DELQA User's Guide

Order Number: EK-DELQA-UG-002

Prepared by Educational Services
of
Digital Equipment Corporation

This document was prepared using VAX DOCUMENT, Version 1.0

CONTENTS

CHAPTER 2 installation

CHAPTER 3 Programming

CHAPTER 4 MAINTENANCE

Appendix A VECTOR ASSIGNMENTS

Appendix B Diagnostics

Appendix C PROGRAMMING EXAMPLES FOR PDP-11 SYSTEMS

INDEX

FIGURES

TABLES

PREFACE

INTRODUCTION

The DELQA module is a communications option which connects the Q-bus to an Ethernet local area network (LAN).

This manual describes how to install, program, and maintain the DELQA. It contains information for first-time servicing and field-service support, and for customer engineers and programmers.

The chapters are as follows.

Chapter 1	introduces the Ethernet LAN and the DELQA module.
Chapter 2	describes how to install a DELQA module.
Chapter 3	describes how to program the DELQA.
Chapter 4	describes how to use the diagnostic utilities to maintain the module
Appendix A	details the DELQA vector address and assignments.
Appendix B	summarizes commands and facilities for the DELQA diagnostics.
Appendix C	gives examples of host software programming of the DELQA.
Appendix D	gives details of DELQA responses to undesired events.
Appendix E	is a glossary.

This revision of the manual contains new information and Chapter 3 has been expanded to contain additional programming notes.

Notes and Warnings

NOTES and **WARNINGS** are defined as follows.

- A **NOTE** contains general information.
- A **WARNING** is designed to prevent personal injury.

Related Publications

Communications Options Mini–Reference Manual: Volume IV (Ethernet) (EK-CMIV4-RM)

DECnet Maintenance Operations Protocol (MOP) Functional Specification V3.0.0 (AA-X436A-TK)

DECnet-RSX System Manager's Guide (AA-H224C-TC)

DECnet-ULTRIX Guide to Network Management (AA-EE38A-TE)

DECnet-VAX System Manager's Guide (AA-H803C-TE)

DEC/X11