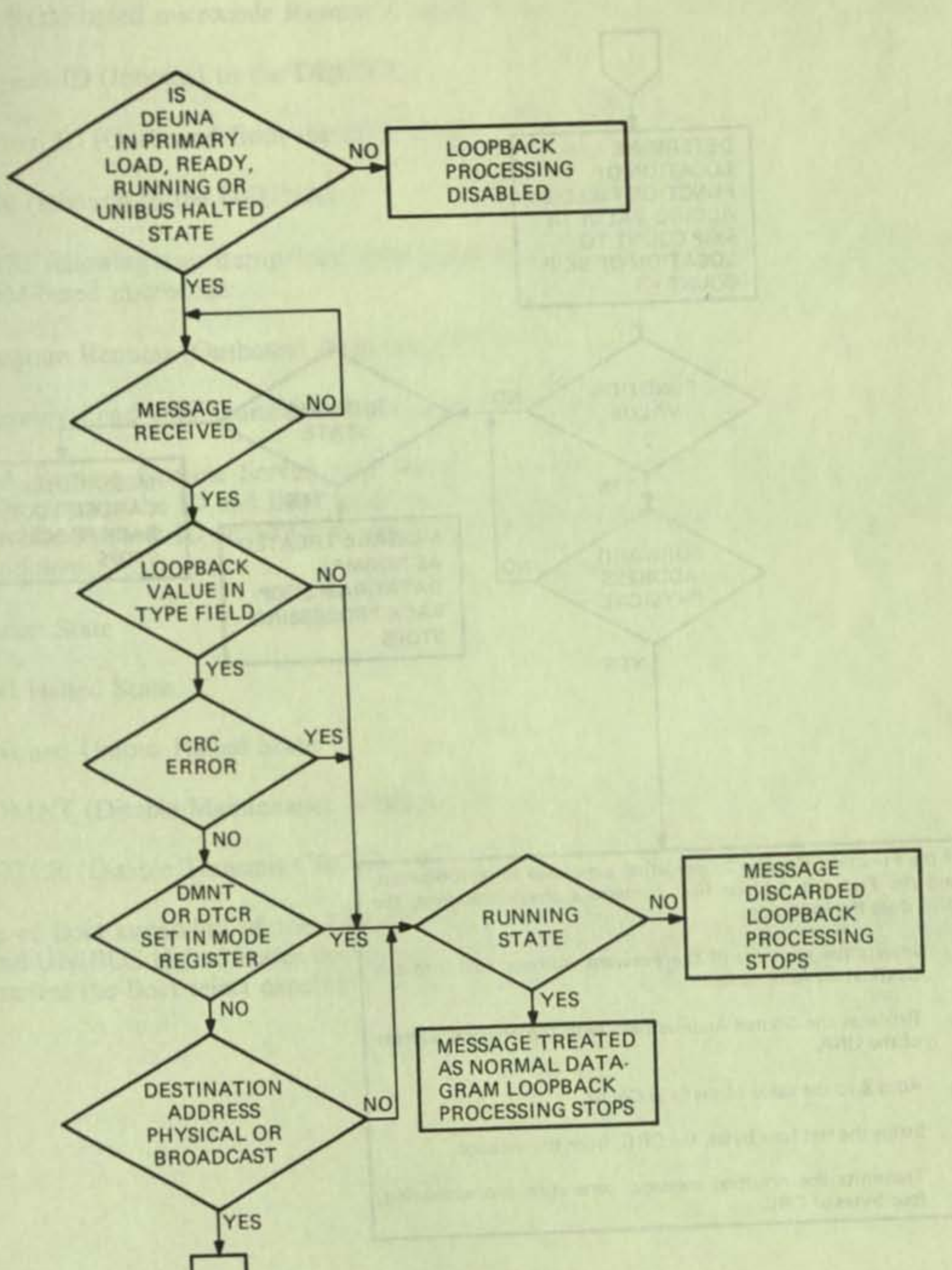
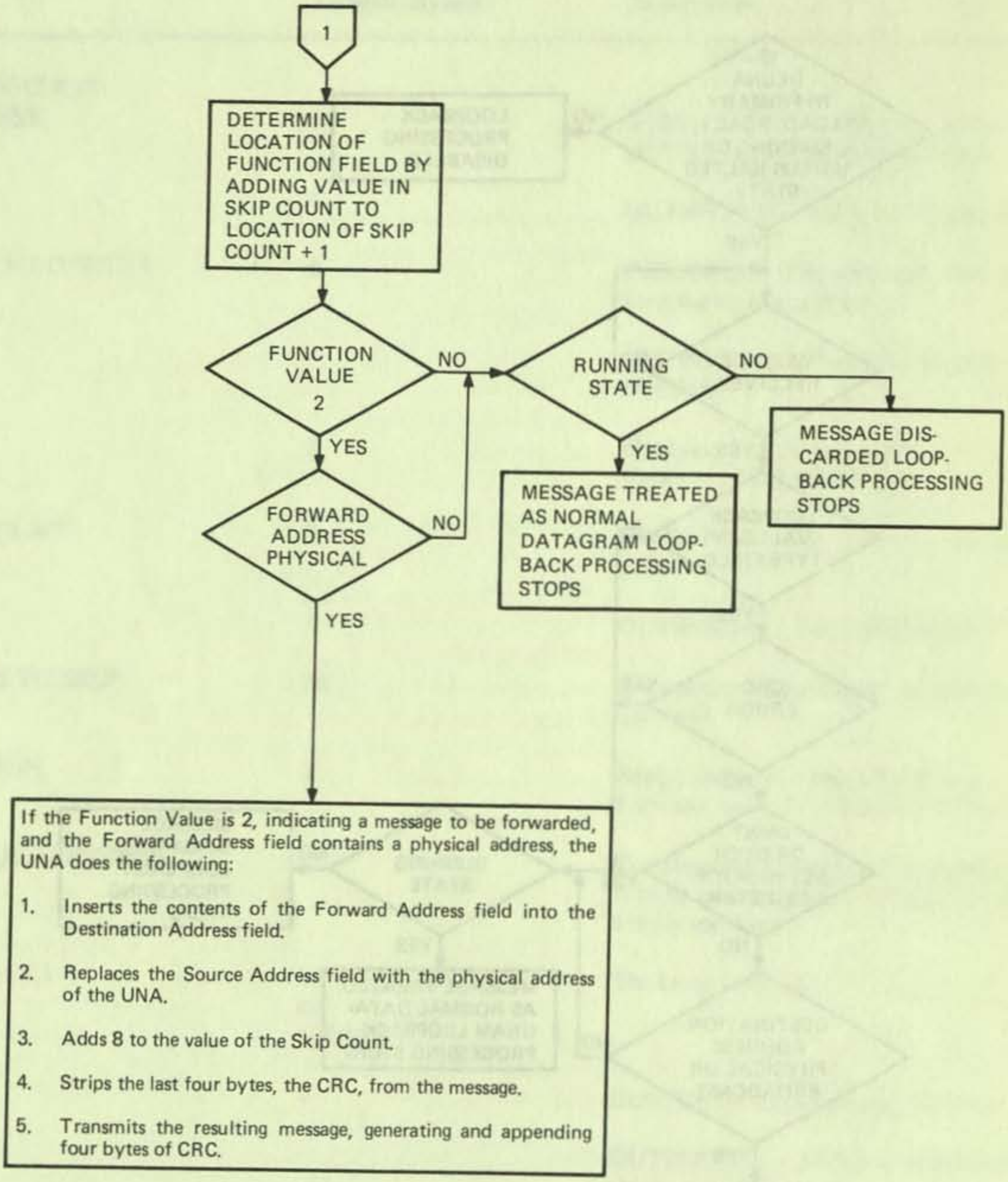

IS DEUNA IN PRIMARY LOAD, READY, RUNNING OR UNIBUS HALTED STATE
NO
LOOPBACK PROCESSING DISABLED
YES
MESSAGE RECEIVED
NO
YES
LOOPBACK VALUE IN TYPE FIELD
NO
YES
CRC ERROR
YES
NO
DMNT OR DTCR SET IN MODE REGISTER
YES
NO
RUNNING STATE
NO
MESSAGE DISCARDED LOOPBACK PROCESSING STOPS
YES
MESSAGE TREATED AS NORMAL DATAGRAM LOOPBACK PROCESSING STOPS
DESTINATION ADDRESS PHYSICAL OR BROADCAST
NO
YES
1
TK-9517

Figure 4-32 Loopback Message Processing Flow (Sheet 1 of 2)


1
DETERMINE LOCATION OF FUNCTION FIELD BY ADDING VALUE IN SKIP COUNT TO LOCATION OF SKIP COUNT + 1
FUNCTION VALUE 2
NO
YES
FORWARD ADDRESS PHYSICAL
NO
YES
RUNNING STATE
NO
YES
MESSAGE TREATED AS NORMAL DATAGRAM LOOP-BACK PROCESSING STOPS
MESSAGE DIS-CARDED LOOP-BACK PROCESSING STOPS
TK-9516

If the Function Value is 2, indicating a message to be forwarded, and the Forward Address field contains a physical address, the UNA does the following:

1. Inserts the contents of the Forward Address field into the Destination Address field.
2. Replaces the Source Address field with the physical address of the UNA.
3. Adds 8 to the value of the Skip Count.
4. Strips the last four bytes, the CRC, from the message.
5. Transmits the resulting message, generating and appending four bytes of CRC.

Figure 4-32 Loopback Message Processing Flow (Sheet 2 of 2)

4.10.2 Remote Console and Down-Line Load

The DEUNA ROM-based microcode Remote Console Server supports the following messages:

- Request ID (Inbound to the DEUNA)
- System ID (Outbound from the DEUNA)
- Boot (Inbound to the DEUNA)

In addition, the following two dump/load type messages, associated with Boot, are supported by the DEUNA ROM-based microcode:

- Program Request (Outbound from the DEUNA)
- Memory Load with Transfer Address (Inbound to the DEUNA)

The DEUNA Remote Console Server may be off or disabled. The ROM-based microcode of the DEUNA only supports the ID and Boot functions of the Remote Console. When it is off, the DEUNA will not honor the Request ID or Boot messages. The DEUNA Remote Console Server is off under the following conditions:

- Reset State
- NI Halted State
- NI and Unibus Halted State
- DMNT (Disable Maintenance Message) bit in the mode register is set
- DTCR (Disable Transmit CRC) bit in the mode register is set

The degree of Boot capability of the DEUNA Remote Console Server in the Primary Load, Ready, Running, and UNIBUS Halted States depends on two on-board switches: the Boot Select Switches. Table 4-39 summarizes the Boot select capability of the DEUNA.

Table 4-39 Boot Select Capability of the DEUNA

Boot Select Switches (M7792)		
BOOT SEL 1	**BOOT SEL 0**	**Boot Option**
ON	ON	Remote Boot Disabled • Enabled Port Command System Boot • Disabled Remote Comm Processor Boot Remote System Boot – Remote Load Remote System Boot – Boot ROM Power Up Boot
OFF	ON	Remote Boot with System Load • Enabled Port Command System Boot Remote Comm Processor Boot Remote System Boot – Remote Load • Disabled Remote System Boot – Boot ROM Power Up Boot
ON	OFF	Remote Boot with ROM • Enabled Port Command System Boot Remote Comm Processor Boot Remote System Boot – Boot ROM • Disabled Remote System Boot – Remote Load Power Up Boot
OFF	OFF	Remote Boot with Power Up Boot and System Load • Enabled Port Command System Boot Remote Comm Processor Boot Remote System Boot – Remote Load Power Up Boot • Disabled Remote System Boot – Boot ROM

Boot Options are as follows:

- **Port Command System Boot** – Result of a Boot Port Command. The DEUNA executes a procedure that down-line loads the system secondary loader into DEUNA WCS microcode. Note that a Port Command Boot is always honored while the DEUNA is in the Ready state.
- **Remote Comm Processor Boot** – Boot message that down-line loads the DEUNA WCS.
- **Remote System Boot** – Remote Load – Result of a Boot message. The DEUNA halts the system and executes a procedure that down-line loads the system secondary loader into DEUNA WCS microcode.
- **Remote System Boot** – Boot ROM – Result of a Boot message. The DEUNA forces the system boot by invoking the system Boot ROM. (The system Boot ROM is not resident on the DEUNA.)
- **Power Up Boot** – Result of system power up. The DEUNA halts the system and executes a procedure that down-line loads the system secondary loader into DEUNA WCS microcode.

The DEUNA honors the Request ID message by sending a System ID message to the requesting station. Refer to Figure 4-33 and Table 4-40 for Request ID Message format and field descriptions. Refer to Figure 4-34 and Table 4-41 for System ID Message formats and field descriptions.

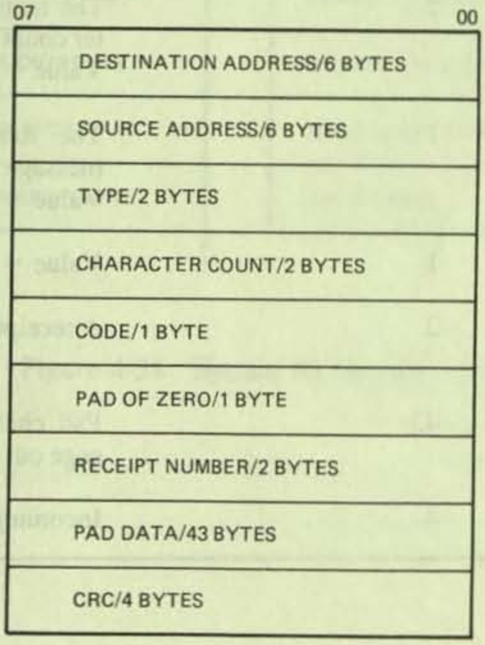


Figure 4-33 Request ID Message Format

The DEUNA also sends a System ID message every eight to ten minutes to the Remote Console Service Multicast address. The Boot message is ignored when the Remote Console Server is Boot Disabled.

When Remote Boot is enabled, the DEUNA honors the Request ID message and sends the System ID message as it does when Boot Disabled. In addition, the DEUNA honors the Boot message by entering the Primary Load State.

If Remote Boot is disabled and the DEUNA is in the Running state, the Boot message is passed to the port-driver as part of the normal datagram service.

Table 4-40 Request ID Message Field Descriptions

Field	Length (Bytes)	Description
DESTINATION ADDRESS	6	The physical address of the DEUNA
SOURCE ADDRESS	6	The physical address of the requesting station
TYPE	2	The Remote Console type Value = (0260) 60-02 hex
CHARACTER COUNT	2	The number of bytes following the character count field less pad data and CRC Value = 04 hex
CODE	1	The function code for the Request ID message Value = 05 hex
PAD OF ZERO	1	Value = 00 hex
RECEIPT NUMBER	2	A receipt number to identify the request
PAD DATA	43	Pad characters, anything to pad the message out to 64 bytes
CRC	4	Incoming block check character

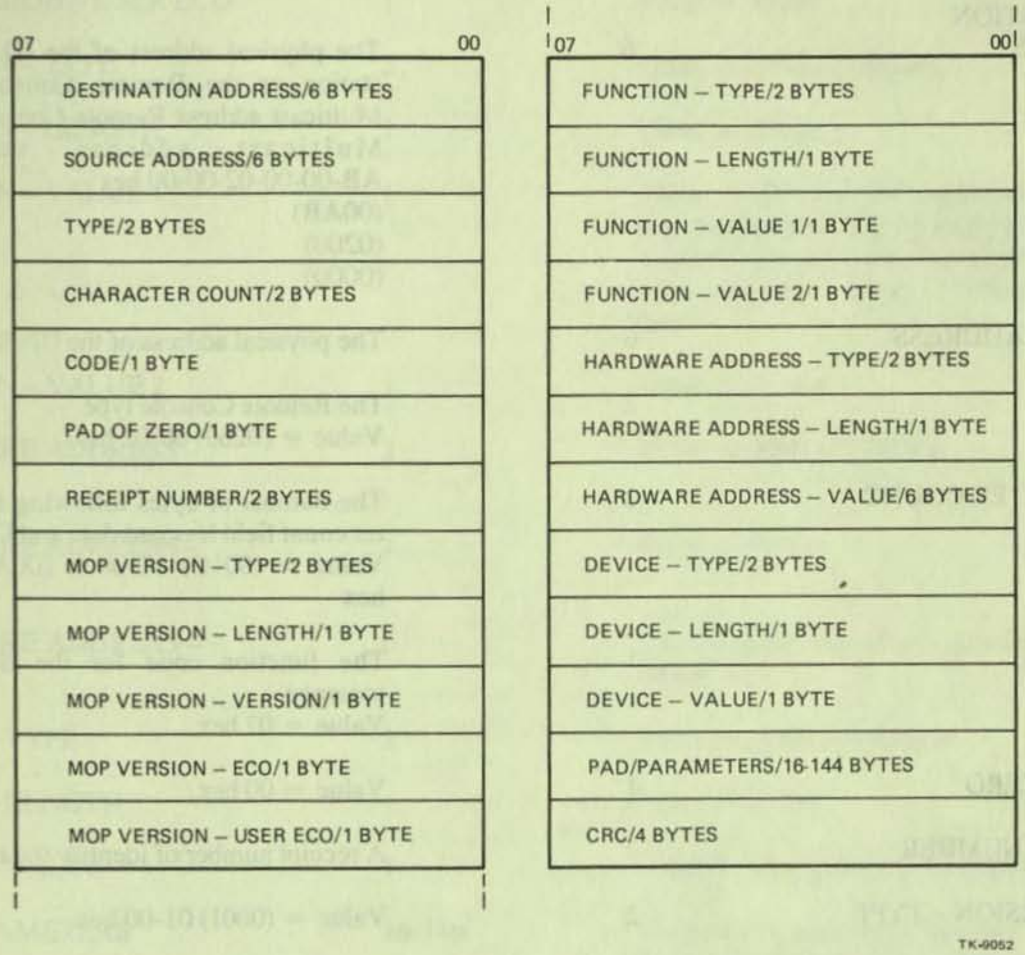

07
00
DESTINATION ADDRESS/6 BYTES
SOURCE ADDRESS/6 BYTES
TYPE/2 BYTES
CHARACTER COUNT/2 BYTES
CODE/1 BYTE
PAD OF ZERO/1 BYTE
RECEIPT NUMBER/2 BYTES
MOP VERSION – TYPE/2 BYTES
MOP VERSION – LENGTH/1 BYTE
MOP VERSION – VERSION/1 BYTE
MOP VERSION – ECO/1 BYTE
MOP VERSION – USER ECO/1 BYTE
07
00
FUNCTION – TYPE/2 BYTES
FUNCTION – LENGTH/1 BYTE
FUNCTION – VALUE 1/1 BYTE
FUNCTION – VALUE 2/1 BYTE
HARDWARE ADDRESS – TYPE/2 BYTES
HARDWARE ADDRESS – LENGTH/1 BYTE
HARDWARE ADDRESS – VALUE/6 BYTES
DEVICE – TYPE/2 BYTES
DEVICE – LENGTH/1 BYTE
DEVICE – VALUE/1 BYTE
PAD/PARAMETERS/16-144 BYTES
CRC/4 BYTES
TK-9052

Figure 4-34 System ID Message Format

Table 4-41 System ID Message Field Descriptions

Field	Length (Bytes)	Description
DESTINATION ADDRESS	6	The physical address of the ID requesting station or the Remote Console Service Multicast address Remote Console Service Multicast address value = AB-00-00-02-00-00 hex (00AB) (0200) (0000)
SOURCE ADDRESS	6	The physical address of the UNA
TYPE	2	The Remote Console type Value = (0260) 60-02 hex
CHARACTER COUNT	2	The number of bytes following the character count field less pad data and CRC. Value = (001C) 1C-00 to (00AE) AE-00 hex
CODE	1	The function code for the System ID message Value = 07 hex
PAD OF ZERO	1	Value = 00 hex
RECEIPT NUMBER	2	A receipt number of identify the request
MOP VERSION – TYPE	2	Value = (0001) 01-00 hex
MOP VERSION – LENGTH	1	Value = 03 hex
MOP VERSION – VERSION	1	Value = 03 hex
MOP VERSION – ECO	1	Value = 00 hex

Table 4-41 System ID Message Field Descriptions (Cont)

Field	Length (Bytes)	Description
MOP VERSION - USER ECO	1	Value = 00 hex
FUNCTION - TYPE	2	Value = (0002) 02-00 hex
FUNCTION - LENGTH	1	Value = 02 hex
FUNCTION - VALUE 1	1	Value = 05 hex, the maintenance functions supported; Loop, Primary Loader Value = 15 hex, the maintenance functions supported, Loop, Primary Loader, Boot
FUNCTION - VALUE 2	1	Value = 00 hex
HARDWARE ADDRESS - TYPE	2	Value = (0007) 07-00 hex
HARDWARE ADDRESS - LENGTH	1	Value = 06 hex
HARDWARE ADDRESS - VALUE	6	The default physical address of the DEUNA
DEVICE - TYPE	2	Value = (0064) 64-00 hex
DEVICE - LENGTH	1	Value = 01 hex
DEVICE - VALUE	1	Value = 01 hex (DEUNA device code)
PAD/PARAMETERS	16-146	Additional parameters supplied by port-driver through the Write System ID port command. If not supplied, zeros are added by the DEUNA to pad the message out to 64 bytes.
CRC	4	Outgoing block check character

4.10.2.1 Remote Boot – Incoming Boot messages invoke procedures within the DEUNA to down-line load DEUNA microcode solely, referred to as Comm Processor Boot and System Boot. System Boots initiated from a remote node may be from the Host system resident Boot ROM or from a Secondary loader down-line loaded into the DEUNA WCS. Refer to Figure 4-35 and Table 4-42 for the Boot message format and field descriptions.

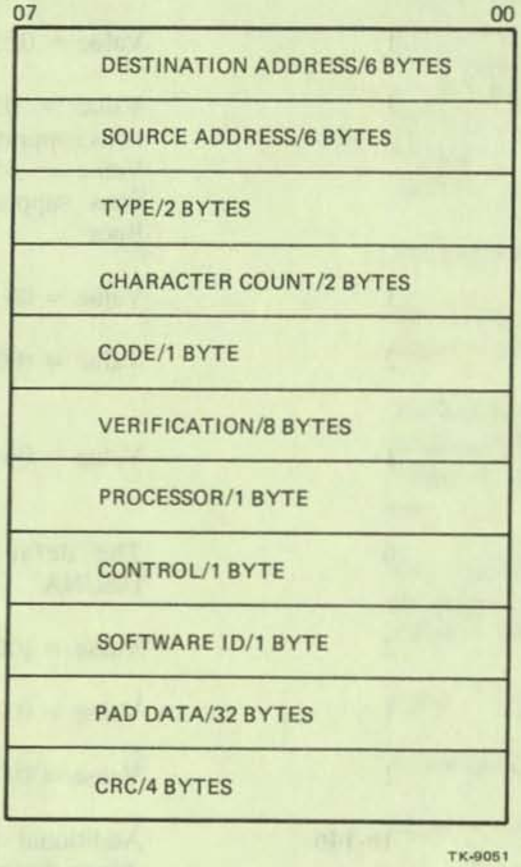


Figure 4-35 Boot Message Format

Table 4-42 Boot Message Field Descriptions

Field	Length (Bytes)	Description
DESTINATION ADDRESS	6	The physical address of the DEUNA.
SOURCE ADDRESS	6	The physical address of the requesting station
TYPE	2	The Remote Console type Value = (0260) 60-02 hex
CHARACTER COUNT	2	The number of bytes following the character count field less pad data and CRC. Value = (000C) 0C-00 hex
CODE	1	The function code for the Boot message. Value = 06 hex.
VERIFICATION	8	The code to be compared against the port-driver supplied verification code. The codes must match before the DEUNA will honor the boot. If the DEUNA has not been supplied with a verification code by the port-driver or supplied with a code of 0, the DEUNA accepts any value in the boot message verification field.
PROCESSOR	1	Value = 00 hex; System boot, enter the Primary Load state. Value = 01 hex; Boot the DEUNA, enter the Primary Load state.
CONTROL	1	Value = 00 hex; Boot from the system default. Value = 01 hex; Boot from the requesting system.

SOFTWARE ID	1	Value = 00 hex; No ID.
		Value = FF hex; Operating system.
		Value = FE hex; Diagnostics.

PAD DATA	32	Pad characters, anything to pad the message out to 64 bytes.
CRC	4	Incoming block check character.

In response to a Boot message, the DEUNA performs the following.

1. A message with remote console type value in its type field and the boot value in the code field is received into a DEUNA buffer.

2. If the Remote Console Server is off, the message is discarded.

3. If the CRC is bad and the DEUNA is in the running state, the message is treated as a normal datagram and boot processing stops. If the CRC is bad and the DEUNA is not in the running state, the message is discarded and boot processing stops; otherwise, boot processing continues.

4. If the DEUNA is in the Primary Load State, executing Loop 1 of the Primary Loader, the message is discarded. See Primary Loader Section 4.10.2.4 for details.

5. If the character count, processor, control, and software ID fields are within the expected limits, boot processing continues. Otherwise, boot processing stops and the message is discarded.

6. If the Boot Select Switches are configured to allow remote boot, processing continues. Otherwise, the boot message is discarded.

7. The DEUNA compares the verification code in the boot message to the verification code supplied by the port command. Refer to Section 4.4.11 for a description of the Write System ID Port command. If they match, processing continues. If the DEUNA has not been supplied with a verification code by a port command, or the DEUNA has been supplied with a verification code of value 0, then any incoming verification code will suffice and processing continues. Otherwise, the boot message is discarded.

8. The DEUNA decodes the Processor field of the Boot message to determine if the Comm or System Processor is to be booted.

9. If the system processor is to be booted, the setting of the boot select switches determines the action taken by the DEUNA.

   a. If the boot switches are configured to allow a boot from the system boot ROM, the DEUNA does the following.

      - Asserts ACLO (causing a system power fail trap).

      - Blocks UNIBUS INIT to itself.

      - Discards any incoming boot messages for 40 seconds.

      - Makes a transition to the READY state.

b. If the boot switches are configured so that a boot from the system boot ROM is NOT allowed, or the Comm Processor is to be booted, the DEUNA does the following.

   - The DEUNA enters the Primary Load State.

   - If the Comm Processor is to be booted and the DEUNA had not been in the Primary Load State previously, the DEUNA sets the USCI bit of PCSR0.

   - If the system processor is to be booted, the DEUNA does the following.

     – Loads the following program into system memory:

```
2/     777
4/     PCSR0 ADDRESS
6/     0
10/    12
12/    0
14/    16
16/    0
20/    22
22/    0
24/    30
26/    340
30/    012706
32/    1000
34/    762
```

     – Asserts ACLO to halt system processor via a powerfail trap. The DEUNA blocks UNIBUS INIT to itself.

   - The DEUNA forms a program request message. Refer to Figure 4-36 and Table 4-43 for the format of the program request message and field descriptions.

   - The DEUNA copies the Software ID field of the boot message into the Software ID field of the Program Request message.

   - If the value 1 is found in the Control field of the Boot message, the DEUNA transfers the address in the Source Address field of the Boot message to the Destination Address field of the Program Request message. If the value 0 is found in the Control field of the Boot message, the DEUNA Load Server Address is written into the Destination Address field of the Program Request message. The DEUNA Load Server Address is supplied by a Port Command. Refer to Section 4.4.12. If the Port Command has not been issued, the DEUNA uses the Load Assistant Multicast Address.

   - After the DEUNA completes the formation of the Program Request message, it executes the Primary Loader. See Primary Loader subsection (4.10.2.4).

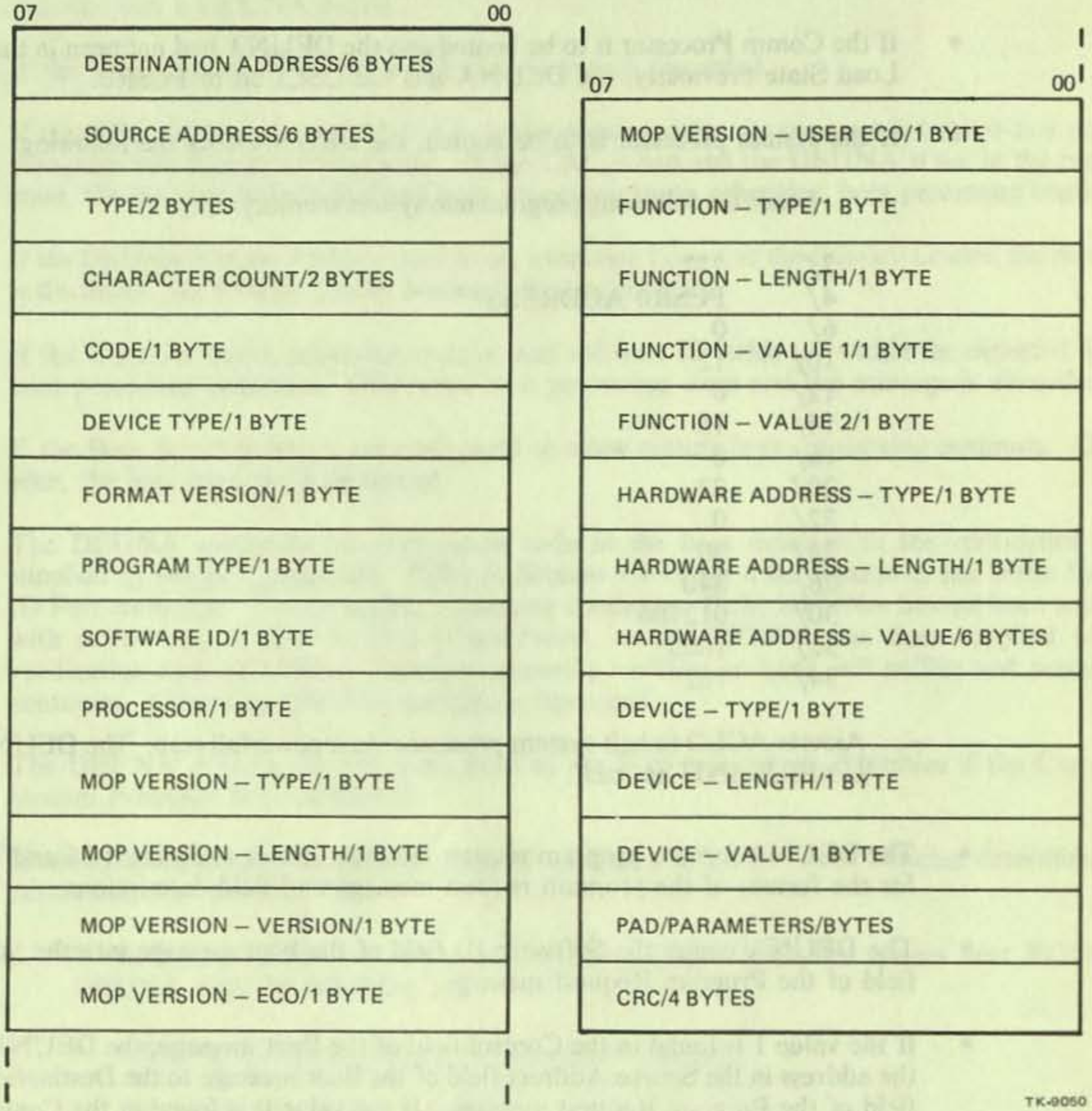


Figure 4-36 Program Request Message Format

Table 4-43 Program Request Message Field Descriptions

Field	Length (Bytes)	Description
DESTINATION	6	The address supplied by the Write Load Server Port Function. The default address is the Load Assistant Multicast address. Load Assistant Multicast address value = AB-00-00-01-00-00 hex. (00AB) (0001) (0000)
SOURCE ADDRESS	6	The physical address of the DEUNA
TYPE	2	The Dump/Load type Value = (0160) 60-01 hex
CHARACTER COUNT	2	The number of bytes following the character count field less pad data and CRC. Value = (001D) 1D-00 hex to (00AF) AF-00 hex
CODE	1	Value = 08 hex
DEVICE TYPE	1	The device type DEUNA Value = 01 hex
FORMAT VERSION	1	Value = 01 hex
PROGRAM TYPE	1	Value = 00 hex DEUNA microcode
PROCESSOR	1	Value = 00 hex; System boot, enter the Primary Load state. Value = 01 hex; Boot the DEUNA, enter the Primary Load state.
MOP VERSION – TYPE	2	Value = (0001) 01-00 hex
MOP VERSION – LENGTH	1	Value = 03 hex
MOP VERSION – VERSION	1	Value = 03 hex

Table 4-43 Program Request Message Field Descriptions (Cont)

Field	Length (Bytes)	Description
MOP VERSION – ECO	1	Value = 00 hex
MOP VERSION – USER ECO	1	Value = 00 hex
FUNCTION – TYPE	2	Value = (0002) 02-00 hex
FUNCTION – LENGTH	1	Value = 02 hex
FUNCTION – VALUE 1	1	Value = 05 hex
		Value = 15 hex. The maintenance functions supported; Loop, Primary Loader, Boot.
FUNCTION – VALUE 2	1	Value = 00 hex
HARDWARE ADDRESS – TYPE	2	Value = (0007) 07-00 hex
HARDWARE ADDRESS – LENGTH	1	Value = 06 hex
HARDWARE ADDRESS – VALUE	6	The physical address of the DEUNA
DEVICE – TYPE	2	Value = (0064) 64-00 hex
DEVICE – LENGTH	1	Value = 01 hex
DEVICE - VALUE	1	The DEUNA device code Value = 01 hex
PAD/PARAMETERS		The set of additional parameters supplied by port-driver through the Write System ID port command. If not supplied, zeros are added by the DEUNA to pad the message out to 64 bytes.
CRC	4	DEUNA generated block check character.

4.10.2.2 Local Boot – The following is the DEUNA operation in response to a Boot Port command.

1. The DEUNA receives the Boot Port Command. If the DEUNA is not already in the Primary Load State, it enters it and sets the DNI bit of PCSR0.

2. The DEUNA forms a Program Request message.

   - The DEUNA writes the Software ID value of the System ID parameter List into the Software ID field of the Program Request message.

   - The DEUNA Load Server Address is written into the Destination Address field of the Program Request message. The DEUNA Load Server Address is supplied by a Port Command. If the Port Command has not been issued, the DEUNA uses the Load Assistant Multicast Address.

3. After the DEUNA completes the formation of the Program Request message, it executes the Primary Loader. See Section 4.10.2.4 for details.

Note that the DEUNA does not attempt to halt the system processor for a Boot on Port Command. It is assumed the system processor will be in the appropriate action after issuing the Port Command.

4.10.2.3 Boot on Power Up – The following is the DEUNA operation in response to a Power Up.

1. The DEUNA enters the Reset State upon Power Up, UNIBUS INIT, device Reset, or the Self-Test port command.

2. The DEUNA compares the footprint of the WCS to detect Power Up. If the footprint compares, indicating not a true power on condition, the DEUNA continues in the Reset State but does not execute the self-test. If footprint does not compare, indicating a true power on condition, the DEUNA does the following:

   - The DEUNA writes the footprint.

   - The DEUNA Tests the Boot Select Switch settings. If Power-up Boot is not selected, the DEUNA continues in the reset State and executes the self-test. If Power-up Boot is selected, the DEUNA does the following:

     a. The DEUNA enters the Primary Load State.

b. The DEUNA loads the following program in system memory:

```
2/       777
4/     PCSR0 ADDRESS
6/         0
10/       12
12/        0
14/       16
16/        0
20/       22
22/        0
24/       30
26/      340
30/   012706
32/     1000
34/      762
```

c. The DEUNA asserts UNIBUS ACLO to halt the system processor through the Powerfail trap. The DEUNA blocks UNIBUS INIT to itself.

d. The DEUNA forms a Program Request message.

- Writes the value − 1 into the Software ID field of the Program Request message.
- The Load Assistant Multicast Address is written into the Destination Address field of the Program Request message.
- After the DEUNA completes the formation of the Program Request message, it executes the Primary Loader. Refer to Section 4.10.2.4.

4.10.2.4 Primary Load State – The Primary Load State of the DEUNA provides for communication processor boot (down-line load of DEUNA Writable Control Store (WCS) based microcode) and system boot. The DEUNA enters the Primary Load State under three possible conditions.

1. Reception of an error-free Boot message. The Remote Console Server is not off and the DEUNA Boot select switches are configured to honor the Boot message.

2. The Remote Console Server is not off and a Boot Port command is received from the Port Driver.

3. The Remote Console Server is not off following a successful powerup and the DEUNA Boot Select switches are configured to allow power-up boot.

Once the DEUNA has entered the Primary Load State, it performs the following.

1. Transmits the Program Request message described under Sections 4.10.2.1 – 4.10.2.3.

2. Waits for a Memory Load with Transfer Address message.

3. If, after five seconds, the DEUNA has not received a correct Memory Load with Transfer Address message, it retransmits the Program Request message. This procedure is repeated for eight transmissions of the Program Request message. During this time, any incoming Boot message is discarded. This is Loop 1 of the Primary Loader. Loop 1 of the Primary Loader is executed regardless of retry faults (collision on 16 attempts).

4. If after eight timeouts no correct Memory Load with Transfer Address message is received, the DEUNA takes the following action.

   - If it entered the Primary Load State in response to a Boot message to boot the comm processor, it exits the Primary Load State, enters the Ready State, and sets the USCI bit of PCSR0.

   - If it entered the Primary Load State in response to a Boot message to boot the system processor, in response to a Boot Port Command, or in response to a Power-Up Boot, the DEUNA does the following.

     - Writes the Software ID field to value 0 and the Destination Address field with the Load Assistant Multicast Address in the Program Request message and transmits it.

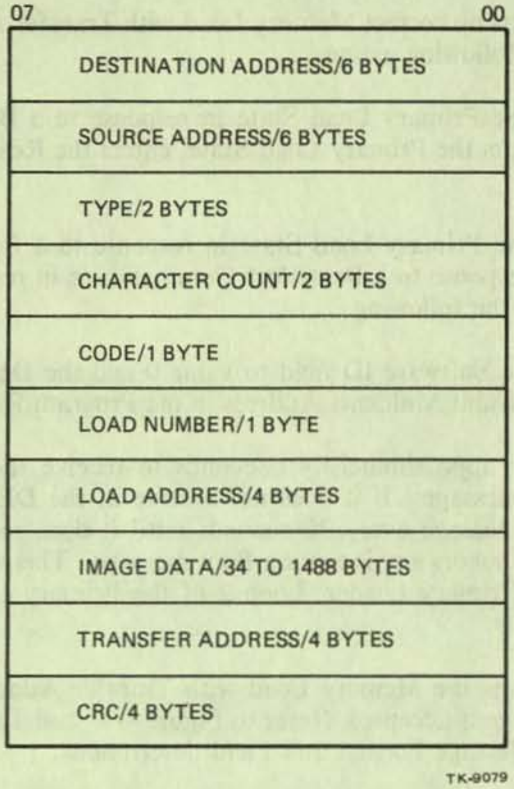
     - Waits for approximately 40 seconds to receive the Memory Load with Transfer Address message. If it does not receive it, the DEUNA retransmits the Program Request message every 30 seconds until it does receive it. During this time, the DEUNA honors any incoming Boot message. This wait and retransmit time is Loop 2 of the Primary Loader. Loop 2 of the Primary Loader is executed regardless of retry faults.

5. The DEUNA receives the Memory Load with Transfer Address message. If the message is error free, the message is accepted. Refer to Figure 4-37 and Table 4-42 for Memory Load with Transfer Address Message Format and Field descriptions.

6. The DEUNA Loads the data image of the Memory Load with Transfer Address message into its WCS starting at the load address supplied with the message.

7. If the Memory Load with Transfer Address message was received to boot the Comm Processor, the DEUNA enters the Ready State and sets the USCI bit of PCSR0.

8. The DEUNA starts executing microinstructions at the transfer address supplied by the Memory Load with Transfer Address message.



DESTINATION ADDRESS/6 BYTES

SOURCE ADDRESS/6 BYTES

TYPE/2 BYTES

CHARACTER COUNT/2 BYTES

CODE/1 BYTE

LOAD NUMBER/1 BYTE

LOAD ADDRESS/4 BYTES

IMAGE DATA/34 TO 1488 BYTES

TRANSFER ADDRESS/4 BYTES

CRC/4 BYTES

Figure 4-37 Memory Load with Transfer Address Message Format

Table 4-44 Memory Load With Transfer Address Message Field Descriptions

Field	Length (Bytes)	Description
DESTINATION ADDRESS	6	Physical address of the DEUNA.
SOURCE ADDRESS	6	Physical address of the Load Server
TYPE	2	The Dump/Load type Value = (0160) 60-01 hex
CHARACTER COUNT	2	Number of bytes following the character count field less pad data and CRC Value = (002C) 2C-00 to (05DA) DA-05 hex
CODE	1	Value = 00 hex
LOAD NUMBER	1	Value = 00 hex
LOAD ADDRESS	4	DEUNA microstore load address for storage of the data image
DATA IMAGE	34 – 1488	Image to be stored in memory
TRANSFER ADDRESS	4	DEUNA microstore starting address of the data image
CRC	4	Received block check character

APPENDIX A
FLOATING DEVICE ADDRESSES AND VECTORS

A.1 FLOATING DEVICE ADDRESSES

UNIBUS addresses from 760010 through 763776 are floating device addresses, (see Figure A-1). They are used as register addresses for communication devices interfacing with a PDP-11 or VAX-11 system.

NOTE
Some devices are not supported by VAX-11/780.

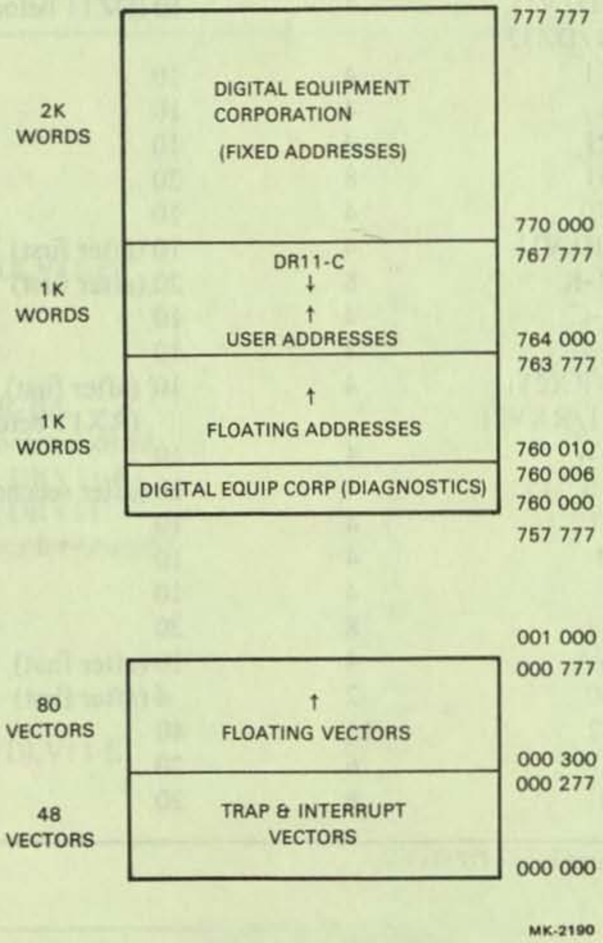


Figure A-1 UNIBUS Address Map

To assign these addresses, a gap of 10_8 must be left between the last address of one device type and the first address of the next device type. The first address of the next device type must start on a module 10_8 boundary. The 10_8 gap must also be left for uninstalled devices that are skipped in the priority ranking list (see Table A-1). Multiple devices of the same type must be assigned contiguous addresses. Device types already in the system may need to be reassigned to make room for additional ones.

Table A-1 Floating Device Address Ranking Sequence

Rank	Option	Decimal Size	Octal Modulus
1	DJ11	4	10
2	DH11	8	20
3	DQ11	4	10
4	DU11,DUV11	4	10
5	DUP11	4	10
6	LK11A	4	10
7	DMC11/DMR	4	10 (DMC before DMR)
8	*DZ11/DZV11, DZS11/DZ32	4	10 (DZ11 before DZ32)
9	KMC11	4	10
10	LPP11	4	10
11	VMV21	4	10
12	VMV31	8	20
13	DWR70	4	10
14	RL11,RLV11	4	10 (after first)
15	LPA11-K	8	20 (after first)
16	KW11-C	4	10
17	Reserved	4	10
18	RX11/RX211 RXV11/RXV21	4	10 (after first) (RX11 before RX211)
19	DR11-W	4	10
20	DR11-B	4	10 (after second)
21	DMP11	4	10
22	DPV11	4	10
23	ISB11	4	10
24	DMV11	8	20
25	DEUNA	4	10 (after first)
26	UDA50	2	4 (after first)
27	DMF32	16	40
28	DMS11	6	20
29	VS100	8	20

* DZ11-E and DZ11-F are treated as two DZ11s.

A.2 FLOATING VECTOR ADDRESSES

UNIBUS addresses from 300 to 777 are floating vector addresses. They are used for communication devices interfacing with a PDP-11 or VAX-11 system.

NOTE

Some devices are not supported by the VAX-11/780 system. Vector size is determined by the device type.

There are no gaps in floating vectors unless required by physical hardware restrictions. In data communications devices, the receive vector must be on a zero boundary; the transmit vector must be on a 4(8) boundary.

Multiple devices of the same type should be assigned vectors sequentially. Table A-2 shows the floating vector ranking assignment sequence.

Table A-2 Floating Vector Ranking Sequence

Rank	Option	Decimal Size	Octal Modulus
1	DC11	4	10*
1	TU58	4	10*
2	KL11	4	10**
2	DL11-A	4	10**
2	DL11-B	4	10**
2	DLV11-J	16	10**
2	DLV11, DLV11-F	4	10**
3	DP11	4	10
4	DM11-A	4	10
5	DN11	2	4
6	DM11-BB/BA	2	4
7	DH11 modem control	2	4
8	DR11-A, DRV11-B	4	10
9	DR11-C, DRV11	4	10
10	PA611 (reader+punch)	8	10
11	LPD11	4	10
12	DT07	4	10
13	DX11	4	10
14	DL11-C	4	10
14	DL11-D	4	10
14	DL11-E/DLV11-E	4	10
15	DJ11	4	10
16	DH11	4	10
17	GT40	8	10
17	VSV11	8	10

* There is no standard configuration for systems with both a DC11 and TU58.

** A KL11 or DL11 used as the console uses a fixed vector.

Table A-2 Floating Vector Ranking Sequence (Cont)

Rank	Option	Decimal Size	Octal Modulus
18	LPS11	12	10
19	DQ11	4	10
20	KW11-W, KWV11	4	10
21	DU11, DUV11	4	10
22	DUP11	4	10
23	DV11+modem control	6	10
24	LK11-A	4	10
25	DWUN	4	10
26	DMC11	4	10
26	DMR11	4	10 (DMC before DMR)
27	DZ11/DZV11, DZS11/DZ32	4	10 (DZ11 before DZ32)
28	KMC11	4	10
29	LPP11	4	10
30	VMV21	4	10
31	VMV31	4	10
32	VTV01	4	10
33	DWR70	4	10
34	RL11/RLV11	2	4 (after the first)
35	TS11	2	4 (after the first)
36	LPA11-K	4	10
37	IP11/IP300	2	4 (after the first)
38	KW11-C	4	10
39	RX11/RX211 RXV11/RXV21	2	4 (after the first) (RX11 before RX211)
40	DR11-W	2	4
41	DR11-B	2	4 (after the first)
42	DMP11	4	10
43	DPV11	4	10
44	ML11	2	4 (MASSBUS device)
45	ISB11	4	10
46	DMV11	4	10
47	DEUNA (REVC)***	2	4 (after the first)
48	UDA50	2	4 (after the first)
49	DMF32	16	4
50	KMS11	6	10
51	PCL11-B	4	10
52	VS100	2	4

* There is no standard configuration for systems with both a DC11 and TU58.

** A KL11 or DL11 used as the console uses a fixed vector.

*** DEUNA (REVB) Decimal=4 Octal=10.

A.3 DEVICE AND VECTOR ADDRESS ASSIGNMENT EXAMPLES

Example 1

The first device requiring address assignment is a DH11 (number 2 in the device address assignment sequence and number 16 in the vector address assignment sequence).

The devices to be assigned addresses are:

2 DH11	1 DMR11
2 DQ11s	1 DEUNA
1 DUP11	

Option	Device Address	Vector Address	Comment
	760010		Gap left for DJ11 (number 1 on device address assignment sequence) which is not used
DH11	760020	300	First DH11
DH11	760040	310	Second DH11
	760060		Gap between last DH11 used and the next device
DQ11	760070	320	First DQ11
DQ11	760100	330	Second DQ11
	760110		Gap between last DQ11 used and the next device
	760120		Gap left for DU11s is not used
DUP11	760130	340	Only one DUP11
	760140		Gap left between DUP11 and next device
	760150		Gap left for LK11-As is not used
DMR11	760160	350	Only one DMR11
	760170		Gap left after the last device with a floating address assignment (in this case, the DMR11) to indicate that none follows
DEUNA	774510	120	First DEUNA uses fixed device and vector addresses

Example 2

The devices to be assigned addresses are:

1 DJ11
1 DH11
2 DQ11s
2 DUP11s
2 DMR11s
2 DEUNAs

Option	Device Address	Vector Address	Comment
DJ11	760010	300	Only one DJ11
	760020		Gap left between DJ11 and the next device
	760030		Gap - The next device, a DH11, must start on an address boundary that is a multiple of 20
DH11	760040	310	Only one DH11
	760060		Gap left between DH11 and the next device
DQ11	760070	320	First DQ11
DQ11	760100	330	Second DQ11
	760110		Gap between last DQ11 used and the next device
	760120		Gap left for DU11s is not used
DUP11	760130	340	First DUP11
DUP11	760140	350	Second DUP11
	760150		Gap left between last DUP11 and the next device
	760160		Gap left for LK11-As is not used
DMR11	760170	360	First DMR11
DMR11	760200	370	Second DMR11
	760210		Gap left between last DMR11 and the next device
DEUNA	774510	120	First DEUNA uses fixed device and vector addresses
DEUNA	760450	400	Second DEUNA uses floating device and vector addresses
	760460		Gap left after the last device with a floating address assignment (in this case, the second DEUNA, to indicate that none follows)

APPENDIX B
REMOTE BOOT AND DOWN-LINE LOAD

B.1 INTRODUCTION

The remote boot and down-line load features implemented in the DEUNA are used to allow the PDP-11 system in which the DEUNA is installed to be booted and to load a system image into the processor. This function is useful with systems requiring remote booting and loading of their system images. Figure B-1 shows a basic PDP-11 system with a DEUNA.

For more information on remote boot and down-line load, refer to Section 4.10.2.

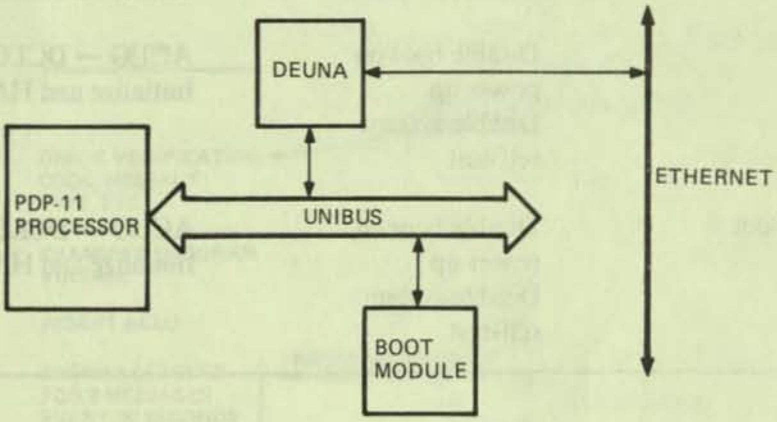


Figure B-1 PDP-11 System

B.2 SYSTEM CONFIGURATION GUIDELINES

When configuring a system to be remote booted and/or down-line loaded use the following guidelines:

1. System Processor

   a. When ACLO is asserted on the UNIBUS, the processor must be set up to assert DCLO (power-fail sequence).

   b. When DCLO is asserted, the processor is initialized and then HALTED. For a boot from ROM function, the processor should start to execute from the boot ROM on the system boot module.

2. System Boot Module (except for boot from boot ROM)

   a. Disable power-up boot.

   b. Disable system self-test.

> **NOTE**
>
> **When configuring a system to meet these guidelines, refer to the processor and boot module manuals for the system.**

Table B-1 summarizes the system configuration guidelines.

Table B-1 Remote and Down-Line Load Configuration Guidelines

DEUNA Boot Function	Boot Module	Processor
Boot with ROM	Configure for boot from ROM	ACLO → DCLO Boot from ROM
Remote Boot	Disable boot on power up Disable system self-test	ACLO → DCLO Initialize and HALT
Remote/Power-up Boot	Disable boot on power up Disable system self-test	ACLO → DCLO Initialize and HALT

B.3 REMOTE BOOT DISABLED

Remote boot is disabled when the BOOT SEL switches are configured as follows:

BOOT SEL 0 = ON
BOOT SEL 1 = ON

When remote boot is disabled, the system processor can only be booted by the DEUNA via a BOOT port command. It cannot be booted via a boot request from another node on the ETHERNET.

B.4 REMOTE BOOT WITH SYSTEM LOAD

Remote boot with system load is enabled when the BOOT SEL switches are configured as follows:

BOOT SEL 0 = ON
BOOT SEL 1 = OFF

When remote boot with system load is selected, the DEUNA accepts a boot message received on the ETHERNET, boots the system processor and down-line loads the system image.

When a boot message for system boot is received from another station on the ETHERNET (NI), the DEUNA performs the following (see Figure B-2).

1. Boot message is received by DEUNA.

2. The DEUNA checks the verification code, message type, etc.

3. The DEUNA transfers a program from ROM via DMA to system memory.

4. The DEUNA asserts ACLO. This simulates a power fail to the system.

5. The DEUNA sends a program request message onto the NI and waits for a memory load with transfer address. The program request message is sent every five seconds for the first eight messages, then every 30 seconds until the memory load with transfer address is performed.

6. The DEUNA checks the memory load message, transfers it to WCS, then executes the instructions starting at the transfer address.

   The program loaded into WCS is the secondary loader. This loader is used to bring a tertiary loader into system memory. The tertiary loader is used to load the system image.

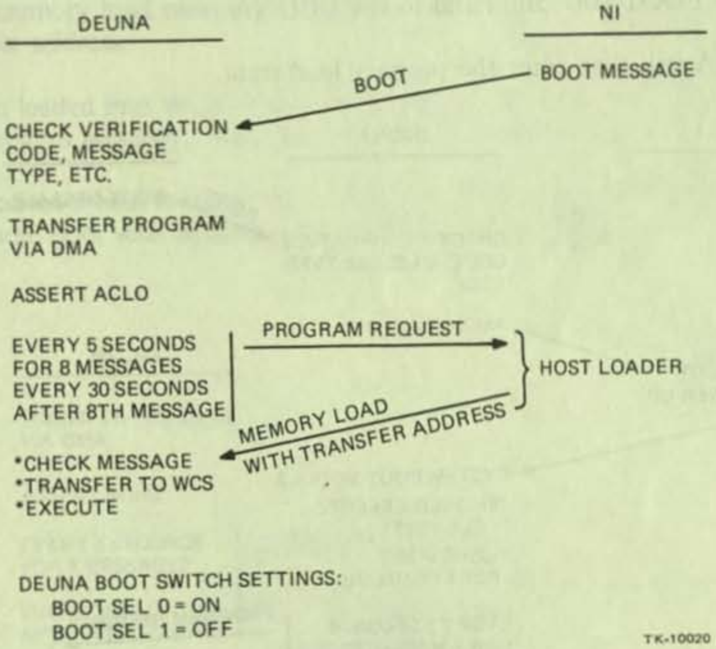

DEUNA
NI
BOOT
BOOT MESSAGE
CHECK VERIFICATION CODE, MESSAGE TYPE, ETC.
TRANSFER PROGRAM VIA DMA
ASSERT ACLO
EVERY 5 SECONDS FOR 8 MESSAGES EVERY 30 SECONDS AFTER 8TH MESSAGE
PROGRAM REQUEST
HOST LOADER
MEMORY LOAD WITH TRANSFER ADDRESS
•CHECK MESSAGE
•TRANSFER TO WCS
•EXECUTE
DEUNA BOOT SWITCH SETTINGS:
BOOT SEL 0 = ON
BOOT SEL 1 = OFF
TK-10020

Figure B-2 Remote Boot with System Load Functional Flow

B.5 REMOTE BOOT WITH ROM

When remote boot with ROM is selected, the DEUNA accepts a boot message received on the ETHERNET, then boots the system via ROM-based instructions contained on the system boot module.

Remote Boot with ROM is selected when the boot select switches are configured as follows:

BOOT SEL 0 = OFF
BOOT SEL 1 = ON

When a boot message for system boot is received from another station on the NI, the following sequence occurs (Figure B-3):

1. The DEUNA checks the verification code, message type, etc..
2. The DEUNA asserts ACLO; this simulates a powerfailure to the system.
3. The system then performs a power-up boot using the ROM-based boot program.
4. The boot program, in addition to booting the system, should:
   - Self-test the system
   - Issue a BOOT port command to the DEUNA
5. The DEUNA will then enter the primary load state.

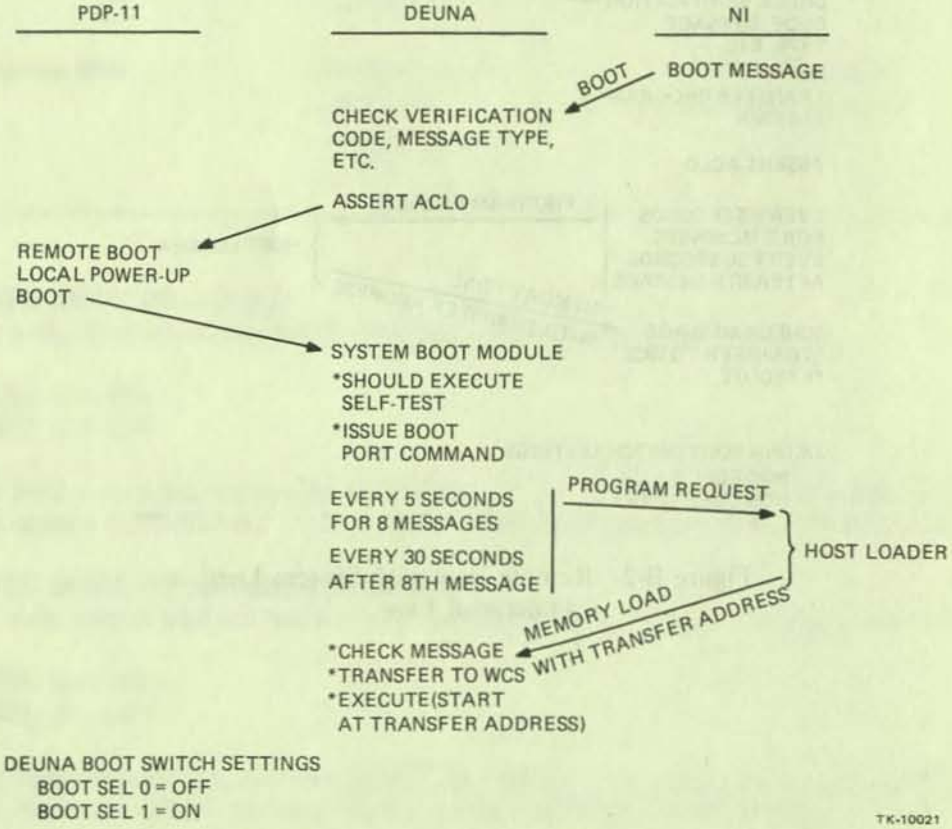

PDP-11
DEUNA
NI
BOOT
BOOT MESSAGE
CHECK VERIFICATION CODE, MESSAGE TYPE, ETC.
ASSERT ACLO
REMOTE BOOT LOCAL POWER-UP BOOT
SYSTEM BOOT MODULE
*SHOULD EXECUTE SELF-TEST
*ISSUE BOOT PORT COMMAND
EVERY 5 SECONDS FOR 8 MESSAGES
EVERY 30 SECONDS AFTER 8TH MESSAGE
PROGRAM REQUEST
HOST LOADER
MEMORY LOAD WITH TRANSFER ADDRESS
*CHECK MESSAGE
*TRANSFER TO WCS
*EXECUTE(START AT TRANSFER ADDRESS)
DEUNA BOOT SWITCH SETTINGS
BOOT SEL 0 = OFF
BOOT SEL 1 = ON
TK-10021

Figure B-3 Remote Boot with ROM Functional Flow

B.6 REMOTE BOOT/POWER-UP BOOT WITH SYSTEM LOAD

When the DEUNA is configured for Remote Boot/Power-Up Boot with System Load, the DEUNA can boot and perform a system load over the ETHERNET in 2 ways:

1. On system power-up
2. On receipt of boot message over the ETHERNET

The boot select switches on the port module of the DEUNA are configured as follows:

BOOT SEL 0 = OFF
BOOT SEL 1 = OFF

When the system is powered up, the DEUNA performs the following (Figure B-4):

1. Transfers a program from ROM via DMA to system memory.

2. Assert ACLO; this simulates a powerfailure to the system.

3. Sends a program request message onto the NI and wait for a memory load with the transfer address. The program request message is sent every five seconds for the first eight messages, then every 30 seconds until the memory load with transfer address is performed.

4. Checks the memory load message, transfers it to WCS, then executes the instructions starting at the transfer address.

   The program loaded into WCS is the secondary loader. This loader is used to bring a tertiary loader into system memory. The tertiary loader is used to load the system image.

When the DEUNA receives a boot message from another station on the ETHERNET, it functions in the same manner as a Remote Boot with System Load. Refer to Section B.3 of this appendix for a description of this function.

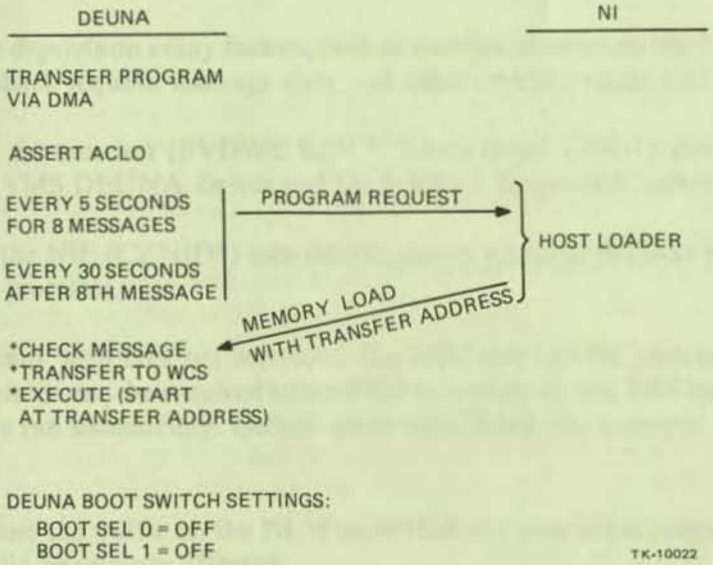

DEUNA
NI
TRANSFER PROGRAM VIA DMA
ASSERT ACLO
EVERY 5 SECONDS FOR 8 MESSAGES
PROGRAM REQUEST
HOST LOADER
EVERY 30 SECONDS AFTER 8TH MESSAGE
MEMORY LOAD WITH TRANSFER ADDRESS
•CHECK MESSAGE
•TRANSFER TO WCS
•EXECUTE (START AT TRANSFER ADDRESS)
DEUNA BOOT SWITCH SETTINGS:
BOOT SEL 0 = OFF
BOOT SEL 1 = OFF
TK-10022

Figure B-4 Power-Up Boot with System Load Functional Flow

B.4 REMOTE BOOT/POWER-UP BOOT WITH SYSTEM LOAD

When the DEUNA is configured for Remote Boot/Power-Up Boot with System Load, the DEUNA can boot and execute a system load over the ETHERNET in 2 ways:

1. On system power-up
2. On receipt of boot message over the ETHERNET

The boot select switches on the port module of the DEUNA are configured as follows:

BOOT SEL 2 = OFF
BOOT SEL 1 = OFF

When the system is powered up, the DEUNA performs the following (Figure B-4):

1. Transfers a program from PROM via DMA to system memory.

2. Asserts ACLO (this simulates a power-failure to the system).

3. Sends a program request message onto the NI and waits for a memory load with the transfer address. The program request message is sent every 4 seconds for the first eight messages, then every 30 seconds until the memory load with transfer address is performed.

4. Checks the memory load message. If valid, it then transfers it to WCS, then executes the instructions starting at the transfer address.

   The program loaded into WCS is the secondary loader. This loader is used to bring a tertiary loader into system memory. The tertiary loader is used to load the system image.

When the DEUNA receives a boot message from another station on the ETHERNET, it functions in the same manner as a Remote Boot with System Load. Refer to Section B.3 of this appendix for a description of this function.

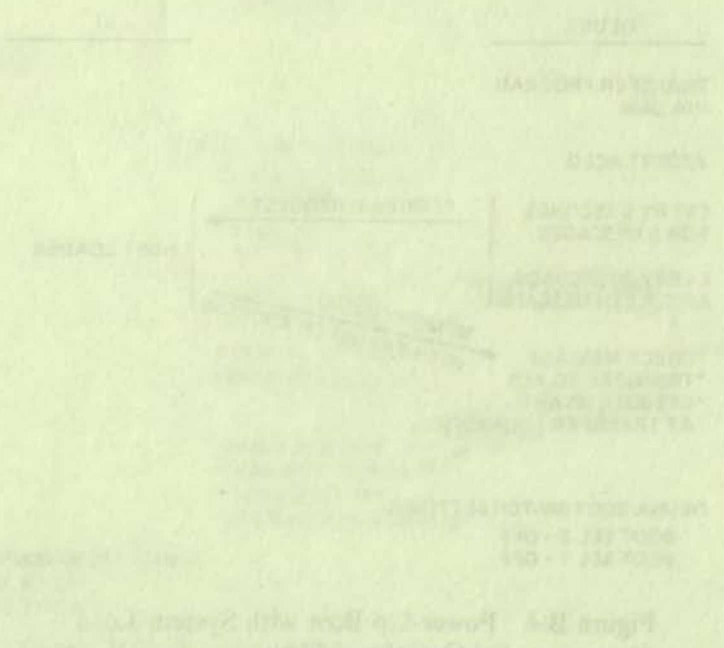


Figure B-4 Power-Up Boot with System Load

APPENDIX C
NETWORK INTERCONNECT EXERCISER

C.1 INTRODUCTION

The Network Interconnect Exerciser (NIE) provides a VAX-11 Level 2R and a PDP-11 standalone diagnostic exerciser for ETHERNET networks. The NIE determines node ability on the network and provides the operator with error analysis. Node installation, verification, and problem isolation can be performed using the NIE.

The NIE is divided into two parts:

- **Default Section** (also called operator intervention section) – This section allows the operator to use the NIE in different modes (for example, size-NI, use loop assist, or full assistance testing of all nodes). This section is operator driven, with the operator selecting the tests and the testing parameters.

- **Unattended Mode** – This mode collects a table of node addresses, then tests the nodes selected using the low level maintenance functions of the DEUNA.

The memory size of the node running the NIE determines how many nodes can be selected for testing at one time. Only current summary information is maintained to retain the maximum number of physical addresses.

The total execution time depends on many factors, such as number of nodes on the NI, the response time of a remote node to a loopback request, message sizes, and other operator-dependent factors.

The VAX-11 version of the exerciser (EVDWC REV *.*) runs on all VAX-11 processors. It is a level 2R diagnostic and uses the VMS DEUNA Driver and the VAX-11 Diagnostic Supervisor (VDS).

The PDP-11 version of the NIE (CZNID*) uses the Diagnostic Runtime Services (DRS) and runs on any PDP-11 UNIBUS type processor.

The NIE runs concurrently with DECnet software. The NIE uses two NI protocol types: loopback and remote console. The operator may be required to run NCP to modify certain DECnet parameters before all parts of the NIE can be run successfully. Certain other restrictions (for example, buffer size) also apply when running DECnet.

Running the NIE increases the traffic on the NI. If more than one copy is running simultaneously, normal operation on the NI could be severely affected.

C.2 RUN-TIME ENVIRONMENT REQUIREMENTS

The VAX-11 Level 2R NIE (EVDWC REV *.*) runs in the standard environment supported by the VAX-11 Diagnostic Supervisor.

- Hardware Required

  VAX-11 processor
  256Kb memory
  UNIBUS adapter
  DEUNA connected to an NI

- Software Required

  VMS Operating System (Version 3.0)
  DEUNA Driver
  VAX-11 Diagnostic Supervisor (REV 6.5 or later)

The PDP-11 standalone NIE (CZNID*) runs in the standard environments supported by the PDP-11 Diagnostic Run-Time Services (DRS).

- Hardware Required

  UNIBUS PDP-11 system
  32Kb memory
  DEUNA connected to an NI

- Software Required

  Diagnostic Runtime Services (DRS)

C.3 FUNCTIONAL DESCRIPTION

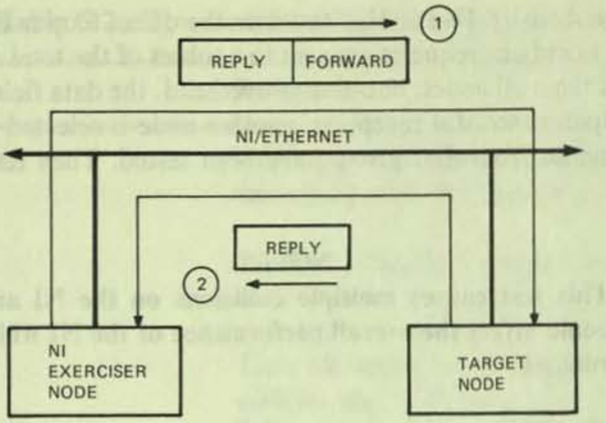
C.3.1 Unattended Mode

The Unattended Mode allows testing of the NI without operator interaction. Default parameters are used for the tests; the tests share a table of physical addresses of the nodes to be tested.

C.3.1.1 Build – The Build subroutine collects the physical addresses of all DEUNA on the NI.

C.3.1.2 Direct Loop Message Test – The ability of a node to respond to a loopback request is checked. A single loop request is sent to each of the nodes identified by the operator for testing. This message uses the minimum size buffer (36 bytes) and waits for a maximum of eight seconds for a reply. Three attempts are made to contact each node.

The structure of the Loop Message and an example of Direct Loopback testing is shown in Figure C-1. For direct looping, a Reply Message is encapsulated in a Forward Message. The Forward Message is sent by the NIE to the target node. The target node receives the Forward Message, extracts the Reply Message, and sends the Reply Message back to the NIE.



NOTE: NUMBERS INDICATE SEQUENCE IN WHICH MESSAGES ARE SENT

TK-9722

Figure C-1 Direct Loop Message Test Example

C.3.1.3 Pattern Test – This test sends six different loop direct messages to each node contained in the node table. Each of the six pattern types is used. During this test, each node will loopback one of the message types in the defualt section. The operator-directed section allows selection of the pattern to be used. Refer to Table C-1 for the Message Pattern Test types.

Table C-1 Message Pattern Test Message Types

Message Type	Message Pattern
Alphanumeric	!“#$%&’()*+,-./0123456789:;⟨=⟩?\abc etc.
Ones	Message of all 1s (111111111111111 ....)
Zeros	Message of all 0s (000000000000000 ....)
1Alt	Message of alternating 1s and 0s (10101010101010 ....)
0Alt	Message of alternating 0s and 1s (01010101010101 ....)
CCITT	“CCITT” psuedo-random test pattern
Operator selected*	Operator-chosen data pattern of less than 72 characters using A-Z, 0-9, and spaces (not used in pattern test).

* The operator-selected pattern is only available in the operator-directed section.

C.3.1.4 Multiple Message Activity Test – This test uses the direct loop maintenance feature to create a large volume of NI traffic. Loopback requests are sent to a subset of the total available nodes (for example, 10). Responses are received from all nodes, but to save overhead, the data field for only one of the nodes is checked for correctness. Upon successful reception, another node is selected (from the group of ten) and testing continues until all nodes from that group have been tested. Then testing continues with another group.

NOTE

This test causes multiple collisions on the NI and could affect the overall performance of the NI while running.

C.3.2 Operator-Directed Section

Selecting tests and test parameters is controlled by a command line interpreter (CLI). Section C.3.2.1 describes the commands an operator can issue when the operator-directed section is started.

C.3.2.1 Operator Conversation – This test uses nodes identified for testing by the operator as target and loop assist nodes (node-pair) and performs testing according to operator input or according to default parameters.

Using the proper commands, the operator can streamline the exerciser to test a particular node-pair. The test could be simple, using the default parameters of message numbers, message length, and data patterns, or more complex, using several messages, various message lengths, and different data patterns.

Only enough letters to make the command unique need be typed by the operator. Table C-2 summarizes the available commands.

Table C-2 Operator Command Summary

Command	Description
Help or ?	Displays a brief summary of NIE commands. There are no arguments. Format: NIE)Help or NIE)? On VAX-11 systems, more extensive help information is available through the Help facility of the Diagnsotic Supervisor. On PDP-11 systems, include more information (NIE)Help) to the operator-directed interface.
EXIT	Returns the operator to the Diagnostic Supervisor (either DR) or DS)). No switches or qualifiers. Format: NIE)Exit

Table C-2 Operator Command Summary (Cont)

Command	Description
SHOW	Prints physical addresses of nodes selected for testing and message parameters (either default or operator input).
	Format: NIE⟩Show ⟨argument⟩
Show Nodes	Lists all nodes in the Node table, including a physical address and a logical name assigned to the node by the NIE. Can be referenced by either physical address or the logical name. Logical names are assigned as N1, N2, N3, etc. The table also identifies the node as target or assist node as assigned by the operator. Unassigned nodes default to target.
Show Message	Lists the message type, message size, and message numbers currently selected.
Show Counters	Lists the counters of the host node.
RUN	Executes the test specified by the argument.
	Format: NIE⟩Run ⟨argument⟩/Pass=nm
Run Direct/pass=nm	Selects the test described in Section C,3.1.2.
Run Looppair/pass=nm	Selects the test described in Section C.3.2.4.
Run Pattern/pass=nm	Selects the test described in Section C.3.1.3.
Run All/pass=nm	Selects the test described in Section C.3.2.5. Allows the operator to select the number of passes for the selected test. If -1 is specified, the test runs continuously. Default=1.
MESSAGE	Allows the operator to change the default parameters of message type, message size, and message number.
	Format: NIE⟩Message=type/size=n/copies=n

Table C-2 Operator Command Summary (Cont)

Command	Description
Message/type/size=n/copies=m	Message types are explained in Table C-1. Message size is variable between 32 and 1466 bytes. Message copies = number of times the message is to be transmitted (1 to 10).
NODE	Allows the operator to enter nodes for testing. Format: NEI⟩Nodes addr/type
Node adr/Target	Adr argument is the the physical address of the node on the NI.
Node adr/Assist	The type argument can be either target or assist. This information in used for the Looppair test and is ignored for the All node test. If this argument is not specified, the default of target is used.
SUMMARY	Prints the summary message of conditions and errors as a result of testing (see Section C.3.2.6). The same summary information can be obtained by typing Summary (VAX-11 system) or Print (PDP-11 system). There are no switches or qualifiers for the Summary command. Format: NIE⟩Summary
BUILD	Builds a table of nodes described in Section C.3.2.2. No switches or qualifiers. To list the node table built from this section, use SUMMARY or SHOW NODES. Format: NIE⟩Build
CLEAR	Format: NIE⟩Clear ⟨argument⟩
Clear Node/adr	Removes a node from the node table.
Clear Node/all	Clears the node table.
Clear Message	Resets the message parameters to the default state.
Clear Summary	Clears the node summary table.

Table C-2 Operator Command Summary (Cont)

Command	Description
IDENTIFY adr*	Performs a request ID to the address included in the command line (see Section C.3.2.3). The argument adr should be a physical address. Format: NIE)Identify adr
SAVE	Saves the contents of the node table. For VAX-11 version of the NIE, the table is saved in file NIE.TBL. The PDP-11 NIE copies the node table into a secondary buffer within the diagnostic. The primary node table can then be modified without destroying the secondary node table. Use UNSAVE to restore the primary table. Format: NIE)Save
UNSAVE	Restores the contents of the node table. The VAX-11 version reads the most recent version of the file NIE.TBL. The PDP-11 version reads the node table from the secondary buffer into the primary buffer. Format: NIE)Unsave

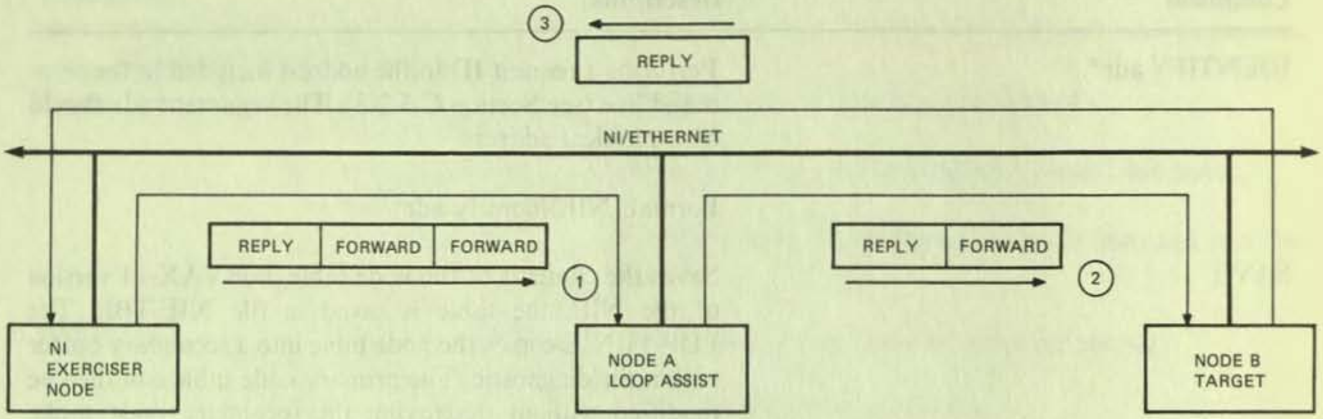
* adr is the physical address of a node on the NI.

C.3.2.2 Collect IDs (Build) – DEUNA nodes transmit a system ID message every eight to ten minutes. It is possible to identify all nodes on the NI and build a table of nodes by listening for the ID messages.

An estimated 40 minutes is required to collect a complete list of nodes on the NI. This test listens for IDs and builds a configuration table until a new node has not been added for 10 minutes or until the build is stopped by the operator. The maximum number of nodes in the node table is 100.

C.3.2.3 Request ID – In response to the operator-directed command IDENTIFY, a request ID is generated to the physical address identified as part of the command line. Three attempts are made to contact the node, and failure is reported to the operator. The information contained in the returned ID message is reported to the operator.

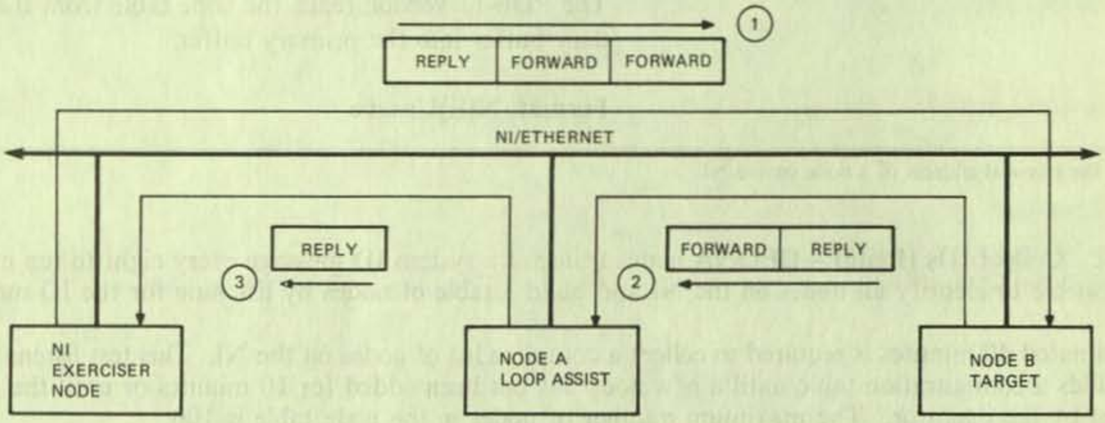
C.3.2.4 Pair-Node Testing – Using the operator-directed interface, the operator enters a pair of nodes for testing. One node is the target node and the other node is the loop assist node. This test uses the loop assist function of the DEUNA. This test can be run without running other parts of the NIE. Therefore, it is necessary to run the full range of loop testing to determine the node with problems. Each node is fully tested using transmit assist, receive assist, and full assist loopback testing. For examples of these tests and message formats, see Figures C-2, C-3, and C-4.



NOTE: NUMBERS INDICATE SEQUENCE IN WHICH MESSAGES ARE SENT

TK-8723

Figure C-2 Transmit Assist Loopback Testing Example



NOTE: NUMBERS INDICATE SEQUENCE IN WHICH MESSAGES ARE SENT

TK-8724

Figure C-3 Receive Assist Loopback Testing Example

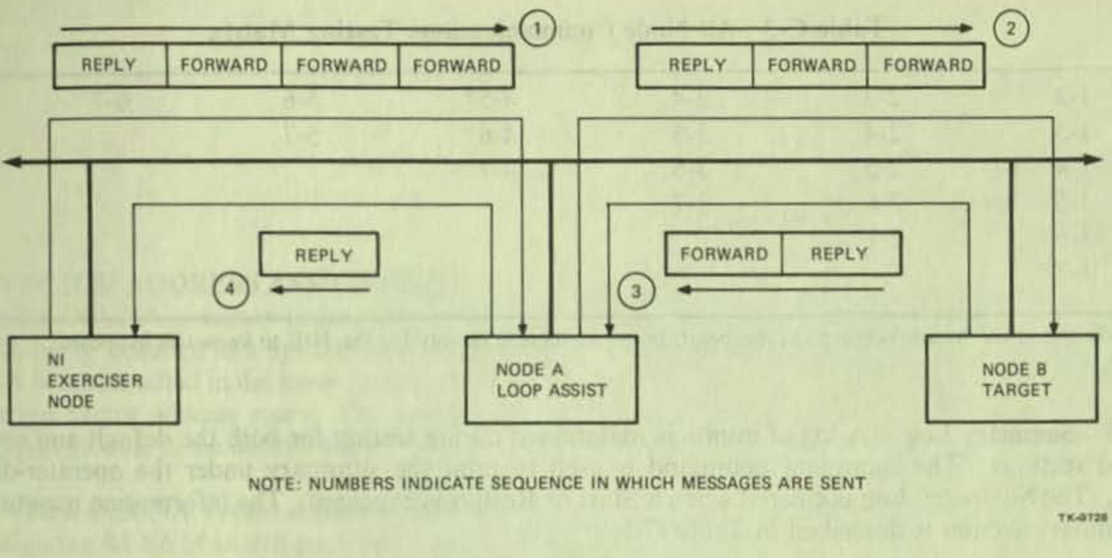


Figure C-4 Full Assist Loopback Testing Example

C.3.2.5 NI All Node Communications Test (End-to-End) – The All node test begins by doing a direct loop to all nodes in the table. If there is a single failure of this portion of the direct test, the All node test is aborted. The operator can then remove the offending node from the table, and restart the test.

Testing all nodes contained in the configuration table is performed using default parameters or operator input parameters. This test provides the most comprehensive testing of the NI. Testing is performed two nodes at a time by attempting two-way communication with a pair.

It is not possible to identify end nodes of an NI segment. Therefore, a test matrix is developed to assure that both end nodes have communicated. Testing each pair of nodes in a predetermined sequence assures that nodes physically positioned at opposite ends of a segment have been tested. This test occupies the longest test time of the NIE.

The formula for determining the number of subtests required is (n(n-1))/2. Figure C-5 is an example of a network with eight nodes. The number of subtests is 7(7-1)/2 = 21 (n=7 because it is not necessary to include the node running the exerciser.) To be certain of covering the entire NI, subtests need to be performed (see Table C-3).

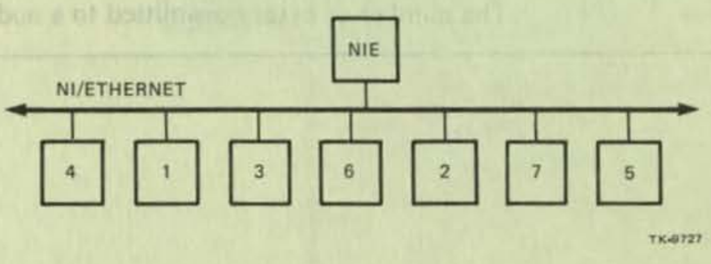


Figure C-5 Example Test Configuration for All Node Communications Test

Table C-3 All Node Communications Testing Matrix

1-2	2-3	3-4	4-5*	5-6	6-7
1-3	2-4	3-5	4-6	5-7	
1-4	2-5	3-6	4-7		
1-5	2-6	3-7			
1-6	2-7				
1-7					

*Complete end-to-end testing occurred at this point; however, there is no way for the NIE to know this happened.

C.3.2.6 Summary Log – A log of events is maintained during testing for both the default and operator-directed sections. The Summary command is used to print the summary under the operator-directed section. The Summary Log is cleared when a Start or Restart is executed. The information maintained in the summary section is described in Table C-4.

Table C-4 Summary Information

Name	Description
Node Physical Address	The physical address of the node on the NI.
Receives Not Complete	The number of packets transmitted without a corresponding reply.
Receives Complete	The number of packets transmitted and received successfully. (Note that messages sent do not always equal messages received if there are problems with the node or if traffic is high enough to cause dropped packets.)
Data Length Error	Number of packets with the bytes expected not equaling the number of bytes received.
Data Comparison Errors	Number of bytes received in error.
Bytes Compares	The number of bytes of data compared.
Bytes Transferred	The number of bytes transmitted to a node (data and header).

APPENDIX D
VECTOR ADDRESS (REVB)

D.1 VECTOR ADDRESS ASSIGNMENT

Assign the DEUNA a vector address from the reserved vector area of memory address space. The first DEUNA being installed in a system must be assigned the vector 120. For the second, and any subsequent DEUNA being installed in the same system, the vector address must be selected from the floating vector area of reserved vector address space. The vector address is assigned by configuring switch pack E62 on the M7792 port module to the desired vector.

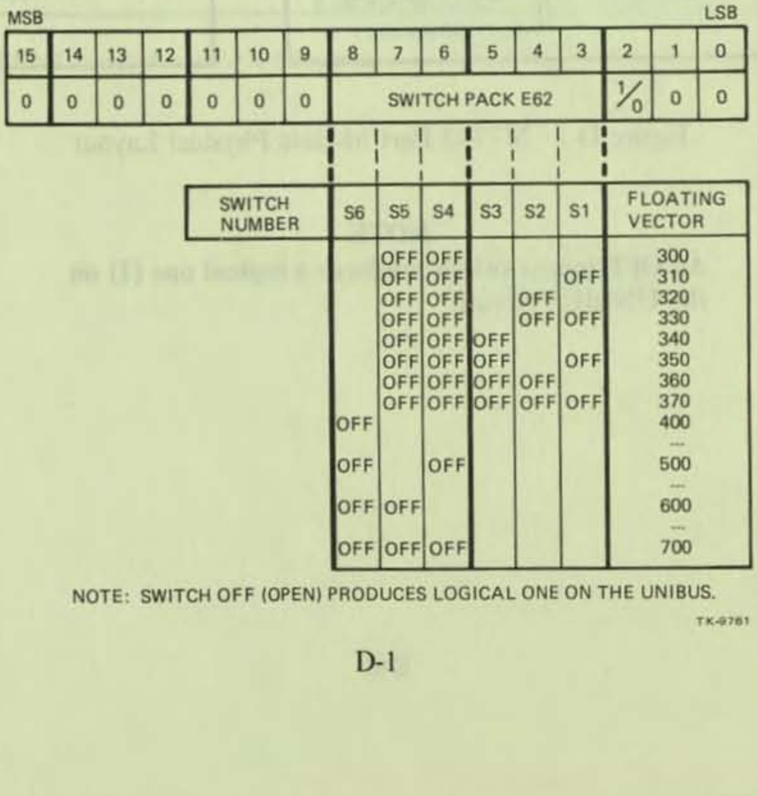
D.1.1 First DEUNA Vector Address (120) – Assign vector address 120 to the first DEUNA in the system by configuring S1-S6 of switch pack E62, on the M7792 port module, as shown below. Note that this vector is also used by the XY11. Refer to Figure D-1 for the location of E62 on the M7792 module.

M7792 - E62

S1	S2	S3	S4	S5	S6
ON	OFF	ON	OFF	ON	ON

D.1.2 Second DEUNA Vector Address (Floating Vector) – Assign a vector to the second (or subsequent) DEUNA by configuring S1-S6 of switchpack E62, on the M7792 port module, to the desired vector determined from the floating vector allocation. Refer to Table D-1 for the correlation between switch number and address bit. The ranking vector address assignment of the DEUNA is forty-seven (47). Refer to Appendix A for more information on floating vector allocation.

Table D-1 Floating Vector Assignment



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SWITCH PACK E62						1/0	0	0

SWITCH NUMBER	S6	S5	S4	S3	S2	S1	FLOATING VECTOR
		OFF	OFF				300
		OFF	OFF			OFF	310
		OFF	OFF		OFF		320
		OFF	OFF		OFF	OFF	330
		OFF	OFF	OFF			340
		OFF	OFF	OFF		OFF	350
		OFF	OFF	OFF	OFF		360
		OFF	OFF	OFF	OFF	OFF	370
	OFF						400
							—
	OFF		OFF				500
							—
	OFF	OFF					600
							—
	OFF	OFF	OFF				700

NOTE: SWITCH OFF (OPEN) PRODUCES LOGICAL ONE ON THE UNIBUS.

TK-9781

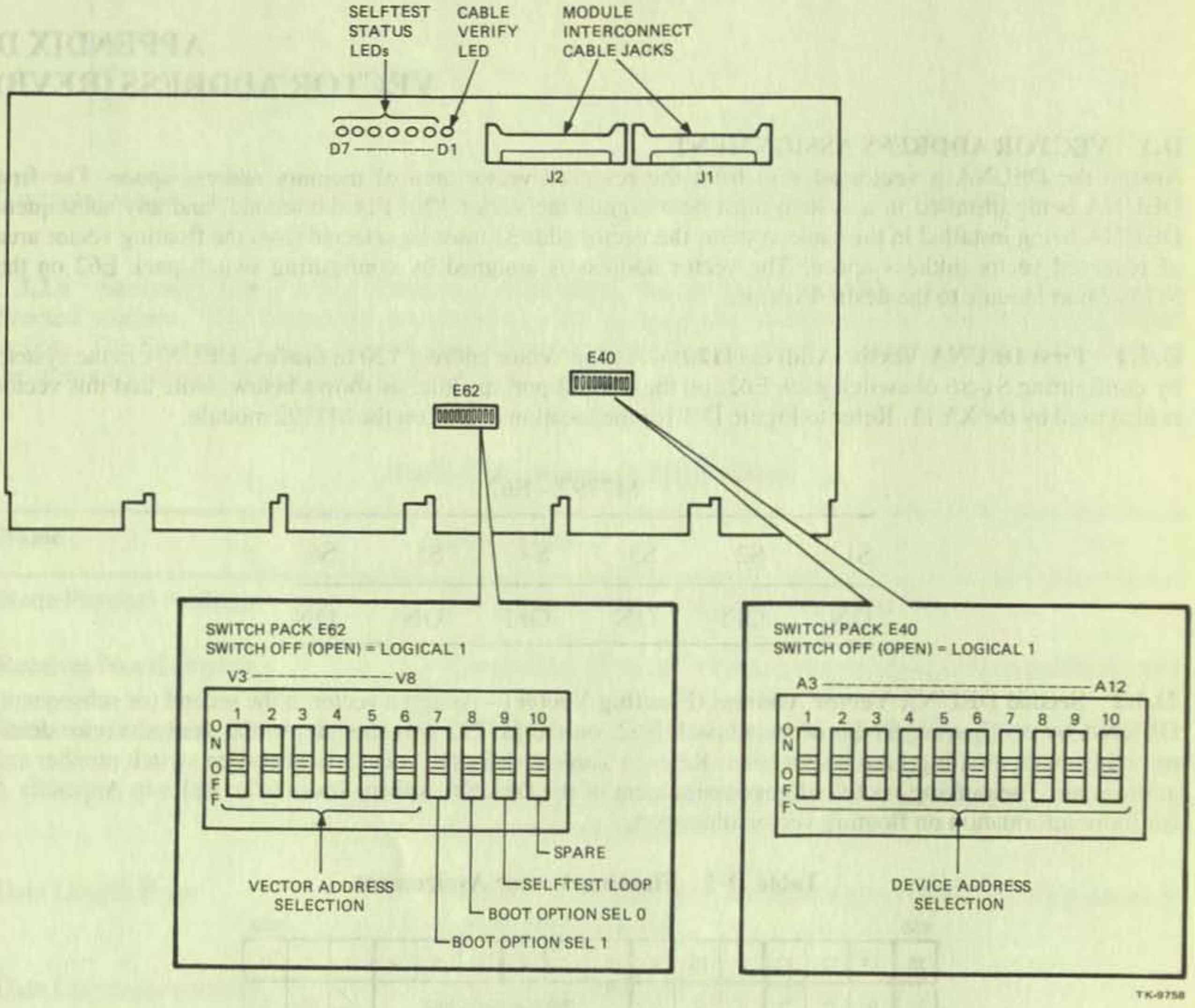


Figure D-1 M7792 Port Module Physical Layout

NOTE

An OFF (open) switch produces a logical one (1) on the UNIBUS circuit.

D.2 BOOT OPTION SELECTION (PDP-11 HOST SYSTEMS ONLY)

The DEUNA provides for remote booting and down-line loading of PDP-11 family host systems. These functions are switch selectable via two boot option select switches located on switch pack E62 on the M7792 port module.

NOTE

Refer to Appendix B for additional information on DEUNA remote booting and down-line loading.

When installing a DEUNA in a PDP-11 family host system, configure switches S7 and S8 on switch pack E62 (M7792 module) for the boot function desired. Table D-2 lists the switch settings and corresponding boot option functions. Refer to Figure D-1 for the location of E62 on the M7792 module.

When installing a DEUNA in a VAX-11 family host system, set both S7 and S8 on E62 (M7792 module) to the ON (disabled) postion.

NOTE

An OFF (open) switch produces a logical one (1). This is the ENABLED state of the switch function.

Table D-2 Boot Option Selection (M7792 E62 – S7 & S8)

BOOT SEL 1	BOOT SEL 0	Function
ON*	ON*	Remote boot disabled
OFF	ON	Remote boot with system load
ON	OFF	Remote boot with ROM
OFF	OFF	Remote boot with power-up boot and system load

* Switch settings for a DEUNA installed in a VAX-11 system.

D.3 SELF-TEST LOOP (FOR MANUFACTURING USE)

The self-test loop is provided on the DEUNA for use during manufacture testing. This is a switch selectable feature that allows the on-board self-test diagnostic program, once it is initiated, to continuously loop itself. This feature is controlled by S9 on switch pack E62 on the M7792 port module and should be disabled during installation.

When installing a DEUNA, disable the self-test loop feature by setting S9 on switch pack E62 (M7792 module) to the ON (closed) position, as indicated in Table D-3. Refer to Figure D-1 for the location of E62 on the M7792 module.

Table D-3 Self-Test Loop Switch (M7792 E62 – S9)

Position	Function
ON* (closed)	DISABLED
OFF (open)	ENABLED

* Switch setting for field operation.

APPENDIX E
DEUNA MICROCODE ECO PROCESS

E.1 INTRODUCTION

The microcode ECO process allows for the microcode of the DEUNA to be updated as network improvements are made. The support for microcode ECO's is as follows:

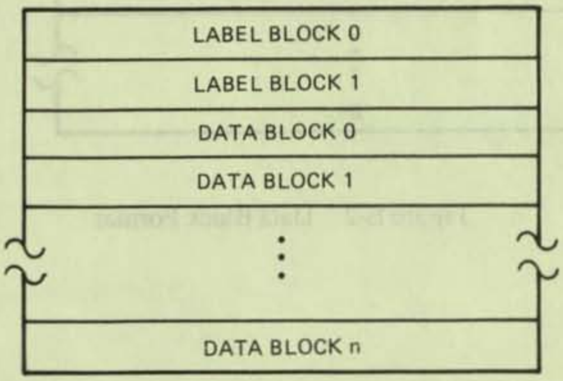
- Microcode ECO's will be included in system distribution kits and autopatch kits.
- A system utility will be provided that automatically reloads the patches on: reboot, recovery from power failures, and software execution of the self-test microcode.

The following sections explain the format of the ECO patch file and the programming steps required to load the patch into the DEUNA and execute it.

E.2 PATCH FILE FORMAT

The patch file consists of the standard RSX label blocks which are followed by data blocks. The format of the patch file is as follows:

1. Two label blocks called LABEL BLOCK 0 AND LABEL BLOCK 1 (Figure E-1).



NOTE: BLOCK = DISK BLOCK

TK-10110

Figure E-1 Patch File Format

2. The two label blocks are followed by the data blocks. Each data block may contain a number of microcode patches (Figure E-2). The last ''patch'' of the last data block in the file will contain a Byte Count of –1 and the DEUNA Start Address.

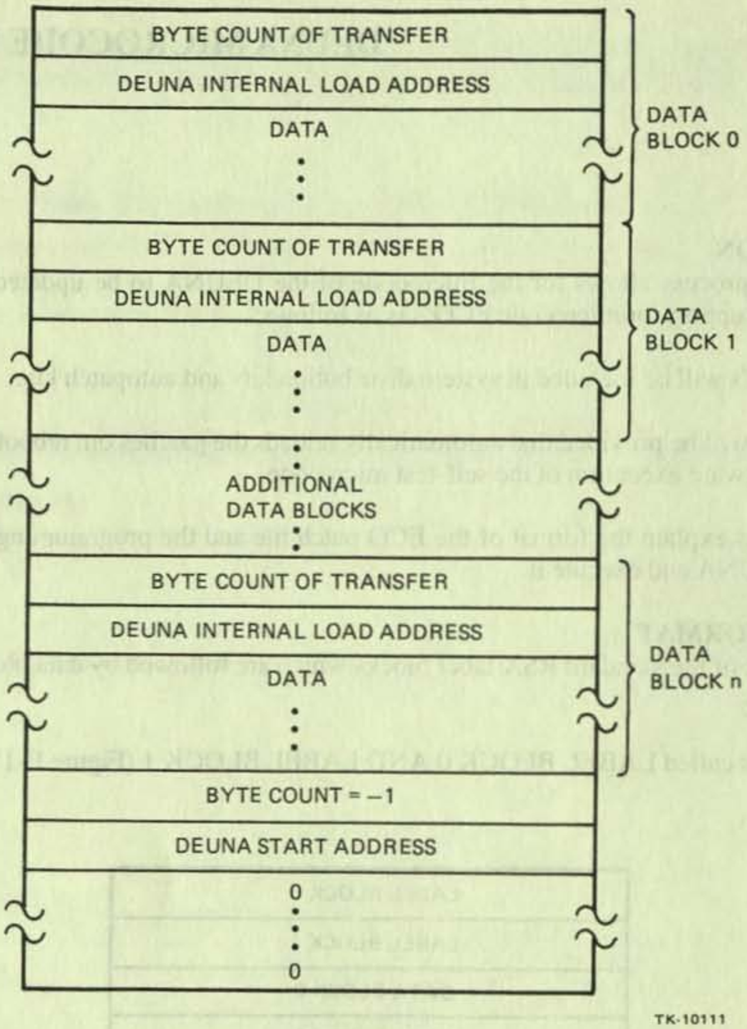


Figure E-2 Data Block Format

NOTE

No patch can extend over a single block of the file Data Block. This means that large changes in the microcode will have to be divided into a series of smaller changes, so that they each fit into a single Data Block.

There can be unused space at the end of each Data Block. This space will be filled with 0's.

E.3 PATCH PROCEDURE

The procedure for loading and executing a patch to the DEUNA microcode is as follows:

1. The DEUNA must be in the READY state.

2. Use the LOAD Ancillary Command (21) to load each patch into the DEUNA. The byte count and Internal load address supplied in each patch are used to set up the LOAD command.

3. Repeat the previous step until all the patches, contained in the file, are loaded into the DEUNA.

4. Using the DEUNA Starting Address supplied in the last patch, issue the LOAD and START MICRO-ADDRESS Ancillary Command (1) to execute the patch.

DEUNA USER'S GUIDE

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc? Is it easy to use?__________

What features are most useful?__________

What faults or errors have you found in the manual?__________

Does this manual satisfy the need you think it was intended to satisfy?__________

Does it satisfy *your* needs?__________ Why?__________

Please send me the current copy of the *Documentation Products Directory*, which contains information on the remainder of DIGITAL's technical documentation.

Name__________ Street__________
Title__________ City__________
Company__________ State/Country__________
Department__________ Zip__________

Additional copies of this document are available from:

Digital Equipment Corporation
Accessories and Supplies Group
P.O. Box CS2008
Nashua, New Hampshire 03061

Attention: Documentation Products
Telephone: 1-800-258-1710

Order No. EK–DEUNA–UG–001

Fold Here

Do Not Tear – Fold Here and Staple

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Educational Services/Quality Assurance
12 Crosby Drive, BU/E08
Bedford, MA 01730

Digital Equipment Corporation • Bedford, MA 01730

EK-DEUNA-TM-PRE

DIGITAL ETHERNET UNIBUS Network Adapter Technical Manual

digital

EK-DEUNA-TM-PRE

DIGITAL ETHERNET UNIBUS Network Adapter Technical Manual

PRELIMINARY

Prepared by Educational Services
of
Digital Equipment Corporation

Preliminary Edition, December 1982

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

Printed in U.S.A.

This document was set on DIGITAL's DECset Integrated Publishing System.

CONTENTS

Page

CHAPTER 1 INTRODUCTION

CHAPTER 2 PORT MODULE FUNCTIONAL DESCRIPTION

CONTENTS (Cont)

Page

CHAPTER 3 LINK MODULE FUNCTIONAL DESCRIPTION

CONTENTS (Cont)

FIGURES

FIGURES (Cont)

TABLES

1.1 SCOPE

This chapter provides an introduction to the DIGITAL ETHERNET UNIBUS Network Adapter (DEUNA). A brief overview of the ETHERNET local area network is included, followed by a description of the DEUNA, its operation, and specifications. Additional documents related to this manual are listed for the reader who wishes more information about the ETHERNET, the DEUNA, or local area networks.

1.2 ETHERNET OVERVIEW

The ETHERNET is a local area network that provides a communications facility for high-speed data exchange among computers and other digital devices located within a moderately sized geographic area. It is intended primarily for use in such areas as office automation, distributed data processing, terminal access, and other situations requiring economical connection to a local communication medium carrying traffic at high-peak data rates.

The primary characteristics of ETHERNET include:

Topology	Branching bus.
Medium	Shielded coaxial cable, Manchester encoded digital base-band signaling.
Data Rate	1Ø million bits per second.
Maximum Separation of Nodes	2.8 kilometers (1.74 miles).
Maximum Number of Nodes	1,Ø24
Network Control	Multiaccess -- fairly distributed to all nodes.
Access Control	Carrier Sense, Multiple Access with Collision Detect (CSMA/CD).
Allocation	Packet length from 64 to 1518 bytes (includes variable data field of from 46 to 15ØØ bytes).

The ETHERNET, like other local area networks, falls into a middle ground between long-distance, low-speed networks that carry data for hundreds or thousands of kilometers and specialized, very high-speed interconnections that are generally limited to tens of meters. Using a branching bus topoloy, ETHERNET provides a local area communications network allowing a 1ØM bits/s data rate over a coaxial cable at a distance of up to 2.8 km (1.74 mi).

A single ETHERNET can connect up to 1,024 nodes together for a local point-to-point/multipoint network. An example of a typical large-scale ETHERNET configuration is shown in Figure 1-1.

Rules for configuring ETHERNET are derived from certain limits that are imposed on the physical channel to ensure the optimal performance of the network. The maximum configuration for an ETHERNET is as follows:

- A segment of coaxial cable can be a maximum of 500 meters (1640.5 feet) in length. Each segment must be terminated at both ends in its characteristic impedance.

- Up to 100 nodes can be connected to any segment of the cable. Nodes on a cable segment must be spaced at least 2.5 meters (8.2 feet) apart.

- The maximum length of coaxial cable between any two nodes is 1,500 meters (4921.5 feet).

- The maximum length of the transceiver cable between a transceiver and a controller is 50 meters (164.05 feet).

- A maximum of 1,000 meters (3281 feet) of point-to-point link is allowed for extending the network.

- Repeaters can be used to continue signals from one cable segment of the ETHERNET to another. A maximum of two repeaters can be placed in the path between any two nodes.

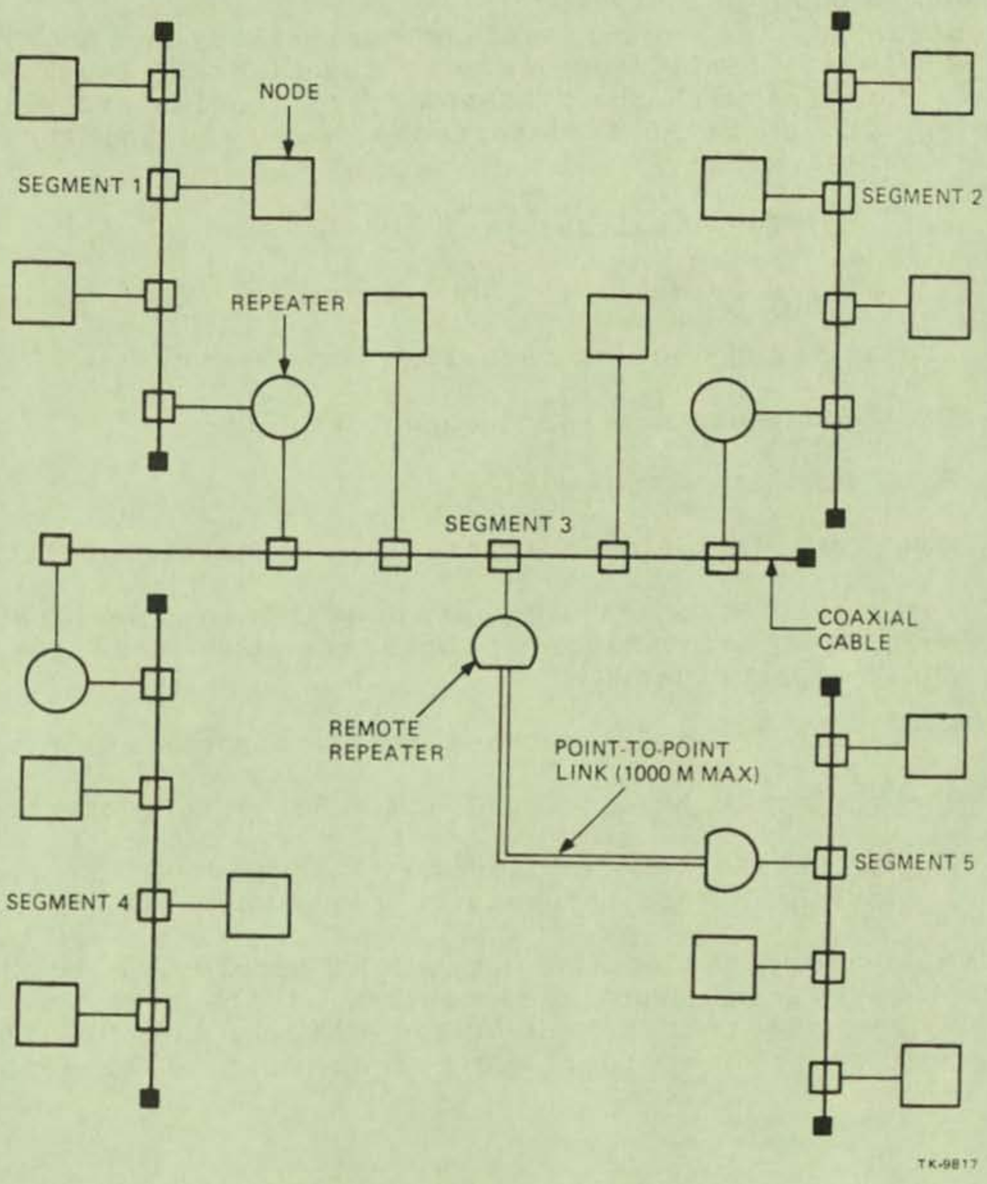


Figure 1-1 Typical Large-Scale ETHERNET Configuration

1.3 DEUNA GENERAL DESCRIPTION

The DEUNA is a data communications controller used to interface VAX-11 and PDP-11 family computers to the ETHERNET local area network. It complies with the ETHERNET specification and allows communication with up to 1024 addressable devices using the ETHERNET shielded coaxial cable.

Features of the DEUNA include:

- 10M bits/s transmission and reception,
- Transmit and receive data link management,
- Data encapsulation and decapsulation,
- Data encoding and decoding,
- Down-line loading and remote load detect capabilities,
- Internal ROM based microdiagnostics to facilitate diagnosis and maintenance of both the DEUNA and the DIGITAL H4000 transceiver,
- Collision detection and automatic retransmission,
- 32-bit Cyclic Redundancy Check (CRC) error detection, and
- 32 KB (16 KW) buffer for continuous datagram reception, transmission, and maintenance requirements.

The DEUNA is comprised of two hex-height modules, a bulkhead interconnect panel, and associated cables. It physically and electrically connects to the ETHERNET cable via the DIGITAL H4000 transceiver and the appropriate transceiver cable as shown in Figure 1-2.

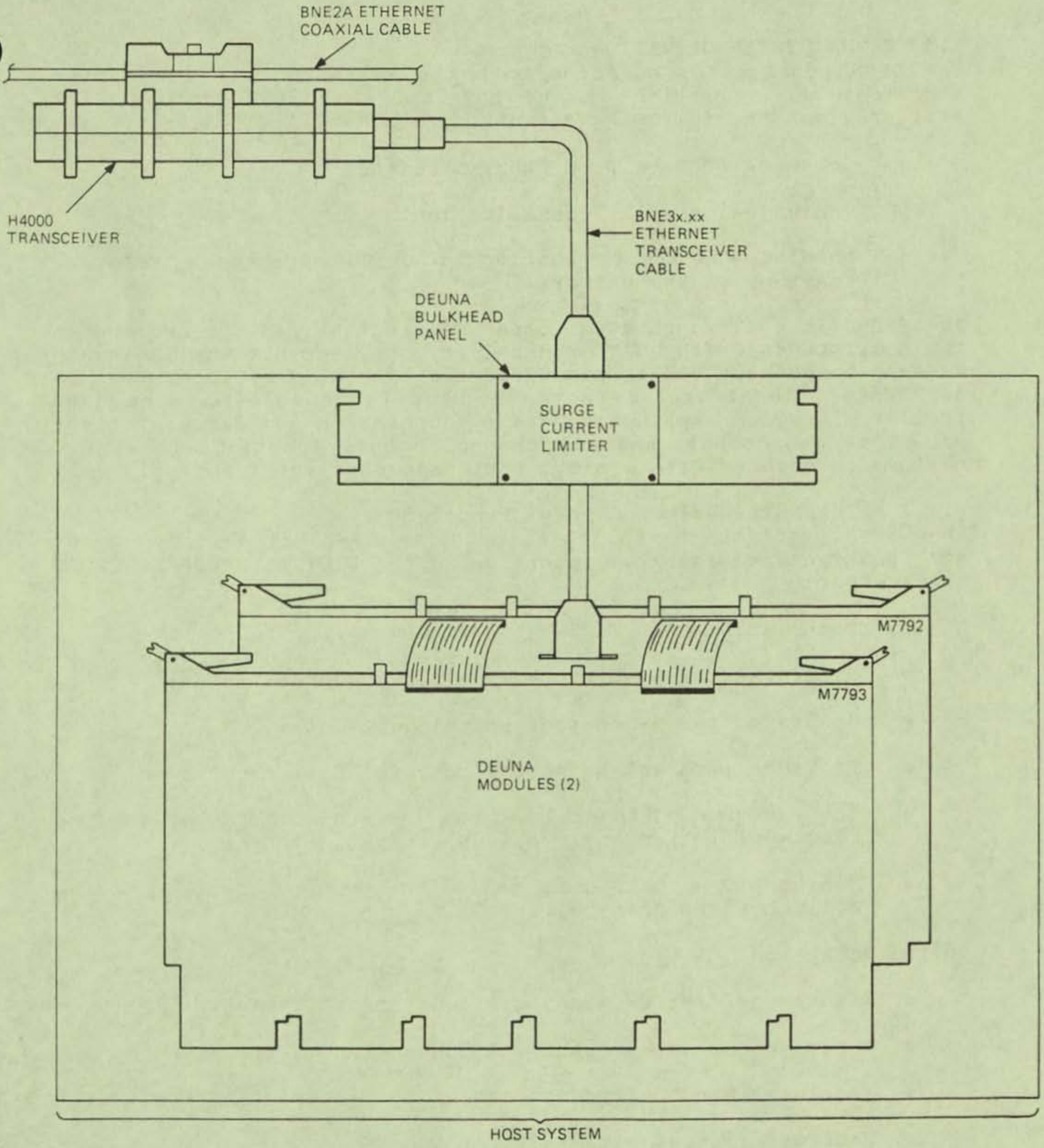


Figure 1-2 DEUNA to ETHERNET Connection

1.4 DEUNA SYSTEM OPERATION

The DEUNA controller performs both the ETHERNET data link layer functions and a portion of the physical channel functions. It also provides the following network maintainability features.

- Loopback of data from other stations.
- Individual system identification.
- Loading and remote booting of UNIBUS systems from other stations on the network.

The DEUNA is a microprocessor based device that, when connected to the DIGITAL H4000 ETHERNET transceiver, provides all the logic necessary to connect VAX-11 and UIBUS PDP-11 family minicomputers to an ETHERNET local area network (Figures 1-3 and 1-4). The controller microcode implements data encapsulation and decapsulation, data link management, and all channel access functions to ensure maximum throughput with minimum host processor intervention.

1.4.1 ETHERNET Physical Channel Functions

The DEUNA provides the following specific ETHERNET physical channel functions necessary to interface to the DIGITAL H4000 ETHERNET transceiver:

During Transmission

- Generates the 64-bit preamble for synchronization.
- Generates the Manchester encoding of data.
- Provides parallel-to-serial conversion of the frame.
- Ensures proper channel access by monitoring and sensing the carrier from any stations' transmission.
- Monitors the self-test collision detect signal from the DIGITAL H4000 transceiver.

During Reception

- Senses carrier from any stations' transmission.
- Provides serial-to-parallel conversion of the frame.
- Performs Manchester decoding of the incoming bit streams.
- Buffers received frames.
- Synchronizes to the preamble and removes it prior to processing.

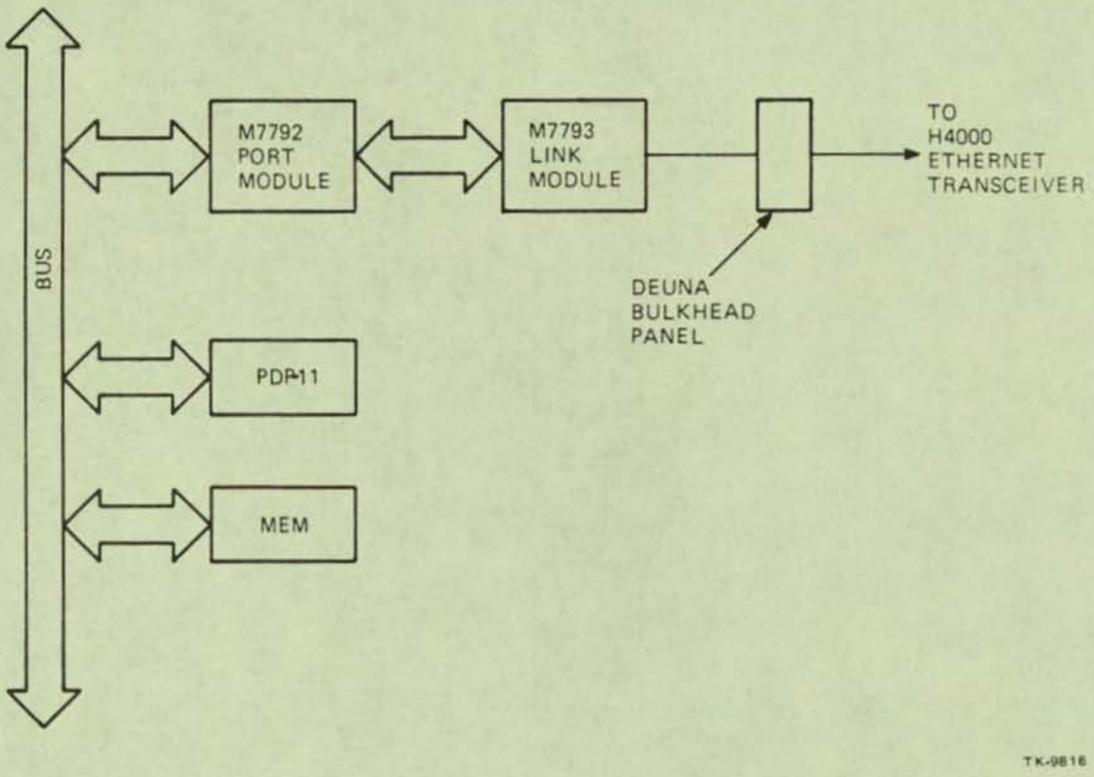


Figure 1-3 PDP-11 Host System Block Diagram

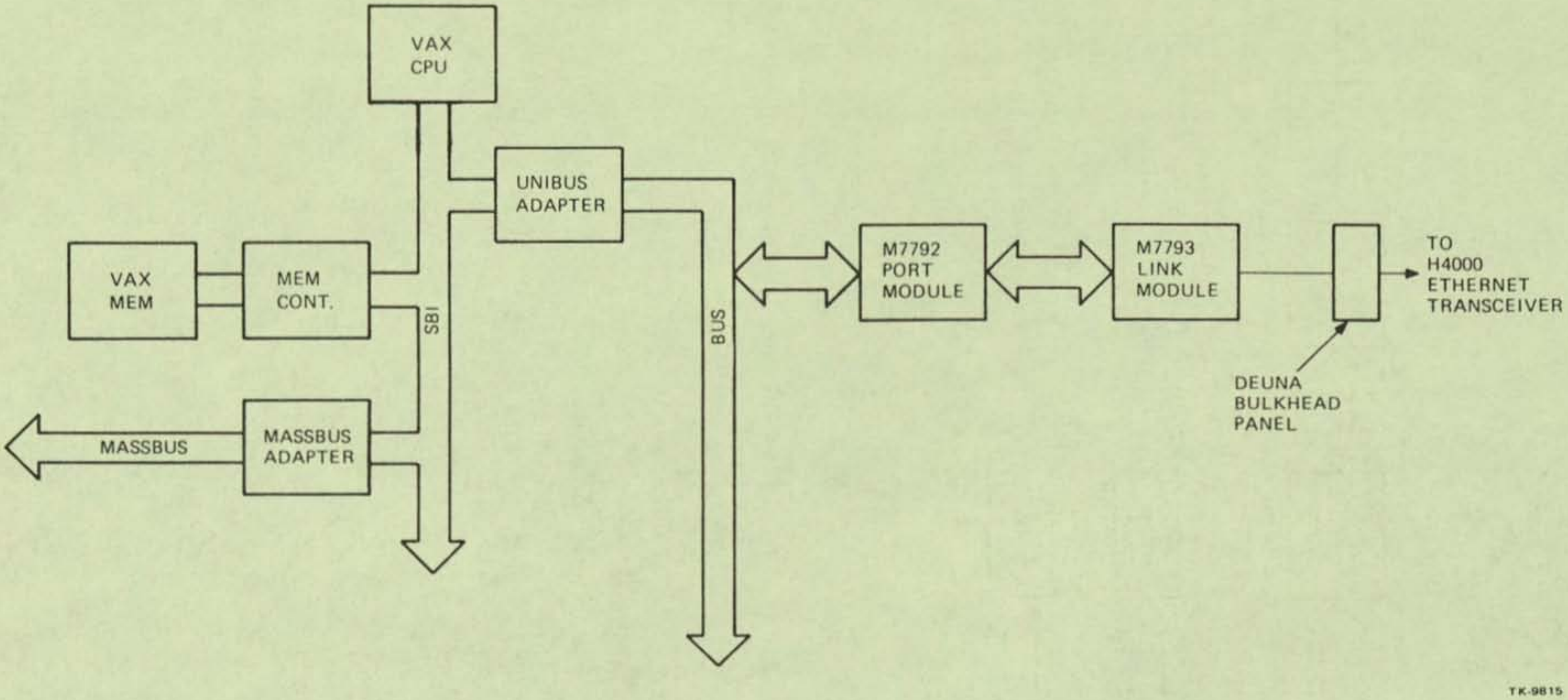


Figure 1-4 VAX-11 Host System Block Diagram

1.4.2 ETHERNET Data Link Functions

The DEUNA provides the following specific ETHERNET data link layer functions.

- Calculates the 32-bit CRC value and places it in the frame sequence field upon transmission
- Attempts automatic retransmission upon collision detection
- Checks incoming frames for proper CRC value
- Performs all address filtration

1.4.3 Data Encapsulation

The ETHERNET frame format for the transmission of data packets is shown in Figure 1-5. Each frame begins with a 64-bit preamble, that is used for synchronization by the receiving station, and ends with a 32-bit frame check sequence. Frames are separated by a specified minimum spacing period of 9.6 microseconds.

The destination address field contains the address(es) of the station(s) where the packet is sent. The address may represent: the physical or logical address of a particular station or group of stations; a multicast, or group address, associated with a set of stations; and a broadcast address for broadcast to all stations on the network.

The source address field specifies the physical address of the transmitting station. Each DEUNA has a unique 48-bit address value determined during manufacture. This value is called the default physical address. The system software can override this value and insert a more apropriate logical address into the source address field upon transmission.

The type field is specified for use by high-level network protocols and it indicates how the content of the data field is to be interpreted. The type field indicates the higher level architecture that can further decapsulate the data.

The data field may have between 46 and 15ØØ bytes of data. The DEUNA can be initialized to automatically insert null characters if the amount of data is less than the minimum 46 byte data size.

The frame check sequence contains a 32-bit Cyclic Redundancy Check (CRC) value that is determined and inserted by the DEUNA during transmission.

1.4.4 Data Decapsulation

The DEUNA continuously monitors the signals transmitted by the DIGITAL H4ØØØ transceiver. After sensing a carrier, the preamble sequence of the received frame is used by the controller for synchronization. It then processes the destination address field through a hardware comparator to determine whether or not the incoming frame is intended for its station. The DEUNA accepts only

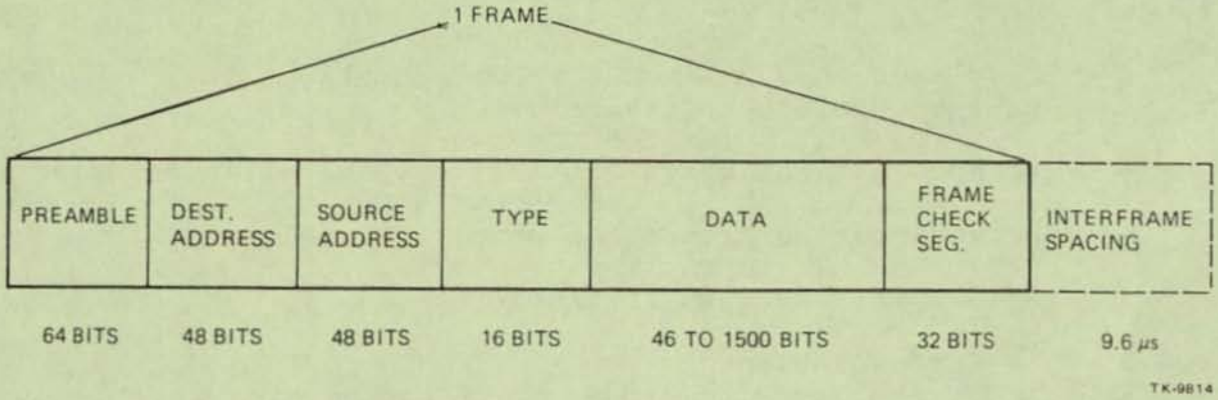


Figure 1-5 Format of an ETHERNET Data Packet

frames that have a destination address that matches one of the following types of address.

1. The physical address of the station

2. The broadcast address for all stations

3. One of the 10 multicast group addresses that the user may assign to the DEUNA

4. Any multicast address

5. All addresses, when desired

The DEUNA performs a hardware comparison of the 48-bit destination address to determine if there is a match with the station's physical address or with one of the ten user designated logical multicast addresses. If necessary, all multicast addresses may be passed to higher level software for decoding when more than ten multicast address groups are required by the user.

To assist in network management functions and to aid in fault diagnosis, the DEUNA can operate in a mode that effectively disregards the internal address filter logic. This allows all frames received from the network to be accepted. The DEUNA verifies the integrity to the received data by recalculating the 32-bit CRC value and comparing it with the CRC that is obtained from the received frame.

1.4.5 Link Management

The method utilized by the ETHERNET for channel access is called carrier sense, multiple access with collision detect (CSMA/CD). The DEUNA controls all of the link management functions necessary to successfully place or remove a frame of data on the ETHERNET network. These functions include:

Carrier Deference

The DEUNA monitors the physical channel for traffic and when the channel is busy, refers to the passing frame by delaying any transmission of its own.

Collision Detection

Collisions occur when two of more controllers attempt to transmit data simultaneously on the channel. The DEUNA monitors the collision sense signal generated by the DIGITAL H4000 transceiver. When a collision is detected, the DEUNA contiues to transmit to ensure that all network stations detect the collision.

Collision Backoff Retransmission

When a controller has attempted transmission and encountered a collision on the channel, it attempts a retransmission a short time later. The schedule for retransmission is determined by a controlled randomization process. The DEUNA attempts to transmit a total of sixteen times and reports an error if it is not successful.

1.4.6 Diagnostics and Maintenance

The DEUNA utilizes both microdiagnostics and extensive system and network diagnostics to greatly minimize the time to isolate and diagnose a network communication fault. On-board self-test microdiagnostics automatically perform a test of the major DEUNA component logic both upon powerup and at the user's discretion. Light emitting diodes on the edge of the port module (M7792) provide an indication of a specific module problem.

The DEUNA does not allow itself to transmit significantly longer than the maximum ETHERNET frame transmit period. It contains an automatic control to prevent monopolizing the ETHERNET channel. The controller can differentiate between normal frame collisions on the physical channel and cable shorts or cable opens. A built-in Time Domain Reflectometry (TDR) circuit is utilized to determine the type of cable fault and its approximate location.

The controller continuously monitors the power applied to the DIGITAL H4000 transceiver to ensure compliance with the transceiver requirements. In addition, the H4000 provides a positive functional verification (heartbeat) after every attempted transmission which indicates its proper operation, including the collision sense circuitry.

Comprehensive system diagnostics provide loopback capability through the DEUNA, transceiver, or the ETHERNET network itself. The DEUNA allows remote stations to loopback through it once the DEUNA has successfully passed the the on-board self-test microdiagnostic. This provides both a local and remote station diagnostic capability. Network error conditions are detected and statistics tabulated for use by higher level network management applications.

1.5 DEUNA SPECIFICATIONS

The DEUNA specifications are outlined in Table 1-1.

Table 1-1 DEUNA Specifications

Specification	Description
Performance	
Operating Mode	Half-duplex
Data Format	ETHERNET specification
Date Rate	10M bits/s
Network Specifications	1024 stations maximum
UNIBUS Conductor Loading	
Module Pair	4 dc loads 2 ac loads
DC Power Requirements	
Port Module	+5 V, 7.0 A
Link Module	+5 V, 9.0 A
	-15 V, 2.0 A (for H4000 transceiver)
Physical Size	
Port and Link Modules	Height (hex): 21.4 cm (8.4 in) Length: 39.8 cm (15.7 in)
Cable Interface Panel	Height: 10.6 cm (4.0 in) Length: 10.6 cm (4.0 in)
Transceiver Cables available in 5 m (16.4 ft), 10 m (32.8 ft), or 20 m (65.6 ft) lengths.	
BNE3A-XX	Low loss PVC jacket/straight connector
BNE3B-XX	Low loss PVC jacket/right angle connector
BNE3C-XX	Low loss TEFLON* jacket/straight connector
BNE3D-XX	Low loss TEFLON* jacket/right angle connector

*TEFLON is a trademark of Dupont de Nemours & Co., Inc.

Table 1-1 DEUNA Specifications (Cont)

Specification	Description
Operating Environment	
Temperature	5°C to 5Ø°C (41°F to 122°F)
Relative Humidity	1Ø to 9Ø% (noncondensing)
Wet Bulb Temperature	32°C (9Ø°F) maximum
Altitude	Sea level to 2.4 km (8,ØØØ ft)
Shipping Environment	
Temperature	-4Ø°C to Ø°C (-4Ø°F to 151°F)
Relative Humidity	Ø to 9Ø% (noncondensing)
Altitude	Sea level to 9 km (3Ø,ØØØ ft)

1.6 RELATED DOCUMENTS

Table 1-2 provides a list of documents related to this manual.

Table 1-2 Related Hardware and Software Documents

Title	Document Numbers
DEUNA User's Guide	EK-DEUNA-UG
H4ØØØ Technical Description	(TBS)
The ETHERNET, A Local Area Network, Data Link Layer and Physical Layer Specifications	AA-K759A-TK
ETHERNET Installation	(TBS)
Introduction to Local Area Networks	EB-22714-18
PDP-11 Bus Handbook	EB-17525

DIGITAL personnel may order hardcopy documents from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA Ø1532

Attn: Publishing and Circulation Services (NRØ3/W3)
Order Processing Section

Customers may order hardcopy documents from:

Digital Equipment Corporation
Accessories and Supplies Group
Cotton Road
Nashua, New Hampshire Ø3Ø6Ø

For information call: 1-8ØØ-257-171Ø

Information concerning microfiche libraries may be obtained from:

Digital Equipment Corporation
Micropublishing Group (BUO/E46)
12 Crosby Drive
Bedford, MA Ø173Ø

CHAPER 2
PORT MODULE
FUNCTIONAL DESCRIPTION

2.1 OVERVIEW

The port module (M7792) is the microprocessor controlled interface between the UNIBUS bus and the link module of the DEUNA. The logic on the port module is divided into three basic functional areas.

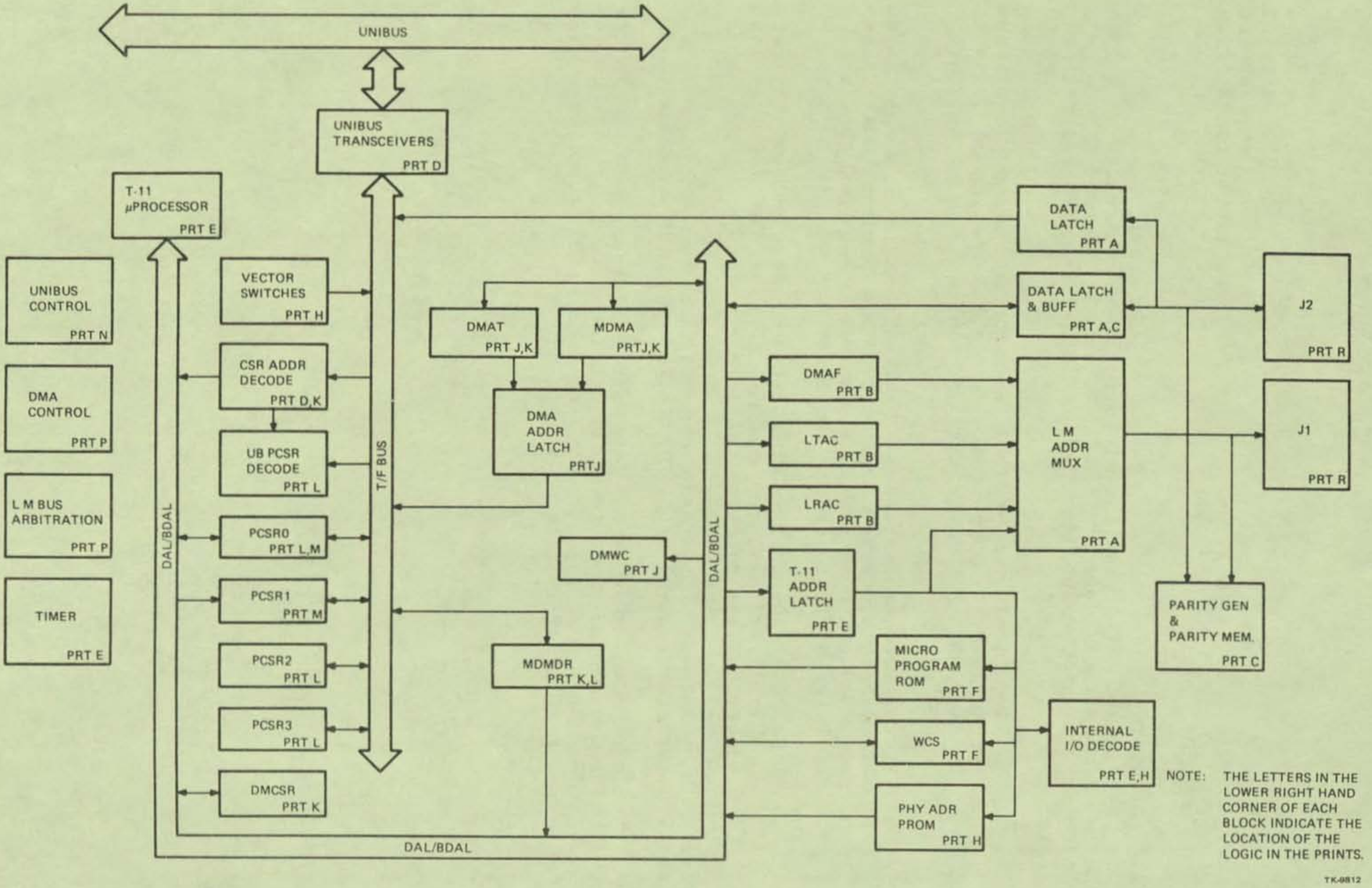
1. UNIBUS Interface -- This section contains the UNIBUS transceiver, port control and status registers (PCSRs), DMA control, UNIBUS interrupt control logic, and UNIBUS control.

2. Microprocessor Section -- This section is made up of the T11 microprocessor, 8K words of ROM for microprogram storage, 4K words of writable control store (WCS), internal register address decode, and timer.

3. Link Memory Control -- This section contains the link memory arbitration logic, control for the 16K words of link memory and the port-to-link interconnect.

The port module is a hex-height module that is installed in a small peripheral controller (SPC) slot of a UNIBUS backplane.

A functional block diagram of the port module is shown in Figure 2-1. The letters in the lower right corner of each block of the diagram indicate the page in the engineering drawings where the logic for that block is located.



NOTE: THE LETTERS IN THE LOWER RIGHT HAND CORNER OF EACH BLOCK INDICATE THE LOCATION OF THE LOGIC IN THE PRINTS.

Figure 2-1 PORT Module Functional Block Diagram

2.2 UNIBUS INTERFACE

The UNIBUS interface logic on the port module is used to control the transfer of data between the host processor and the DEUNA. This logic generates the signals required of a bus master and bus slave on the UNIBUS. The DEUNA functions as bus master when data is to be transferred to or from the host processors memory via direct memory access (DMA). The DEUNA performs DMAs for the transfer of:

- Data received from the ETHERNET and
- Data to be transmitted on the ETHERNET.

The DEUNA functions as a bus slave when the host processor accesses the port control and status registers (PCSRs) for the transfer of control and status information.

> **NOTE**
> **For a detailed explanation of UNIBUS architecture and protocol, refer to the PDP-11 Bus Handbook (EB-17525).**

The port also controls the UNIBUS ACLO signal. It does this by setting a bit in the link mode register on the link module (see Section 3.3.2). This is used to get control of the host processor during a down-line load.

2.2.1 DMA Control

The DMA control logic is divided into two sections:

1. RX DMA -- Used when a message has been received from the ETHERNET and is ready to be transferred to the host processors memory.
2. T11 UNIBUS DMA -- Used when the T11 has to:
    - Read the ring structures in host memory,
    - Read data buffers in host memory for transmission on the ETHERNET, and
    - Write status information into the data buffers in host memory when the transmission is finished.

The control of each of these processes is implemented via programmed array logic (PAL) with the T11 UNIBUS DMA having a higher priority than the RX DMA process. This priority is established because the T11 UNIBUS DMA process transfers its data in smaller segments and therefore does not use the UNIBUS for long periods of time. This results in little effect on the RX DMA process and helps to maximize thoughput.

A description of each of the PALs used in the DMA control is contained in the engineering drawings for the DEUNA.

2.2.1.1 **RX DMA** -- The DEUNA transfers received messages to host memory via the UNIBUS using DMAs. This is done asynchronous to the process or processes going on in the DEUNA. The port microprocessor (T11) starts the DMA transfers by loading a group of registers with the necessary address and word count information. Once this information is loaded, the T11 starts the DMA process by setting a bit in the DMA control and status register (DMCSR). This starts the DMA transfers under the control of the RX DMA PAL and the UNIBUS control logic.

The following registers, located on the port module, are used for the RX DMA process:

- DMCSR -- DMA control and status register,
- DMAT -- DMA-to-address register,
- DMAF -- DMA-from-address register, and
- DMWC -- DMA word count register.

The T11 controls the transfer of data from the buffers located on the link module to the host memory in the following manner:

1. The T11 loads the DMAT, DMAF, and the DMWC with the proper information.
2. The T11 sets the DMA GO bit in the DMCSR.
3. The DMA logic takes over and moves the data via NPRs to the host memory.
4. When all the data is transferred or when the RX DMA logic receives an error, it interrupts the T11.

The RX DMA logic only transfers words on the UNIBUS. The host software is responsible for throwing away the extra byte when transferring an odd byte buffer.

A description and layout of each of the registers used is given in the following sections.

2.2.1.1.1 **DMA Control and Status Register (DMCSR)** -- The DMCSR is used by the T11 to enable the DMA logic and to report DMA status to the T11. Figure 2-2 shows the DMCSR bit format and Table 2-1 gives a description of each of the bits.

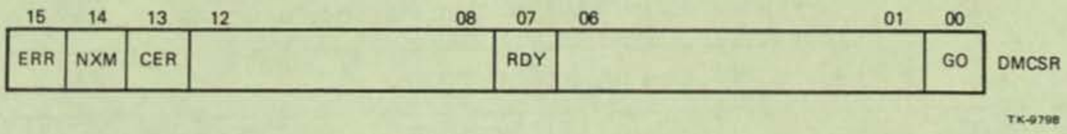


Figure 2-2 DMCSR Bit Format

Table 2-1 DMCSR Bit Descriptions

Bit	Field	Description
DMCSR<ØØ>	GO	Go Bit -- This bit is set after the address and word count are loaded. On setting the DMA, the engine begins to arbitrate for the UNIBUS and starts data transfer to host memory.
DMCSR<Ø7>	RDY	Ready Bit -- This bit creates an interrupt to the T11 to indicate that the word count has expired and the current DMA process is complete.
DMCSR<13>	CER	Collision Error -- When set indicates that the heartbeat from the H4ØØØ or similar transceiver was not detected.
DMCSR<14>	NXM	Non-Existent Memory -- When set causes the DMA logic to interrupt the T11. Indicates a UNIBUS timeout to the address contained in the DMA-to-address register.
DMCSR<15>	ERR	DMA Logic Error -- Set when UPE or NXM are set.

2.2.1.1.2 DMA to Address Registers (DMATØ and DMAT1) -- The DMAT registers are loaded by the T11 with the starting address of the receive buffer in host memory. DMATØ contains the lower 16 bits of the address. DMAT1 contains the upper 2 bits of the address. These registers are a 17-bit counter that is incremented by two after each NPR cycle.

NOTE

Bit Ø of DMATØ is always a Ø. This is because the RX DMA logic only performs word transfers.

Figure 2-3 shows the format of each of the registers.

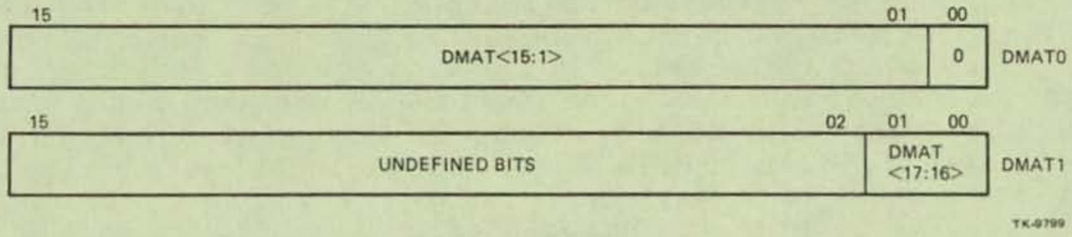


Figure 2-3 DMAT Bit Format

2.2.1.1.3 DMA-from-Address Register (DMAF) -- The DMAF register contains the receive buffer address in the link memory from which data is to be transferred. It is made up of a register for the upper four bits and a counter for the lower ten bits. The T11 loads the upper four bits from the link completed buffer address FIFO (refer to Section 2.4.3 for an explanation of the LCBAF). When the upper four bits are loaded, the lower ten bits are cleared. The counter is incremented by two after each NPR cycle. The address cannot overflow into the next buffer. Figure 2-4 shows the bit format of the DMAF register.

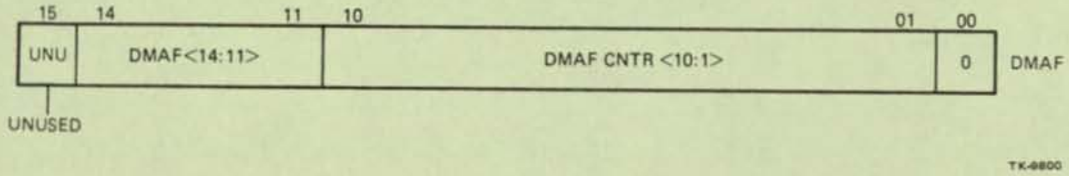


Figure 2-4 DMAF Bit Format and Descriptions

2.2.1.1.4 DMA Word Count Register (DMWC) -- The DMWC is loaded by the T11 with the number of words to be transferred to host memory by the DMA logic. The DMWC is implemented in a counter. After each NPR cycle it is decremented by two. When the register goes to zero, the DMA GO bit in the DMCSR is cleared thereby stopping the DMA logic. The RDY bit in the DMCSR is set causing an interrupt to the T11. Figure 2-5 shows the bit format of the DMWC register.

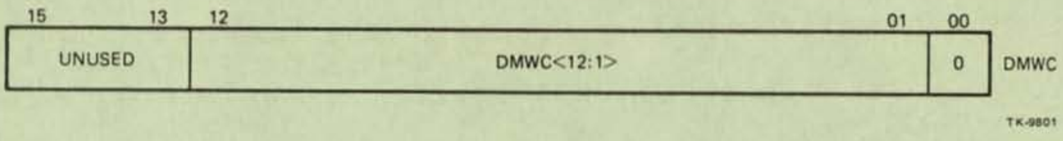


Figure 2-5 DMWC Register Bit Format and Bit Descriptions

2.2.1.2 T11 UNIBUS DMA -- The T11 UNIBUS DMA is used by the port microprocessor to access the host memory in order to perform the following functions:

- Read ring structure data,
- Read data buffers from host memory for transmission by the link, and
- Write status information to data buffers upon completion of transmission.

A T11 UNIBUS DMA transactions occurs in the following sequence:

1. The T11 loads the UNIBUS address registers, MDMAØ and MDMA1.
2. The T11 either reads or writes the microprocessor DMA data register. (MDMDRØ incrementing or MDMDR1 decrementing. Refer to Section 2.2.3.1.)
3. The reading or writing of the data register causes the microprocessor DMA to acquire the UNIBUS and transfer the data to/from host memory. During the DMA process, the T11 is stalled until the transfer is complete.

2.2.1.2.1 Microprocessor DMA Address Registers (MDMAØ and MDMA1) -- The micoprocessor DMA address registers are made up of a 17-bit counter that contains the address in host memory to or from which the data is to be transferred. MDMAØ contains the lower 16 bits of the address and MDMA1 contains the upper 2 bits of the address.

NOTE

Bit Ø of MDMAØ is always a Ø because the DMA logic only performs word transfers.

Figure 2-6 shows the Microprocessor DMA Address Register Bit Format.

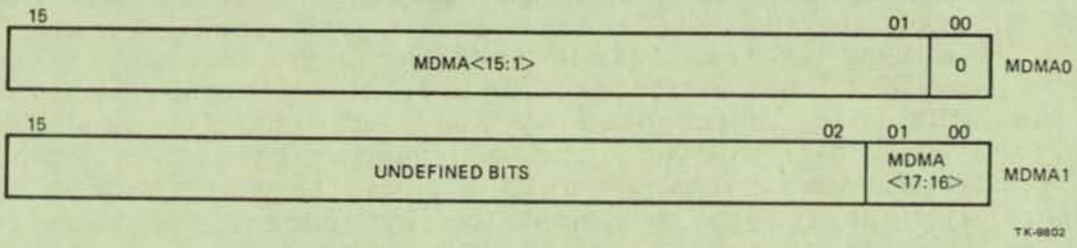


Figure 2-6 Microprocessor DMA Address Register Bit Format and Description

2.2.1.2.2 Microprocessor DMA Data Registers (MDMDRØ and MDMDR1) -- The microprocessor DMA data registers are used as data ports for the data that is transferred to/from host memory. If the T11 reads/writes the first register, MDMDRØ, the address contained in MDMAØ and MDMA1 is incremented by two. If the T11 reads/writes the second register, MDMDR1, the address contained in MDMAØ and MDMA1 is decremented by two. This allows the T11 to do multiple transfers without loading the host memory address for each transfer. The reading or writing of MDMDRØ or MDMDR1 by the T11 generates an NPR request to the UNIBUS. Refer to Figures 2-7 and 2-8 for MDMDRØ and MDMDR1 bit formats.

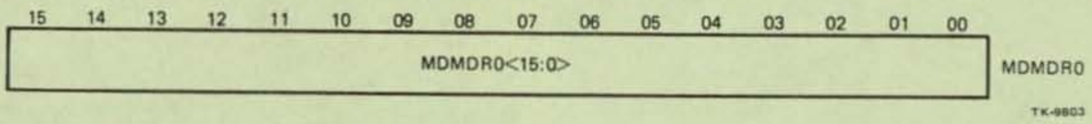


Figure 2-7 Data Register MDMDRØ (Incrementing)

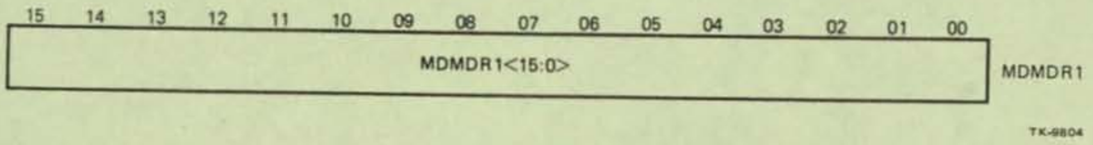


Figure 2-8 Microprocessor DMA Data Register MDMDR1

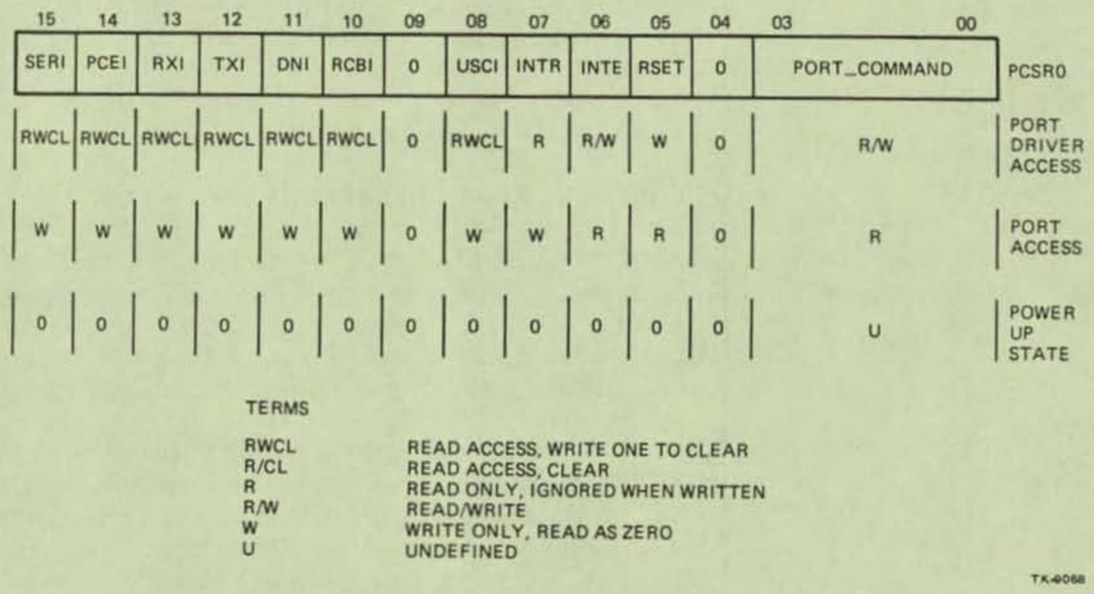
2.2.2 Port Control and Status Registers

The port control and status registers (PCSR) are used by the port module to receive commands from the host processor and report the results of the command along with other status information (interrupts, etc.).

There are four PCSRs, each with a specific function. The following sections show the format of the PCSRs and give a description of their function.

For a more detailed explanation of functions performed by the PCSRs, refer to Chapter 4, "Programming", of the DEUNA User's Guide (EK-DEUNA-UG).

2.2.2.1 Port Control and Status Register Ø (PCSRØ) -- Figure 2-9 shows the format of PCSRØ and Table 2-2 lists the functions of the bits.



15	14	13	12	11	10	09	08	07	06	05	04	03 — 00	
SERI	PCEI	RXI	TXI	DNI	RCBI	0	USCI	INTR	INTE	RSET	0	PORT_COMMAND	PCSR0
RWCL	RWCL	RWCL	RWCL	RWCL	RWCL	0	RWCL	R	R/W	W	0	R/W	PORT DRIVER ACCESS
W	W	W	W	W	W	0	W	W	R	R	0	R	PORT ACCESS
0	0	0	0	0	0	0	0	0	0	0	0	U	POWER UP STATE

TERMS

RWCL	READ ACCESS, WRITE ONE TO CLEAR
R/CL	READ ACCESS, CLEAR
R	READ ONLY, IGNORED WHEN WRITTEN
R/W	READ/WRITE
W	WRITE ONLY, READ AS ZERO
U	UNDEFINED

Figure 2-9 PCSRØ Format

Table 2-2 PCSR0 Bit Descriptions

Bits	Name	Description
<15>	SERI	Status Error Interrupt -- Indicates the presence of an error condition flagged in status register accessible by the port command function. Set by the DEUNA, cleared by the port-driver through the read and clear status port function.
<14>	PCEI	Port Command Error Interrupt -- Indicates the occurrence of either a function error or a UNIBUS timeout during the execution of a port command. Bit 7 of PCSR1 distinguishes between the two error conditions. Set by the DEUNA, cleared by the port-driver.
<13>	RXI	Receive Ring Interrupt -- Attention bit for ring updates. Set the by the DEUNA cleared by the port-driver. Indicates, when set, that the DEUNA has placed a message(s) on the ring.
<12>	TXI	Transmit Ring Interrupt -- Attention bit for ring updates. Set by the DEUNA, cleared by the port-driver. Indicates, when set, that transmission has been suspended. All messages it found on the transmit ring have been set, or an error was encountered during a transmission.
<11>	DNI	Done Interrupt -- Interrupts when the DEUNA completes a port command. (Note: the port command NO-OP does not cause the DNI bit to set.) Set by the DEUNA, cleared by the port-driver.
<10>	RCBI	Receive Buffer Unavailable Interrupt -- Interrupts when the DEUNA discards an incoming message due to receive ring buffers being unavailable. Once set by the DEUNA, RCBI is not set again until after the DEUNA has received a PDMD port command and has discarded a subsequent message. Set by the DEUNA, cleared by the port-driver.
<09>	zero	

Table 2-2 PCSRØ Bit Descriptions (Cont)

Bits	Name	Description
<Ø8>	USCI	Unsolicited State Change Interrupt -- Interrupts when the DEUNA performs the following actions: Fatal Error -- A transition into the NI and UNIBUS halted state from the ready, running, UNIBUS halted, or NI halted states. This state change is caused by the DEUNA detecting an internal fatal error, that is, internal parity error. Communication Processor Boot -- A transition into the primary load state caused by the reception of a remote boot request of the communication processor (DEUNA microcode). Communication Processor Boot -- A transition into the ready state from the primary load state following the reception of the memory load with transfer address message, as part of a remote boot request. The three conditions are distinguished by examining the state field of PCSR1. Set by the DEUNA, cleared by the port-driver.
<Ø7>	INTR	Interrupt Summary -- The logical OR of PCSRØ <15:Ø8>. Set by the DEUNA.
<Ø6>	INTE	Interrupt Enable -- Set or cleared by the port-driver, unchanged by the DEUNA.

NOTE

In order to overcome synchronization problems with the port command field when writing the INTE bit, the DEUNA hardware locks the port command field during write accesses that change the INTE bit from a one to a zero or change the INTE bit from a zero to a one. Issuing the DEUNA, a port command, and changing the state of the INTE bit must occur in two different write accesses.

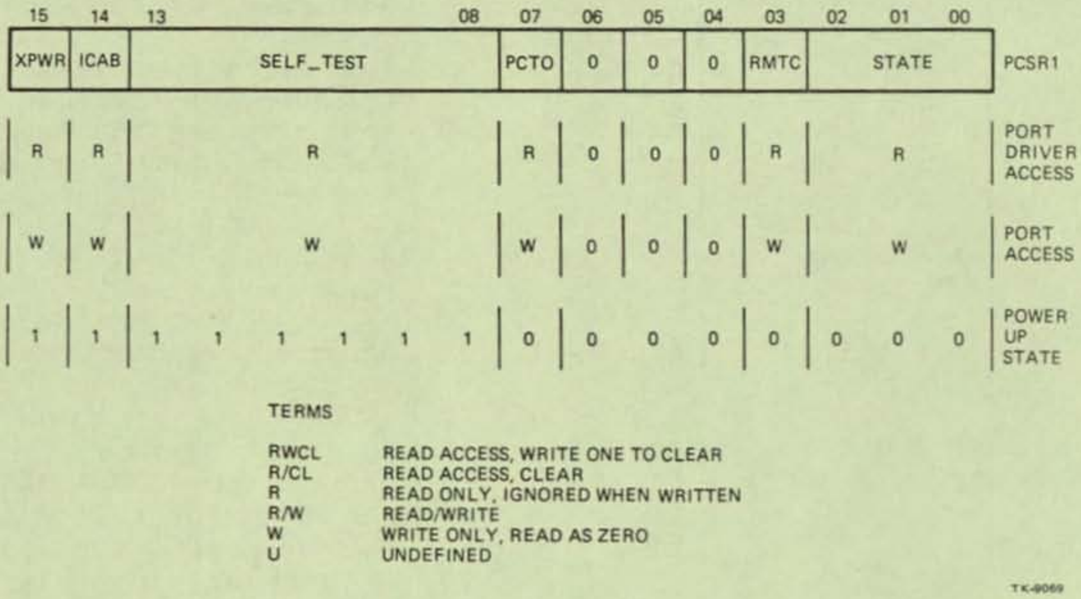
Table 2-2 PCSRØ Bit Descriptions (Cont)

Bits	Name	Description	
<Ø5>	RSET	DEUNA Reset -- Clears the DEUNA and returns it to the power up state when written with a one byte port-driver. This bit is write-only. After a successful reset, PCSRØ <11> (DNI) = 1 and PCSRØ <Ø7> (INTR) = 1.	
<Ø4>	zero		
<Ø3:ØØ>	PORT_COMMAND		
	Ø Ø Ø Ø	NO-OP	No operation
	Ø Ø Ø 1	GET PCBB	Instructs the DEUNA to fetch the address of the port control block from PCSRs 2 and 3. The DEUNA accesses PCSRs over the UNIBUS conductor, and retains a copy of the address internally. If the address of the port control block is changed, this command must be repeated to inform the DEUNA.
	Ø Ø 1 Ø	GET CMD	Instructs the DEUNA to fetch and execute a command found in the first word of the port control block. The address of the port control block was obtained through the get PCBB command.
	Ø Ø 1 1	SELF TEST	Instructs the DEUNA to enter the reset state and execute self-test.
	Ø 1 Ø Ø	START	Enables transmission and reception of frames from the port-driver. This command is ignored by the DEUNA if it is in the running state. Clears any current buffer status that

Table 2-2 PCSRØ Bit Descriptions (Cont)

Bits	Name	Description	
			the DEUNA has stored internally and resets the ring pointers to the base addresses of the rings.
	Ø 1 Ø 1	BOOT	Instructs the DEUNA to enter the primary load state and initiate the down-line load of additional DEUNA microcode.
	Ø 1 1 Ø	Not Used	Reserved code, causes a NO-OP.
	Ø 1 1 1	Not Used	Reserved code, causes a NO-OP.
	1 Ø Ø Ø	PDMD	Polling Demand -- Instructs the DEUNA to check the descriptor rings. The DEUNA polls the receive descriptor ring only if it had not previously acquired a free buffer.
	1 Ø Ø 1	Not Used	Reserved code, causes a NO-OP, sets DNI.
	1 Ø 1 Ø	Not Used	Reserved code, causes a NO-OP, sets DNI.
	1 Ø 1 1	Not Used	Reserved code, causes a NO-OP, sets DNI.
	1 1 Ø Ø	Not used	Reserved code, causes a NO-OP, sets DNI.
	1 1 Ø 1	Not Used	Reserved code, causes a NO-OP, sets DNI.
	1 1 1 Ø	Not Used	Reserved code, causes a NO-OP, sets DNI.
	1 1 1 1	STOP	Suspends the operation of the DEUNA to transition to the ready state. Causes no action if the DEUNA is not in the running state.

2.2.2.2 Port Control and Status Register 1 (PCSR1) -- Figure 2-1Ø shows the format of PCSR1 and Table 2-3 lists the functions of the bits.



15	14	13					08	07	06	05	04	03	02	01	00	
XPWR	ICAB	SELF_TEST						PCTO	0	0	0	RMTC	STATE			PCSR1
R	R	R						R	0	0	0	R	R			PORT DRIVER ACCESS
W	W	W						W	0	0	0	W	W			PORT ACCESS
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	POWER UP STATE

TERMS

RWCL	READ ACCESS, WRITE ONE TO CLEAR
R/CL	READ ACCESS, CLEAR
R	READ ONLY, IGNORED WHEN WRITTEN
R/W	READ/WRITE
W	WRITE ONLY, READ AS ZERO
U	UNDEFINED

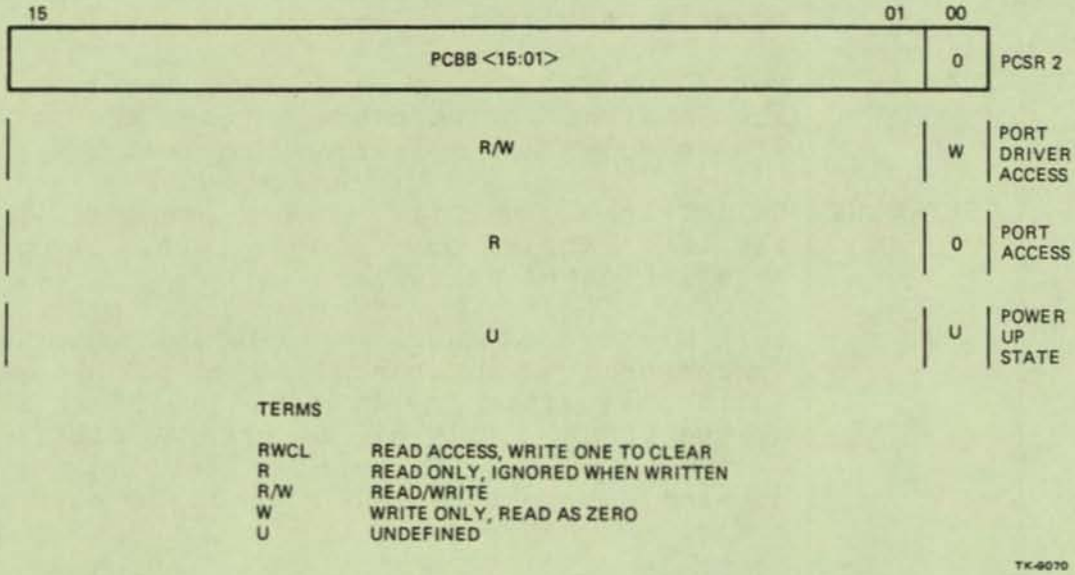
TK-9069

Figure 2-1Ø PCSR1 Format

Table 2-3 PCSR1 Bit Descriptions

Bits	Name	Description
<15>	XPWR	Transceiver Power OK -- A zero indicates that a failure exists in either the transceiver power supply or the fuse on the link module.
<14>	ICAB	Port/Link Cabling OK -- A zero indicates that the interconnecting cable between the port and link modules has a seating problem.
<13:Ø8>	SELF-TEST	Self-Test Error Code -- The encoded test of the DEUNA failed during self-test. A code of zero indicates no failure.
<Ø7>	PCTO	Port Command Timeout -- A UNIBUS timeout was encountered while executing a port command. Valid only after the PCEI bit of PCSRØ is set by the DEUNA. This bit is used to distinguish between a DEUNA failure to complete a port command due to a UNIBUS timeout or a function error.
<Ø6:Ø4>	Zeros	
<Ø3:ØØ>	STATE	Ø Ø Ø Ø Reset Ø Ø Ø 1 Primary Load Ø Ø 1 Ø Ready Ø Ø 1 1 Running Ø 1 Ø Ø Not Used Ø 1 Ø 1 UNIBUS Halted Ø 1 1 Ø NI Halted Ø 1 1 1 NI and UNIBUS Halted Fatal internal error, that is parity error. An interrupt condition. When the DEUNA is in this state, the FATI bit of PCSRØ is also set. Cleared by the port-driver setting the RSET bit. 1 1 1 1 Secondary Loader

2.2.2.3 **Port Control and Status Register 2 (PCSR2)** -- Figure 2-11 shows the format of PCSR2 and Table 2-4 lists the functions of the bits.



TERMS

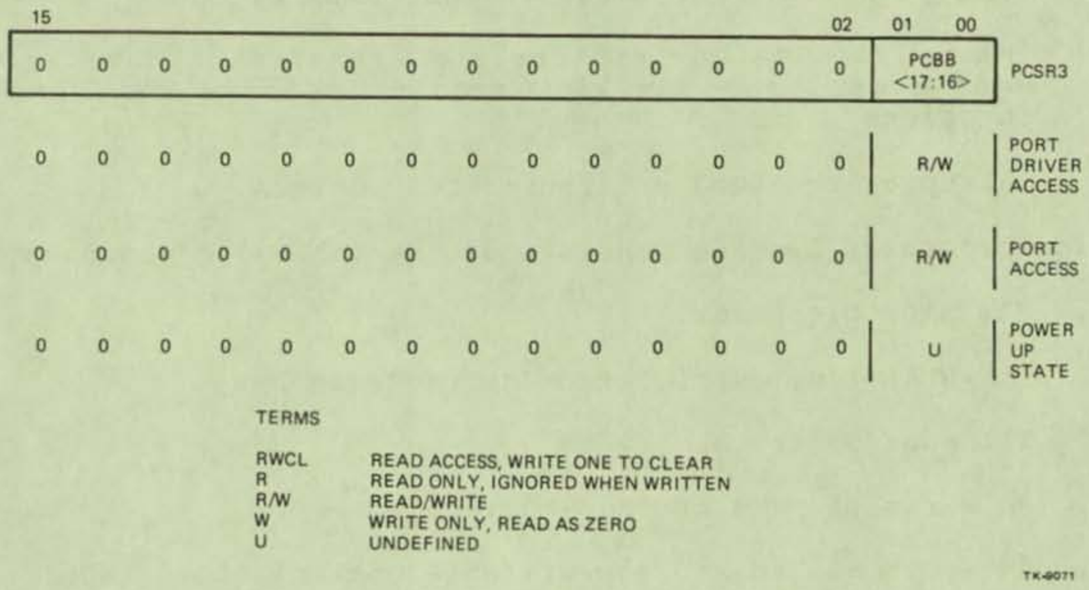
RWCL READ ACCESS, WRITE ONE TO CLEAR
R READ ONLY, IGNORED WHEN WRITTEN
R/W READ/WRITE
W WRITE ONLY, READ AS ZERO
U UNDEFINED

Figure 2-11 PCSR2 Format

Table 2-4 PCSR2 Bit Description

Bits	Name	Description
<15:ØØ>	PCBB	The low order 16 bits of the address of the port control block base. The PCBB is read by the port as an even number.

2.2.2.4 Port Control and Status Register 3 (PCSR3) -- Figure 2-12 shows the format of PCSR3 and Table 2-5 lists the function of the bits.



15													02	01 00	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	PCBB <17:16>	PCSR3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	PORT DRIVER ACCESS
0	0	0	0	0	0	0	0	0	0	0	0	0	0	R/W	PORT ACCESS
0	0	0	0	0	0	0	0	0	0	0	0	0	0	U	POWER UP STATE

TERMS

RWCL	READ ACCESS, WRITE ONE TO CLEAR
R	READ ONLY, IGNORED WHEN WRITTEN
R/W	READ/WRITE
W	WRITE ONLY, READ AS ZERO
U	UNDEFINED

TK-8071

Figure 2-12 PCSR3 Format

Table 2-5 PCSR3 Bit Description

Bits	Name	Description
<15:02>		Zeros
<01:00>	PCBB	The high order 2 bits of the address of the port control block base.

2.3 MICROPROCESSOR SECTION

The function of the microprocessor section of the port module is to:

- Manage the ring structure in host memory,
- Set up the DMA control for the transfer of data between host memory and the link module (receive and transmit functions), and
- Interpret received or transmitted packets.

The microprocessor section consists of the following components:

1. T11 microprocessor,
2. DAL/BDAL-time multiplexed data/address bus,
3. T11 address latch,
4. 8K words of PROM storage-microcode,
5. 4K words of RAM storage-writable control store (WCS), and
6. Internal I/O decode-used when T11 has to access a register on the PORT module.

2.3.1 Microprocessor

The DEUNA uses a microprocessor located on the port module to control its operation. The microprocessor used is a DCT11-AA (T11). The T11 is a single chip microprocessor that uses the LSI-11 instruction set. The T11 communicates to the port module over a time multiplexed bidirectional bus called the data address lines (DAL). It also receives process and status information via a separate set of interrupt inputs. Each interrupt and its function is listed in Table 2-6.

The T11 can access a total of 32K words of memory. This address space is divided into areas for:

- Microprogram storage,
- Writable control store (WCS),

- Transmit and receive buffer space, and
- Input/output control.

Figure 2-13 shows the configuration of the T11's address space.

For more information on the operation of the T11 microprocessor, refer to the DCT11-AA Microprocessor User's Guide (EK-DCT11-UG).

Table 2-6 T11 Interrupts

Interrupt	Signal Name	Description
Receive Miss	MISS INTR	There is no receive buffer available for an incoming message.
Memory Parity	LNK MEM PAR ERR	There is a parity error in the link buffer memory.
PCSR Write	PCSR INTR	The host processor has written a command into PCSRØ.
UNIBUS Error	UBERR INTR	There is a UNIBUS timeout.
Transmit Done	XMIT DONE	The link has finished transmitting a buffer.
Receive Buffer Done	RCV BUFF DONE	There is a receive buffer waiting to be sent to host memory.
Timer	TIMER INTR	Interrupts T11 every second for timing information.
DMA Ready	DM INTR	DMA machine ready to be started.

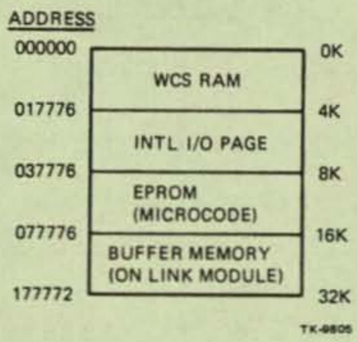


Figure 2-13 T11 Address Space

2.3.2 Internal Registers

The internal registers of the port module are used by the T11 for setting up and controlling the operation of the DEUNA. These registers reside in the I/O page of the T11.

The I/O decode logic of the port enables the selected internal register when it is addressed by the T11. This logic monitors the output of the T11 address latch and the read and write signals generated by the T11.

Table 2-7 gives a list of the internal register addresses and the type of access allowed.

Table 2-7 Internal Register Address Assignments

Address	Name	Access	Description
21ØØØ	PCSRØ	R/W	Port control and status reg. Ø
21ØØ2	DMCSR	R/W	DMA control and status reg.
21ØØ4	DMATØ	R/W	DMA-to-address register Ø
21ØØ6	DMAT1	R/W	DMA-to-address register 1
21Ø1Ø	MDMAØ	R/W	MicroCPU DMA-to-adrs. reg. Ø
21Ø12	MDMA1	R/W	MicroCPU DMA-to-adrs. reg. 1
21Ø14	MDMDRØ	R	MicroCPU DMA data reg. Ø
21Ø16	MDMDR1	R	MicroCPU DMA data reg. 1
21Ø2Ø	PCSR1	WO	Port control and status reg. 1
21Ø22	DMAF	WO	DMA-from-address register
21Ø24	DMWC	WO	DMA word count register
21Ø26	MDMDRØ	WO	Read Inc UB data port
21Ø3Ø	LTAC	WO	Link transmit adrs. counter reg.
21Ø32	LRBAF	WO	Link rec. buffer address FIFO
21Ø34	LCSR	WO	Link control and status reg.
21Ø36	MDMDR1	WO	Write Dec. UB data port
21Ø4Ø	PCSRSW	RO	Port switchpack reg.
21Ø42	UNUSED	RO	

Table 2-7 Internal Register Address Assignments (Cont)

Address	Name	Access	Description
21Ø44	LCBAF	RO	Link completed buffer add. FIFO
21Ø46	PCSR1	RO	Port cntl. and status reg. 1
21Ø5Ø	UNUSED	RO	
21Ø52	UNUSED	RO	
21Ø54	UNUSED	RO	
21Ø56	UNUSED	RO	
21Ø6Ø	PHYADØ	RO	Physical address byte Ø
21Ø62	PHYAD1	RO	Physical address byte 1
21Ø64	PHYAD2	RO	Physical address byte 2
21Ø66	PHYAD3	RO	Physical address byte 3
21Ø7Ø	PHYAD4	RO	Physical address byte 4
21Ø72	PHYAD5	RO	Physical address byte 5
21Ø74	PHYAD6	RO	Physical address byte 6
21Ø76	PHYAD7	RO	Physical address byte 7

2.3.3 Default Station Address (Physical Address)

The microprocessor section of the DEUNA contains a PROM which the T11 can read on power up to get the default address of the node. When the T11 reads the physical address from the PROM it transfers it to the station address RAM on the link module.

The physical address in the station address RAM can be changed by the host by a change physical address command.

2.3.4 Physical Address Registers

These registers are used to read the physical address from the physical address PROM. Figure 2-14 shows the configuration of these registers.

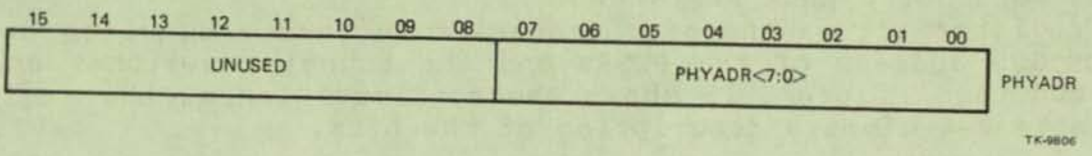


Figure 2-14 Physical Address Register Bit Configuration

2.3.5 Port Switchpack Register

This register allows the microprocessor to read the switch selected UNIBUS address of the PCSRs and the function switches on the port module. Figure 2-15 shows the configuration of the register and Table 2-8 gives a description of the bits.

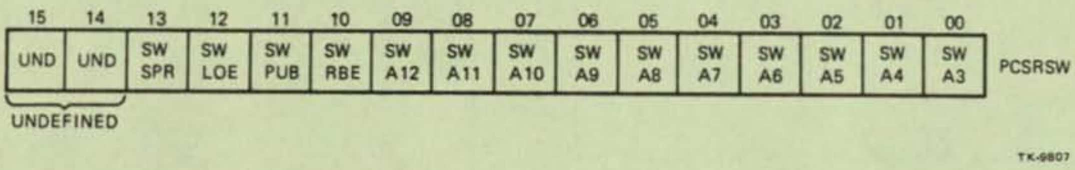


Figure 2-15 Port Switchpack Register Bit Configuration

Table 2-8 Port Switchpack Register Bit Descriptions

Bit	Field	Description
PCSRSW<Ø9:ØØ>	SW AXX	UNIBUS address address
PCSRSW<1Ø>	SW RBE	Remote boot enable switch
PCSRSW<11>	SW PUB	Power-up boot switch
PCSRSW<12>	SW LOE	Loop-on-self-test error switch
PCSRSW<13>	SW SPR	Spare switch
PCSRSW<15:14>	UND	Undefined

2.3.6 Timer

The timer is made up of a one shot that generates an interrupt to the T11 every second. This allows the T11 to time events through the use of software routines.

2.3.7 Internal Buses

The port uses three sets of internal buses for the transfer of information within the DEUNA. These buses are:

1. DAL/BDAL (Data/Address Lines, Buffered Data/Address Lines).

   - Time multiplexed -- carry data during part of the timing cycle and address during the other part of the timing cycle.

   - BDAL is a buffered extension of the DAL for loading purposes.

2. T/F BUS (To/From Bus) -- transfers data between the UNIBUS bus and the DEUNA.

3. LMD BUS (Link Memory Data Bus) -- data bus to link memory buffers.

4. LINK MEM A (Link Memory Address Bus) -- address bus to link memory buffers.

2.4 LINK MEMORY CONTROL

The link memory section is the part of the port module which communicates with the link module. This section contains control for the 16K words of RAM that are located on the link module (parity generation and memory are on the port module). This memory is divided into 16 buffers that are used to buffer packets of data being transmitted to or received from the ETHERNET via the link module.

The link memory section contains the logic necessary to:

1. Arbitrate for use of the link memory,
2. Keep track of which buffers are available for use, and
3. Generate the memory addresses for writing or reading data from memory.

2.4.1 Link Memory Arbitration

Link memory is accessed by four different processes:

1. Link transmit state machine,
2. Link receive state machine,
3. DMA control, and
4. T11.

Arbitration for use of link memory by any of these processes is performed by the link memory arbitration PAL. A description for the PAL is given in the DEUNA engineering drawings.

2.4.2 Link Transmit Address Counter (LTAC)

The link transmit address counter is used when a transmit buffer is to be transmitted onto the ETHERNET.

The link transmit address counter consists of two sections:

1. Link Transmit Address Counter Register -- loaded by T11 with the four-bit buffer address.
2. Link Transmit Address Counter -- this is a 1Ø-bit counter that is used to generate the lower 1Ø bits of the transmit buffer address.

They are configured as shown in Figure 2-16.

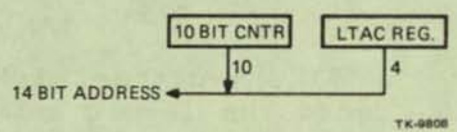


Figure 2-16 LTAC Configuration

When a transmit buffer in link memory is to be transmitted onto the ETHERNET, the following action takes place.

1. The T11 loads the LTAC register with the four-bit buffer address. This clears the 1Ø-bit counter and notifies the transmit state machine on the link module that there is a buffer to be transmitted.

2. Transmit state machine increments counter by two after reading the word to be transmitted until the buffer is empty.

The transmit state machine can clear the 1Ø-bit counter if it needs to do a transmit retry.

Figure 2-17 shows the bit configuration of the LTAC.

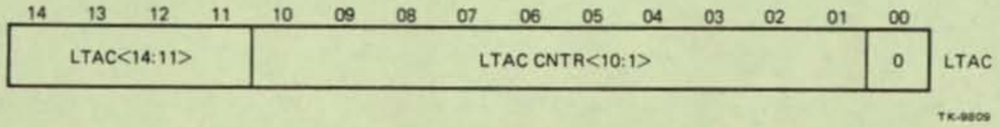


Figure 2-17 LTAC Bit Configuration

2.4.3 Link Receive Address Counter

The link receive address counter is used to generate the buffer addresses for messages received from the ETHERNET by the DEUNA.

The link receive address counter is made up of three sections.

1. The link receive buffer address FIFO (LRBAF),
2. The link completed buffer address FIFO (LCBAF), and
3. The link receive address counter.

They are configured as shown in Figure 2-18.

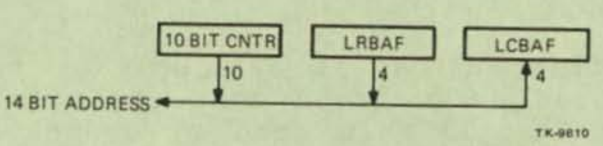


Figure 2-18 Receive Address Counter Configuration

The LRBAF and the LCBAF are four-bit by 64 location FIFOs. The LRBAF has the upper four bits of all available buffer addresses placed into it by the T11. When a receive buffer is needed by the link, the following functions are performed.

1. The address counter is cleared.

2. The buffer address at the output of the LRBAF and the output of the counter are used to generate the link memory address.

3. The counter is incremented until the buffer is completed.

4. When the buffer is completed by the link it advances the LRBAF which loads the address of the completed buffer into the LCBAF and clears the counter. The address bubbles through the FIFO.

5. When a buffer address is available at the output of the LCBAF the T11 is notified that there is a completed receive buffer. This is done by generating an interrupt (RCV BUF DONE) to the T11.

6. The T11 then processes the completed buffer and returns the buffer address to the LRBAF.

Figure 2-10 shows the configuration of the LRBAF and the LCBAF.

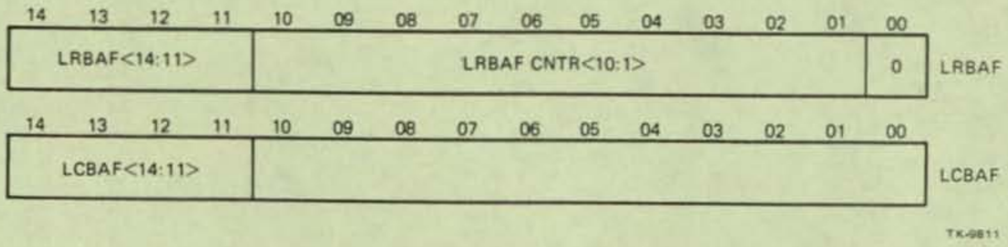


Figure 2-19 LRBAF, LCBAF Bit Configuration

2.4.4 T11 Addressing of Link Buffer Memory

When the T11 addresses link buffer memory, the memory arbitor arbitrates for use of the memory. When the T11 receives use of the memory the requested data is passed to the T11. During the data transfer the T11 is stalled.

2.4.5 Port-to-Link Interface

The DEUNA is comprised of two modules which have to be UNIBUS SPC compatible. This does not allow for a backplane interconnect. Therefore, the DEUNA port and link modules are connected by a Berg type connector over the handles cables. The signals on these cables comprise the port link interconnect.

The signals on these cables are broken down into five classes:

1. Link memory bus signals,
2. Link memory address control signals,
3. Link command register signals,
4. Link discrete status signals, and
5. Clock and initialize signals.

The following sections describe the port-to-link interface signals.

2.4.5.1 Link Memory Bus --

Signal	Source	Description
BUS LMD <15:ØØ>	BIDIR	Link Data Bus -- Sixteen bidirectional data lines between the link and port modules.
LINK MEM A <14:Ø1>	PORT	Link Memory Address Bus -- Fourteen address lines between the port and link modules.
BUS READ	BIDIR	Read/Write -- Used to indicate the direction of the transfer.
RX REQUEST	LINK	Receiver Request -- Used by the link receive state machine to request the link memory.
RX ACK	PORT	Receiver Acknowledge -- Used by the port to acknowledge the link request.
TX REQUEST	LINK	Transmit Request -- Used by the link transmit state machine to request the link memory.

TX ACK	PORT	Transmit Acknowledge -- Used by the port to acknowledge the link request.

2.4.5.2 Link Memory Address Control Signals -- A detailed description of these signals is given in Chapter 3 of this manual.

Signal	Source	Definition
INC TX POINTER	LINK	Increment Transmit Pointer -- Used by the link to increment the transmit address pointer.
RES TX POINTER	LINK	Restore Transmit Pointer -- Used by the link to restore the transmit address counter to the beginning address.
INC RX POINTER	LINK	Increment Receiver Pointer -- Used by the link to increment the receive address counter.
RES RX POINTER	LINK	Restore Receiver Pointer -- Used by the link to restore the receive pointer to the beginning.
ADV RX POINTER	LINK	Advance Receiver Pointer -- Used by the link to get the next receive address buffer.

2.4.5.3 Command Register Control --

Signal	Source	Description
CMDW	PORT	Command Register Write -- Enables the command register to be written from the link memory bus.
CMDE	PORT	Command Register Execute -- Tells the link to execute the command in the link command register.

2.4.5.4 Link Discrete Status --

Signal	Source	Description
CERR	LINK	Collision Test Error -- Indicates the collision output failed to activate during the collision test following a transmission (heartbeat).

SET MISS	LINK	Missed Packet -- Indicates that the link failed to write a received packet into link memory because a buffer was unavailable.
TATT	PORT	Transmitter Attention -- Tells the link that the port has completed a buffer for transmission.
TX DONE	LINK	Transmit Done -- Indication to the port that the link has finished transmitting a buffer.
ICAB1	LINK	Installed Cable 1 & 2 -- Used by the port to re-ensure the interconnecting cable is plugged in properly.
ICAB2	LINK	
FUSE CHECK	LINK	Transceiver Power -- Used by the port to check that the power to the transceiver is available.
RX FREE BUF	PORT	Receiver Buffer Free -- Used by the link to find out if there are any free receive buffers.
WR RESET	PORT	Reset -- Used by the link to do a UNIBUS reset.

2.4.5.5 Clock and Reset --

Signal	Source	Description
1ØMHZ	LINK	Clock -- 1Ø MHz square wave.
INIT	PORT	Buffered Initialize -- Buffered UNIBUS INIT.

CHAPTER 3
LINK MODULE
FUNCTIONAL DESCRIPTION

3.1 INTRODUCTION

The link module (M7793) is the interface between the DEUNA and the ETHERNET transceiver. It is microprogram controlled and provides the following functions.

- Physical channel interface
- Parallel-to-serial conversion of data on transmit
- Serial-to-parallel conversion of data on receive
- Collision detection and retry
- CRC generation and checking
- Station address detection
- Link memory bus control

The link in connection with the port provides the logic necessary to interface the UNIBUS Bus with the ETHERNET.

A functional block diagram of the link module is shown in Figure 3-1. The letters, in parenthesis, on the block diagram give the location of the logic for that functional block in the engineering drawings.

3.2 LINK MEMORY BUS

The link memory bus provides the communication path between the link module and the port module. The bus is made up of 54 lines that are divided into four signal groups.

- Memory Bus
- Discrete Control
- Discrete Status
- Clock

Tables 3-1, 3-2, 3-3, and 3-4 list the link memory bus signal names, their source, and a description of their function.

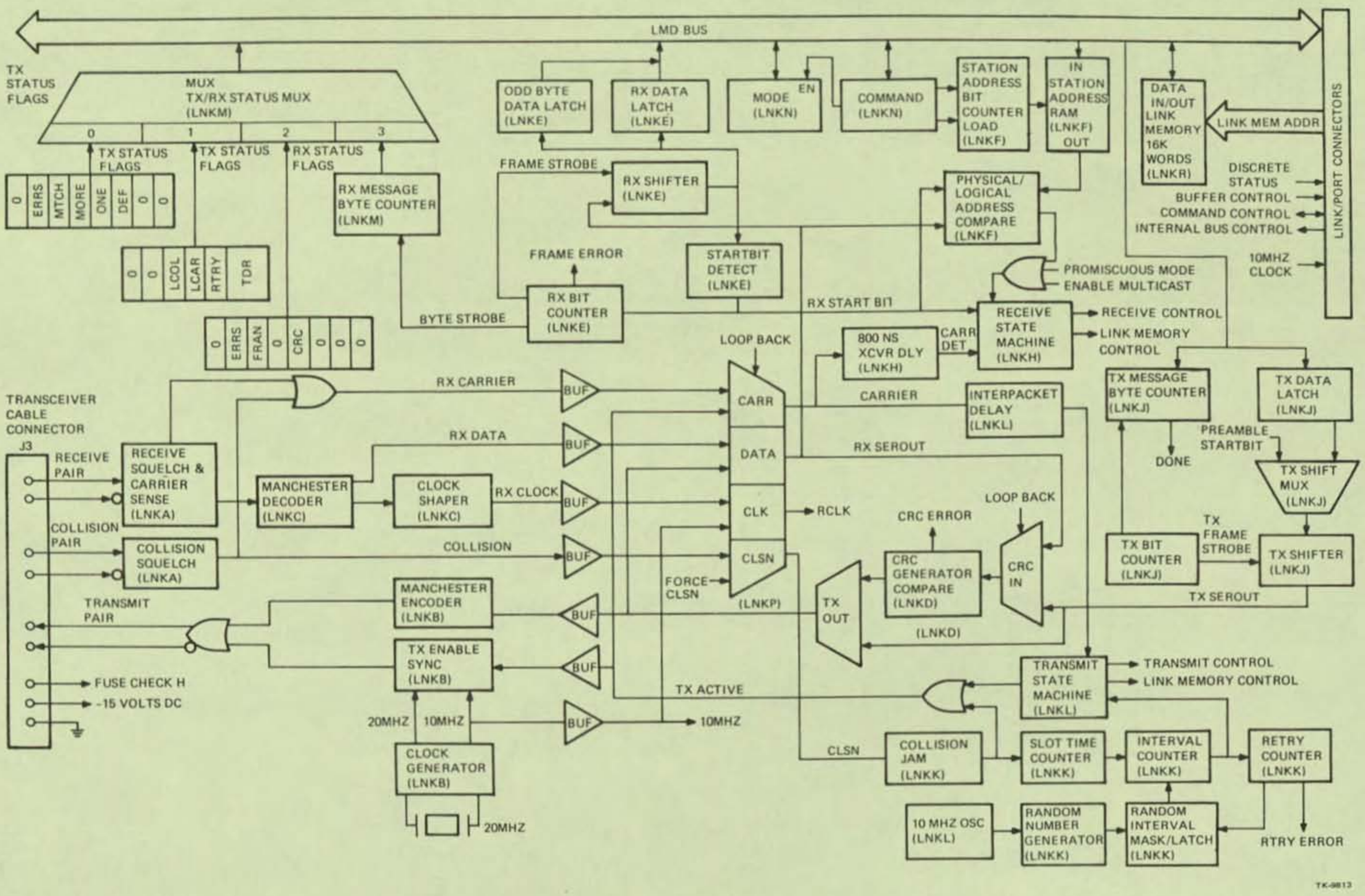


Figure 3-1 Link Module Functional Block Diagram

Table 3-1 Memory Bus Signals

Signal	Source	Description
LINK MEM <14:Ø1>	Port	Link Memory Address Bus - Fourteen address lines used to address the memory buffers on the link module.
BUS LMD <15:ØØ>	BIDIR	Link Memory Data Bus - Sixteen bidirectional data lines between the link and port module.
TX REQUEST	Link	Transmit Data Request - Used by the link to start arbitration for the Link Memory Bus.
RX REQUEST	Link	Receive Data Request - Used by the link to start arbitration for the link memory data bus.
TX ACK	Port	Transmit Data Acknowledge - Used by the port to inform the link that it has granted the link memory bus for a transmit operation.
RX ACK	Port	Receive Data Acknowledge - Used by the port to inform the link that it has granted the link memory bus receive operation.
BUS READ	BIDIR	Read/Write - Used to indicate the direction of the transfer. When set data is transferred from a link memory buffer.

Table 3-2 Discrete Control Bus Signals

Signal	Source	Description
INIT	Link	Synchronized Initialize - Clock synchronized power up initialize.
WR RESET	Port	Software Reset - Comes from port PCSRØ.
CMDW	Port	Command Register Write - Enables the command register on the link to be written from the link memory bus. This signal is valid for 1ØØ ns.
TATT	Port	Transmitter Attention - The port notifies the link that a transmit buffer is ready for transmission. Set by the port cleared by TX DONE.

Table 3-2 Discrete Control Bus Signals (Cont)

Signal	Source	Description
RES TX POINTER	Link	Reset Transmit Pointer - Tells the port to reset the transmit address pointer on the port. This signal is valid for 100 ns.
INC TX POINTER	Link	Increment Transmit Pointer - Tells the port to increment the transmit address pointer on the port. This signal is valid for 100 ns.
RES RX POINTER	Link	Reset Receive Pointer - Tells the port to reset the receiver address pointer on the port. This signal is valid for 100 ns.
INC RX POINTER	Link	Increment Receiver Pointer - Tells the port to increment the receiver address pointer on the port. This signal is valid for 100 ns.
ADV RX POINTER	Link	Advance Receiver Pointer - Tells the port to advance the receive buffer address pointer on the port. This signal is valid for 100 ns.
CABLE VERIFY IN	Port	Cable Verify Input - This circuit provides a closed loop electrical path with cable verify output that is used to indicate that the cable between the link and the port is installed and connected properly.
CABLE VERIFY OUT	Port	Cable Verify Output

Table 3-3 Discrete Status Bus Signals

Signal	Source	Description
CERR	Link	Collision Test Error - The transceiver collision output failed to activate during the collision test following transmission (heartbeat). Set during a collision test error. This signal is valid for 100 ns. This signal is valid for the H4000 or equivalent transceiver.
MISS	Link	Missed Packet - Receiver failed to write a packet addressed to the port into the link memory because a buffer was unavailable. This signal is valid for 100 ns.

Table 3-3 Discrete Status Bus Signals (Cont)

Signal	Source	Description
TX DONE	Link	Transmit Done - Indication to the port that the link has finished transmitting a buffer. This signal is valid for 1ØØ ns.
FUSE CHECK	Link	Transceiver Power OK - A ONE indicates that a failure exists in either the transceiver power supply or in the cabling to the bulkhead assembly.
FREE RX BUFF	Port	Free Receiver Buffer - A buffer is available in the link memory to put an incoming packet. Set by the port, cleared by ADV RX pointer.

Table 3-4 Clock Signal

Signal	Source	Description
1Ø MHz	Link	Clock - The link clock is a 1ØØ nanosecond square wave derived from a free running 1Ø MHz clock located in the ECL section of the link.

3.3 LINK REGISTERS

The operation of the link module is controlled by the port module though the use of two registers. The two registers are the link command register and the mode register. These registers are used to initialize, start, stop, and select the mode of operation of the link module. In addition to the command and mode registers, the link contains the station address RAM. The station address RAM is used to hold the addresses of the node for decoding by the address detection logic.

3.3.1 Command Register

The link command register is used by the port module to initialize, start, and stop the link module. This register is accessed by the port microprocessor by asserting CMDW H on the link memory bus. This register is write only by the port and is set to all zeros on power up or when initialized.

Figure 3-2 shows the format of the register and Table 3-5 describes the function of each bit.

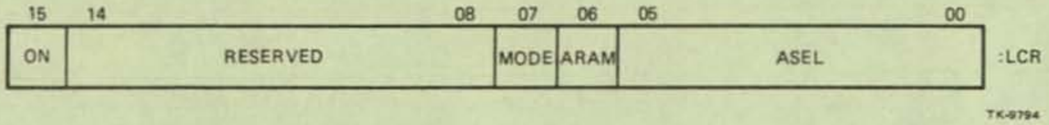


Figure 3-2 Format of Link Command Register

Table 3-5 Link Command Register Bit Descriptions

Bits	Field	Description
<15>	ON	Enable Link Module - When set, this bit enables both the receive and the transmit state machines. Set and cleared by the port. Powers on in the zero state.
<14:8>	Reserved	
<7>	Mode	Enable Mode Register - When set, this bit enables the write access of the mode register over the link memory data bus when CMDE H is asserted by the port. Set and cleared by the port.
<6>	ARAM	Enable Station Address RAM - When set, this bit allows the station address RAM to be written when CMDE H is asserted by the port. Set and cleared by the port.
<5:Ø>	ASEL	Address Select - Specifies the memory location within the station address RAM containing the physical and logical address: the data section of the station address begins at location ASEL=2Ø (octal). Set and cleared by the port.

NOTE

The first word, ARAMØ, of the Station Address RAM begins at location ASEL=2Ø (octal). This is due to the binary counter logic used in the address comparator section.

3.3.2 Link Mode Register

The port uses the mode register to control the transmit and receive operations of the link module. It is written when the mode bit of the link command register is set and bus signal CMDE is asserted. The register is set to all zeros on power up or when the link is initialized.

Figure 3-2 shows the format of the register and Table 3-6 describes the function of each bit.

15	14	13	12	11	10	09	06	05	04	03	02	01	00	
PROM	ENAL	RES		ENCR	ACLO	RESERVED		DRTY	COLL	DTCR	LOOP	RES	HDPX	:MODE

TK-0795

Figure 3-3 Link Mode Register Format

Table 3-6 Bit Descriptions for Link Mode Register

Bits	Field	Description
<15>	PROM	Instructs the link to accept all incoming frames regardless of the destination address field. Written and cleared by the port.
<14>	ENAL	Instructs the link to accept all incoming frames with multicast destinations. Written and cleared by the port.
<13:12>	RES	Reserved
<11>	ENCR	Enable Collision Test Error. When set, any collison test errors will be reported back to the port. Set and cleared by the port.
<1Ø>	ACLO	Enable ACLO. When set, ACLO asserts ACLO on the UNIBUS Bus and disables INIT on the DEUNA. Set by the port cleared by the link.
<9:6>	RESERVED	
<5>	DRTY	Disable Retry Logic. When set, the link attempts only one transmission of a packet. This is a maintenance self-test function. Written and cleared by the port.
<4>	COLL	Simulate a collision on the wire during loopback mode. This is a maintenance self-test function. Written and cleared by the port.
<3>	DTCR	Disable Transmit CRC Logic. If DTCR=1, the CRC logic is dedicated to the receiver. If DTCR=Ø, the CRC logic is dedicated to the transmitter. This feature is used as a loopback maintenance function. Written and cleared by the port.
<2>	LOOP	Enable Loopback. When set, this bit enables loopback internal to the link, and the CRC logic dedicated to the receiver or transmitter as selected by DTCR. Written and cleared by the port.
<1>	RES	Reserved

Table 3-6 Bit Descriptions for Link Mode Register (Cont)

Bits	Field	Description
<Ø>	HDPX	Half-Duplex Mode. Indicates when clear that the link will receive messages transmitted to itself over the wire. Messages received in this manner do not undergo CRC check and a CRC error status is returned with them. Indicates when set that the link will not receive messages transmitted to itself. However, the link recognizes the transmitted messages as being addressed to itself and sets the MTCH bit in the transmit ring following the transmission attempt. Set and cleared by the port. Cleared upon power up.

3.3.3 Station Address RAM (ARAM)

The station address RAM contains the physical, logical, and broadcast addresses of the node. There can be a maximum of 12 addresses. Each address is 48 bits in length. These addresses are loaded by the port and read by the receive state machine.

Data is written to the ARAM over the link memory bus when the ARAM bit of the command register is set and the port asserts CMDE H.

Figure 3-4 shows the format of the station address RAM and Table 3-7 describes the register bits.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
RESERVED				ADRN0	ADRM0	ADRL0	ADRK0	ADRJ0	ADRH0	ADRF0	ADRE0	ADRD0	ADRC0	ADRB0	ADRA0	ARAM 0
RESERVED				ADRN1	ADRM1	ADRL1	ADRK1	ADRJ1	ADRH1	ADRF1	ADRE1	ADRD1	ADRC1	ADRB1	ADRA1	ARAM 1
RESERVED				ADRN 47	ADRM 47	ADRL 47	ADRK 47	ADRJ 47	ADRH 47	ADRF 47	ADRE 47	ADRD 47	ADRC 47	ADRB 47	ADRA 47	ARAM 47

TK-8792

Figure 3-4 Station Address RAM Format

Table 3-7 Station Address RAM Bit Descriptions

Word	Bits	Field	Description
ARAM Ø	<11:ØØ>	ADRXØ	These bits specify the first bits of each of the physical/logical/broadcast address in the station address RAM. Set and cleared by the port.
ARAM 1	<11:ØØ>	ADRX1	These bits specify the second bits of each of the physical/logical/broadcast addresses in the station address RAM. Set and cleared by the port.
ARAM 2- ARAM 47	<11:ØØ>	ADRX2-47	These bits specify the 2nd to 47th bits of each of the physical/logical/broadcast addresses in the station address RAM. Set and cleared by the port.

3.4 PHYSICAL CHANNEL INTERFACE

The physical channel is implemented in ECL technology and directly interfaces to the ETHERNET transceiver. The physical channel provides Manchester encoding and decoding of all serial data.

3.4.1 Transceiver Signals

The transceiver signals are those signals required by the H4ØØØ transceiver. The following signals the ones used to communicate between the transceiver and link.

1. Collision Presence -- This signal is used to notify the transmit and retry logic of the link of a collision on the ETHERNET.

2. Receive -- This is the data received from the ETHERNET.

3. Transmit -- This is the data to be transmitted from the link.

4. Power -- Power required for the operation of the transceiver.

3.4.2 Receiver

3.4.2.1 Receiver Squelch and Carrier Sense -- Carrier sense is asserted when one or more stations are attempting transmission on the cable, regardless of whether the station sensing carrier is transmitting at that time. Carrier sense will turn on and remain on as long as data is present on the cable.

The carrier sense signal passes through the carrier MUX and is delayed 800 ns to allow proper synchronization of the preamble.

The delayed carrier signal is used as an input to:

- CRC checker,
- Receive shifter,
- Start bit detector, and
- Receive state machine.

The nondelayed carrier signal at the output of the carrier MUX is used as an input to:

- Time Domain Reflectometer (TDR), and
- Interpacket gap counter.

3.4.2.2 **Manchester Decoder** -- The Manchester decoder is used to separate the incoming phase encoded bit stream from the coaxial cable into a data stream and a clock signal. The Manchester data output is used as an input to the CRC checker and the RX shifter. The RX clock generated by the Manchester decoder is used as inputs to the clock shaper, CRC checker, and the RX shifter.

3.4.2.3 **Clock Shaper** -- The clock shaper is used to reshape the Manchester decoder clock output to ensure a minimum clock period and pulse width. The clock shaper protects the receive clock from distortion due to noise at the receive input.

3.4.2.4 **Collision Squelch** -- The collision squelch is similar in operation to the receive squelch. Its output is ORed with the output of the receive squelch circuitry.

Collision is asserted when two or more stations are attempting transmission on the coaxial cable, regardless of whether the station sensing collision is transmitting at that time. The collision squelch is used as an input to the TDR counter, collision jam, and the carrier multiplexer.

This signal is synchronized to the 10 MHz system clock by a dual rank synchronizer before entering any logic operating off the system clock.

3.4.3 Transmitter

The transmitter section of the physical channel interface on the link performs the encoding of data and enables the transmitter. This logic is comprised of the Manchester encoder and transmit enable circuitry.

3.4.3.1 **Manchester Encoder** -- The Manchester encoder is used to translate physically separate signals of lock and data into a single, self synchronizing serial bit stream, suitable for transmission on the coaxial cable. The inputs to the Manchester encoder are a 10 MHz clock and the output of the TX shifter. The Manchester encoder is controlled by the transmit state machine, and collision jam.

3.4.3.2 Transmit Enable Sync -- The TX enable sync logic enables the transmission of data when either the transmission slottime counter has expired and at the end of an interpacket delay.

TX enable sync is controlled by the transmit state machine and collision jam.

3.5 TRANSMIT SECTION

The transmit section of the link module prepares data for transmission onto the ETHERNET. After transmission, this logic will report status on the data transmitted. In order to accomplish this, the transmit section performs the following functions.

- Buffering of transmit data and status information between the host processor and the physical channel
- Parallel-to-serial data conversion
- Preamble generation
- CRC generation

The following paragraphs explain the functional sections of the transmit logic.

3.5.1 Data Section (Link Memory Buffers)

The link memory transmit buffer is made up of two sections, the data section and the status section.

There are two ways that the link module interacts with a link memory transmit buffer.

- Data Section -- This is the data being transmitted. It is written by the port and read by the link.
- Transmit Status Information Section -- Upon successful completion of transmission of a frame or after 16 unsuccessful attempts to transmit a frame, the link will write status information to the link memory transmit buffer.

Figure 3-5 shows the format of the transmit buffer before transmission and Table 3-8 gives a description of the buffer bits.

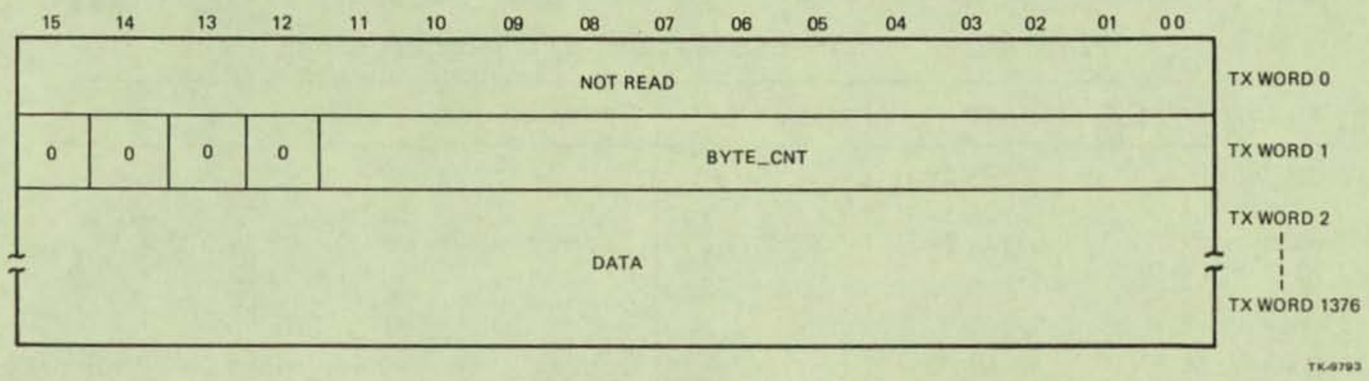


Figure 3-5 Transmit Buffer Format Before Transmission

Table 3-8 Transmit Buffer Bit Descriptions

Word	Bits	Field	Description
TX Word Ø	<15:ØØ>	Not Read	
RX Word 1	<11:ØØ>	BYTE CNT	Transmit Byte Count register. Written and cleared by the port.
TX Word 2- TX Word 1376	<15:ØØ>	Data	Written by the port. When transmitting an odd number of bytes, data found in bits <Ø7:ØØ> in the last entry location of the buffer is sent last.

3.5.2 TX Data Latch

The TX data latch is used to transfer transmit data from the link memory data bus to the TX shift multiplexer. The TX data latch is controlled by the transmit state machine and link memory bus controller.

3.5.3 TX Message Byte Counter

The TX byte counter is implemented as a 12-bit counter that is loaded by the transmit state machine from information contained in the link memory buffer. The TX byte counter contains the number of data bytes to be transmitted over the physical channel and is decremented to zero by 1Ø MHz clock. The count output of the TX byte count register is an input to the transmit state machine.

3.5.4 TX Frame and Byte Sync

The TX frame and byte sync signals provide a 1ØØ ns pulse signal every 16- and 8-clock periods respectively. The TX frame and byte sync signals are implemented as an UP counter and a terminal count detect circuit. These signals are initialized by the transmit state machine. During the odd byte case TX frame is advanced eight bits just before sending the four byte CRC. TX frame and byte sync are controlled by the 1Ø MHz clock and the TX byte count register.

3.5.5 TX Shift MUX

The TX shift multiplexer is used to selectively transfer a 16-bit word of either preamble or transmit data to the TX shifter. The TX shift multiplexer is controlled by the transmit state machine.

Sel 1	Output
Sel TX Data L=Ø	Transmit Data
Sel TX Data L=1	Preamble Data

TX Multiplexer Selection Chart

3.5.6 TX Shifter

The TX shifter converts parallel data from the TX shift multiplexer into a serial output data stream that goes to the CRC generator and the Manchester encoder. The TX shifter is parallel loaded and is controlled by the TX clock and the transmit state machine.

3.5.7 TX Output MUX

The TX output MUX is used to select the output of the TX shifter or the CRC generator for output to the transmitter.

3.5.8 TX Status Information

The transmit status information is written into the link memory transmit buffer by the link either after a successful attempt to transmit a frame or after 16 attempts to transmit a frame have failed. The first two words of the transmit buffer are used to store this information. Figure 3-6 shows the format of these words in relation to the rest of the buffer. Table 3-9 describes each of the status bits.

For information about the transmit data buffer before transmission, refer to paragraph 3.5.1 of this chapter.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
0	ERRS	MTCH	MORE	ONE	DEF	0	0	0	0	0	0	0	0	0	0	TX WORD 0
0	0	0	LCOL	LCAR	RTRY	TDR										TX WORD 1
NOT WRITTEN																TX WORD 2 TX WORD 1376

TK-9796

Figure 3-6 Transmit Buffer Format

Table 3-9 Transmit Status Bit Descriptions

Word	Bits	Field	Description
TX Word Ø	<14>	ERRS	Error Summary - The logical OR of LCOL, LCAR, or RTRY was set. Written and cleared by the link.
TX Word Ø	<13>	MTCH	Station Match - Set by the link when the destination address of the message matches one of the addresses of the UNA.
TX Word Ø	<12>	MORE	Multiple Retries Needed - Set when more than one and less than 16 retries were needed to transmit a frame. Written and cleared by the link.
TX Word Ø	<11>	ONE	One Collision - Set when exactly one retry was needed to transmit a frame. Written and cleared by the link.
TX Word Ø	<1Ø>	DEF	Deferred - Set when the transmitter experienced no collisions but had to defer while trying to transmit a frame. Written and cleared by the link.
TX Word 1	<12>	LCOL	Late Collision - A collision has occurred after the slot time of the channel has elapsed. Written and cleared by the link.
TX Word 1	<11>	LCAR	Loss of Carrier - Carrier was either not present on the channel during transmission or transceiver power was not present. Written and cleared by the link.
TX Word 1	<1Ø>	RTRY	Retry - Transmitter has failed in 16 attempts to transmit the frame due to collisions on the medium. Written and cleared by the link.
TX Word 1	<9:Ø>	TDR	Time Domain Reflectometry Value - Valid only when RTRY or LCAR is set. Written and cleared by the link. All ones indicates an overflow condition.

Table 3-9 Transmit Status Bit Descriptions (Cont)

Word	Bits	Field	Description
TX Word 2- TX Word 1376	<15:ØØ>	NOT WRITTEN	

3.5.9 Transmit State Machine

The transmit state machine controls are:

- The link data path during transmission, and
- The access of the buffers in the link memory.

The transmit state machine is implemented in PALs and consists of the following states:

1. Transmit Enable State - Entered by the transmit state machine when the port asserts TATT. Exited after carrier has gone away and the interpacket gap timer has elapsed.

2. Preamble/Start Bit - Entered after the transmit enable state. The preamble consists of 64 bits of alternating ones and zeros ending in a double one. The preamble is loaded into the TX shifter as four 16-bit words to be shifted serially out onto the wire.

   If the transmitter is enabled and there are no collisions on the wire, the transmit state machine will increment the TX pointer and then load the transmit byte count during the loading of the first word of preamble.

3. Data State - Entered after the fourth word of preamble is loaded into the TX shifter. During this state, data is transferred from the link memory data bus to the TX shifter to be serially shifted onto the wire. This state remains active until the TX byte count register has expired or a collision occurs.

4. CRC State - Entered after the data state if the DTCR bit is not set and exited after 32 bits of CRC are transmitted or a collision occurs.

5. Write Status - Entered after the CRC state. During this state the transmit state machine writes the transmit status into the link memory buffer residing on the port. If there are no collisions and no collision errors, then the transmit state machine resets the TX pointer, write status Word Ø, and write status Word 1.

6. Retry - Entered if there is a collision on the wire. During this state the transmit state machine continues transmitting, a process known as jamming, for 32-bit times. At the end of enforcing the jam, the transmit state machine delays for attempting to retransmit again. This delay is based upon some multiple number of slot times. This state is further described in the RETRY section.

7. Done.

3.6 RETRY LOGIC

The retry logic controls the scheduling of the retransmission of packets when a collision has occurred. This logic uses the binary exponential backoff algorithm. Basically the algorithm waits a generally increasing random number of slot times before retransmission. The random number must be between Ø and 2**K, where K is the min(n,1Ø) for the nth transmission.

3.6.1 Collision Jam

Collision jam keeps the transmitter on for 32-bit times after a collision is detected and the preamble has finished transmitting.

Collision jam is asserted by the leading edge of collision detect and is used as an input to the retry slot time counter, the carrier multiplexer, and the TX enable sync.

3.6.2 Slot Time Counter

The slot time counter is a 51.2 microseconds modulus counter. The slot time counter begins its count upon recognition that the retry state machine is in the backoff state. The output of the slot time counter is used as an input to the retry interval counter.

3.6.3 1Ø MHz Oscillator

The 1Ø MHz oscillator is implemented as an RC voltage controlled oscillator. The oscillator provides the clock for the random number generator.

An RC oscillator is used so that the probability of the retry logic of other nodes on the ETHERNET becoming synchronized is decreased.

3.6.4 Random Number Generator

The random number generator is implemented as a 1Ø-bit binary counter that continuously counts from power-up and is never reset. The 1Ø outputs of the random number generator are the inputs to the random interval mask/latch.

3.6.5 Random Interval Mask/Latch

The random interval mask is combinational logic which masks out bits in the random number according to the number of retries needed to successfully transmit a packet. The mask ensures that the random number is between Ø and 2**K, where K is the min(n,1Ø) for the nth transmission. Inputs to the random interval mask are the 1Ø output lines from the random number generator, and the retry counter. The output from the mask is latched into the random interval latch.

3.6.6 Interval Counter

The retry interval counter is a binary counter that counts the number of slot times that have elapsed. Counting ceases when the number of slot time intervals is equal to the number of random slot times provided by the random interval mask/latch.

3.6.7 Retry Counter

The retry counter counts the number of retransmissions that have occurred. The retry counter is incremented by the interval counter and reset by the transmit state machine. The outputs of the retry counter are the inputs to the transmit status register, and the retry interval mask.

3.6.8 Retry State Machine

The retry state machine, not shown on the block diagram, is used to control the retry process during a collision. The retry state machine is implemented in a PAL and consists of the following states.

1. Jam State - This state is entered if a collision is encountered during transmission of data on the wire. During this state, the transmit section remains transmitting for 32-bit times if the collision occurred during the data state. If, however, the collision occurred during the preamble state, the transmitter will continue transmitting the preamble and then jam for 32 bits. During the jam state, the CRC is disabled.

2. Backoff State - This state is entered after the jam state.

   At the end of enforcing jam, the transmitter delays before attempting to retransmit again. This delay is an integral multiple of slot times. The number of slot times to delay before the nth retransmission attempt is chosen as a uniformly distributed random integer r in the range of Ø<= r <= 2**k where k=min(N,1Ø). If all 16 attempts to transmit fail, the event is reported back as a RTRY error.

3. Force Collision - This is a maintenance self-test function and is valid if the loop and COLL bits in the mode register are both set and the function is reset by clearing these bits.

   Force test allows the microprogrammer to single step through the collision retry algorithm one attempt at a time by simulating a collision on the wire without being physically linked to it. Each attempt to transmit forces a collision internally to the link module. The transmit state machine then goes through the collision jam and retry states. The retry counter is then incremented and the transmit state machine then writes the appropriate status information to link memory.

3.6.9 Time Domain Reflectometry

The TDR counter is ten bits wide modulus counter. It is cleared by the transmit state machine and counts upon the recognition of carrier during transmission. Counting ceases either due to a collision, loss of carrier, or if it has reached its modulus. The value of the TDR is written into memory by the microprocessor. TDR is used to determine the location of suspected cable faults.

3.7 RECEIVE SECTION

The receive logic on the link module is used to:

- Convert serial data to parallel data
- Count the number of bytes received
- Write the received data into the link memory buffers
- Write status information and message length into the receive buffers

3.7.1 Data Section (Link Memory Buffers)

The link memory receive buffer is located in link memory and is written only by the link. It contains the data and status information provided by the physical channel and the receiver state machine. Figure 3-7 shows the format of the link memory receive buffer. Table 3-1Ø describes the status and data bits of the buffer.

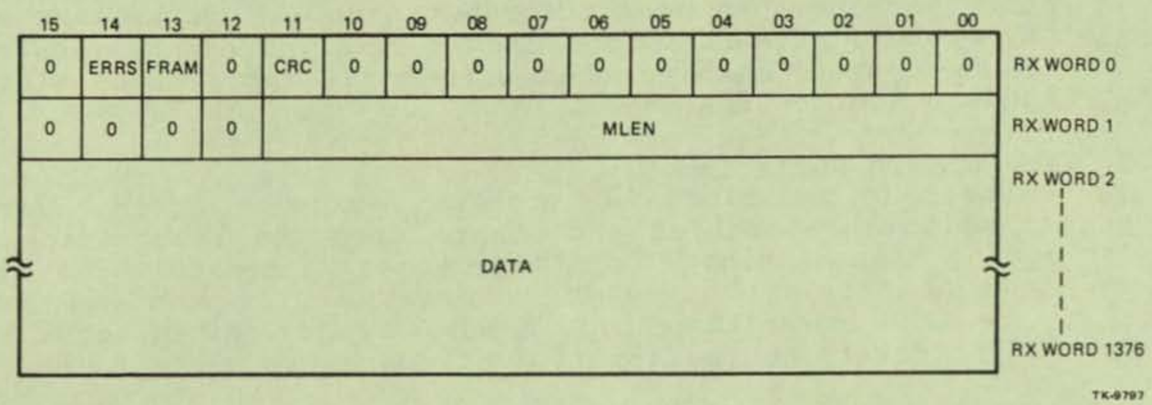


Figure 3-7 Receive Buffer Format

Table 3-1Ø Receive Buffer Status and Data Bit Description

Word	Bits	Field	Description
RX Word Ø	<14>	ERRS	Error Summary - The logical OR of FRAM, CRC. Written and cleared by the link.
RX Word Ø	<13>	FRAM	Frame Error - Indicates that the incoming frame contained a non integer multiple of 8 bits and the CRC value at the last 8-bit boundary was in error. Written and cleared by the link.
RX Word Ø	<11>	CRC	Cyclical Redundancy Check - Frame check error, data is not valid. Written and cleared by the link.
RX Word 1	<11:Ø>	MLEN	Receiver Byte Count Register - Written by the link. This register latches at all ones indicating a babbling node on the network or broken byte count detect logic.
RX Word 2- RX Word 1376	<15:ØØ>	DATA	Written by the link. During the odd byte case, data would be found in bits <7:ØØ> of the last word written.

3.7.2 Receive MUX

The RX multiplexer is used to select data from the ETHERNET or the output of the TX output MUX. The RX multiplexer is controlled by a loopback signal.

Sel 1	Output
Loopback=Ø	Receive Data
Loopback=1	Transmit Data

Multiplexer Output Selection Chart

3.7.3 Receive Shifter

The RX shifter is a 16-bit wide device that frames the incoming serial bit stream into a word stream. A normal reception sequence consists of the continuous shifting of the alternating "ones" and "zeros" that comprise the preamble through the shifter. Upon the recognition of the double "one" pattern that indicates the start of data, the data is then framed into the RX shifter.

The RX shifter is controlled by the receive state machine and RX clock. The output of the RX shifter is transferred to the RX data latch.

3.7.4 RX Data Latch

The RX data latch is used to transfer receive data from the RX shifter to the link memory data bus. The RX latch is implemented as three 8-bit counters whose inputs are the outputs of the RX shifter. Two of these latches work together to transfer data found on 16-bit boundaries to the link memory data bus. The third latch, strobed every 8 bits, is used during the odd byte case to transfer the last byte of data to the link memory data bus. Data strobed into the RX latch is transferred onto the link memory data bus using the handshake provided by the link memory bus controller. The RX latch is controlled by the receive state machine, carrier, bit Ø of the RX byte count register (to detect odd bytes), and the link memory data bus control logic.

3.7.5 RX Frame and Byte Sync

The RX frame and byte sync signals provide a 1ØØ ns pulse signal every 16- and 8-clock period respectively. The RX frame and byte sync signals are implemented as an up counter and a terminal count detect circuit. These signals are initialized by the recognition of start bit. The frame and byte sync are further gated with RCLK L to minimize skew.

3.7.6 RX Byte Counter

The RX byte count is implemented as a 12-bit counter that may be accessed over the link memory data bus. The RX byte counter contains the number of data bytes that are received from the physical channel.

The RX byte counter is incremented by RX clock and is controlled by the receive state machine. The counter will latch up to all ones for an overflow condition. The output of the RX byte counter is passed to the link memory data bus by the TX/RX status multiplexer.

3.7.7 Receive State Machine

Control of the link data path during reception is provided by the receive state machine. The receiver state machine is implemented in PALs and consists of the following states.

1. Receiver Enabled - Entered by the receiver state machine upon setting the on bit in the command register, or upon completing the transfer of an incoming frame, or after the bad packet state, or after the miss state. The receiver state machine stays in this state until carrier is no longer present on the wire.

2. No Carrier - Entered after receive enabled state when carrier is no longer present on the wire and exited when the carrier signal comes up.

3. Carrier - Entered after the no carrier state upon presence of carrier on the wire. During this state the receive state machine looks for a valid preamble, a free receiver buffer, and checks for a runt, no address match, or a start bit.

4. Pointer Reset - Entered after the no carrier state upon presence of carrier on the wire. This state resets the receiver address pointer on the link memory. Exited after one clock period.

5. Pointer Increment - Entered after the pointer reset state. This state increments the receiver address pointer on the link memory to point to the data section of the buffer.

6. Data Request - Entered during the carrier state after recognition of a valid start bit. Exited upon loss of carrier and a bad packet, or status write and a valid packet. During this state, the receive state machine transfers data from the wire to link memory and increments the RX pointers.

7. Bad Packet - Entered after the carrier state if either the packet was less than 64 bytes (runt packet) or the packet did not pass address recognition.

8. Valid Packet - Entered after the carrier state if the packet passed address detection was not a runt packet and there was a free receiver buffer to put the packet in.

9. Miss - Entered after the carrier state if the packet passed address detection, was not a runt packet, and there was no free receiver buffer available.

10. Write Status - Entered after the valid packet state. During this state, status information is written to the link memory buffer.

11. End of Reception - Entered after the write status state. Exited after one clock period.

3.7.8 Interpacket Delay

The interpacket delay prevents the transmission of data for at least 9.6 microseconds after the last carrier detect.

The interpacket delay is asserted by the trailing edge of carrier and is used as an input to the transmit state machine.

3.8 STATION ADDRESS DECODE

The station address detect logic checks the destination address of the incoming packet to determine if the packet is addressed to this node. A packet passes address detection if at least one of the following is true:

1. Logical address match: the destination address of the packet exactly matches one of the 11 possible logical addresses of the node.

2. Physical address match: the destination address of the packet exactly matches the physical address of the node.

3. Promiscuous mode: this mode accepts all packets regardless of the destination address.

4. Enable all multicast: this mode accepts all packets with multicast address regardless of the destination address.

The station address match is then used by the receive state machine. This signal is synchronized to the 10 MHz system clock by a dual rank synchronizer before entering any logic operating off the system clock.

3.8.1 Physical/Logical Address Detection

Physical/logical address detection is done by serial comparing each bit of the destination address on the wire against the contents of the 48*12 station address RAM. The serial compares of the physical and logical addresses are all done in parallel and are enabled by the receiver state machine.

The physical/logical address is written into the station address RAM by 48 sequential memory writes over the link memory data bus.

3.8.2 Promiscuous Mode

In this mode the receiver logic will accept all packets that are sent, regardless of the destination field of the packet.

3.8.3 Enable All Multicast

This mode accepts all packets with multicast addresses regardless of the destination address.

3.9 CRC LOGIC

The CRC logic implements the 32-bit CRC using the AUTODIN-II polynomial as the generating polynomial. The generation and checking of the CRC is done using a 32-bit register implemented in PALs which acts as a shift register, XOR gates, and combinational logic for control.

CRC logic is half-duplex during transmission, reception, and loopback. During loopback the CRC logic is dedicated to the transmit section of the link unless DTCR is set in the link mode register. (If DTCR is set, the CRC logic is dedicated to the receiver.)

For checking the CRC at the end of a packet, a residue detector is used to monitor the data as it shifts through the CRC generator. The residue detector is strobed on 8-bit boundaries. If there are no CRC errors, the output of the CRC to the residue detector is the value of:

11ØØØ111 ØØØØØ1ØØ 11Ø111Ø1 Ø1111Ø11

(Where the leftmost bit corresponds to the X**31 term of the polynomial and the rightmost to the X**Ø term.) Any other value indicates an error.

The input to the CRC generator is either the transmit data stream or the receive data stream. The CRC generator/checker is controlled by the transmit state machine, receive state machine, RX clock, TX clock, and loopback.

NOTE
Output of the CRC PALs are asserted "low".

3.1Ø TX/RX STATUS

The TX/RX status multiplexer is used to transfer status information from TX Word Ø, TX Word 1, RX Word Ø, or RX Word 1 to the link memory data bus for writing into the apropriate link memory buffer that resides on the port. The TX/RX status multiplexer is enabled and controlled by the link memory data bus control logic and the receive and transmit state machines.

Sel 1	Sel Ø	Output
Ø	Ø	TX Status Word Ø
Ø	1	TDR, TX Status Word 1
1	Ø	RX Status Word Ø
1	1	RX byte count

TX/RX Status Multiplexer Selection Chart

3.11 LINK MEMORY

The link memory section is the part of the link module that communicates with the port module. This section contains 16K words of RAM which is used by the link module to buffer packets that are to be received or transmitted on the ETHERNET. This 16K of memory is broken down into sixteen 1536 byte buffers. The first four bytes of each buffer are used to convey status information about the packet.

Addressing of link memory is provided by the port module over one of two over-the-top cables connecting the port to the link.

This memory is arbitrated for and accessed by four different processes:

1. Link transmit state machine.

2. Link receive state machine.

3. DMA engine (described in the UNA Port Module Functional Description).

4. T11 (described in the UNA Port Module Function Description).

The link memory arbitrator resides on the port module.

3.12 LINK MEMORY BUS CONTROLLER

The link memory bus controller is a simple state machine that provides the necessary handshake involved in transferring data between link memory and the transmitter or receiver.

CHAPTER 4
MICROCODE

4.1 OVERVIEW

The microcode provides the microcode instructions necessary to control the T11 microprocessor contained on the port module. This code in conjunction with the T11 is responsible for data encapsulation and decapsulation, data link management, and all channel access functions. This allows for maximum data throughput with a minimum of intervention by the host processor.

In order to understand how the microcode of the DEUNA functions, it is necessary to understand how the DEUNA is programmed. Information on how the DEUNA is programmed can be found in Chapter 3 of the DEUNA User's Guide (EK-DEUNA-UG).

4.2 STRUCTURE

The microcode of the DEUNA is structured as a series of concurrent, cooperating processes that are executed under the control of a supervisor program. These processes are created at the time the DEUNA is powered up and are entirely self-contained. Each process is capable of performing its specific function without assistance from any other process.

4.3 SUPERVISOR

The supervisor is made up of the routines that are needed to:

- Control the scheduling of the different processes used in the DEUNA, and

- Maintain the status and data needed for the operation of the DEUNA.

There are two different types of routines executed by the supervisor, interrupt routines and subroutines.

1. Interrupt Routines -- These routines are executed as a result of a specific interrupt generated by the hardware of the DEUNA. These routines will normally run to completion at the level of the interrupt.

2. Subroutines -- These routines are called by a specific process while that process is running. These routines are accessed by way of a dispatch table contained in ROM. This table is written into the WCS of the DEUNA during initialization.

4.3.1 Initialization

The initialize routine is the first supervisor routine to be executed after the completion of self test.

The function of the initialize routine is to:

1. Reset the hardware of the DEUNA to a known state;
2. Build the supervisor dispatch tables in Writeable Control Store (WCS);
3. Create the data structures in WCS required by the microcode;
4. Clear all the internal counters, the multicast list, mode register, and descriptor ring lengths;
5. Load the physical and broadcast address into the station address RAM on the link module;
6. Enable all hardware interrupts;
7. Load the address of the receive buffers and allocate the transmit buffers; and
8. Start the null process (this executes at priority zero).

4.3.2 Scheduling

The supervisor performs the scheduling of processes through the use of a request mask. When the T-11 receives an interrupt requesting a particular process to be run, the interrupt service routine sets a bit in the request mask. The next time the null process runs it will scan the request mask to see if any low priority processes are scheduled to be run.

All the processes will execute at the CPU priority of zero with the exception of the datagram receive process. As a result there is no context switching between low priority processes. This means that each process, with the exception of the datagram receive process, will run to completion before the request mask is scanned again. The receive process runs at the priority of the hardware interrupt.

When a process has completed it will return to the supervisor by executing an RTI instruction or calling the supervisor command complete routine.

Table 4-1 gives a list of the processes and the order of execution (priority).

Table 4-1 Priority of Processes

Process	Priority
Datagram Receive	1
Port Command	2
Timer	3
Loop and Maintenance	4
Datagram Transmit	5
Null	6

4.3.3 Datagram Receive Process

The datagram receive process is used to transfer receive datagrams from the receive buffers on the link to host memory. This process is the highest priority process because it has the greatest impact on the throughput of the Ethernet and the DEUNA.

The receive process is started by the buffer filled interrupt or by the START port command. The process is ended when:

1. The datagram was written into host memory
2. Status information was written into the descriptor
3. A new buffer descriptor was read from the ring entry

The receive process executes at a hardware priority level of five and can only be interrupted by DMA done, power failure, or errors. Because the amount of processing performed is short (get buffer, start DMA machine), it is possible for other processes to run between the time the DMA machine is started and the DMA done interrupt is generated.

The receive process is initiated in two ways:

- A datagram was received and an interrupt was sent to the T-11.
- A poll demand or start command was received from the host. Either causes an interrupt to generate, and the receive process to start.

The receive process performs the following:

1. Poll receive ring to get a buffer in host memory.
2. Load and start DMA machine.

3. When DMA is done, execute an RTI instruction or run the null process.

Figure 4-1 and Figure 4-2 show the function of the microcode for the receive process.

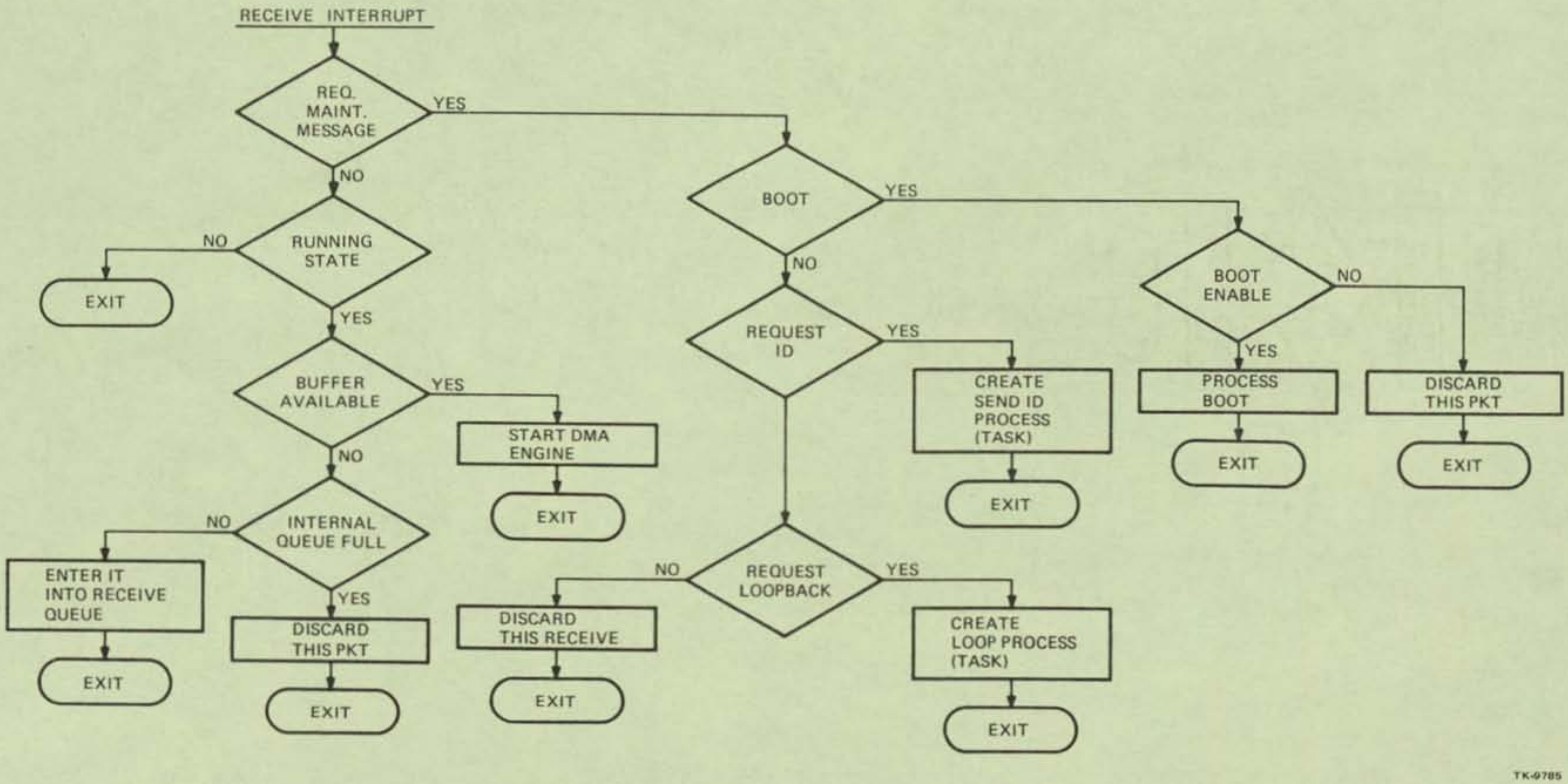


Figure 4-1 Receive Flow Diagram

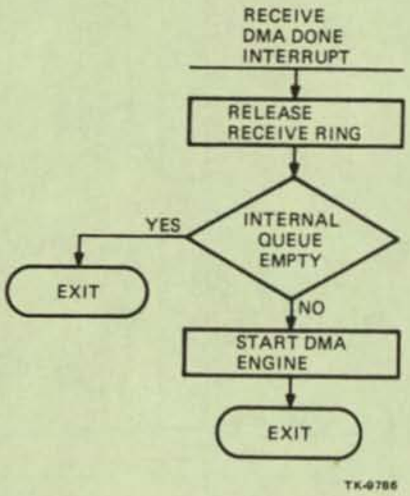


Figure 4-2 Receive DMA Done Flow Diagram

4.3.4 Command Execution Process

The command execution process is used to receive commands from the host processor. The host sends commands to the DEUNA via a structure in host memory called the port control block (PCB). The host tells the DEUNA that it has placed a command in the PCB by writing to PCSRØ. This causes an interrupt to be generated. When the interrupt is received by the T-11, the supervisor will read the command from the PCB and schedule the requested process for execution.

The command process reads the low byte of PCSRØ and uses the code in bits <Ø3:ØØ> to select one of the port command routines.

Figure 4-3 shows the different command processes.

4.3.4.1 Port Commands -- The following port commands are used by the DEUNA.

1. Get PCBB -- The DEUNA reads the address of the PCBB from PCSR2 and PCSR3 and stores it in the WCS.

2. Get CMD -- Requests execution of the ancilliary command process.

3. Self-Test -- Invokes the internal ROM based diagnostic feature of the DEUNA. All datagram activity are aborted and the DEUNA returns to the ready state.

4. START -- The transmit and receive processes are activated and the ring pointers are reset to the base of the rings.

5. BOOT -- The UNA enters the primary load state and requests a program from the load server address.

6. POLL Demand -- The transmit and receive processes are activated if not already active. The transmit and receive rings in host memory are polled.

7. STOP -- The DEUNA completes the current transmit and receive operations and does not fetch any more ring entries until a START command is received.

   This command is implemented by:

   a. Clearing the status flag that indicates the DEUNA is in the running mode, and

   b. Setting the state of both rings to inactive.

   This also causes any datagrams in the link memory buffers to be lost.

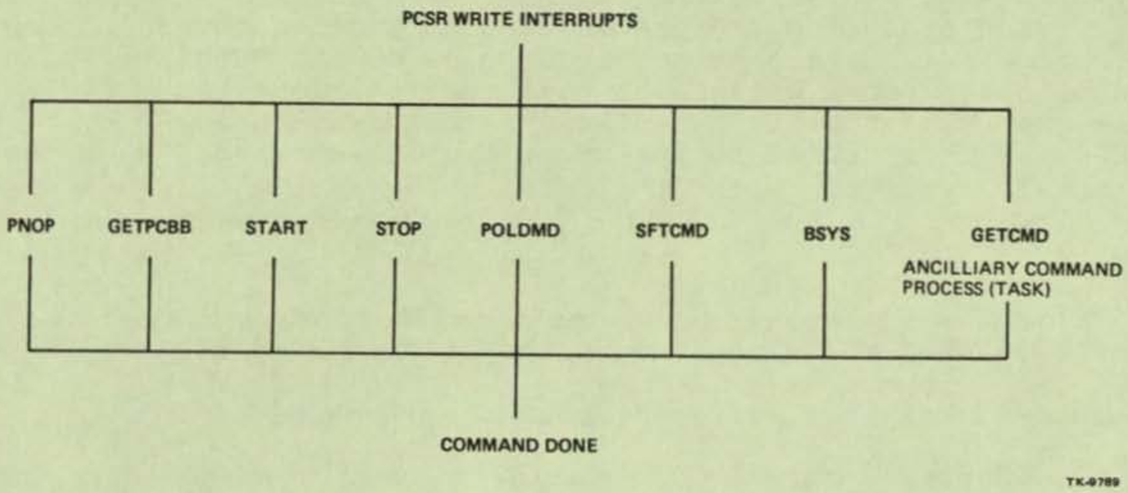


Figure 4-3 Port Command Processes

4.3.4.2 Ancilliary Commands -- The ancilliary commands are subroutines that are called by the get CMD port command. Each of the subroutines executes its specific task and then exits to:

- The command done supervisor routine, or
- Set an error flag and call the appropriate function error routine.

The ancilliary commands executed by the DEUNA are as follows.

1. No Operation (NOP) -- Calls the command done routine.

2. Load and Start at Address (LDSTA) -- Executes a JSR PC instruction directly to the address specified by the PCB.

3. Read Default Physical Address (RDEFPA) -- The physical address contained in ROM on the port module is written to the PCB.

4. Write Physical Address (WRTPA) -- The physical address specified by the PCB is placed in the location reserved for the current physical address (PHYADR). The formatting routine is called to build the data format needed by the address filter of the link module and the data is loaded into the link.

   The link must be halted to execute this command.

5. Read Physical Address (READPA) -- The current physical address is written to the PCB.

6. Write Multicast List (WRTMLT) -- The multicast list is read and stored in a table in WCS. This list along with the broadcast and physical addresses is formatted and written into the address filter in the link.

   The link must be halted to execute this command.

7. Read Multicast List (RDMLT) -- The multicast list is written to the UNIBUS data block specified by the PCB.

8. Read Ring Format (RDRFMT) -- The ring format block of the DEUNA is written to the UNIBUS data block specified by the PCB.

9. Write Ring Format (WTRFMT) -- The ring format is read from the UNIBUS data block and written into WCS of the DEUNA. To maximize performance, the address of the last entry in each ring is calculated. These addresses along with the length of the rings in bytes are saved. The address of both of the rings is written into the ring descriptor for the next entry to be fetched from each of the rings (receive and transmit).

The DEUNA can not be in the running state when this command is executed.

10. Read Counters (RDCNTR) and Read and Clear Counters (RCLCNT) -- The counters that are maintained in WCS are written to host memory. If the command is a read and clear command, the counters are read and then cleared.

11. Dump Internal Memory (DMPMEM) -- A block of data contained in the memory of the DEUNA is specified by the command and transferred to a data buffer in host memory.

12. Load Internal Memory (LDMEM) -- A specified block of data in host memory is copied into the memory on the DEUNA.

13. Read/Write System ID Parameters (RDPARM), (WTPARM) -- The system parameters list is copied from either:

    - A data buffer in host memory to the DEUNA, or

    - The DEUNA to a data buffer in host memory.

14. Read Load Server Address (RDSERV) -- The load server address currently in use by the DEUNA is written into the PCB.

15. Write Load Server Address (WTSERV) -- The load server address in the PCB is written to the DEUNA.

4.3.5 Timer Process

The timer process is executed every second in response to an interrupt generated by the timer on the port module of the DEUNA. The timer is used to:

1. Send an ID message to the ETHERNET every 10 minutes,

2. Keep track of seconds since the counters maintained by the DEUNA were last zeroed. This keeps track of activity in the DEUNA, and

3. Provide timing for various boot operations.

4.3.6 Loop and Maintenance Process

The loop and maintenance process is used to loop data back onto the network, send system ID messages, and perform system boots. The processes are handled as follows:

1. Loop Messages -- Loop service is provided by the microcode to verify that the DEUNA is properly connected to the network and is able to receive and transmit messages.

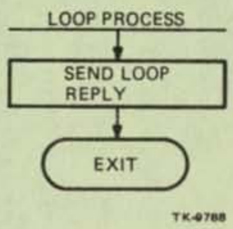


Figure 4-4 Loop Process Flow Diagram

The microcode screens the received messages, in internal memory, to see if there are any loop type messages. (All messages that are not loop type messages are handled as normal datagrams.) If a loop type message is found and is error free it is handled as follows:

a. The microcode modifies some of the address fields.

b. Places the receive buffer into a transmit buffer.

c. Transmits the message.

d. The receive buffer is returned to the receive free buffer queue.

These type of messages are not passed to the host for processing by higher level software.

Figure 4-1 and Figure 4-4 show the function of the microcode for the loop process.

2. System Identification Messages -- When enabled, the microcode will build and transmit a system identification message. This message is transmitted to the network every 8 to 1Ø minutes to identify the node to the network. This address is also sent if a request station ID message is received. A request station ID message is not processed as a datagram.

   Figure 4-1 and 4-5 shows the function of microcode for the system ID process.

3. Boot Messages -- When enabled, the microcode monitors the incoming receive messages for a boot message. If a boot message is received, the following action takes place:

   a. Datagram service is turned off,

   b. A request program load message is sent to the requesting station, and

   c. The WCS is down-line loaded and program execution is started out of the WCS.

   This procedure may be used to load remote console code or to load the system secondary loader. If the system is to be booted, as determined by the boot message, the microcode will halt the system by asserting ACLO and starting the power fail sequence before it transmits the program request message.

   Figure 4-1 shows the function of the microcode for the boot functions.

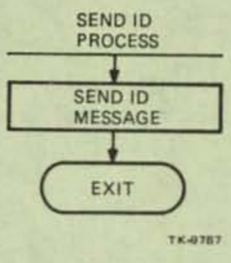


Figure 4-5 Station ID Flow Diagram

4. Power-up Boot -- If the microcode is enabled to do a power-up boot, the microcode will:

   a. Halt the system,

   b. Start power fail sequence,

   c. Transmit program request message, and

   d. Wait for secondary loader.

   If the system boot port command is received, the microcode will handle the request the same way except it does not halt the system.

5. Remote Boot -- For a remote boot from the system ROM (not located on DEUNA), the microcode asserts ACLO.

4.3.7 Transmit Datagram Process

The function of the transmit process is to read a datagram located in host memory and load it into a buffer in link memory for transmission onto the ETHERNET.

The transmit process is activated when the DEUNA receives a poll demand and will be deactivated when the DEUNA comes to a ring entry that is not owned by it.

The transmit process functions as follows.

1. A poll demand or start command generate an interrupt which starts the transmit process.

2. A transmit buffer in link memory is allocated.

3. A ring entry is fetched from the host and is stored in the transmit descriptor of the DEUNA called NEXT.

4. The data described by the ring entry is loaded into the transmit buffer in link memory.

5. The link is given the address of the buffer to be transmitted on the ETHERNET.

6. The link transmit function is started.

7. The ring descriptor in the DEUNA is renamed CURRENT.

8. When the link has finished transmitting the buffer, a transmit done interrupt is generated.

9. The transmit status from the link is stored in the ring entry addressed by the CURRENT ring descriptor in the DEUNA.

10. The ring entry is released and the CURRENT descriptor is marked empty.

11. A return is executed and the NULL process will run. If the transmit process is still the highest process in the request mask the transmit ring will be polled and the process repeated.

Figure 4-6 and Figure 4-7 shows the function of the microcode for the transmit process.

4.3.8 Null Process

The null process scans the request mask to see if any low priority process is scheduled to run. All the low priority processes run sequentially. Each process runs to completion before the request mask is scanned again.

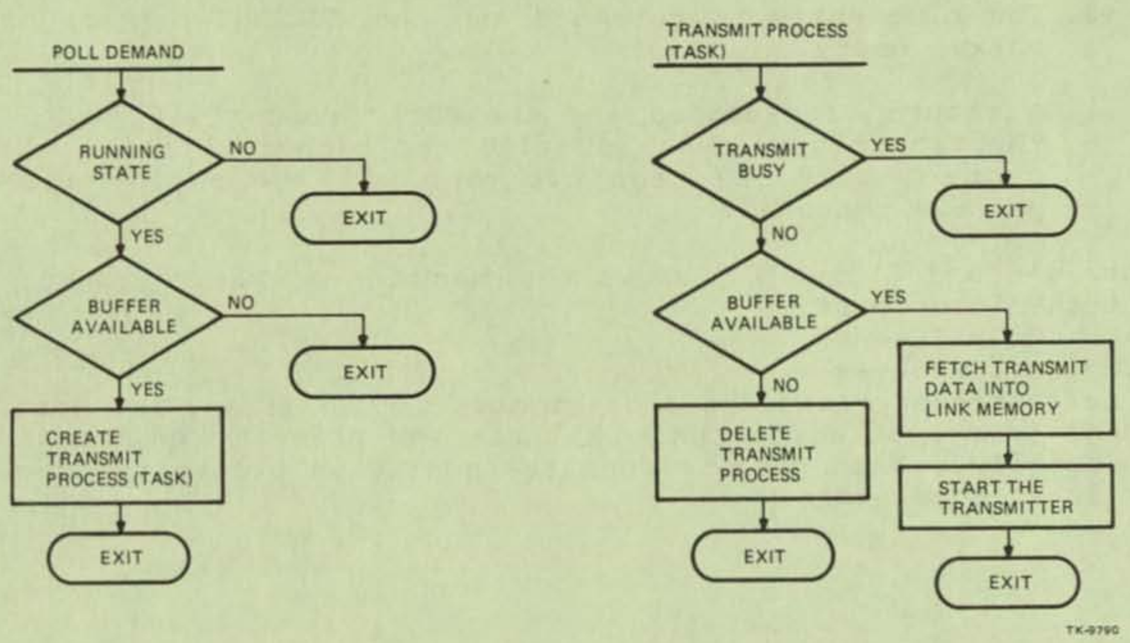


Figure 4-6 Transmit Flow Diagram

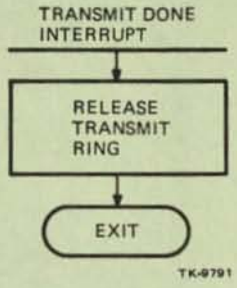


Figure 4-7 Transmit Done Flow Diagram

Digital Equipment Corporation • Bedford, MA 01730

EK-UDA 50-SV-001

UDA50 SERVICE MANUAL

digital

EK-UDA 50-SV-001

UDA50 SERVICE MANUAL

Prepared by Educational Services

Digital Equipment Corporation

The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

This document was set on DIGITAL's DECset-8000 computerized typesetting system.

- **Class A Computing Devices:**

Notice: This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference in which case the user at his own expense may be required to take measures to correct the interference.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DEC
DECUS
DIGITAL
Digital Logo
PDP
UNIBUS
VAX
DECnet
DECsystem-10
DECSYSTEM-20
DECwriter
DIBOL
EduSystem
IAS
MASSBUS
OMNIBUS
OS/8
PDT
RSTS
RSX
VMS
VT

CONTENTS

Page

CHAPTER 1 INTRODUCTION

CHAPTER 2 UDA50 FAULT ISOLATION

Page

FIGURES

TABLES

CHAPTER 1
INTRODUCTION

1.1 SCOPE OF MANUAL

The UDA50 Service Manual describes the maintenance and troubleshooting procedures needed to support the UDA50 Disk Controller. The manual covers both UDA50-resident diagnostic and UDA50 host-resident diagnostic operating procedures. When troubleshooting disk subsystem problems, refer to the service manuals of the disk products in the subsystem for device-specific service information.

1.2 UDA50 MAINTENANCE PHILOSOPHY

The maintenance philosophy planned for the UDA50 Disk Controller is module replacement. Field Service personnel should not attempt to replace or repair component parts within these modules.

1.3 UDA50 FIELD REPLACEABLE PARTS

The UDA50 Disk Controller consists of two hex modules, two flat ribbon intermodule cables, an unshielded Standard Interface (SI) cable assembly, an I/O bulkhead assembly, and some assorted hardware. Figure 1-1 illustrates the major Field replaceable Units (FRUs) in a UDA50 assembly.

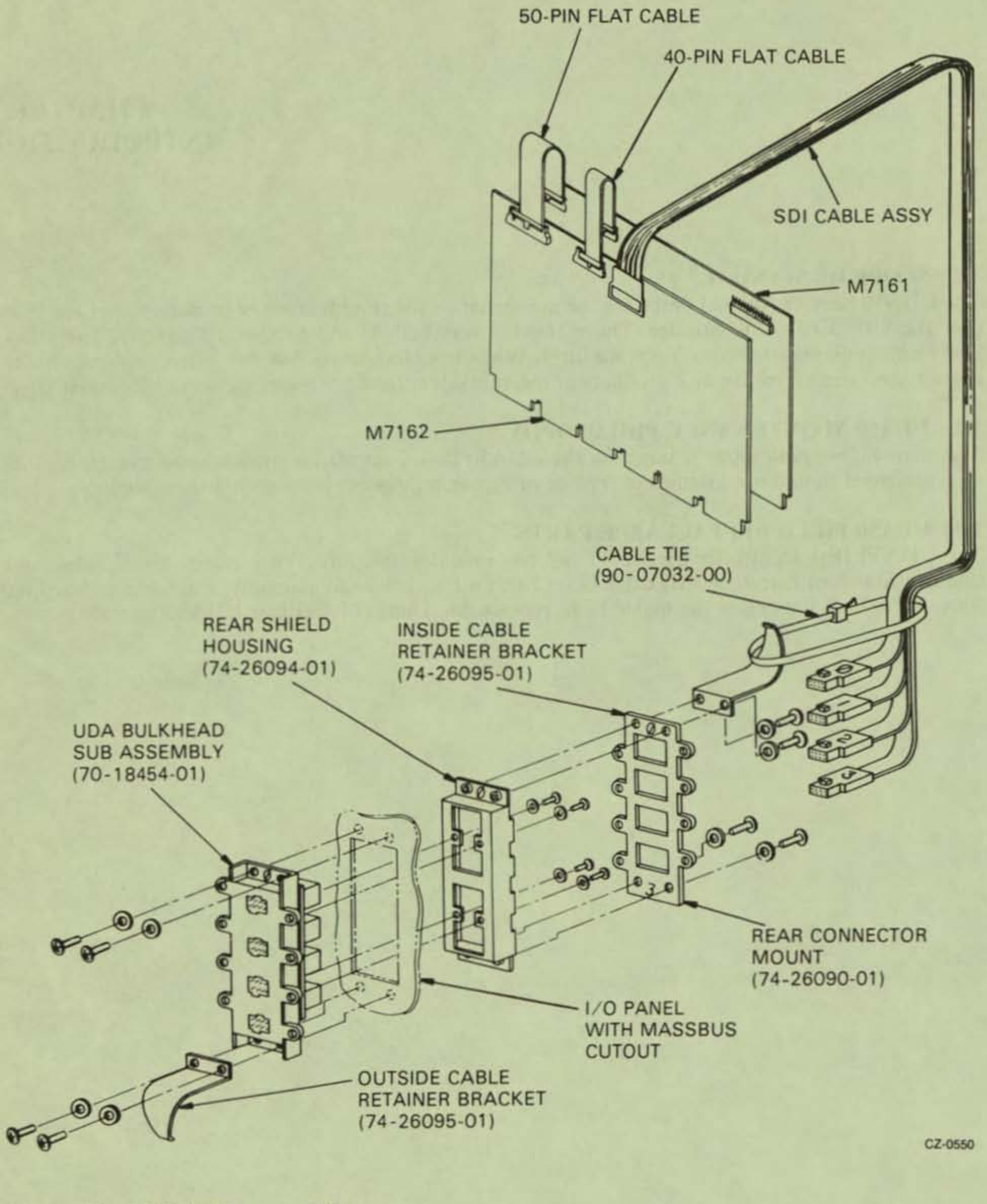


Figure 1-1 UDA50 Illustrated Parts

1.4 UDA50 MAINTENANCE FEATURES

The UDA50 Disk Controller has the following maintenance features.

- UDA50-resident diagnostics
- UDA50 LED maintenance displays
- UDA50 host-resident diagnostics

The UDA50-resident diagnostic is a PROM-based microcode program that performs UDA50 self-diagnosis upon power-up or hard initialization.

A UDA50 maintenance display is located on each UDA50 module. Each display consists of four LEDs. These LEDs display current resident diagnostic activity and error codes caused by malfunctions. Figure 1-2 shows the location of the maintenance LEDs on each module.

The UDA50 host-resident diagnostics contain four tests that isolate subsystem faults to the UNIBUS or disk drives. A system exerciser program is also provided to test the performance of the entire disk subsystem.

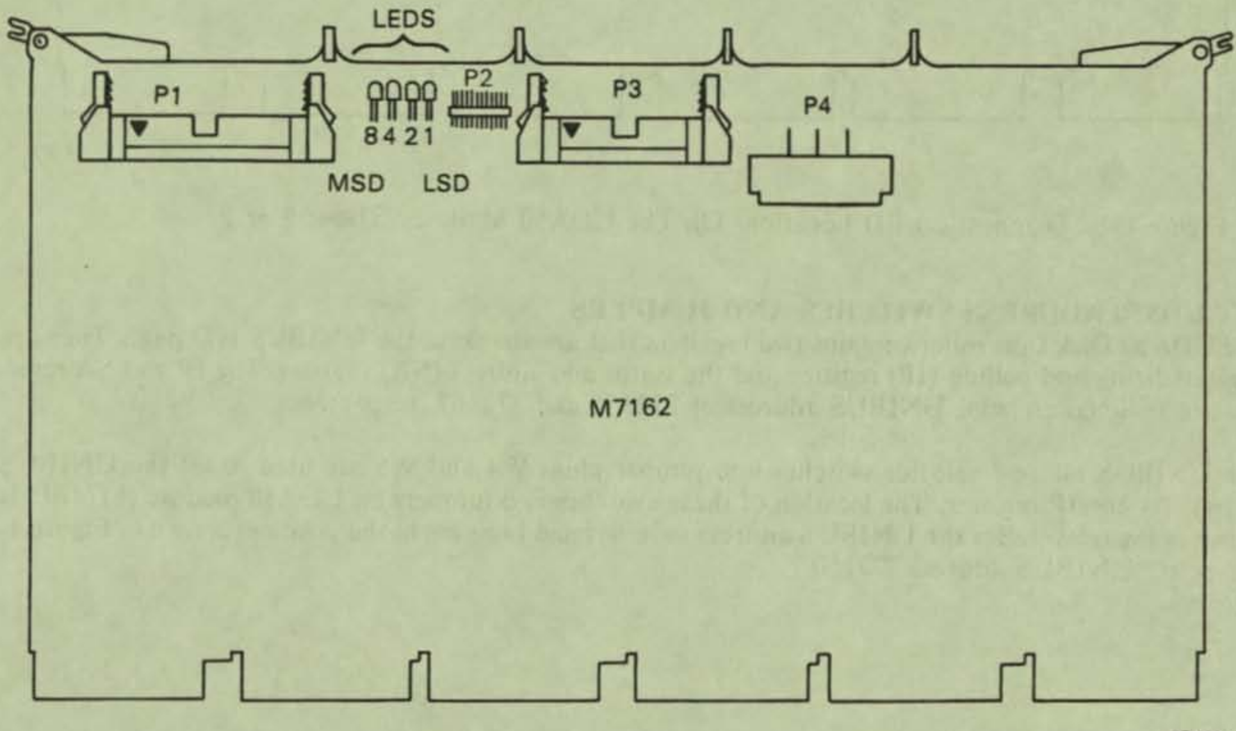


Figure 1-2 Diagnostic LED Locations On The UDA50 Modules Sheet 1 of 2

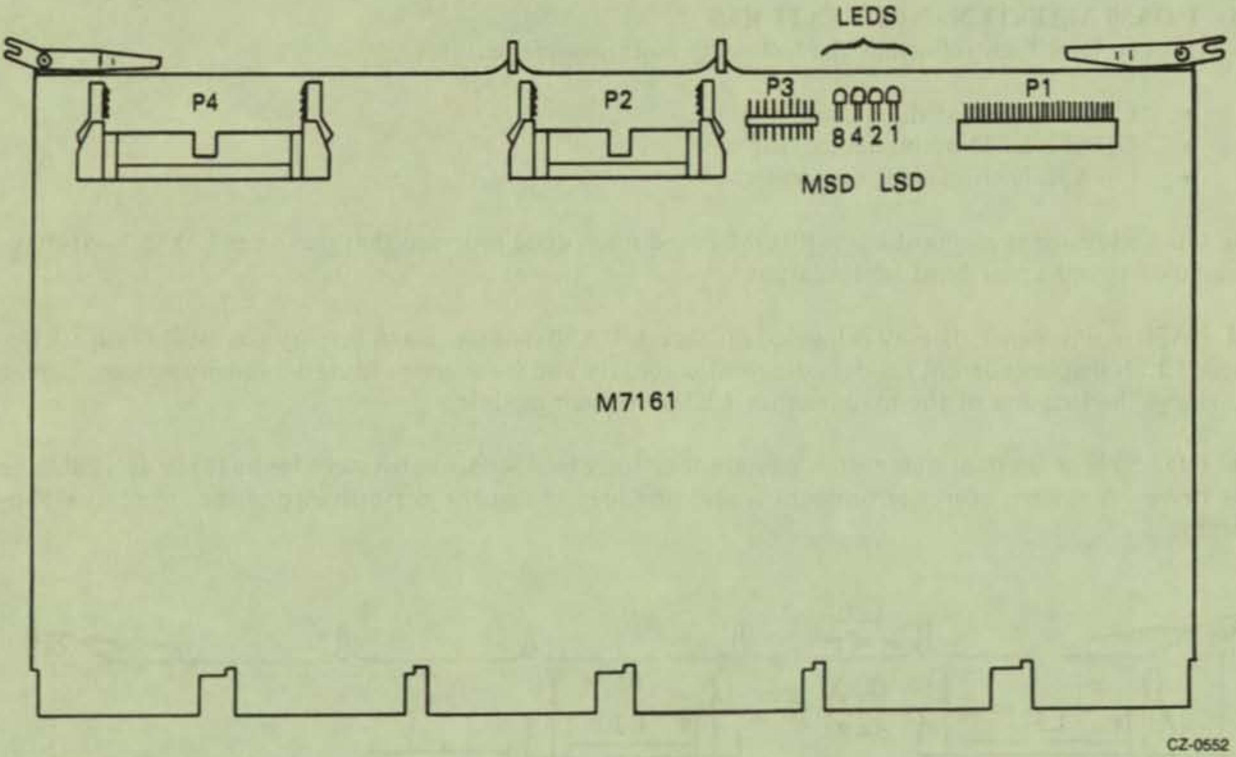


Figure 1-2 Diagnostic LED Locations On The UDA50 Modules Sheet 2 of 2

1.5 UDA50 ADDRESS SWITCHES AND JUMPERS

The UDA50 Disk Controller contains two registers that are visible to the UNIBUS I/O page. They are the initializing and polling (IP) register and the status and address (SA) register. The IP and SA registers are assigned an octal UNIBUS address of 772150 and 772152, respectively.

The UNIBUS address selector switches and jumper plugs W4 and W5 are used to set the UNIBUS address for the IP register. The location of these switches and jumpers on UDA50 module (M7161) is shown in Figure 1-3. Set the UNIBUS address switches and jumpers to the positions shown in Figure 1-4 to select UNIBUS address 772150.

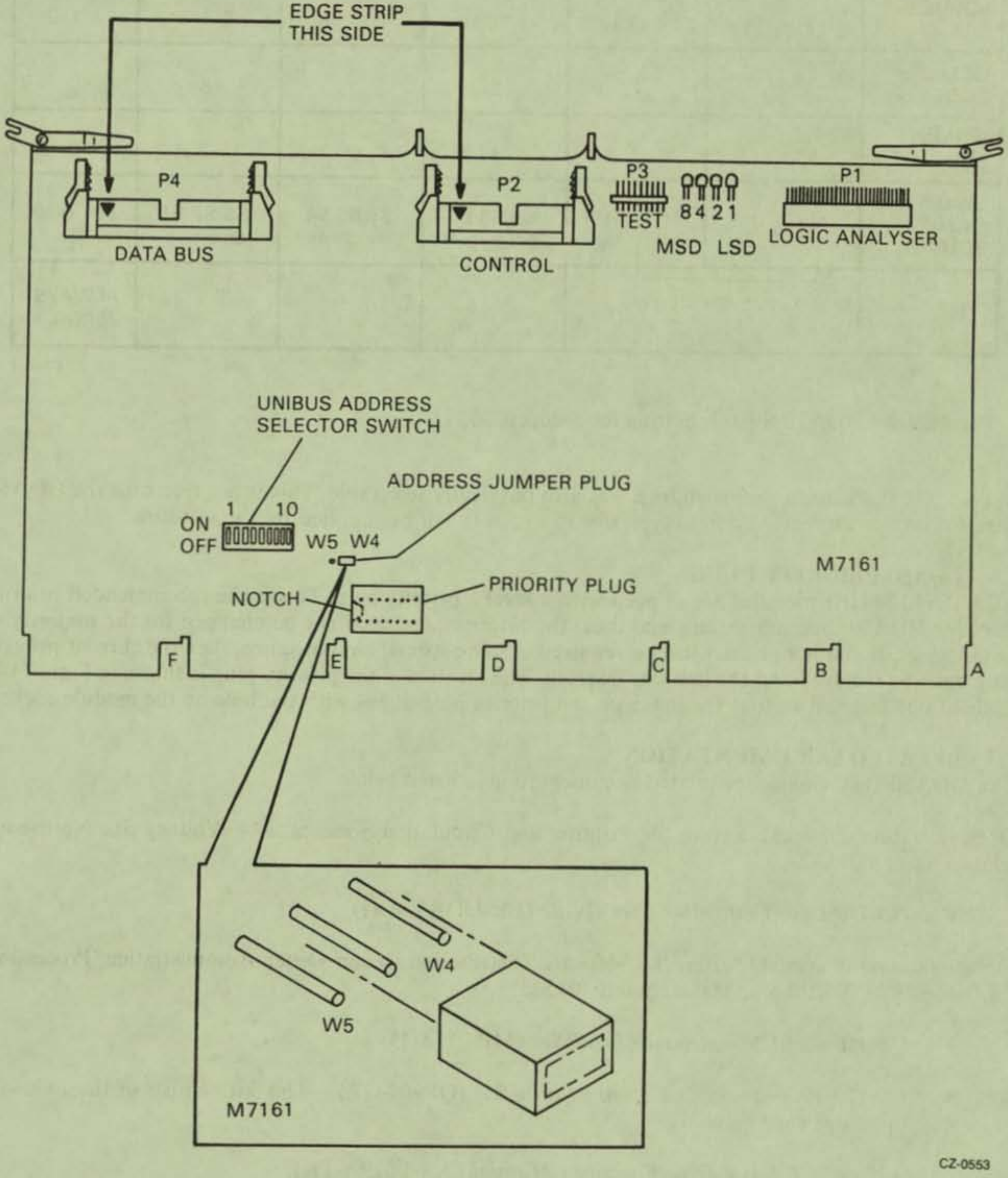


Figure 1-3 UNIBUS Address Selector Switch and Jumper Locations

UNIBUS ADDRESS BITS	17 16 15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
OCTAL CODE	7	7	2	1	5	0
BINARY CODE	1 1 1	1 1 1	0 1 0	0 0 1	1 0 1	0 0 0
UDA50 SWITCH SETTING	1 1 1	1 1 S10 ON	S9 S8 S7 OFF ON OFF	S6 S5 S4 OFF OFF ON	S3 S2 S1 ON OFF ON	W4 0 0 IN
	ALWAYS ONES					ALWAYS ZEROS

CZ-0554

Figure 1-4 UDA50 Switch Setting for Address 772150

In past disk products, a vector address was also physically selectable. This is not true with the UDA50 Disk Controller. A vector address typically 154 (octal) will be supplied by the software.

1.6 UDA50 PRIORITY PLUG

All UDA50 M7161 modules are shipped with a level 5 priority plug. This is the recommended priority level for UDA50 disk subsystems and thus, the priority plug need not be changed for the majority of installations. If another priority level is required in some special circumstance, then the current priority plug must be removed and the new one inserted. The location of the priority plug is shown in Figure 1-3. It should be inserted so that the notch on the priority plug aligns with the hole on the module socket.

1.7 RELATED DOCUMENTATION

The UDA50 Disk Controller related documentation is listed below.

Documentation is available from the Printing and Circulation Services, 444 Whitney St., Northboro, Massachusetts 01532.

- *UDA50 Disk Controller User Guide* (EK-UDA50-UG)

Documentation is available from the Software Distribution Center Order Administration/Processing, 20 Forbes Rd., Northboro, Massachusetts 01532.

- *UDA50 Field Maintenance Print Set* (MP-01331)
- *UDA50 Programmer's Documentation Kit* (QP905-GZ) – This kit consists of the following three software manuals.
    - *MSCP Basic Disk Functions Manual* (AA-L619A-TK)
    - *Storage System Diagnostic and Utilities Protocol* (AA-L620A-TK)
    - *Storage System UNIBUS Port Description* (AA-L621A-TK)

- *UDA50 Maintenance Documentation Kit* (QP904-GZ) – This kit consists of a small looseleaf binder, the *UDA50 Maintenance Guide*, and the current drive maintenance guides that operate on the UDA50.

  - *UDA50 Maintenance Guide* (AA-M185A-TC) – The above maintenance guide is 8 × 5-1/2 inch looseleaf.

  - *RA80 Maintenance Guide* (AA-M186A-TC) – The above maintenance guide is a looseleaf.

  - *Maintenance Guide Looseleaf Binder* (AV-L980A-TK)
(this note goes with Figure 1-4)

NOTE

UNIBUS Address Bit 2 is selected by jumper plugs W4 and W5. Only one jumper plug can be in place at a time. When jumper W4 is IN, Bit 2 equals 0. When jumper W5 is IN, Bit 2 equals 1.

CHAPTER 2
UDA50 FAULT ISOLATION

2.1 UDA50-RESIDENT DIAGNOSTICS

There are two ways of obtaining resident diagnostic information from the UDA50 Disk Controller. The first is through the UDA50 LED error codes. The second is by examining the contents of the UDA50 status/address (SA) register. The SA register contents are also supplied to the host CPU for error logs and diagnostic error reports.

2.1.1 UDA50 LED Error Codes

Table 2-1 lists the LED error codes and indicates which FRU is most likely at fault.

Table 2-1 LED Error and Symptom Codes

M7161 LEDs 8 4 2 1	M7162 LEDs 8 4 2 1	Error Symptoms	Most Likely Failure
o o o •	x x x x	Hex 1; undefined	Undefined
o o • o	o o o o	Hex 2; microcode stuck in init step 2	M7161 or software
o o • •	o o o o	Hex 3; microcode stuck in init step 3	M7161 or software
o • o o	o o o o	Hex 4; microcode stuck in init step 4 or UNIBUS timeout error	M7161 or host inactive
o • o BLINK	o o o o	Hex 4/5; test complete	No problem
o • • o x x x x	x x x x o • • o	Hex 6; undefined	Undefined
o • • • x x x x	x x x x o • • •	Hex 7; undefined	Undefined

Table 2-1 LED Error and Symptom Codes (Cont)

M7161 LEDs 8 4 2 1	M7162 LEDs 8 4 2 1	Error Symptoms	Most Likely Failure
● o o o	o o o o	Hex 8; wrap bit 14 set in SA register	M7161 or software
● o o ● o o o o	o o o o ● o o ●	Hex 9; board one error	M7161
● o ● o ● o ● o	o o o o ● o ● o	Hex A; board two error	M7162
● o ● ● x x x x	x x x x ● o ● ●	Hex B; undefined	Undefined
x x x x	● ● o o	Hex C; timeout error check error code in SA register	Many causes
● ● o ● x x x x	x x x x ● ● o ●	Hex D; RAM parity error	M7162
● ● ● o x x x x	x x x x ● ● ● o	Hex E; ROM parity error	M7161
● ● ● ●	● ● ● ●	Hex F; sequencer error	M7161
Cycling pattern	Cycling pattern	UDA responds to host if cycling lasts less than 2 seconds after host attempts UDA initialization	No problem
		UDA does not respond to host if cycling pattern lasts more than 2 seconds after host attempts UDA initialization	M7161

Note: ● = LED ON o = LED OFF x = Don't care condition

When two codes are given for the same error, both indicate the same failure.

2.1.2 Status/Address Register Error Codes

More detailed information on UDA50 functional and diagnostic error codes is reported through the SA register. The contents of this register may be examined manually through the CPU console at the UDA50 UNIBUS address plus 2. This address is normally 772152. Table 2-2 lists the SA error codes and indicates the most likely FRU at fault.

Table 2-2 SA Register Error Codes

Error Code (Octal)	Error Description	Most Likely FRU Failed
100001	UNIBUS packet read error	M7161*
100002	UNIBUS packet write error	M7161*
100003	UDA ROM and RAM parity error	M7161 or M7162
100004	UDA RAM parity error	M7162
100005	UDA ROM parity error	M7161
100006	UNIBUS ring read error	M7161*
100007	UNIBUS ring write error	M7161*
100010	UNIBUS interrupt master failure	M7161
100011	Host access timeout error	M7161*
100012	Host exceeded command limit	M7161*
100013	UDA SI hardware fatal error	M7162
100014	DM XFC fatal error	M7162
100015	Hardware timeout of instruction loop	M7161*
100016	Invalid virtual circuit identifier	M7161*
100017	Interrupt write error on UNIBUS	M7161*
104040	D processor ALU	M7161
104041	D processor control ROM parity error	M7161
105102	D processor with no BD #2 or RAM parity error	M7162
105105	D processor RAM buffer error	M7162
105152	D processor SI error	M7162
105153	D processor write mode wrap serdes error	M7162
105154	D processor read mode serdes, RSGEN & ECC error	M7162
106040	U processor ALU error	M7161
106041	U processor control register error	M7161
106042	U processor DFAIL/control ROM parity/BD #1 test CNT	M7161
106047	U processor constant PROM error with D processor running SI test	M7161
106055	Unexpected trap found, abort diagnostic	M7161
106071	U processor constant PROM error	M7161
106072	U processor control ROM parity error	M7161
106200	Step 1 data error (MSB not set)	M7161 or RE-INIT

Table 2-2 SA Register Error Codes (Cont)

Error Code (Octal)	Error Description	Most Likely FRU Failed
107103	U processor RAM parity error	M7162
107107	U processor RAM buffer error	M7162
107115	Test count was wrong (BD #2)	M7162
112300	Step 2 error	M7161
122240	NPR error	M7161
122300	Step 3 error	M7161
142300	Step 4 error	M7161

* Possibly the host CPU is at fault.

2.2 UDA50 SUBSYSTEM DIAGNOSTICS

The UDA50 host-resident diagnostics for both the PDP-11 CPU family and the VAX CPU family are described briefly in the following paragraphs. A more detailed description of these diagnostic programs is found in the diagnostic listings that are available from the Software Distribution Center.

If the diagnostic programs report errors, refer to the troubleshooting procedure in Paragraph 2.3.

2.2.1 PDP-11 Subsystem Diagnostics

2.2.1.1 CZUDEA0 – Formatter program – Most disk drives will be shipped with formatted disks. On formatted disk drives, it will not be necessary to run the formatter program. Refer to the disk drive user guide to determine if you must run the formatter program before the diagnostic program.

2.2.1.2 CZUDCA0 – UDA50 Host-Resident Diagnostic – This diagnostic consists of the following four tests:

- Test 1 – UNIBUS addressing test
- Test 2 – Disk-resident diagnostic test
- Test 3 – Disk functional test
- Test 4 – Disk exerciser test

This program will ask some hardware and software questions of the user. A sample printout of these questions, when the default conditions are elected, is shown below.

```
CHANGE HW (L) ? N *

# UNITS (D) ? 1

UNIT 0
UNIBUS ADDRESS OF UDA (0) 172150 ?
VECTOR (0) 154 ?
BR LEVEL (D) 5 ?
UNIBUS BURST RATE (D) 0 ?
DRIVE NUMBER (D) 0 ?
EXERCISE ON CUSTOMER DATA AREA IN TEST 4 (L) N ?

CHANGE SW (L) ? N *

ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSIS (L) N ?

REMAINING SOFTWARE QUESTIONS APPLY TO TEST 4 ONLY

ERROR LIMIT (D) 32 ?
READ TRANSFER LIMIT IN MEGABYTES - 0 FOR NO LIMIT (D) 0 ?
SUPPRESS PRINTING SOFT ERRORS (L) Y ?
DO INITIAL WRITE ON START (L) Y ?
ENABLE ERROR LOG (L) N ?
```

* The following questions will not be asked when you answer this question with a "N". They are listed here to show you what will be asked if you answer "Y", and what the defaults are.

2.2.1.3 CXDUBA0 – DECX11 Module – The DECX11 module operates in two modes:

- It performs data transfer over the UNIBUS to the UDA50 internal buffer.
- It performs reads and writes to the customer data area of the disks.

2.2.2 VAX Subsystem Diagnostics

2.2.2.1 ZZ-EVRLB – Formatter Program – Most disk drives will be shipped with formatted disks. On these disk drives, it will not be necessary to run the formatter program. Refer to the disk drive user guide to determine if you must run the formatter before the diagnostic program.

2.2.2.2 ZZ-EVRLA – UDA50 Host-Resident Diagnostics – The VAX UDA50 host-resident diagnostic contains the same four tests as the PDP-11 version.

- Test 1 – UNIBUS addressing test
- Test 2 – Disk-resident diagnostic test
- Test 3 – Disk functional test
- Test 4 – Disk exerciser test

Use the default conditions for this test during system installation and no hardware or software questions will be asked. If the default conditions are not elected, the program will ask the user a series of software questions. These software questions are described in the diagnostic listing available from the Software Distribution Center.

2.2.2.3 ZZ-EVRLC – SI Generic Disk Exerciser – This program tests the read and write ability of any SI type disk drive, and will display differences in the read and write data to the operator.

2.3 UDA50 SUBSYSTEM TROUBLESHOOTING

A brief UDA50 subsystem troubleshooting flowchart is illustrated in Figure 2-1. The troubleshooting procedure recommends examining the error log first.

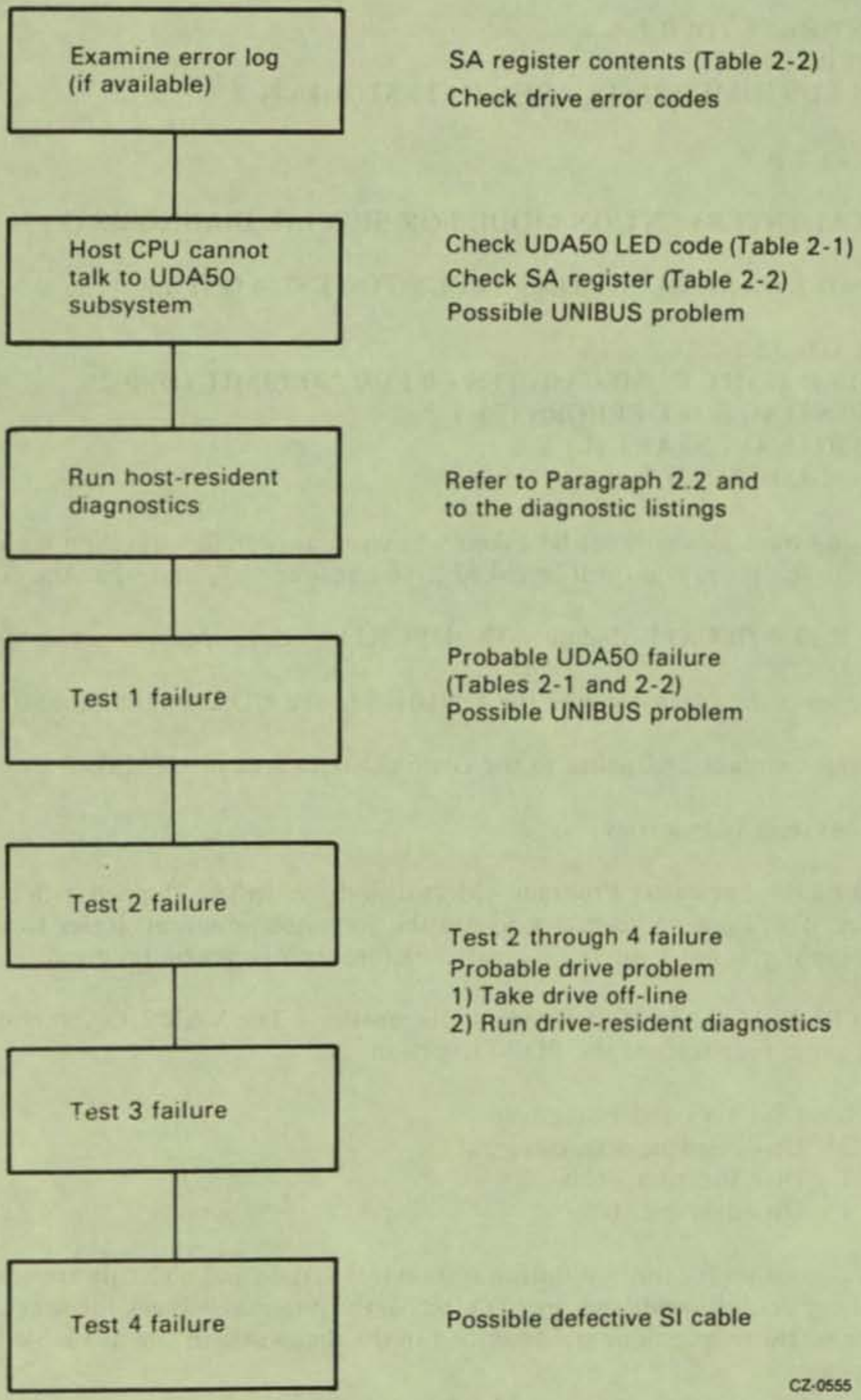

Examine error log
(if available)
SA register contents (Table 2-2)
Check drive error codes
Host CPU cannot talk to UDA50 subsystem
Check UDA50 LED code (Table 2-1)
Check SA register (Table 2-2)
Possible UNIBUS problem
Run host-resident diagnostics
Refer to Paragraph 2.2 and to the diagnostic listings
Test 1 failure
Probable UDA50 failure
(Tables 2-1 and 2-2)
Possible UNIBUS problem
Test 2 failure
Test 2 through 4 failure
Probable drive problem
1) Take drive off-line
2) Run drive-resident diagnostics
Test 3 failure
Test 4 failure
Possible defective SI cable
CZ-0555

Figure 2-1 UDA50 Subsystem Troubleshooting Flowchart

2.3.1 UDA50-Resident Diagnostics

The UDA50-resident diagnostics are initiated when power is applied to the UDA50 Disk Controller. The CPU should be halted during this test. The four LED indicators on each UDA50 module will display a cycling pattern in the LEDs. The cycling pattern in the LEDs signifies the completion of a successful UDA50 diagnostic test. Figure 1-2 shows the location of the four LEDs on each UDA50 module.

If the UDA50 LEDs do not display the cycling pattern after power is applied, look up the LED error code in Table 2-1 to locate the problem.

2.3.2 UDA50 Host-Resident Diagnostic

A brief description of the UDA50 host-resident diagnostics is presented in Paragraph 2.2. Use the UDA50 host-resident diagnostic to isolate problems to the UNIBUS or the disk drives. These diagnostics will send back error messages concerning drive status or real-time drive state. Since the drive status error messages are unique to each disk drive, they will be described in the drive maintenance guide and service manual. The real-time drive state error messages describe what is happening in the drive. Paragraph 2.3.3 describes the error message information.

2.3.3 Subsystem Error Message Information

Error messages will be typed out during the UDA50 host-resident diagnostic if a problem is detected. Two sample printouts are shown below. Sample 1 shows typical drive error printout. Note that the real-time drive state (RTDS) and the drive status are given in the last two lines of sample 1. Sample 2 shows a typical UDA50 error printout. The last line of sample 2 gives the contents of the SA register.

Sample 1:

Printout of a Drive Error

```
CZUDC HRD ERR 00044 ON UNIT 00 TST 004 SUB 000 PC: 021044
DISK EXERCISER DM PC:5110 UDA AT 172150 DRIVE 032 RUNTIME 0:00:23
ENTIRE RCT AREA SEARCHED, COULD NOT FIND RBN TO REPLACE
LBN WITH HEADER COMPARE ERROR
SEARCING FOR LBN: 900

CZUDC SFT ERR 00006 ON UNIT 00 TST 004 SUB 000 PC: 021044
DISK EXERCISER DM PC:5324 UDA AT 172150 DRIVE 032 RUNTIME 0:00:37
TIMEOUT OF DRIVE DURING WRITE ATTEMPT
WRITE ATTEMPT RETRIES: 0
L/DBN NUMBER 5252
ACTUAL L/R/DBN 0
TRK 1 GRP 0 CYL 6
ORIGIN OF LAST SEEK WAS CYL 5 GROUP 1
REAL TIME DRIVE STATE 8001
STATUS: 0001 1100 0000 0A00 0000 0613 1020
```

Sample 2:

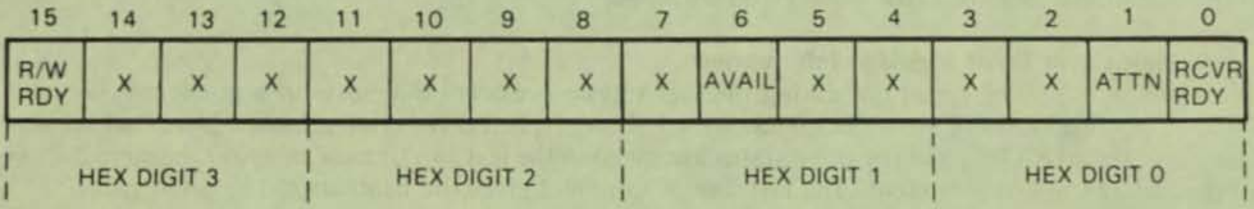
Printout of a UDA50 Error

```
CZUDC DVC FTL ERR 00005 ON UNIT 00 TST 001 SUB 002 PC: 023710
UDA INITIALIZE ERROR
UDA RESIDENT DIAGNOSTICS DETECTED FAILURE
UDASA REGISTER = 106040
```

2.3.3.1 Real-Time Drive State Message Interpretation – The real-time drive state message consists of 4 hexadecimal digits. Only four state bits within these hexadecimal digits are of diagnostic value to the field service engineer. The rest of the bits are too transitory and are masked out before the RTDS message is printed. The following are the four important RTDS state bits.

- Read/write ready (R/W RDY)
- Drive available (AVAIL)
- Attention (ATTN)
- Receiver ready (RCVR RDY)

The location of these four state bits within the hexadecimal code is shown in Figure 2-2. The interpretation of the RTDS message requires an understanding of the causes and effects of each bit in the RTDS message. It also requires an understanding of what is meant by drive on-line, drive off-line, drive available and drive unavailable. Definitions of each of the of the four RTDS message bits and the on-line and available states are given.



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W RDY	X	X	X	X	X	X	X	X	AVAIL	X	X	X	X	ATTN	RCVR RDY
HEX DIGIT 3				HEX DIGIT 2				HEX DIGIT 1				HEX DIGIT 0			

X = Do not care condition

CZ-0556

Figure 2-2 Real-Time Drive State Interpretation

The following four terms define the state of the drive as seen from the controller.

- Drive Off-line – The drive is not operational and may not communicate with the controller via the drive control protocol.

- Drive Unavailable – The drive is operating, is visible to, and may at times communicate with the controller. However, the controller may not fully utilize the drive because the drive is "drive on-line" to another controller.

- Drive Available – The drive is visible to, capable of communicating with, and capable of executing an on-line command. However, the drive is not currently drive on-line to any specific controller.

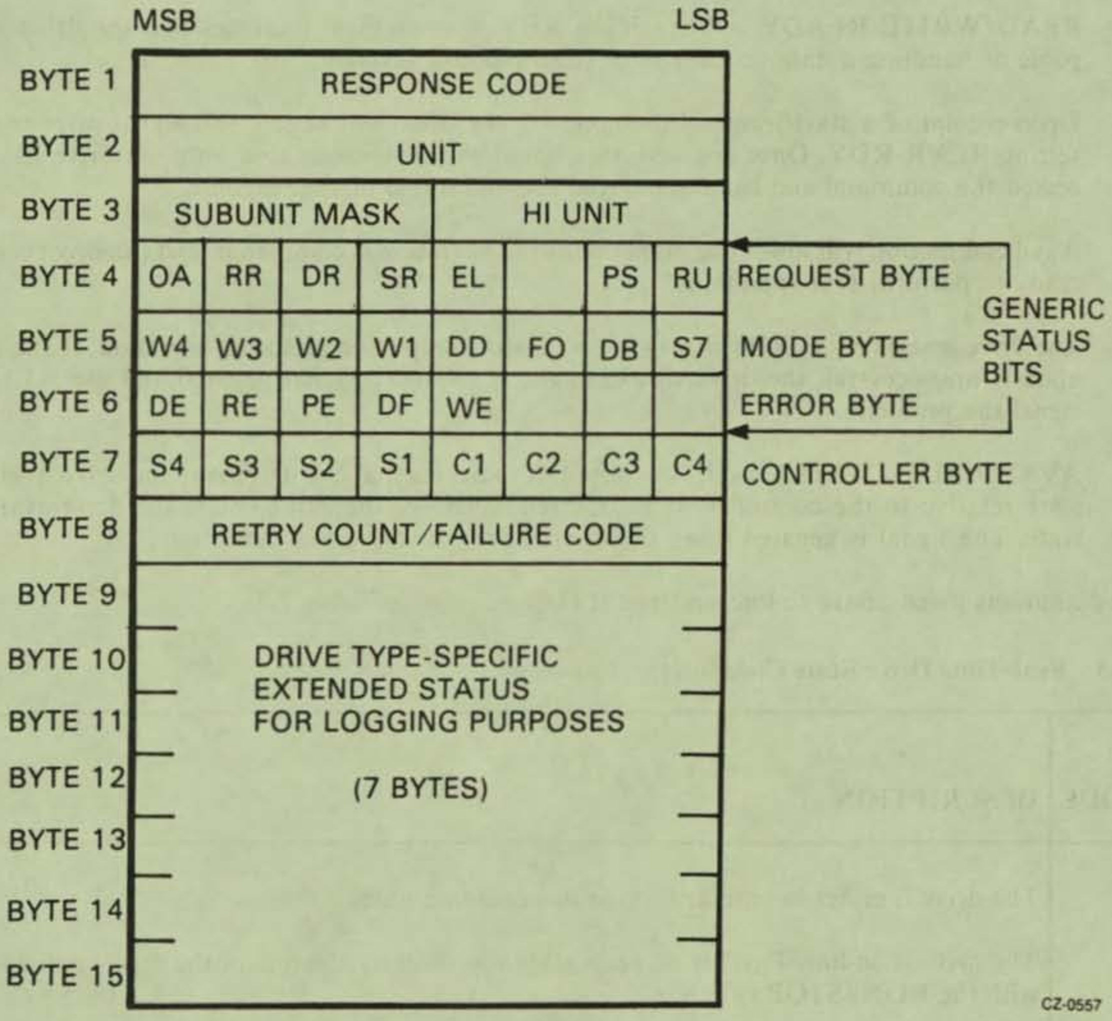
- Drive On-line – The drive is dedicated to the exclusive use of a particular controller, and is not available to any alternate controller.

The following paragraphs explain the causes, effects, and interrelationships of the four state bits within the RTDS message.

- RECEIVER READY – When RCVR RDY is asserted, it indicates that the drive is ready to receive a command on the SI interface WRITE/COMMAND line. RCVR RDY is negated while the drive is processing a command.

- ATTENTION – This notifies the controller that a potentially significant status change has occurred in the drive.

  When in the drive on-line state, the drive asserts this signal whenever any of its generic status bits (see Figure 2-3) change, except for the following three cases.



	MSB							LSB	
BYTE 1	RESPONSE CODE								
BYTE 2	UNIT								
BYTE 3	SUBUNIT MASK				HI UNIT				
BYTE 4	OA	RR	DR	SR	EL		PS	RU	REQUEST BYTE
BYTE 5	W4	W3	W2	W1	DD	FO	DB	S7	MODE BYTE
BYTE 6	DE	RE	PE	DF	WE				ERROR BYTE
BYTE 7	S4	S3	S2	S1	C1	C2	C3	C4	CONTROLLER BYTE
BYTE 8	RETRY COUNT/FAILURE CODE								
BYTE 9	DRIVE TYPE-SPECIFIC EXTENDED STATUS FOR LOGGING PURPOSES (7 BYTES)								
BYTE 10									
BYTE 11									
BYTE 12									
BYTE 13									
BYTE 14									
BYTE 15									

Figure 2-3 Drive Status Bytes

1. A generic status bit changes as a direct consequence of the corţect operation of a command.

2. A generic status bit changes as the result of an error in the reception, validation, or execution of a command.

3. The "RE" status bit changes due to a transmission error outside of a command. The "RE" bit is described in byte 6 of the drive status message.

An on-line drive may assert ATTN regardless of whether a command is in progress or not. The drive will continue to assert this signal until it receives a valid get status command from the controller. At this point, the drive will negate the ATTN signal.
A drive in the available state, that is spinning, always asserts the ATTN signal. The ATTN signal is negated if any condition arises that would prevent the available drive from spinning up under controller command.

- READ/WRITE READY – When R/W RDY is asserted, it indicates that the drive is capable of handling a data transfer to or from the disk surface.

  Upon receipt of a start frame of a command, the drive will negate this signal prior to reasserting RCVR RDY. Once negated, this signal will remain negated until the drive has processed the command and has transmitted the end frame of the response.

  Any head motion will lower this signal until the operation is completed, and the drive is again ready to perform I/O operations.

  The drive asserts R/W RDY after the successful completion of a seek operation. If the operation is unsuccessful, the drive will keep the R/W RDY signal negated and use ATTN to signal the problem.

- AVAILABLE – When AVAIL is asserted, it indicates that the drive is in the drive available state relative to the controller. It is asserted whenever the drive enters the drive available state. The signal is negated when the drive leaves the drive available state.

Use the definitions given above to interpret the RTDS message in Table 2-3.

Table 2-3 Real-Time Drive State Code Interpretation

RTDS HEX CODE	DESCRIPTION
0000	The drive is either in initialization or in an off-line state.
0001	The drive is on-line. Possibly an error state was recently cleared, or the drive spun down with the RUN/STOP switch out.
0002	This code indicates an invalid drive state. ATTN is asserted and the drive cannot receive controller commands with RCVR RDY negated.
0003	The drive is on-line and one of two conditions exist. 1. The disks are spinning, and there is an error state. 2. The disks are not spinning and there is a switch change active.
0040	This code indicates an invalid drive state. RCVR RDY should be asserted if the drive is in the available state.
0041	The drive is available, but not spinable. The RUN/STOP switch is not pushed in, or there could be an open interlock that prevents spin up.

Table 2-3 **Real-Time Drive State Code Interpretation**

RTDS HEX CODE	DESCRIPTION
0042	This code indicates an invalid drive state. ATTN is asserted and the drive cannot receive controller commands with RCVR RDY negated.
0043	The drive is available and spinable.
8000	This code indicates an invalid drive state. R/W RDY should not be asserted with RCVR RDY negated.
8001	This is the normal drive on-line state.
8002	This code indicates an invalid drive state. ATTN is asserted and RCVR RDY is negated, preventing the drive from receiving controller commands.
8003	The drive is on-line and one of two conditions exist: 1. There is a change of switch state. 2. The drive is reporting a successful retry of a seek with recalibration.
8040	This code indicates an invalid drive state. R/W RDY and AVAIL should never be asserted together. Also, ATTN should be asserted when the drive is available and spinable.
8041	This code indicates an invalid drive state. R/W RDY and AVAIL should never be asserted together. Also, ATTN should be asserted when the drive is available and spinable.
8042	This code indicates an invalid drive state. R/W RDY and AVAIl should never be asserted together. Also, ATTN is asserted and the drive cannot receive controller commands with RCVR RDY negated.
8043	This code indicates an invalid drive state. R/W RDY and AVAIL should never be asserted together.
FFFF	The controller is unable to get a valid drive state.

2.3.3.2 Status Message Bytes – The status line found in error message sample 1 is the result of the diagnostic performing a get status command. Fourteen of fifteen status bytes are printed out by the error message. Figure 2-3 shows the breakdown of the fifteen status bytes. The first byte is not printed out since it is a response code to the get status command. Bytes 9 through 15 contain drive-specific status bits and the drive service manual or maintenance guide should be consulted for interpretation. Table 2-4 gives a bit description of status message bytes.

Table 2-4 Byte Description of Status Message Bit Description

Status Byte	Bit Description
Byte 1	**Response Code Field** – Byte 1 is the response code to a controller command.
Byte 2	**Unit Number** – The unit number consists of two hexadecimal digits representing the unit number of the selected disk drive returning the status (0-254).
Byte 3	**Subunit Mask** – The subunit mask is a four-bit representation of the sub-unit that is returning the status message. The right-most bit position represents sub-unit 0. The left-most bit position represents sub-unit 3. Only one bit can be set at a time. UDA50 subsystems can handle only drives that can contain up to two sub-units. Therefore, the valid numbers in this status byte can only be a hexadecimal 1 or 2. Figure 2-4 shows the bit layout. For drives that contain no sub-units (e.g. the RA80), the right-most bit position is always set indicating sub-unit 0.
Byte 3	**High Unit Number** – Byte 3 contains the upper four bits to a 12-bit (3 hexadecimal digits) unit number.
Byte 4	**OA** – A logical one in this position indicates the drive is unavailable to the UDA50. A logical zero indicates the drive is available to the UDA50.
Byte 4	**RR** – A logical one in this position indicates that the drive requires an internal readjustment. Some drives do not use this bit.
Byte 4	**DR** – A logical one in this position indicates that there is a request for a diagnostic to be loaded in the drive microprocessor memory. A logical zero indicates that no diagnostic is being requested of the host system.
Byte 4	**SR** – A logical one in this position indicates that the drive spindle is up to speed. A logical zero indicates the drive spindle is not up to speed.
Byte 4	**EL** – A logical one in this bit position indicates that there is loggable information in the extended status area (bytes 9-15). A logical zero indicates that no information is available in the extended status area.
Byte 4	**PS** – A logical one in this bit position indicates that the drive port select switch for this controller is pushed in (selected). A logical zero indicates that the switch is out.
Byte 4	**RU** – A logical one in this position indicates that the RUN/STOP switch is pushed in (RUN). A logical zero indicates the switch is out (STOP).
Byte 5	**W4-W1** – Logical ones in any of these four bit positions represent the write-protect status for the sub-unit represented. (e.g., a 0001 indicates that sub-unit 0 within the selected drive is write-protected.)
Byte 5	**DD** – A logical one in this bit position indicates that the drive has been disabled by a controller error routine or diagnostic. The FAULT light is on when this bit is set. A logical 0 indicates that the drive was enabled by a controller error routine or diagnostic.

Table 2-4 Byte Description of Status Message Bit Description (Cont)

Status Byte	Bit Description
Byte 5	**FO** – A logical one in this position indicates that the drive can be formatted.
Byte 5	**DB** – A logical one in this position indicates that the diagnostic cylinders on the drive can be accessed.
Byte 5	**S7** – A logical one in this bit position indicates that the 576 byte sector format is selected. A logical zero indicates that the 512 byte sector format is selected.
Byte 6	**DE** – A logical one in this position indicates that a drive error has occurred and the drive FAULT lamp may be on.
Byte 6	**RE** – A logical one in this position indicates that an error occurred in the transmission of a command between the drive and the UDA50. The error could be a checksum error or an incorrectly formatted command string.
Byte 6	**PE** – A logical one in this position indicates that improper command codes or parameters were issued to the drive.
Byte 6	**DF** – A logical one in this postion indicates a failure in the initialization routine of the drive.
Byte 6	**WE** – A logical one in this position indicates a write lock error has occurred.
Byte 7	**S4-S1** – This is a four-bit representation of the sub-units that have their attention available messages suppressed in the UDA50. The right-most bit position represents sub-unit 0. The left-most bit position represents sub-unit 3. If one of the bits is set, it indicates that the controller is not to interrupt the host CPU with an attention available message when the specified sub-unit raises its available real-time drive status line to the UDA50. The S4-S1 bits reflect the results of a change controller flags command in which attention-available messages are not desired for certain sub-units.
Byte 7	**C1-C4** – This is a four-bit drive status code indicating various states of drive operation. At the present time only three codes are valid. A code of 0000 = drive normal operation. A code of 1000 = the drive is off-line due to being under control of a diagnostic. A code of 1001 = the drive is off-line due to another drive having the same unit identifier (e.g. serial number, drive type, class etc.).

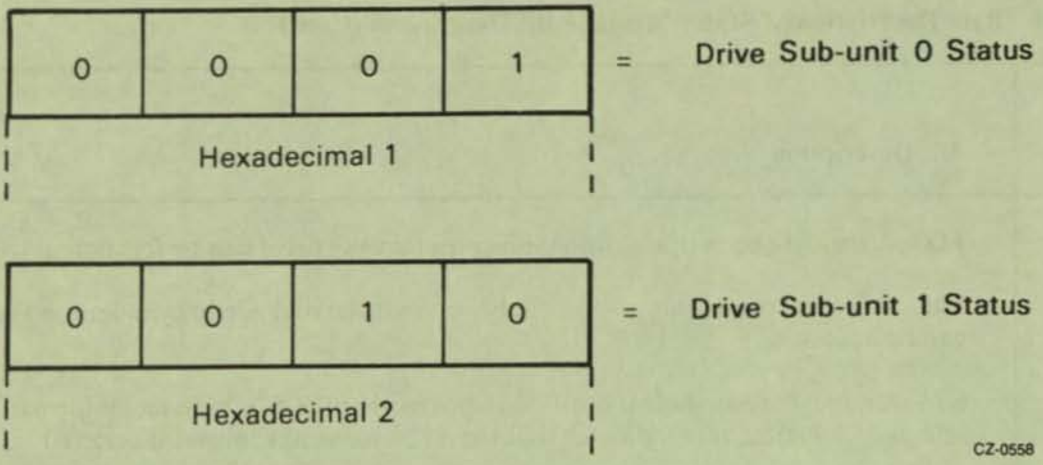


Figure 2-4 Sub-unit Mask Bit Layout Examples

2.3.3.3 Status Message Interpretation – A printout of a drive error was given in paragraph 2.3.3 sample 1. The last line of this error printout gave a status message as follows.

```
STATUS:  00|01    11|00    00|00    0A|00    00|00    06|13    10|20
BYTE:    15|14    13|12    11|10     9| 8     7| 6     5| 4     3| 2
```

Use Figure 2-5 to break down the status message byte code. Then use the following byte descriptions to interpret the above status message.

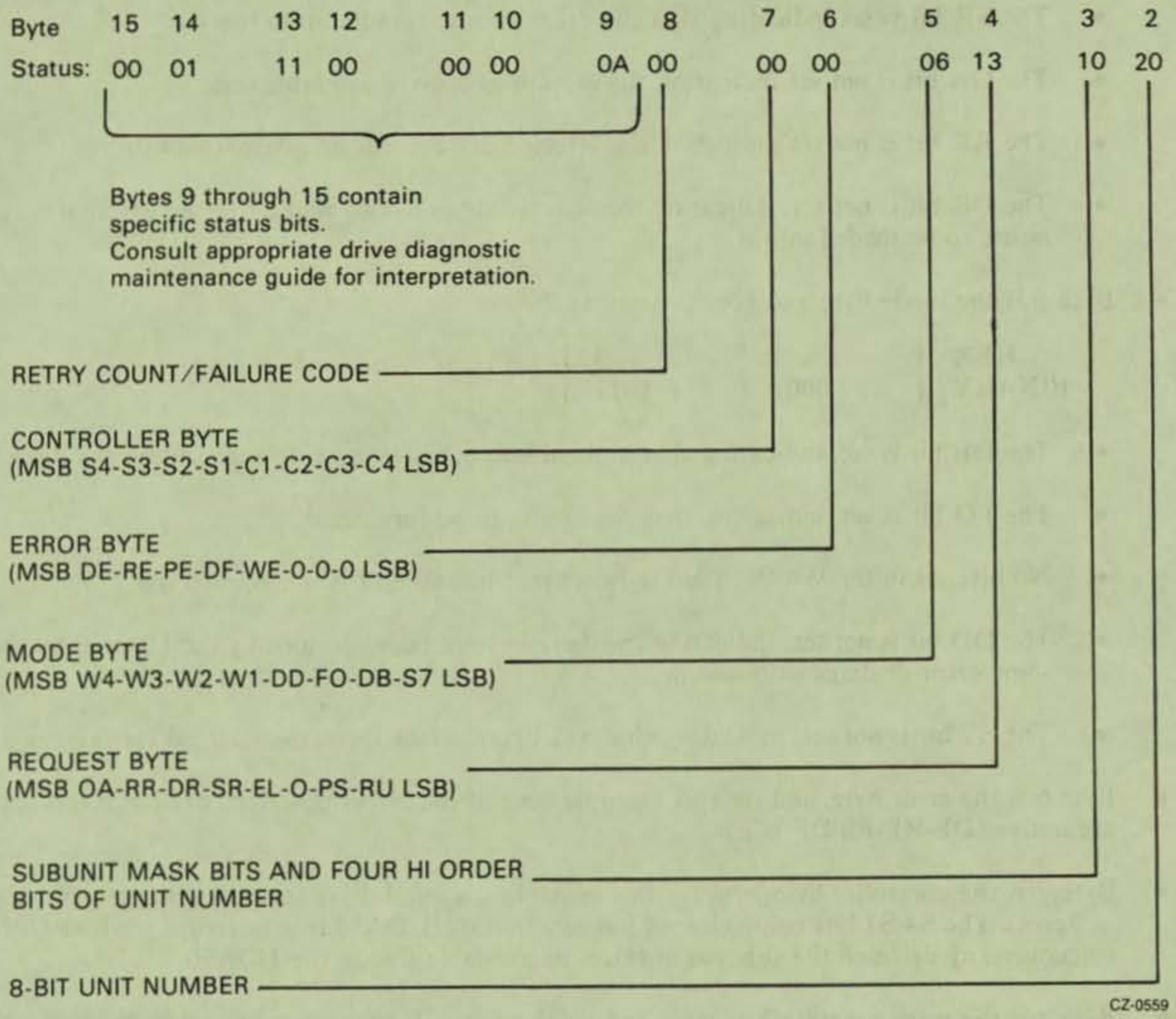

Byte 15 14 13 12 11 10 9 8 7 6 5 4 3 2
Status: 00 01 11 00 00 00 0A 00 00 00 06 13 10 20
Bytes 9 through 15 contain specific status bits. Consult appropriate drive diagnostic maintenance guide for interpretation.
RETRY COUNT/FAILURE CODE
CONTROLLER BYTE (MSB S4-S3-S2-S1-C1-C2-C3-C4 LSB)
ERROR BYTE (MSB DE-RE-PE-DF-WE-0-0-0 LSB)
MODE BYTE (MSB W4-W3-W2-W1-DD-FO-DB-S7 LSB)
REQUEST BYTE (MSB OA-RR-DR-SR-EL-O-PS-RU LSB)
SUBUNIT MASK BITS AND FOUR HI ORDER BITS OF UNIT NUMBER
8-BIT UNIT NUMBER
CZ-0559

Figure 2-5 Status Message Interpretation

- Byte 1 is the get status response code and is not printed out.

- Byte 2 and the lower half of byte 3 comprise a 3 hexadecimal digit unit number. In the example, the unit number is 020 (hexadecimal) or 32 (decimal).

- Byte 3 (upper half) reflects the sub-unit mask and informs us that the drive sending the status is sub-unit 0 (0001).

- Byte 4 is the request byte and breaks down as follows.

```
   HEX |      1 |      3 |
BINARY |   0001 |   0011 |
```

  - The RU bit is set which informs us that the drive has the RUN switch depressed.

  - The PS bit is set which informs us that the port select switch for the UDA requesting the status is depressed. (The drive is available to the UDA50).

  - The SR bit is set indicating that the drive has the spindle up to speed.
  - The OA bit is **not** set indicating the drive is at a drive available state.
  - The RR bit is **not** set indicating the selected drive needs no internal adjustment.
  - The DR bit is **not** set, indicating the selected drive has no request for an external diagnostic to be loaded into it.

- Byte 5 is the mode byte and breaks down as follows.

HEX	1	3
BINARY	0001	0011

  - The DB bit is set, indicating that a diagnostic cylinder is being accessed on the drive.
  - The FO bit is set, indicating that the drive can be formatted.
  - No bits set in the W4-W1 field indicate that no sub-unit is write-protected.
  - The DD bit is **not** set, indicating the drive has not been disabled by the UDA50 due to some error or diagnostic routine.
  - The S7 bit is **not** set, indicating that 512 byte/ sector format is selected for the drive.

- Byte 6 is the error byte, and for this example none of the errors described earlier in this text are active (DE-RE-PE-DF-WE).
- Byte 7 is the controller byte, and for this example a normal drive status is observed (C1-C4 = zeros). The S4-S1 bits being cleared indicate that the UDA50 is to interrupt the host CPU whenever any drive on the subsystem raises its available line to the UDA50.
- Byte 8 is the retry count/failure code and for this example, no retries by the diagnostic were attempted.

UDA50 DISK SUB SYSTEM
SERVICE MANUAL
EK-UDA50-SV-001

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? ______________________________

____

____

____

What features are most useful? ______________________________

____

____

____

What faults or errors have you found in the manual? ______________________________

____

____

____

Does this manual satisfy the need you think it was intended to satisfy? ______________

Does it satisfy *your* needs? ______________ Why? ______________

____

____

____

□ Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name ______________ Street ______________
Title ______________ City ______________
Company ______________ State/Country ______________
Department ______________ Zip ______________

Additional copies of this document are available from:

Digital Equipment Corporation
444 Whitney Street
Northboro, Ma 01532
Attention: Communications Services (NR2/M15)
Customer Services Section

Order No. EK-UDA50-SV-001

Fold Here

Do Not Tear - Fold Here and Staple

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO 33 MAYNARD, MA.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
301 Rockrimmon Boulevard South
Colorado Springs, Colorado 80919

Educational Services
Development and Publishing

Digital Equipment Corporation • Bedford, MA 01730

LiT

EK-UDA50-UG-002

UDA50 USER GUIDE

digital

EK-UDA50-UG-002

UDA50 USER GUIDE

Prepared by Educational Services

Digital Equipment Corporation

1st Edition, December 1981
2nd Edition, October 1982

The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

This document was set on DIGITAL's computerized typesetting system.

- **Class A Computing Devices:**

Notice: This equipment generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference in which case the user at his own expense may be required to take measures to correct the interference.

CONTENTS

Page

FIGURES

Page

FIGURES (Cont)

TABLES

CHAPTER 1
INTRODUCTION

1.1 UDA50 DISK CONTROLLER

The UDA50 is a two-module disk controller that operates on the UNIBUS. It controls up to four Standard Disk Interconnect (SDI) disk drives. Each SDI disk drive is connected to the UDA50 by a separate shielded SDI cable. The basic configuration for the UDA50 Disk Subsystem is illustrated in Figure 1-1.

The operation of the UDA50 is controlled by two resident processors known as the U and the D processors. The U processor controls the interface between the UNIBUS and the UDA50 controller. The D processor controls the interface between the SDI disk drives and the UDA50 Disk Controller.

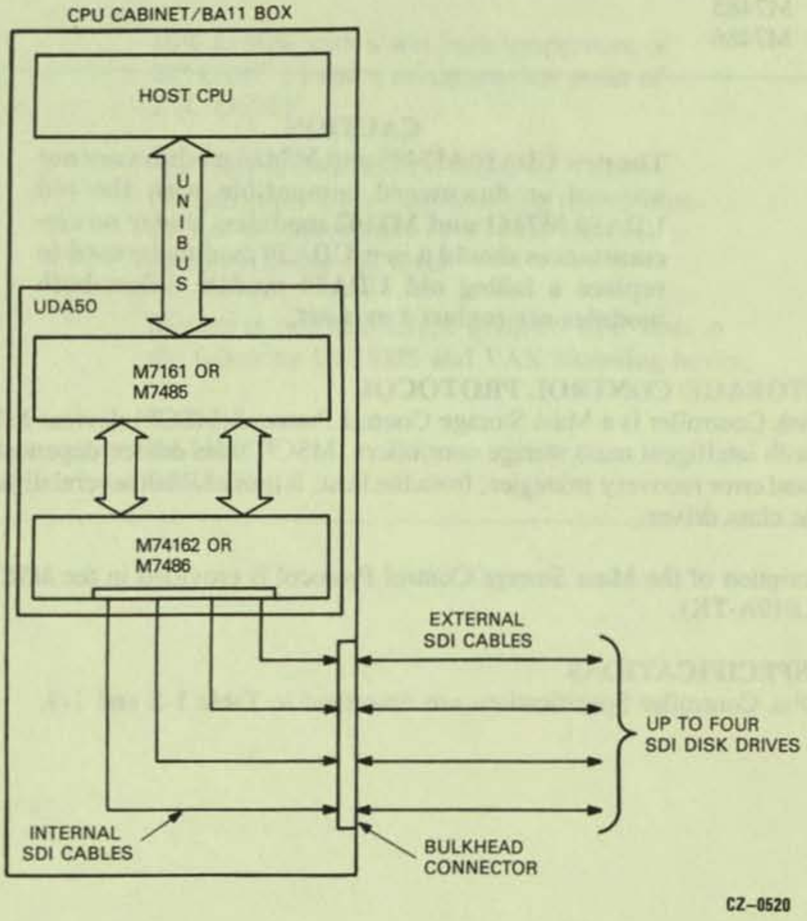


Figure 1-1 UDA50 Disk Subsystem Configuration

1.2 UDA50 UPGRADE

The UDA50 has been upgraded to increase its performance. The difference between the old UDA50 and the upgraded UDA50 will be called out in this manual where applicable.

1.3 UDA50 DIFFERENCES

The upgraded UDA50 consists of two new modules. Refer to Table 1-1. Some of the features which increase the performance of the new UDA50 are:

- Increased RAM size on the M7486 module from 4K to 16K.
- Minor hardware modification to make buffering easier.
- Etched-in capability to use larger PROMS.
- Selectable jumpers for UNIBUS delays, for various systems.

Table 1-1 UDA50 Modules

OLD UDA50 MODULES	NEW UDA50 MODULES
M7161	M7485
M7162	M7486

CAUTION

The new UDA50 M7485 and M7486 modules are not upward or downward compatible with the old UDA50 M7161 and M7162 modules. Under no circumstances should a new UDA50 module be used to replace a failing old UDA50 module unless both modules are replaced as a set.

1.4 MASS STORAGE CONTROL PROTOCOL

The UDA50 Disk Controller is a Mass Storage Control Protocol (MSCP) device. MSCP is a communication protocol used with intelligent mass storage controllers. MSCP hides device-dependent requirements, such as disk geometry and error recovery strategies, from the host. It thus enables several different device drivers to be replaced by one class driver.

A detailed description of the Mass Storage Control Protocol is provided in the *MSCP Basic Disk Functions Manual* (AA-L619A-TK).

1.5 UDA50 SPECIFICATIONS

The UDA50 Disk Controller Specifications are described in Table 1-2 and 1-3.

Table 1-2 Old UDA50 Specifications

Characteristics	Specification
Physical components	UDA module #1 (M7161-YA) UDA module #2 (M7162) 50-pin flat cable assembly 40-pin flat cable assembly SDI cable assembly I/O bulkhead assembly
Power consumption	80 watts nominal
Heat dissipation	Approximately 256 Btu/hour
Electrical voltage and current requirements	11 amps at +5 volts 60 millamps at +15 volts 2 amps at —15 volts
Operating temperature range	10° C to 40° C (5° F to 109° F) with a temperature gradient of 20° C/hour (36° F/hour)
Operating relative humidity range	10% to 90% with a wet bulb temperature of 28° C (82° F) and a minimum dew point of 2° C (36° F)
Operating altitude range	Sea level to 2400 meters (8000 ft). Derate the maximum allowable operating temperature by 1.8° C/1000 meters (1° F/1000 feet) for operation above sea level
Mounting restrictions	Mounts in two hex-height UNIBUS SPC slots in the following UNIBUS and VAX mounting boxes: BA11-A BA11-K BA11-L

Table 1-3 New UDA50 Specifications

Characteristics	Specifications
Physical components	UDA module 1 (M7485) UDA module 2 (M7486) 50-pin flat cable assembly 40-pin flat cable assembly SDI cable assembly I/O bulkhead assembly
Power consumption	83 watts nominal
Heat dissipation	Approximately 256 Btu/hour
Electrical voltage and current requirements	12 amps at +5 volts 60 millamps at +15 volts 1.4 amps at —15 volts
Operating temperature range	10° C to 40° C (50° F to 109° F) with a temperature gradient of 20° C/hour (36° F/hour)
Operating relative humidity range	10% to 90% with a wet bulb temperature of 28° C (82° F) and a minimum dew point of 2° C (36° F)
Operating altitude range	Sea level to 2400 meters (8000 ft). Derate the maximum allowable operating temperature by 1.8° C/1000 meters (1° F/1000 feet) for operation above sea level
Mounting restrictions	Mounts in two hex-height UNIBUS SPC slots in the following UNIBUS and VAX mounting boxes: BA11-A BA11-K BA11-L

1.6 RELATED DOCUMENTATION

The UDA50 Disk Controller related documentation is listed below.

Documentation is available from the Printing and Circulation Services, 444 Whitney St., Northboro, Massachusetts 01532.

- *UDA50 Disk Controller User Guide* (EK-UDA50-UG)
- *UDA50 Disk Controller Service Manual* (EK-UDA50-SV)

Documentation is available from the Software Distribution Center Order Administration/Processing, 20 Forbes Rd., Northboro, Massachusetts 01532.

- *UDA50 Field Maintenance Print Set* (MP-01331)
- *UDA50 Programmer's Documentation Kit* (QP905-GZ) - This kit consists of the following three software manuals.
  - *MSCP Basic Disk Function Manual* (AA-L619A-TK)
  - *Storage System Diagnostic and Utilities Protocol* (AA-L620A-TK)
  - *Storage System UNIBUS Port Description* (AA-L621A-TK)
- *UDA50 Maintenance Documentation Kit* (QP904-GZ) - This kit consists of a small looseleaf binder, the *UDA50 Maintenance Guide* , and the current drive maintenance guides that operate on the UDA50.
  - *UDA50 Maintenance Guide* (AA-M185B-TC) - The above maintenance guide is a 6¾ × 4 inch looseleaf.
  - *RA80 Maintenance Guide* (AA-M186A-TC) - The above maintenance guide is a 6¾ × 4 inch looseleaf.
  - *RA81 Maintenance Guide* (AA-M879A-TC) - The above maintenance guide is a 6¾ × 4 inch looseleaf.
  - *Maintenance Guide Looseleaf Binder* (AV-L980A-TK)

[illegible] RELATED DOCUMENTATION

[illegible]

[illegible]

- [illegible]
- [illegible]

[illegible]

- [illegible] (MP-01231)
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

CHAPTER 2
INSTALLATION

2.1 INTRODUCTION

The installation procedure for the UDA50 Disk Controller requires the insertion of two hex-height modules into a UNIBUS backplane. If the UDA50 Disk Controller is to control disk drives located outside the same cabinet that the controller is in, then an I/O bulkhead assembly must be mounted on the rear of the CPU cabinet. SDI cables must be connected between the UDA50 Disk Controller and the I/O bulkhead assembly and from the I/O bulkhead to each disk drive. Figure 2-1 gives an illustrated parts breakdown of the UDA50 assembly.

Use the following checklist to perform the UDA50 installation. The list indicates the paragraph number where each procedure is described.

- Join the CPU cabinet to the drive cabinet (2.2) ________________
- Check the I/O page address switches and jumpers (2.3.1) ________________
- Check the UDA50 priority plug (2.3.4) ________________
- Insert and clamp the SDI cable to J4 (2.3.5) ________________
- Select the backplane slot (2.3.6) ________________
- Prepare the backplane slot (2.3.70 ________________
- Insert the UDA50 modules (2.3.8) ________________
- Install two flat ribbon cables (2.3.9) ________________
- Install the I/O bulkhead connector (2.3.9.1) ________________
- Install the internal SDI cable to the I/O bulkhead (2.3.9.3) ________________
- Install the external SDI cable to the I/O bulkhead (2.3.9.4) ________________
- Install the bootstrap ROM (2.4) ________________
- Perform the field acceptance test (2.5) ________________

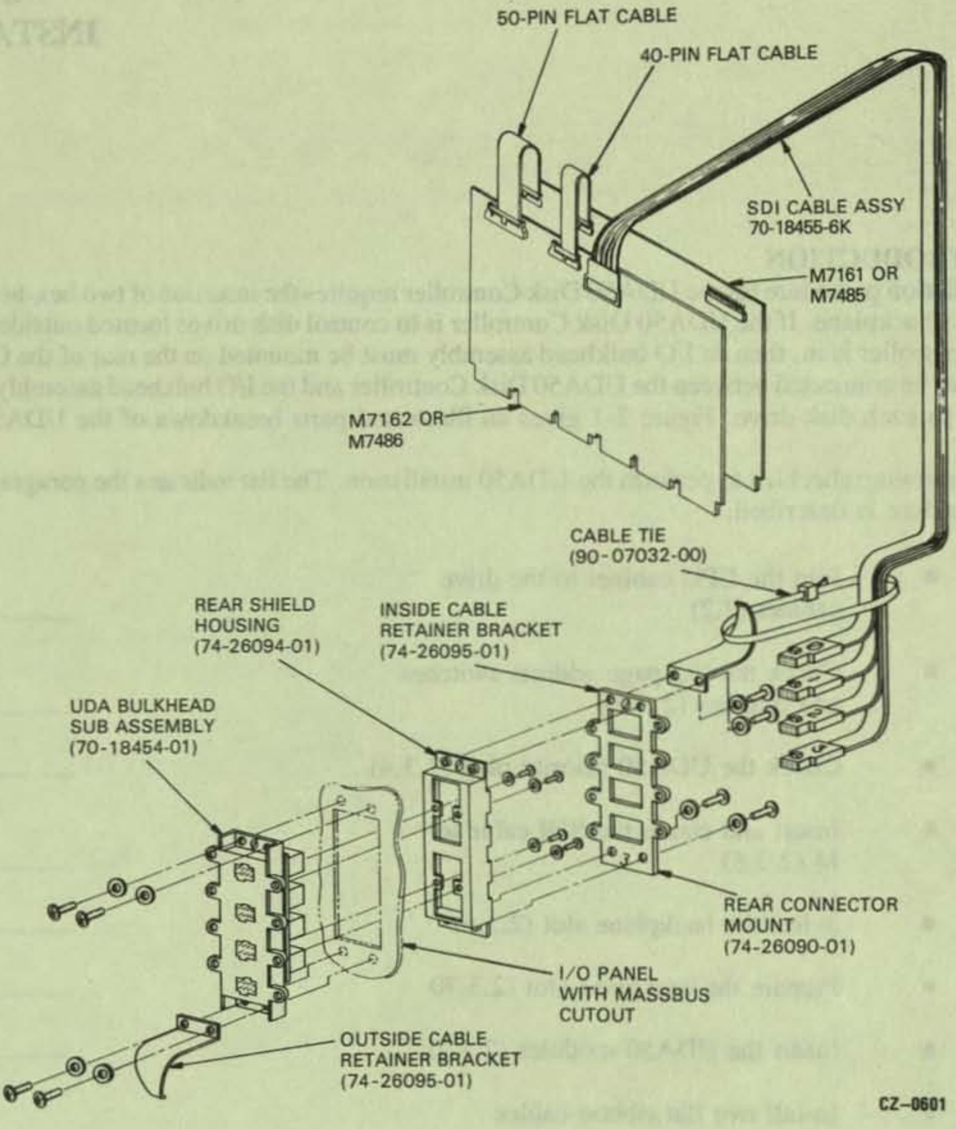


Figure 2-1 UDA50 Illustrated Parts

2.2 CABINET JOINING PROCEDURE

The disk drives that operate with the UDA50 Disk Controller come in an H9642 cabinet with a joiner panel. The joiner panel allows the disk drive cabinet to be joined to the cross-product CPU cabinet in which the UDA50 is located. Figure 2-2 shows how two cross-product cabinets are joined. Use the following procedure to join cross-product cabinets.

1. Open the front door of the CPU cabinet. If the CPU cabinet does not have a front door, remove the lowest front filler panel to expose the end panel lock shown in Figure 2-3.

2. Remove the front left-end panel lock from the CPU cabinet.

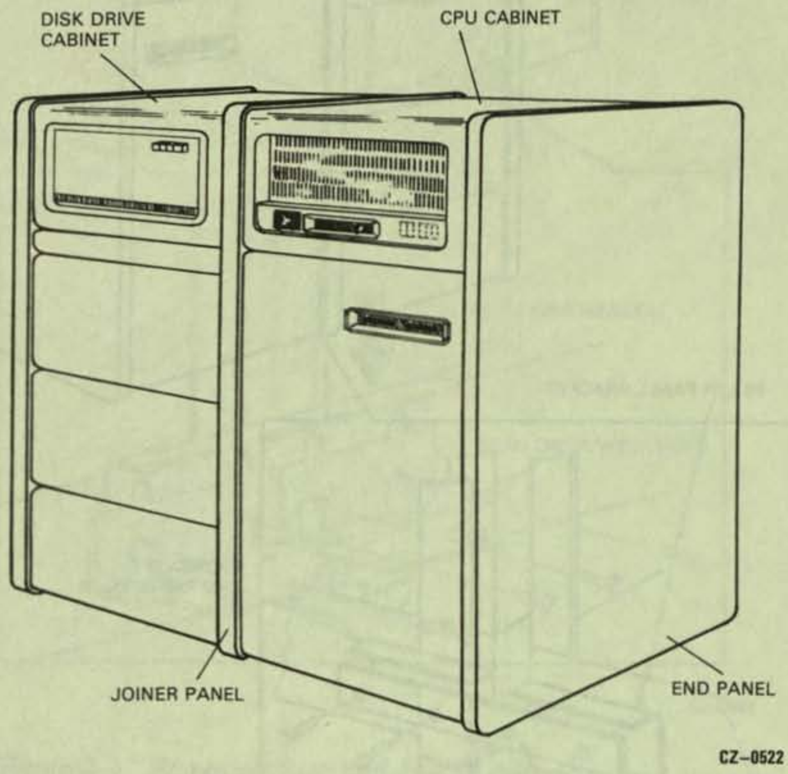


Figure 2-2 Joining Cross Product Cabinets

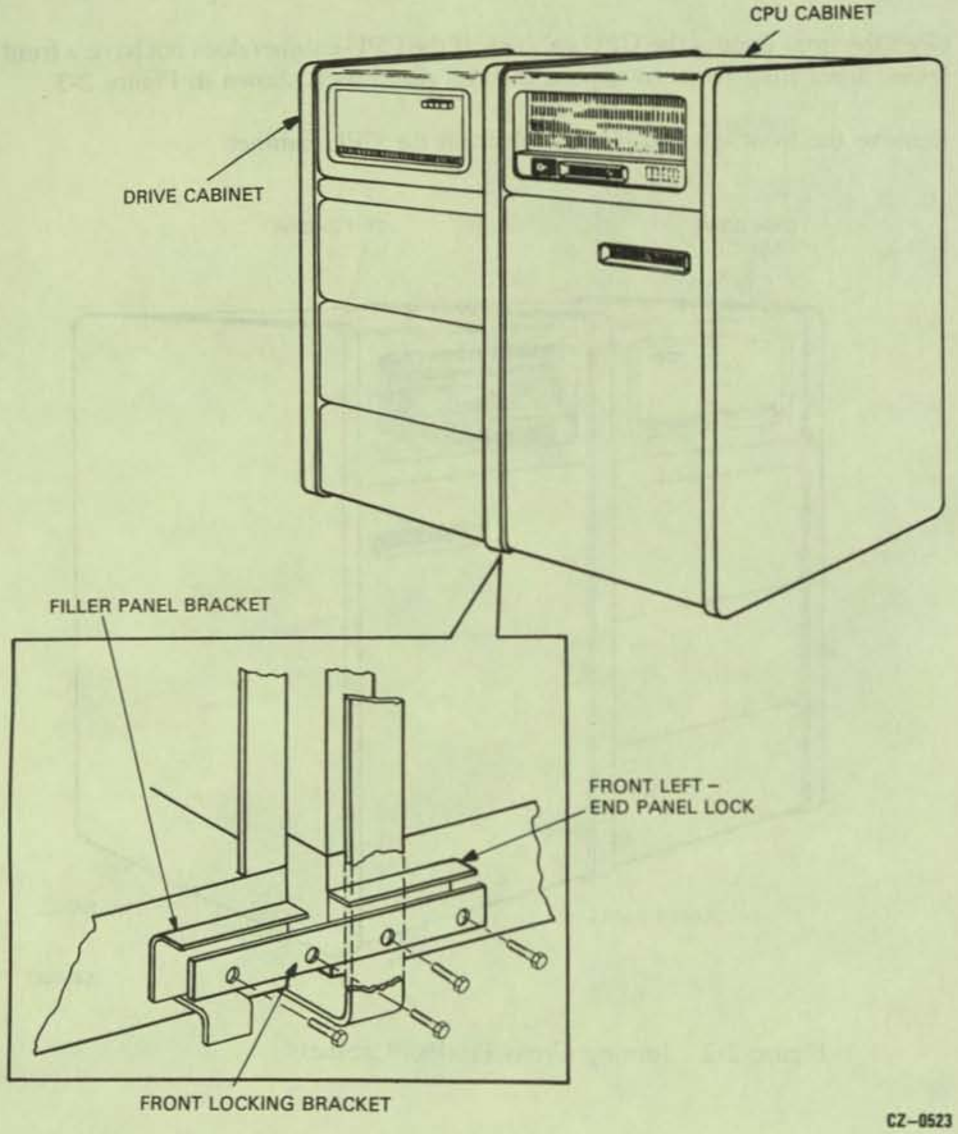


Figure 2-3 Removing Front End Panel Lock

3. Open the rear door of the CPU cabinet.

4. Loosen the screws that secure the right rear-end panel lock. Refer to Figure 2-4.

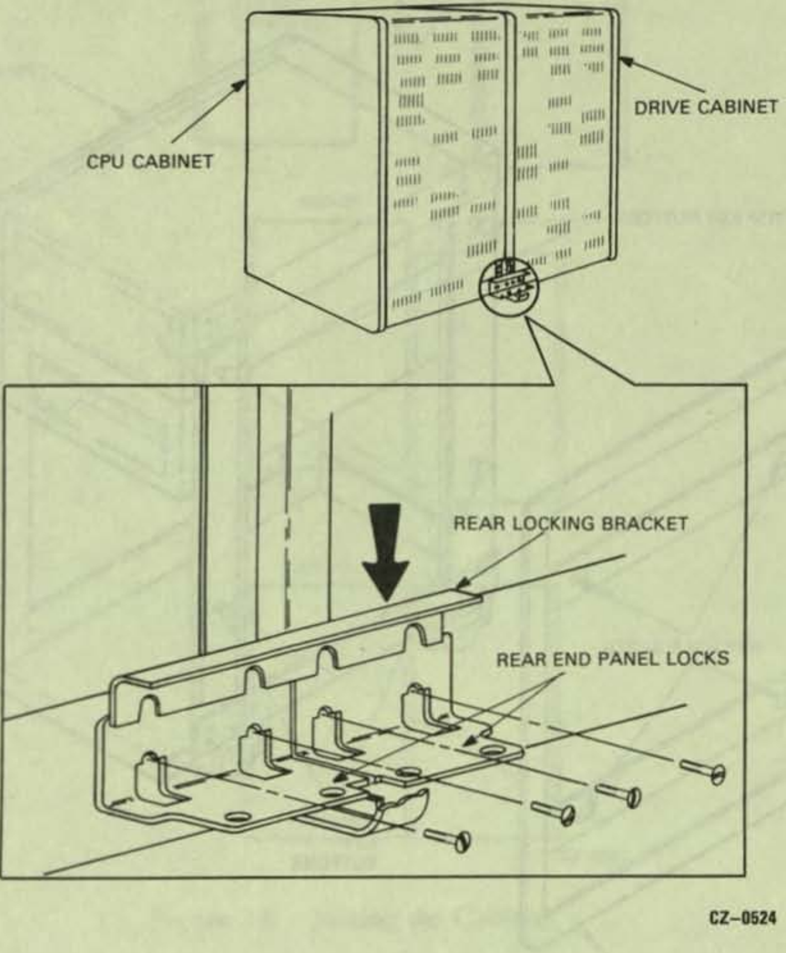


Figure 2-4 Removing Rear End Panel Lock

5. Remove the CPU cabinet end panel and its ground strap, if there is one.

6. Remove the two lower key buttons from the left side of the CPU cabinet uprights. These are removed by unscrewing the Phillips head screws in their center. Refer to Figure 2-5.

7. Slide the two cabinets together and engage the top two key buttons on the CPU cabinet in the keyhole slots on the drive cabinet joiner panel. Adjust the cabinets until their fronts are flush. Refer to Figures 2-5 and 2-6.

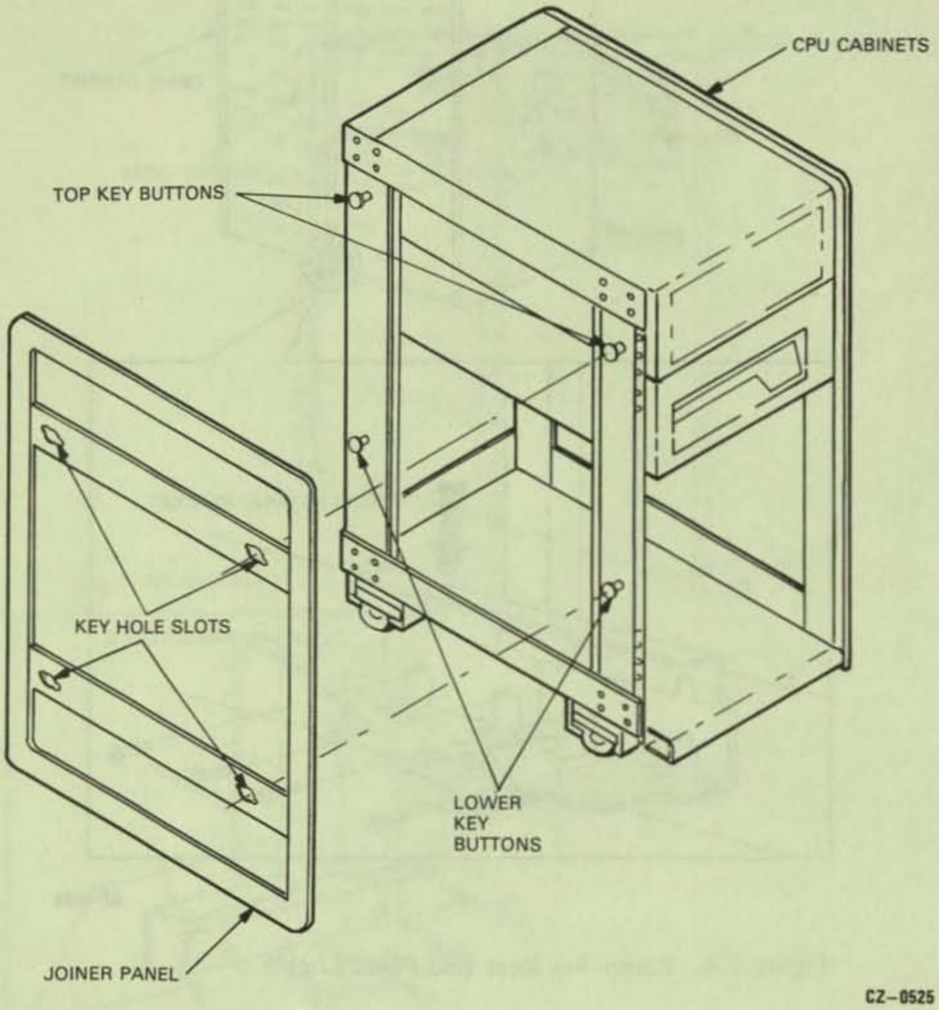


Figure 2-5 Removing Lower Key Buttons

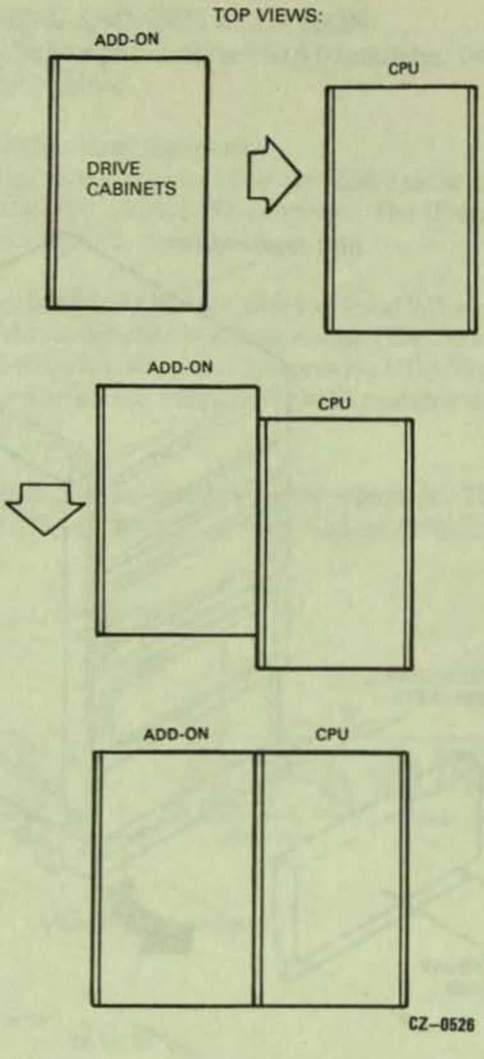


Figure 2-6 Joining the Cabinets

8. Remove the bottom front filler panel from the disk drive cabinet by removing the two screws at its base. Refer to Figure 2-7.

9. Remove the front right filler panel bracket from the disk drive cabinet.

10. Place the front locking bracket over the filler panel bracket and end panel lock as shown in Figure 2-3. Then bolt the two cabinets together with the existing hardware.

11. Open the rear door of the disk drive cabinet and loosen the screws that secure the rear left-end panel lock. Refer to Figure 2-4.

12. Slide the rear locking bracket over the end panel locks as shown in Figure 2-4. Then tighten the four screws.

13. Install the end panel that was removed from the left side of the CPU cabinet onto the left side of the drive cabinet. If an end panel is needed for drives in cross product cabinets, order part number H9544-AA. Be sure to reattach any ground straps to the end panel that might have been removed.

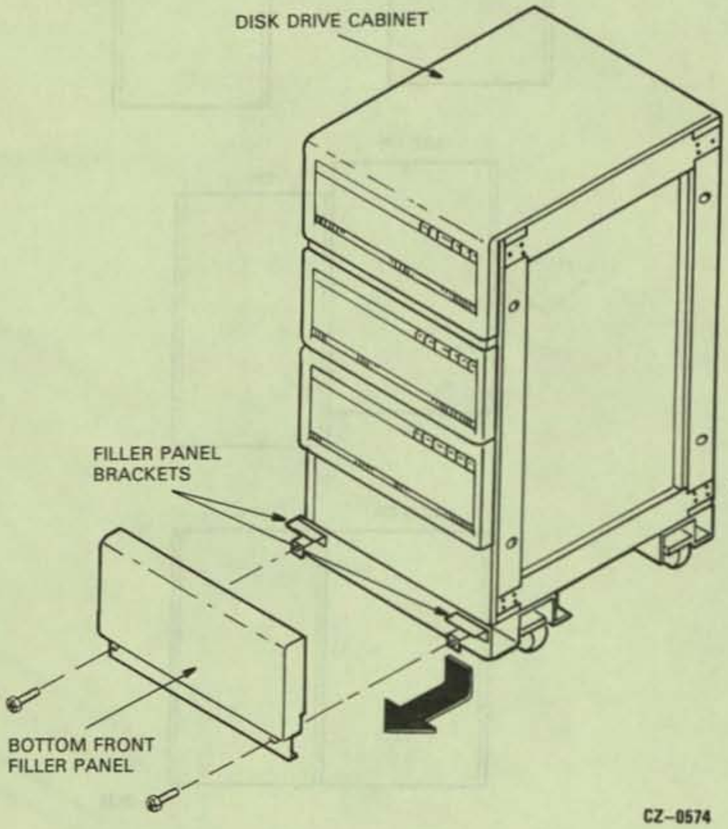


Figure 2-7 Removing the Filler Panel Brackets

2.3 MODULE PREPARATION AND INSTALLATION

The following paragraphs describe how to install the UDA50 modules, I/O bulkhead, and cables once the CPU and disk drive cabinets have been joined.

2.3.1 I/O Page Address Switches and Jumpers

The UDA50 Disk Controller contains two registers that are visible to the I/O page. They are the initializing and polling (IP) register and the status and address (SA) register. The IP and SA registers are assigned an octal UNIBUS address of 772150 and 772152, respectively.

The UNIBUS address selector switches and jumper plugs W4 and W5 are used to set the UNIBUS address for the IP register. The location of these switches and jumpers on UDA50 module (M7161) is shown in Figures 2-8A and 2-8B. The location of these switches and jumpers on UDA50 module (M7485) is shown in Figure 2-8A. Set the UNIBUS address switches and jumpers for both modules to the positions shown in Figure 2-9 to select UNIBUS address 772150.

In past disk products, a vector address was also physically selectable. This is not true with the UDA50 Disk Controller. A vector address of 154 (octal) will be supplied by the software.

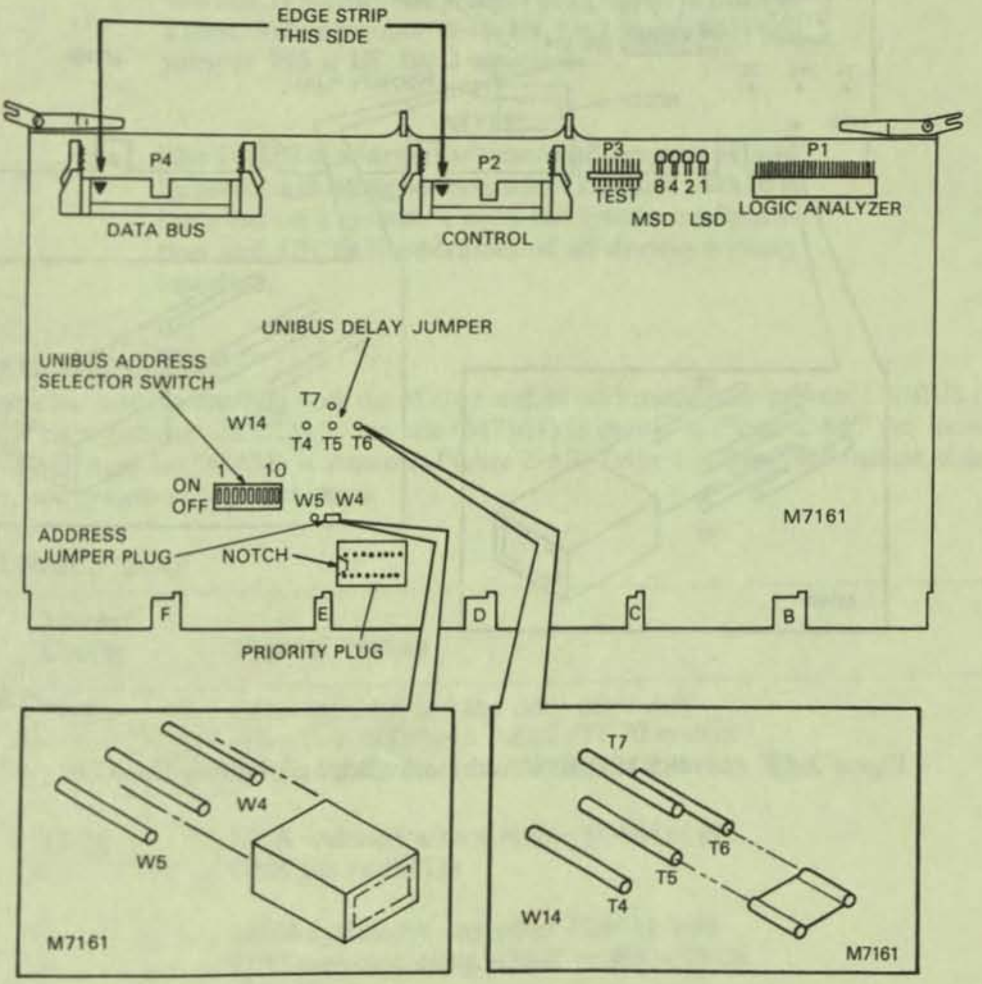


Figure 2-8A M7161 UNIBUS Address Switch and Jumper Locations

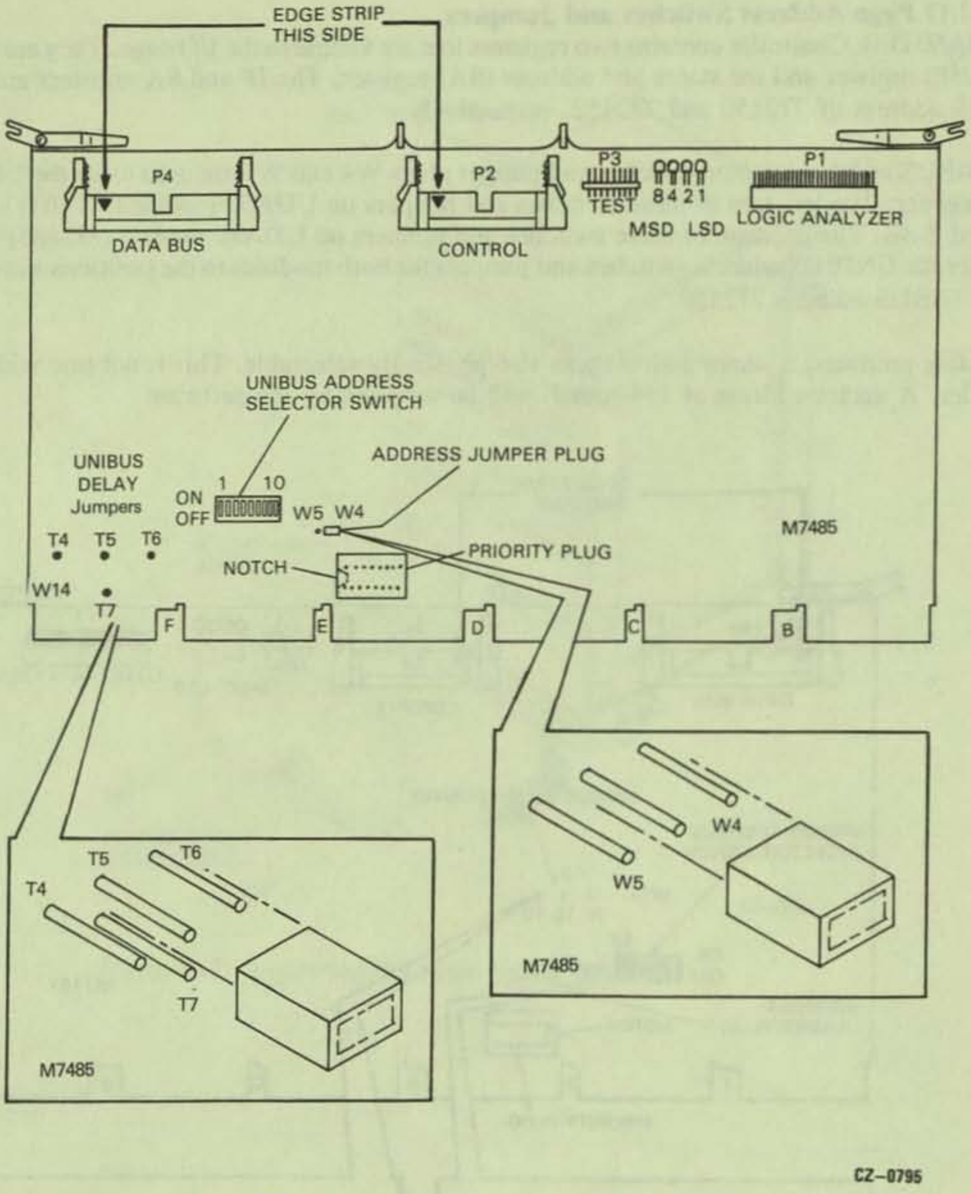
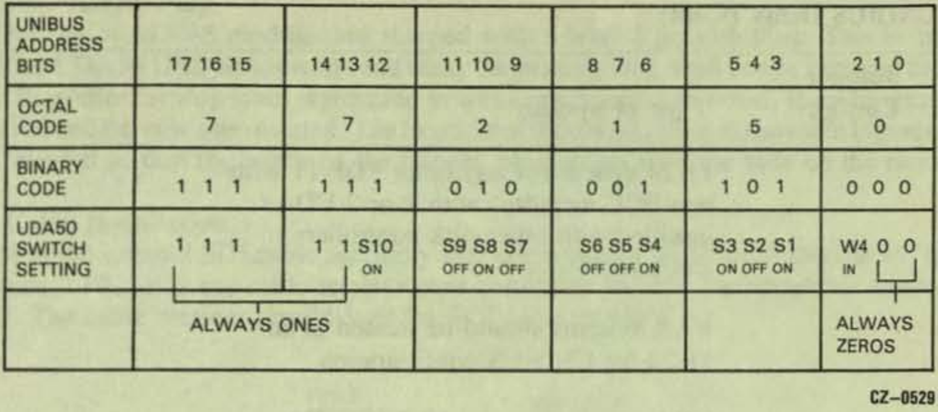


Figure 2-8B M7485 UNIBUS Address Switch and Jumper Locations



UNIBUS ADDRESS BITS	17 16 15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
OCTAL CODE	7	7	2	1	5	0
BINARY CODE	1 1 1	1 1 1	0 1 0	0 0 1	1 0 1	0 0 0
UDA50 SWITCH SETTING	1 1 1	1 1 S10 ON	S9 S8 S7 OFF ON OFF	S6 S5 S4 OFF OFF ON	S3 S2 S1 ON OFF ON	W4 0 0 IN
	ALWAYS ONES					ALWAYS ZEROS

Figure 2-9 UDA50 Switch Setting for Address 772150

NOTE

UNIBUS address bit 2 is selected by jumper plugs W4 and W5. Only one jumper plug can be in place at a time. When jumper W4 is IN, bit 2 equals 0. When jumper W5 is IN, bit 2 equals 1.

NOTE

The UNIBUS address switches and jumpers should be set for a floating address when a second UDA50 is installed on a system. Check the system configuration and UNIBUS addresses of all devices already installed.

2.3.2 UNIBUS Overloading

A set of jumpers has been inserted on both the M7161 and M7485 modules to prevent UNIBUS overloading. The location of these jumpers on UDA50 module (M7161) is shown in Figure 2-8A. The location of these jumpers on UDA50 module (M7485) is shown in Figure 2-8B. Table 2-1 shows the amount of delay, jumper configuration, and system configuration.

Table 2-1 UNIBUS Delay

Amount of Delay	Jumper Config.	Type of System
0 usec	T4-T5	UDA installed and the only other disk drive is a RL02 or a RK07 (11/70 system with a RK07 and 1mb DMR will not work*)
6.2 usec	T5-T6	UDA installed with multiple DMR11s or DMC11s or DZ11s 11/44 system (or any other PDP-11 with ECC memory) using RM02 or RP04/05/06 disk drives 11/44 with RL02 or RK07 disk drives

Table 2-1 UNIBUS Delay (Cont)

Amount of Delay	Jumper Config.	Type of System
		11/24 system (or any other PDP-11 with non-ECC memory) with 1 or 2 UDAs installed with other disk controllers and a DZ11
		VAX systems should be treated as an 11/24 for UNIBUS configuration
		UDA installed on the UNIBUS with one or more real time data acquistion devices, and real time data overrun or underflow is observed †
10 usec	T5-T7	11/44 system with RL02 and RK07 disk drives
		11/70 system with a UDA/RL02/DMR11 (1mb) mix

* The UDA/RK07/DMR11 configuration gives data late errors from the RK07 regardless of the UDAs jumper setting. Because of this, either an RK07 or a UDA, but not both can be configured on the 11/70 when a 1mb DMR11 is present.

† If underflow or overrun conditions are observed after setting the UDAs jumper to the 6.2 usec. position, the UDA's jumper must be set to the 10 usec. position (T5-T7).

2.3.3 UNIBUS Overload Exceptions

There are exceptions to using the UNIBUS delay in preventing overload and the number of UDAs that can be installed on a system. They are:

1. The UDA should not be installed on a UNIBUS system which has a bus repeater because the repeater slows the UNIBUS. Other devices such as RK07, RM02, and RP04/05/06 may also experience data late conditions.
2. The UDA must be installed after all non-buffered devices on the UNIBUS.
3. On PDP-11 systems, there may be no more than two UDAs installed on a UNIBUS. However on VAX systems, no more than one UDA should be installed on a UNIBUS with non-buffered UNIBUS peripheral devices.

NOTE

The old UDA50 M7161 module has the UNIBUS delay jumpers installed starting with M7161 module revision E. Check this module and its delay jumpers if you are having UNIBUS overload problems.

2.3.4 UDA50 Priority Plug

All UDA50 M7161 or M7485 modules are shipped with a level 5 priority plug. This is the recommended priority level for UDA50 Disk Subsystems and thus, the priority plug need not be changed for the majority of installations. If another priority level is required in some special circumstance, then the current priority plug must be removed and the new one inserted. The location of the priority plug is shown in Figures 2-8A and 2-8B. It should be inserted so that the notch on the priority plug aligns with the hole on the module socket.

2.3.5 SDI Cable Installation

Insert plug P4 of the internal SDI cable assembly into connector J4 on UDA50 module M7162 or M7486 as shown in Figure 2-10. Slide the cable retainer over connector J4 until the connector protrudes through the plastic cutout. The cable retainer should lock the SDI cable in place.

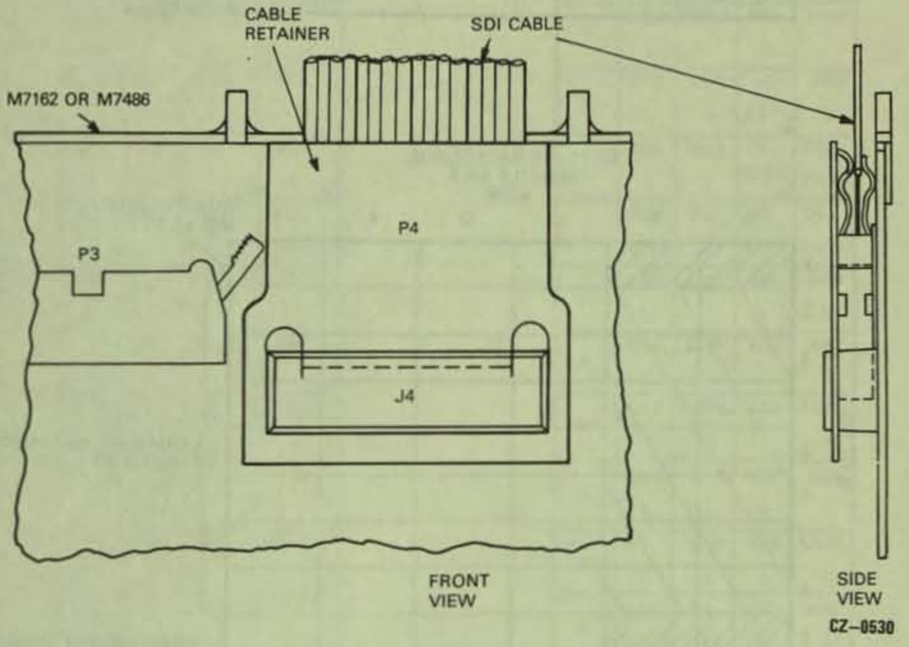


Figure 2-10 M7162 or M7486 SDI Cable Assembly Installation

2.3.6 UNIBUS Backplane SPC Slots

The UDA50 Disk Controller will operate in either the standard UNIBUS or modified UNIBUS hex-height small peripheral connector (SPC) slots. Locate two adjacent empty SPC slots for the two UDA50 modules. Illustrations of the UNIBUS backplanes and pin assignments are shown in Figures 2-11 and 2-12, respectively.

The early SPCs did not utilize direct memory access (DMA) data transfers to and from memory; therefore, the signals now used for this purpose are not part of the original SPC pin assignments. Newer options, such as the UDA50 Disk Controller, do utilize DMA transfers. There is a new pin assignment called SPC PRIME that includes these signals. Refer to Figure 2-13. If the UDA50 Disk Controller is used in an older (non-SPC PRIME) slot, it is then necessary to make sure that the signals shown in Table 2-2 are wired on the backplane.

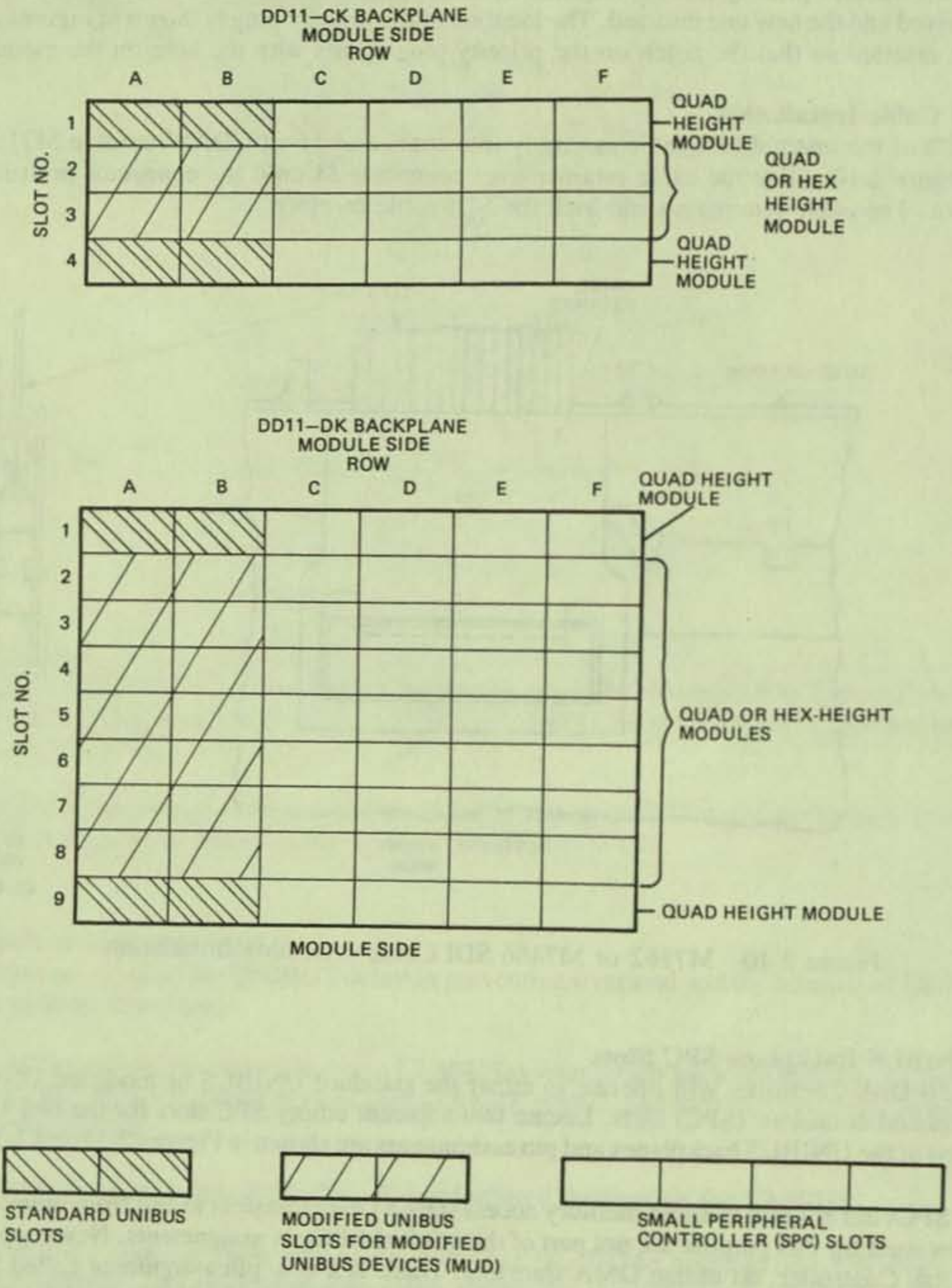


Figure 2-11 Optional Backplane Slot Assignments

STANDARD UNIBUS PIN DESIGNATIONS

	ROW A		ROW B	
Side Pin	1	2	1	2
A	INIT L	+5V	BG6 H ◢	+5V
B	INTR L	GND ◢	BG5 H ◢	GND ◢
C	D00 L	GND	BR5 L	GND
D	D02 L	D01 L	GND ◢	BR4 L
E	D04 L	D03 L	GND ◢	BG4 H ◢
F	D06 L	D05 L	AC LO L	DC LO L
H	D08 L	D07 L	A01 L	A00 L
J	D10 L	D09 L	A03 L	A02 L
K	D12 L	D11 L	A05 L	A04 L
L	D14 L	D13 L	A07 L	A06 L
M	PA L	D15 L	A09 L	A08 L
N	GND ◢	PB L	A11 L	A10 L
P	GND ◢	BBSY L	A13 L	A12 L
R	GND ◢	SACK L	A15 L	A14 L
S	GND ◢	NPR L	A17 L	A16 L
T	GND	BR7 L	GND	C1 L
U	NPG H ◢	BR6 L	SSYN L	C0 L
V	BG7 SO ◢	GND ◢	MSYN L	GND ◢

MODIFIED UNIBUS PIN DESIGNATIONS

	ROW A		ROW B	
SIDE PIN	1	2	1	2
A	INIT L	+5V	RESV PIN ◢	+5V
B	INTR L	TP ◢	RESV PIN ◢	TP ◢
C	D00 L	GND	BR5 L	GND
D	D02 L	D01 L	+5 BAT ◢	BR4 L
E	D04 L	D03 L	INT SSYN ◢	PAR DET ◢
F	D06 L	D05 L	AC LO L	DC LO L
H	D08 L	D07 L	A01 L	A00 L
J	D10 L	D09 L	A03 L	A02 L
K	D12 L	D11 L	A05 L	A04 L
L	D14 L	D13 L	A07 L	A06 L
M	PA L	D15 L	A09 L	A08 L
N	PAR P1 ◢	PB L	A11 L	A10 L
P	PAR P0 ◢	BBSY L	A13 L	A12 L
R	+15 BAT ◢	SACK L	A15 L	A14 L
S	-15 BAT ◢	NPR L	A17 L	A16 L
T	GND	BR7 L	GND	C1 L
U	+20 (CORE) ◢	BR6 L	SSYN L	C0 L
V	+20 (CORE) ◢	+20 (CORE) ◢	MSYN L	-5 (CORE) ◢

NOTE: ◢ INDICATES A REDESIGNATED PIN.

CZ-0532

Figure 2-12 Standard and Modified Backplane Pin Assignments

	ROW C		ROW D		ROW E		ROW F	
SIDE PIN	1	2	1	2	1	2	1	2
A	NPG (IN)	+5V	TP	+5V	GND A	+5V	ABG OUT	+5V
B	NPG (OUT)	-15V	TP	-15V	ASSYN IN H	-15V	ABG IN	-15V
C	PA L	GND	A SEL 6	GND	A12 L	GND	SSYN L	GND
D	LTC	D15 L	A OUT LOW	BR7 L	A17 L	A15 L	BBSY L	FO1 N1
E	TP	D14 L	A SEL 4	BR6 L	MSYN L	A16 L	FO1 V2	D02 L
F	TP	D13 L	A SEL 0	BR5 L	A02 L	C1 L	D05 L	D06 L
H	D11 L	D12 L	A IN	BR4 L	A01 L	A00 L	D07 L	A INT ENB B
J	A INT B	D10 L	A SEL 2	A BR OUT	SSYN L	C0 L	NPR L	GND A
K	TP	D09 L	A OUT	BG7 SO	A14 L	A13 L	D08 L	A INT B
L	A INT ENBB	D08 L	INIT L	BG7 OUT	A11 L	TP	D03 L	FO1 L2
M	TP	D07 L	AINT ENBA	BG6 SO	AIN	AOUT HIGH	INTR L	FO1 M2
N	DC LO	D04 L	A INT A	BG6 OUT	A OUT LOW	A08 L	FO1 N1	D04 L
P	HALT REQ	D05 L	TP	BG5 S0	A10 L	A07 L	ABR OUT	FO1 P2
R	HALT GRT	D01 L	TP	BG5 OUT	A09 L	ASEL 4	FO1 L2	FO1 N1
S	PB L	D00 L	TP	BG4 S0	ASEL 6	ASEL 0	FO1 M2	FO1 P2
T	GND	D03 L	GND	BG4 OUT	GND	ASEL 2	GND	SACK L
U	+15	D02 L	TP	ABG IN	A06 L	A04 L	A INT A	ABR OUT
V	AC LO	D06 L	ASSYN IN H	ABG OUT	A05 L	A03 L	A INT ENB A	FO1 FO1

CZ-0533

Figure 2-13 SPC PRIME Backplane Pin Assignments

Table 2-2 Backplane Signal Checks

Backplane Pins	Signal Names	Used On
Pin CA1	NPG IN	M7161 or M7485
Pin CB1	NPG OUT	M7161 or M7485
Pin FJ1	NPR	M7161 or M7485
Pin CV1	AC LO1	M7161 or M7485
Pin CU1	+15V	M7162 or M7486

2.3.7 UNIBUS Backplane Slot Preparation

If the slot has SPC PRIME pinning, the NPG jumper will have to be removed. The NPG line is the UNIBUS grant line for devices that perform data transfers without processor intervention. Continuity of the NPG line is provided by wirewrap jumpers on the backplane. When an NPR device is placed in a slot, the corresponding jumper wire from pin CA1 to pin CB1 of that slot must be removed. The routing of the NPG signal through the backplane is shown in Figure 2-14. Grant priority decreases from slot 1 to slot 9 (slot 1 has the highest priority).

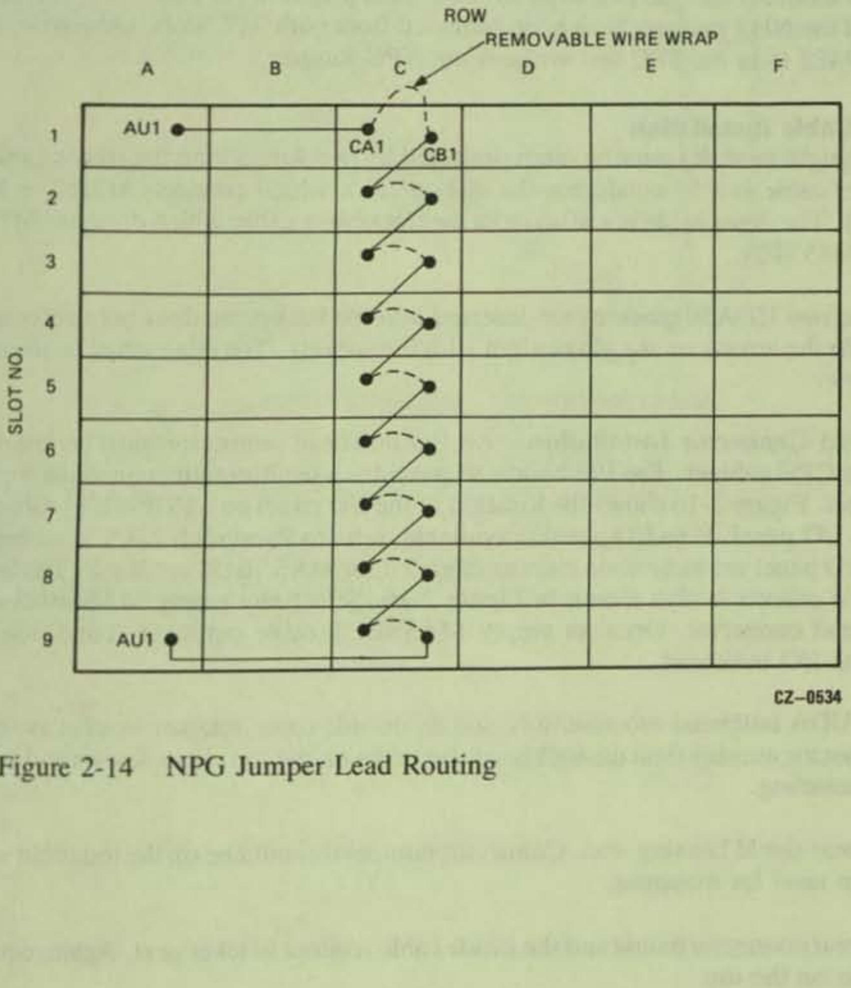


Figure 2-14 NPG Jumper Lead Routing

The NPG jumper must be removed from the slot that will be occupied by UDA50 module M7161 or M7485. Module M7161 or M7485 will not operate with the NPG jumper in place. The NPG jumper may also be removed from the slot occupied by the second UDA50 Module, M7162 or M7486, since the module will provide continuity on the NPG line. Module M7162 or M7486 will operate whether or not the NPG jumper is in place. If both NPG jumpers are removed from these two module slots, the module location can be interchanged and the UDA50 will still operate. However, remember that the NPG continuity line will be interrupted whenever Module M7162 or M7486 is removed.

NOTE

If an NPR device is removed from a slot, the jumper wire from pin CA1 to pin CB1 must be reconnected.

The bus grant lines (BG4 through BG7) for devices requiring processor intervention during data transfers are routed through each small peripheral control section in slot D. Each of the four grant signals is routed on a separate line. Grant priority for each level decreases from slot 1 to slot 9.

A bus grant jumper card (G727, G7270, or G7271) must be placed in connector D of any unoccupied SPC section. If an SPC section is left open, bus grant continuity will be lost.

2.3.8 UDA50 Module Insertion

Insert the two UDA50 modules into the two adjacent SPC slots prepared for them. The two modules may be inserted in any order if the NPG jumpers have been removed from both SPC slots. Otherwise, make sure that module M7161 or M7485 is in the SPC slot without the NPG jumper.

2.3.9 Flat Ribbon Cable Installation

The two UDA50 hex-height modules must be interconnected by two 4 inch long flat ribbon cables as shown in Figure 2-15. The outer cable is a 50-conductor flat ribbon cable which connects M7162 or M7486 (P1) to M7161 or M7485 (P4). The inner cable is a 40-conductor flat ribbon cable which connects M7162 or M7486 (P3) to M7161 or M7485 (P2).

The order in which the two UDA50 modules are inserted into the backplane does not matter to these cables. Install the cables so that the arrows on the plugs align with the sockets. The edge stripe on the cables is on the same edge as the arrow.

2.3.9.1 I/O Bulkhead Connector Installation – An I/O bulkhead connector must be installed on the I/O panel at the rear of the CPU cabinet. The I/O bulkhead provides a feedthrough connection for all SDI cables leaving the CPU cabinet. Figure 2-16 shows the location of the I/O panel on a PDP-11/44 cabinet. Other CPU cabinets use this same I/O panel. If no I/O panel is available, refer to Paragraph 2.3.9.5 (Alternate SDI Cable Installation). On the I/O panel are three wide cutouts intended for MASSBUS cable use. The location of these three MASSBUS cable cutouts is also shown in Figure 2-16. Select any empty MASSBUS cable cutout to mount the I/O bulkhead connector. Once an empty MASSBUS cable cutout is found, use the following procedure to install the I/O bulkhead.

1. Install the UDA bulkhead sub-assembly and its outside cable retainer bracket as shown in Figure 2-17. Connector number 0 on the bulkhead should be on the top. Four screws and lock washers are used for mounting.

2. Install the rear shield housing next. Connector number 0 should be on the top. Four screws and lock washers are used for mounting.

3. Install the rear connector mount and the inside cable retainer bracket next. Again, connector number 0 should be on the top.

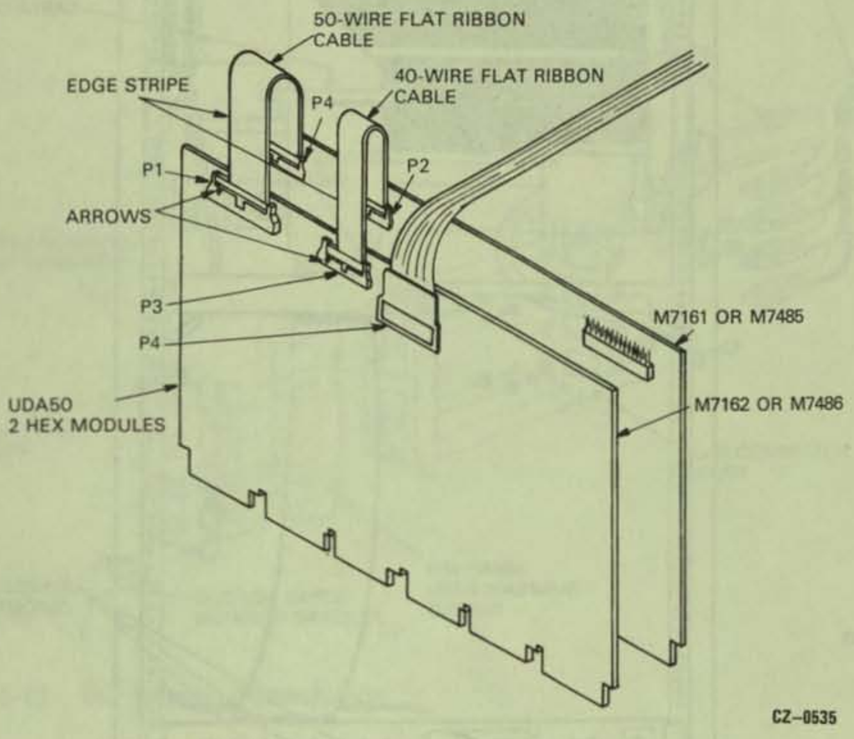


Figure 2-15 UDA50 Intermodule Flat Ribbon Cables

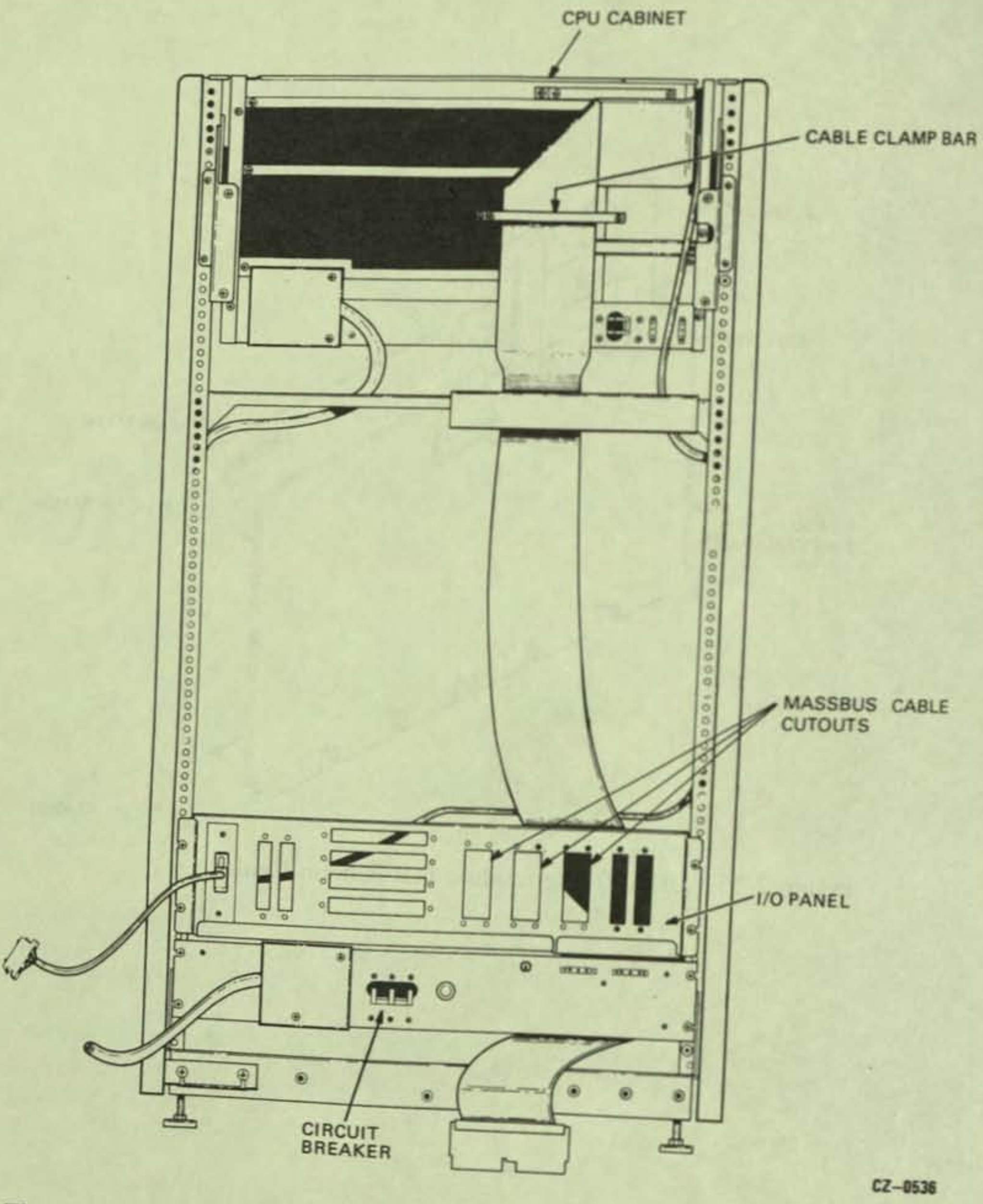


Figure 2-16 I/O Panel and MASSBUS Cable Slot Locations

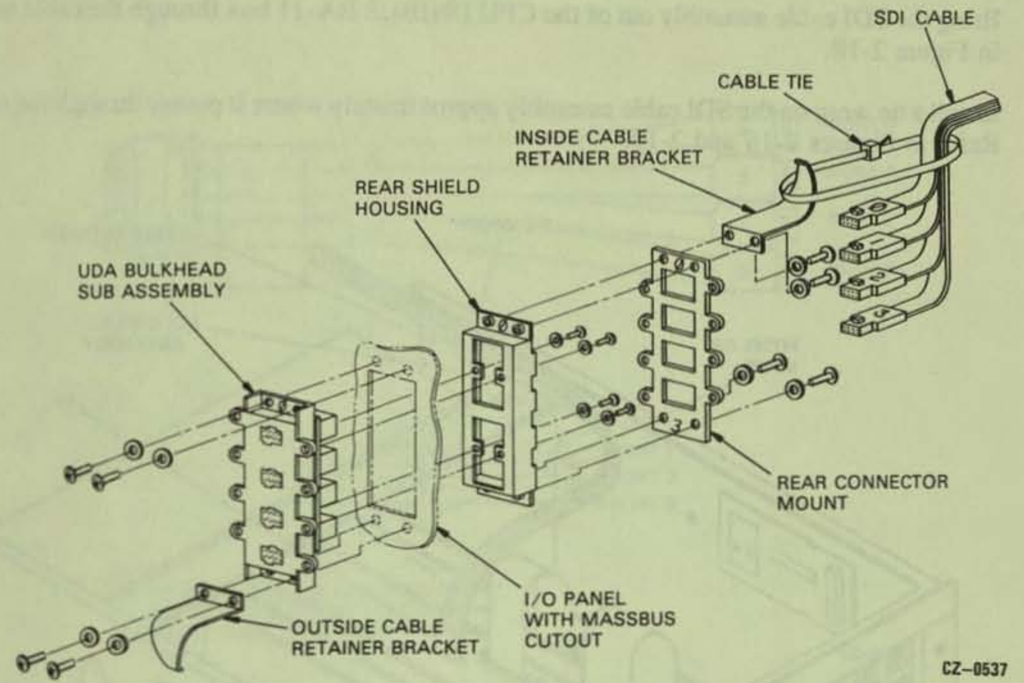

SDI CABLE
CABLE TIE
INSIDE CABLE RETAINER BRACKET
REAR SHIELD HOUSING
UDA BULKHEAD SUB ASSEMBLY
REAR CONNECTOR MOUNT
I/O PANEL WITH MASSBUS CUTOUT
OUTSIDE CABLE RETAINER BRACKET
CZ-0537

Figure 2-17 I/O Bulkhead Installation

2.3.9.2 SDI Cabling Procedures – Standard Disk Interconnect (SDI) cables must be installed both inside and outside the CPU cabinet. The internal SDI cabling procedure will be described first.

2.3.9.3 Internal SDI Cables – One end of the internal SDI cable is already connected to UDA50 Module M7162 and M7486. This is described in paragraph 2.3.5. Now the other end of the SDI cable assembly must be plugged into the I/O bulkhead on the I/O panel at the rear of the CPU cabinet. Use the following procedure to install this cable.

1. Bring the SDI cable assembly out of the CPU UNIBUS BA-11 box through the cable trough shown in Figure 2-18.

2. Install a tie wrap on the SDI cable assembly approximately where it passes through the cable trough. Refer to Figures 2-18 and 2-19.

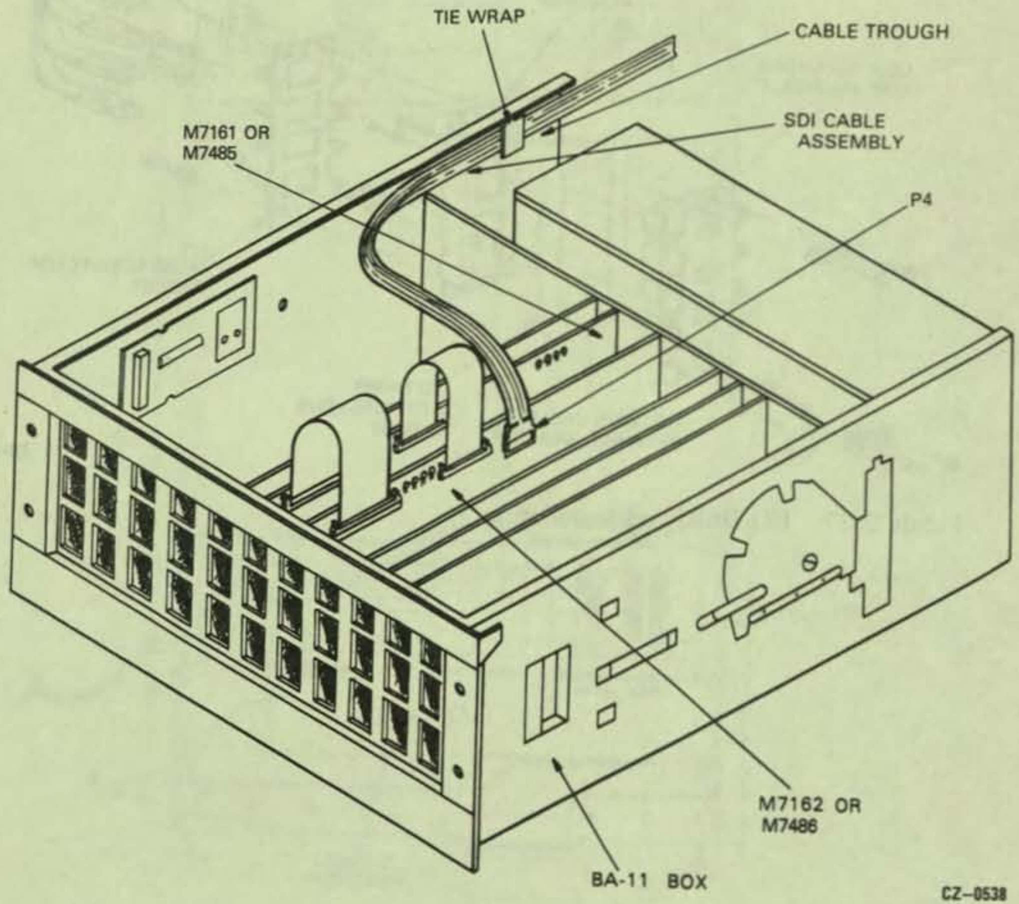


Figure 2-18 SDI Cable Routing Inside BA-11 Box

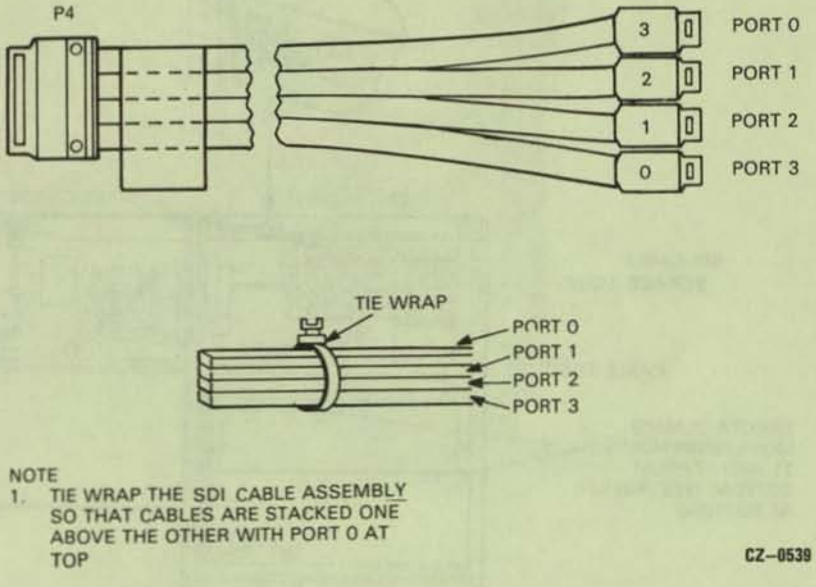

P4
3
2
1
0
PORT 0
PORT 1
PORT 2
PORT 3
TIE WRAP
PORT 0
PORT 1
PORT 2
PORT 3
NOTE
1. TIE WRAP THE SDI CABLE ASSEMBLY SO THAT CABLES ARE STACKED ONE ABOVE THE OTHER WITH PORT 0 AT TOP
CZ–0539

Figure 2-19 SDI Cable Assembly with Tie Wrap

3. Tie wrap the SDI cable at point A where it exits the rear of the BA-11 box. Refer to Figure 2-20.

4. Install the two Dakota clamps as shown in Figure 2-20 and insert the SDI cable assembly in them.

5. Install the remaining seven cable ties on the SDI cable assembly as shown in Figure 2-20. The seventh cable tie is hidden behind the bottom I/O panel.

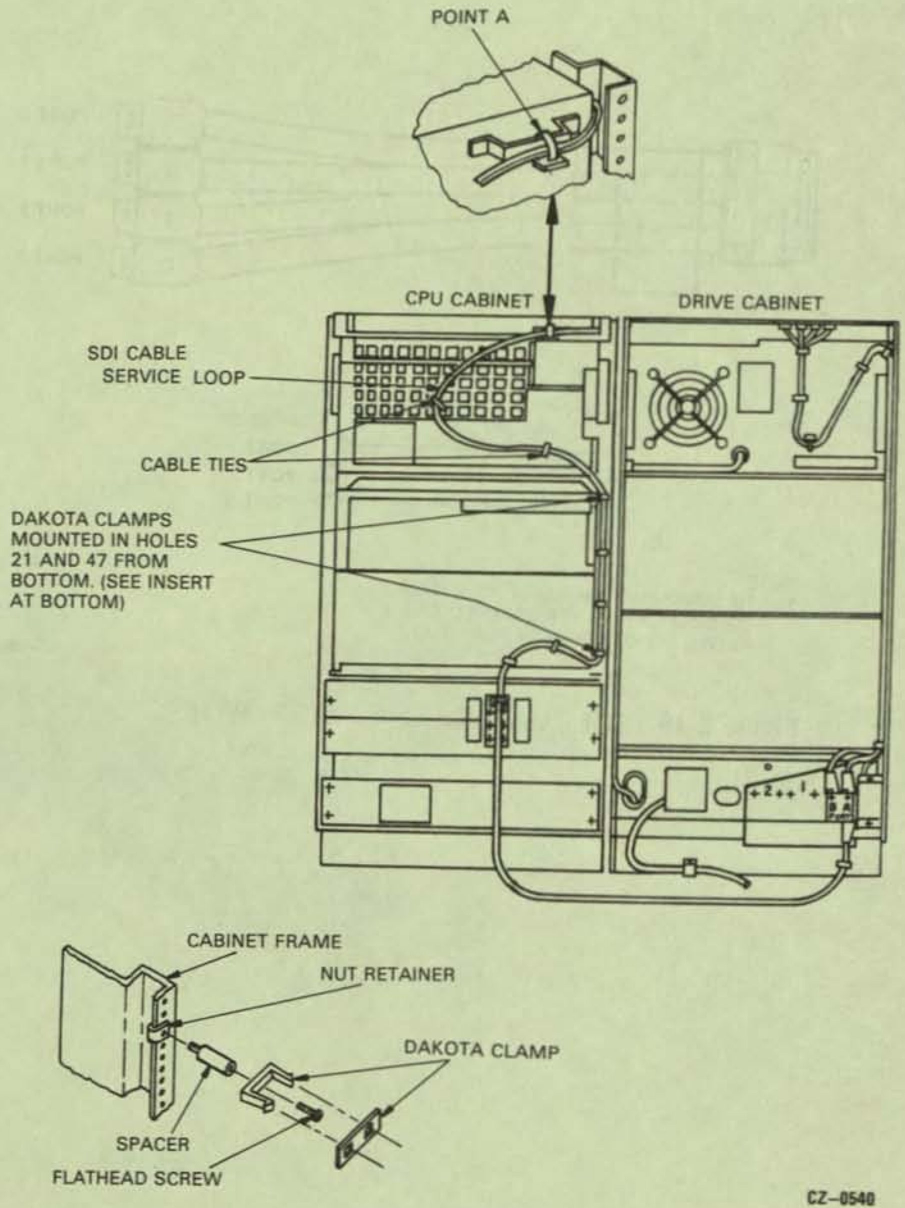


Figure 2-20 SDI Cable Assembly Installation

6. Insert the SDI cable plugs into the I/O bulkhead with the port 0 cable in the top connector. The I/O bulkhead connectors are numbered 0, 1, 2, and 3 from the top. Clamp the SDI cables to the retainer bracket. Refer to Figure 2-21.

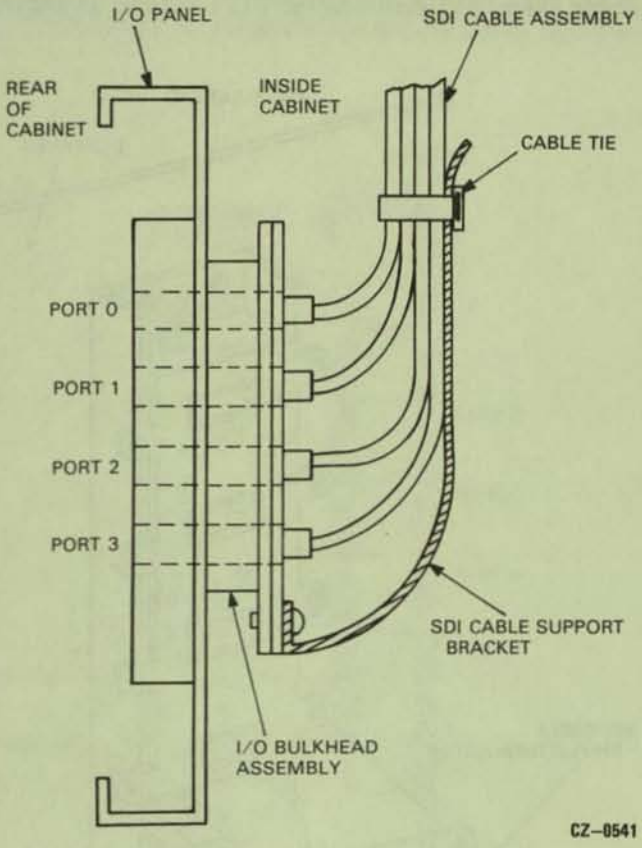


Figure 2-21 SDI Cable Retainer Bracket

2.3.9.4 External SDI Cables – The external SDI cables are shielded cables that must be grounded to the I/O bulkhead by mounting the shield terminators with screws. Use the following procedure to install these cables.

1. Plug the first SDI cable into the top I/O bulkhead connector (Port 0).

2. Screw the SDI cable shield terminator to the I/O bulkhead as shown in Figure 2-22.

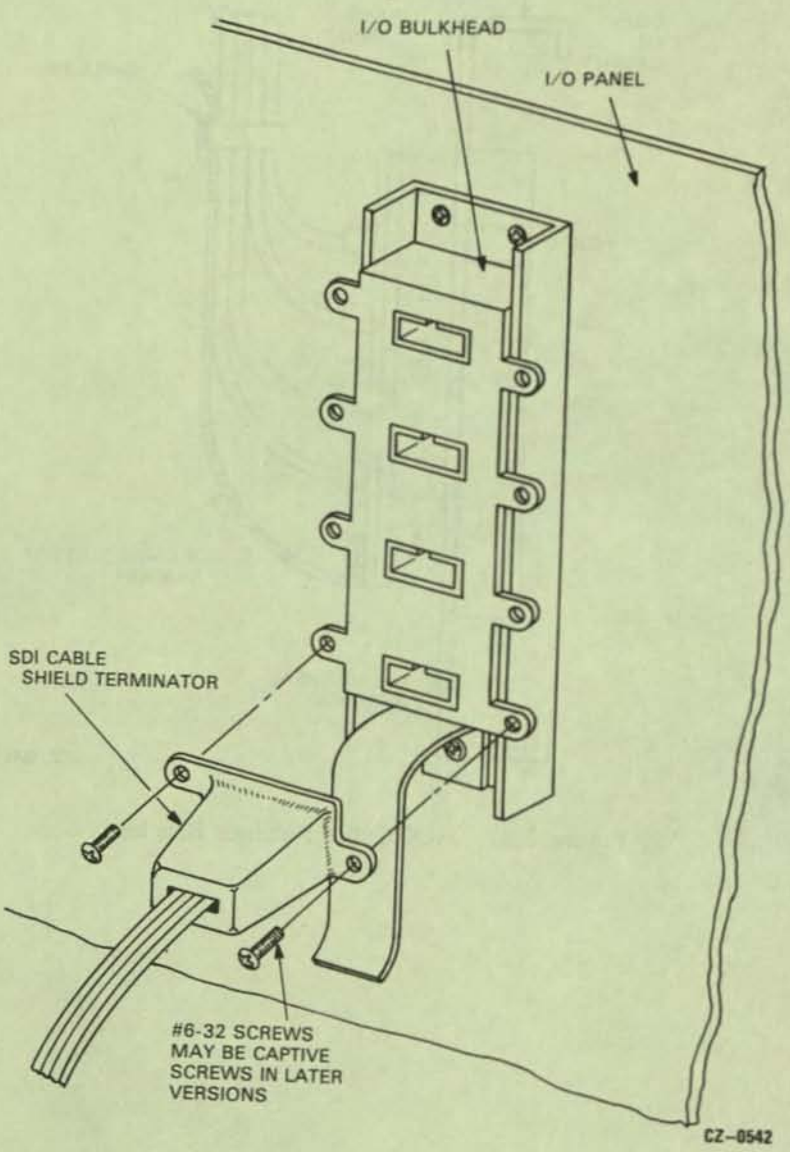

I/O BULKHEAD
I/O PANEL
SDI CABLE SHIELD TERMINATOR
#6-32 SCREWS MAY BE CAPTIVE SCREWS IN LATER VERSIONS
CZ-0542

Figure 2-22 SDI Cable Shield Terminator Installation

3. Install an SDI cable for each disk drive, starting at I/O bulkhead connector 0 and going down sequentially.

4. Secure the SDI cables to the SDI cable retainer bracket shown in Figure 2-23.

5. Install the drive end of the SDI cables into the drive I/O bulkhead connectors as described in the disk drive user guide. The UDA50 port 0 SDI cable should attach to drive 0, UDA50 port 1 to drive 1, etc.

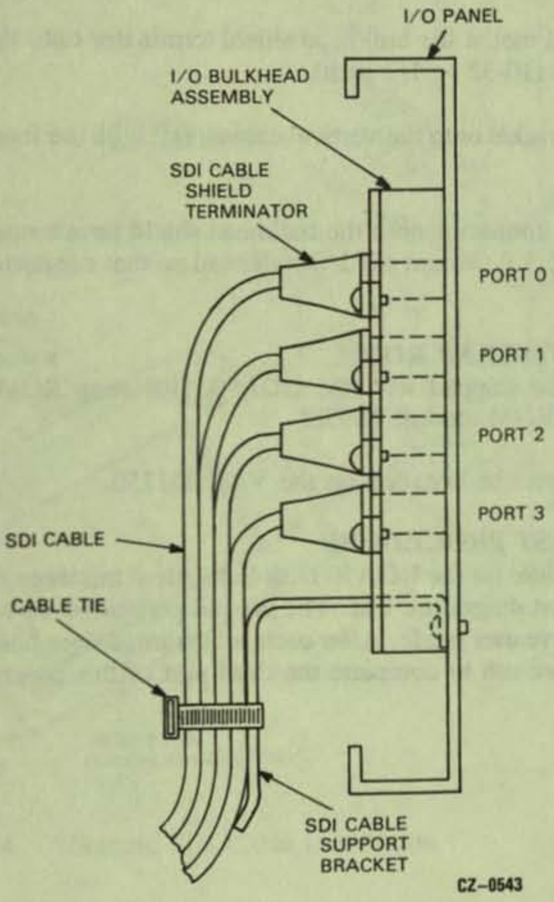


Figure 2-23 Clamping the SDI Cables to the Support Bracket

2.3.9.5 Alternate SDI Cable Installation – The SDI cable installation procedures described in paragraph 2.3.9.3 and 2.3.9.4 should be used whenever an I/O panel is available and room permits. When no I/O panel is present, an alternate means of SDI cable installation is provided. This alternate procedure requires the parts shown in Figure 2-24. Use Figure 2-24 as a reference and perform the following steps.

1. Select a suitable location on either rear vertical cabinet rail where this alternate I/O bulkhead can be mounted without interfering with existing equipment. Choose the lowest available location in the cabinet.
2. Push on the four u-nuts to align with the holes in the vertical rail bracket.
3. Select the best angle and mount the bulkhead shield terminator onto the vertical rail bracket with two' Phillips head sems (10-32 × 1/2 inch).
4. Mount the vertical rail bracket onto the vertical cabinet rail with the four Phillips head sems (10-32 × 1/2 inch).
5. Install the I/O bulkhead connector onto the bulkhead shield terminator using the same procedure described in paragraph 2.3.8. Mount the I/O bulkhead so that connector number 0 is towards the right.

2.4 INSTALLATION OF BOOTSTRAP ROM

The proper bootstrap ROMs will be shipped with the UDA50. Bootstrap ROM # 23-767A9-00 must be installed on the PDP-11 bootstrap ROM module M93l2.

Bootstrap ROM # 23-990A9-00 must be installed on the VAX 11/750.

2.5 FIELD ACCEPTANCE TEST PROCEDURE

The field acceptance and test procedure for the UDA50 Disk Subsystem has three parts. The first part is to run the UDA50 Disk Controller resident diagnostic test. The second part involves running the disk drive field acceptance test found in the disk drive user guide. After each subsystem device has been tested separately, the UDA50 host-resident diagnostics are run to complete the third part of this procedure.

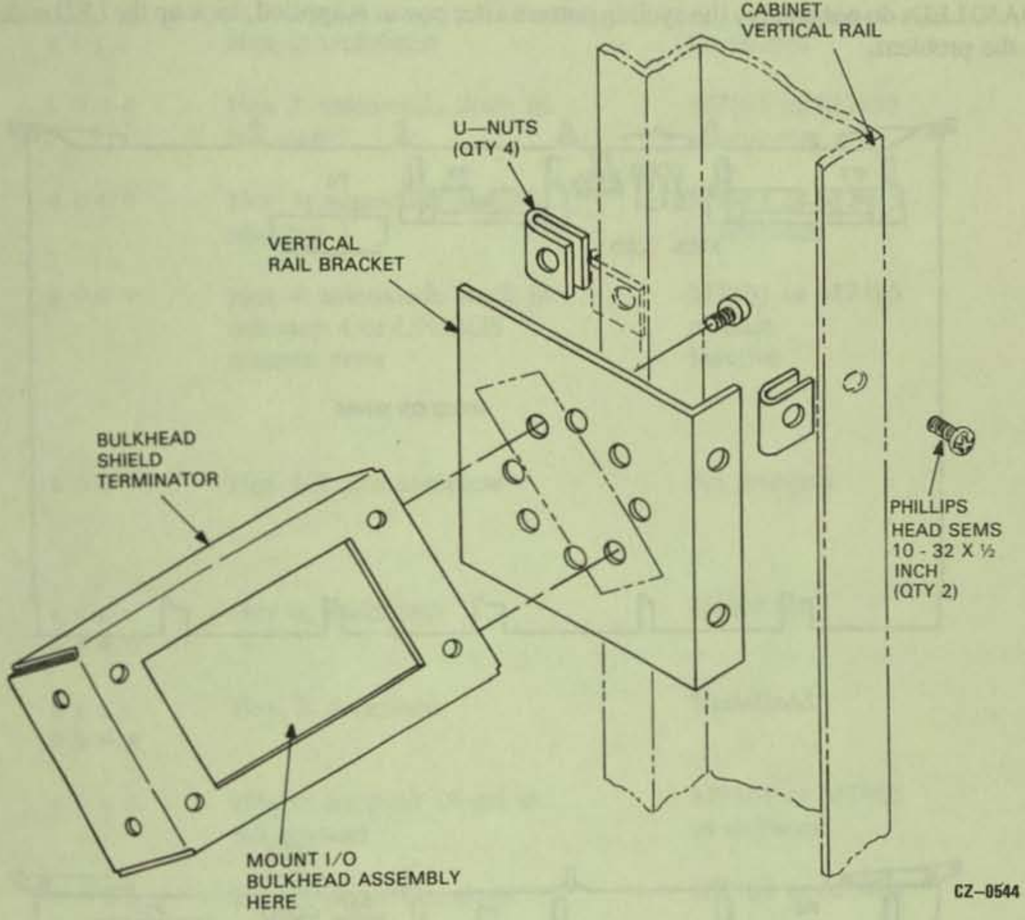


Figure 2-24 Alternate SDI Cable Installation

2.5.1 UDA50-Resident Diagnostics

The UDA50-resident diagnostics is initiated when power is applied to the UDA50 Disk Controller. The CPU should be halted during this test. The four LED indicators on each UDA50 module should display a cycling pattern in the LEDs. The cycling pattern in the LEDs signifies the completion of a successful UDA50 diagnostic test. Figure 2-25 shows the location of the four LEDs on each UDA50 module.

If the UDA50 LEDs do not display the cycling pattern after power is applied, look up the LED code in Table 2-3 to locate the problem.

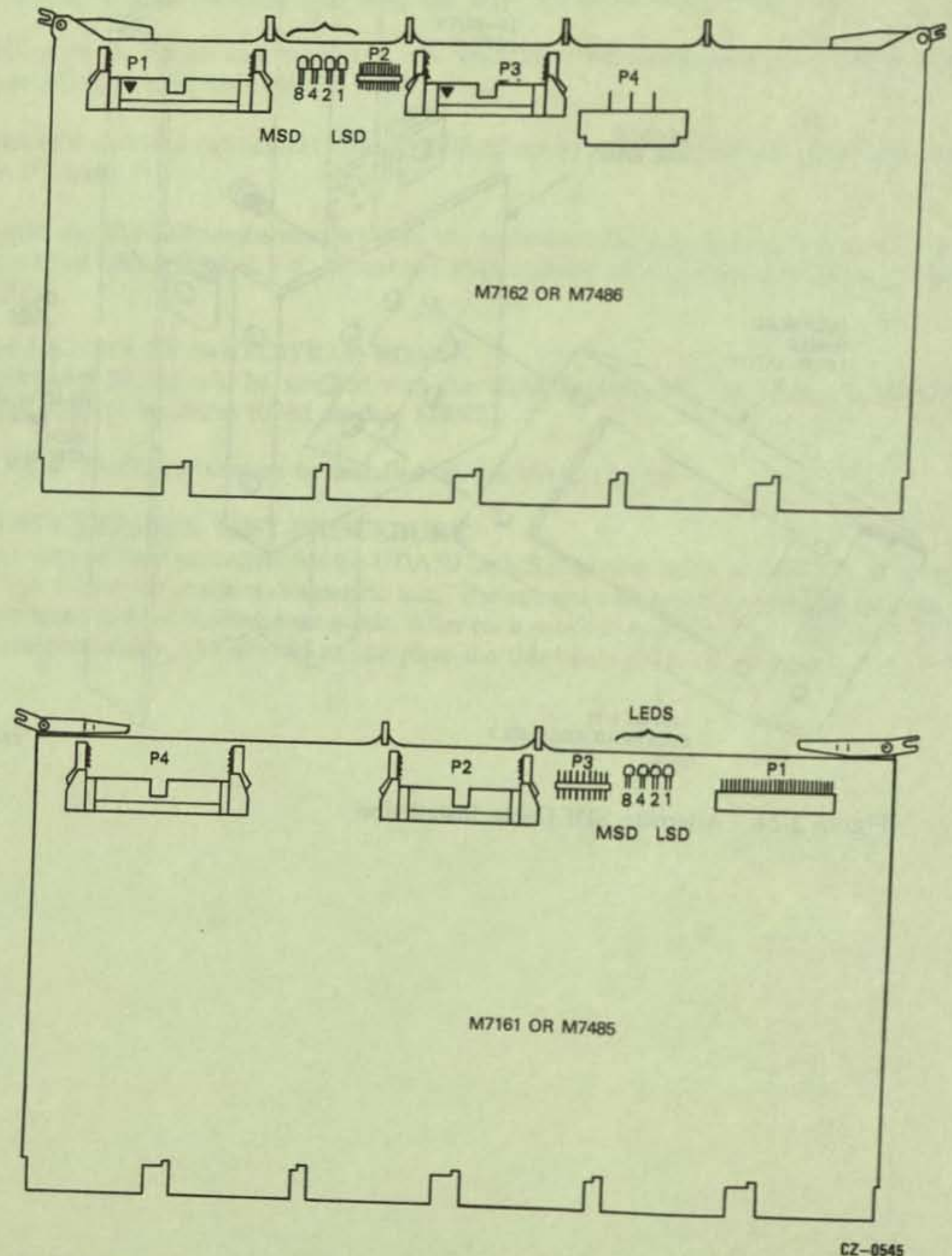


Figure 2-25 Diagnostic LED Locations on UDA50 Modules

Table 2-3 **LED Error and Symptom Codes**

M7161 or M7485 LEDs 8 4 2 1	M7162 or M7485 LEDs 8 4 2 1	Error Symptoms	Most Likely Failure
○ ○ ○ ●	x x x x	Hex 1; undefined	Undefined
○ ○ ● ○	○ ○ ○ ○	Hex 2; microcode stuck in init step 2	M7161 or M7485 or software
○ ○ ● ●	○ ○ ○ ○	Hex 3; microcode stuck in init step 3	M7161 or M7485 or software
○ ● ○ ○	○ ○ ○ ○	Hex 4; microcode stuck in init step 4 or UNIBUS timeout error	M7161 or M7485 or host inactive
B L ○ ● ○ I N K	○ ○ ○ ○	Hex 4/5; test complete	No problem
○ ● ● ○ x x x x	x x x x ○ ● ● ○	Hex 6; undefined	Undefined
○ ● ● ● x x x x	x x x x ○ ● ● ●	Hex 7; undefined	Undefined
● ○ ○ ○	○ ○ ○ ○	Hex 8; wrap bit 14 set in SA register	M7161 or M7485 or software
● ○ ○ ● ○ ○ ○ ○	○ ○ ○ ○ ● ○ ○ ●	Hex 9; board one error	M7161 or M7485
● ○ ● ○ ● ○ ● ○	○ ○ ○ ○ ● ○ ● ○	Hex A; board two error	M7162 or M7486
● ○ ● ● x x x x	x x x x ● ○ ● ●	Hex B; undefined	Undefined
x x x x ● ● ○ ○	● ● ○ ○ x x x x	Hex C; ROM parity error	M7161 or M7485
● ● ○ ● x x x x	x x x x ● ● ○ ●	Hex D; RAM parity error	M7162 or M7486
● ● ● ○ x x x x	x x x x ● ● ● ○	Hex E; ROM or RAM parity error	M7161 or M7485 M7162 or M7486
● ● ● ●	● ● ● ●	Hex F; sequencer error	M7161 or M7485

Table 2-3 LED Error and Symptom Codes (Cont)

M7161 or M7485 LEDs 8 4 2 1	M7162 or M7485 LEDs 8 4 2 1	Error Symptoms	Most Likely Failure
Cycling pattern	Cycling pattern	UDA responds to host if cycling pattern lasts less than 2 seconds after host sends step 1 data.	No problem
		UDA does not respond to host if cycling pattern lasts more than 2 seconds after host sends step 1 data.	M7161 or M7485

Note: ● = LED ON ○ = LED OFF x = Does not care condition

When two codes are given for the same error, both indicate the same failure.

2.5.2 Drive-Resident Diagnostics

Each disk drive should be tested separately by running the drive-resident diagnostics. The procedure for running the resident diagnostics is found in the installation chapter of the disk drive user guide. Perform the drive field acceptance tests found in the installation chapter and then go to Paragraph 2.4.3 for the subsystem diagnostic procedures.

2.5.3 Subsystem Diagnostics

The subsystem diagnostic procedures for the UDA50 controller are different, depending on whether they are used on a PDP-11 CPU or a VAX CPU. The following paragraphs first cover the PDP-11 diagnostics and then secondly, the VAX diagnostics.

NOTE

If the diagnostic program reports errors, refer to the *UDA50 Service Manual* or *Maintenance Guide*.

2.5.3.1 PDP-11 Subsystem Diagnostics –

- CZUDEC0 (UDA disk formatter)

  Most disk drives will be shipped with formatted disk packs. On these disk drives, it will not be necessary to run the formatter program. Refer to the disk drive user guide to determine if you must run the formatter before the diagnostic program.

- CZUDCC0 (UDA and disk drive diagnostic)

  This diagnostic consists of the following four tests:

  - Test 1 - UNIBUS Addressing Test
  - Test 2 - Disk-Resident Diagnostic Test
  - Test 3 - Disk Functional Test
  - Test 4 - Disk Exerciser Test

  The hardware and software questions asked by this diagnostic are shown in the following two samples along with their default conditions < X>.

NOTE
Refer to the software documentation for detailed description, error messages, etc.

Sample hardware questions:

```
CHANGE HW (L)      ? N

# UNITS (D)   ? 1

UNIT 0
UNIBUS ADDRESS OF UDA (O) 172150
VECTOR (O) 154?
BR LEVEL (D) 5 ?
UNIBUS BURST RATE (D)   0?
DRIVE NUMBER (D) 0?

EXERCISE ON CUSTOMER DATA AREA IN TEST 4 (L) N ? N
```

Sample software questions:

```
CHANGE SW (L)      ? N

ENTER MANUAL INTERVENTION MODE FOR SPECIAL DIAGNOSDIS (L) Y ? N

REMAINING SOFTWARE QUESTIONS APPLY TO TEST 4 ONLY

ERROR LIMIT (D) 32 ?
READ TRANSFER LIMIT IN MEGABYTES - 0 FOR NO LIMIT (D) 0 ?
SUPRESS PRINTING SOFT ERRORS (L) Y ?
DO INITIAL WRITE ON START (L) Y ?
ENABLE ERROR LOG (L) N ?
```

2.5.3.2 VAX Subsystem Diagnostics –

- ZZ-EVRLB (UDA50 disk formatter)

  Most disk drives will be shipped with formatted disk packs. It will not be necessary nor desirable to run the formatter program on these disk drives. Refer to the disk drive user guide to determine if you must run the formatter before the diagnostic program.

- ZZ-EVRLA (UDA50 disk subsystem diagnostic)

  The VAX UDA host-resident diagnostic contains the following four tests.

  - Test 1 - UNIBUS addressing test
  - Test 2 - Disk-resident diagnostic test
  - Test 3 - Disk functional test
  - Test 4 - Disk exerciser test

  Use the verify section of this diagnostic for system installation.

- ZZ-EVRLC (Generic disk drive exerciser)

  This program tests the read and write ability of any SDI type disk drive, and will display differences in the read and write data to the operator.

NOTE

Refer to the software documentation for detailed descriptions, error messages, etc.

CHAPTER 3
UDA50 PROGRAMMER INFORMATION

3.1 GENERAL PROGRAMMING INFORMATION

The UDA50 operates according to the rules defined in three separate documents. The following is a list of these documents.

- *MSCP Basic Disk Functions Manual* (AA-L619A-TK)
- *Storage System Diagnostic and Utilities Protocol* (AA-L620A-TK)
- *Storage System UNIBUS Port Description* (AA-L621A-TK)

All three documents may be purchased separately or as a kit called the *UDA50 Programmer's Documentation Kit* (QP905-GZ) from the Software Distribution Center, Order Administration/Processing, 20 Forbes Road (NR4), Northboro, Massachusetts 01532.

3.2 UDA50-SPECIFIC PROGRAMMING INFORMATION

The following information is UDA50-specific and is necessary for anyone needing to write his own software for the UDA50.

- The address of the UDA50 IP register is 772150 (octal).
- The address of the UDA50 SA register is 772152 (octal).
- The UDA50 supports a host-settable interrupt vector address. A vector address of 154 (octal) is assigned to the UDA50.
- The UDA50 has a command limit value of 13. This includes 12 MSCP commands plus 1 immediate-only command.
- The UDA50 supports an NPR burst value of 1 to 32 long words. One long-word is the default condition.
- The UDA50 supports only 512 byte disk formats.
- The UDA50 supports both the MSCP and the diagnostic and utilities protocols (DUP).
- The diagnostic option capabilities available on the UDA50 are the purge and poll and the diagnostics wrap.
- The UDA50 supports maintenance read and maintenance write to and from the UDA RAM.
- The UDA50 supports last fail log packets.

UDA50 USER GUIDE
EK-UDA50-UG-002

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use?

What features are most useful?

What faults or errors have you found in the manual?

Does this manual satisfy the need you think it was intended to satisfy?

Does it satisfy *your* needs? Why?

□ Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name
Title
Company
Department

Street
City
State/Country
Zip

Additional copies of this document are available from:

Digital Equipment Corporation
444 Whitney Street
Northboro, Ma 01532
Attention: Communications Services (NR2/M15)
Customer Services Section

Order No. EK-UDA50-UG-002

CX

Fold Here

Do Not Tear - Fold Here and Staple

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
EDUCATIONAL SERVICES/QUALITY ASSURANCE
12 CROSBY DRIVE (BUO/E08)
BEDFORD, MA 01730

Digital Equipment Corporation • Bedford, MA 01730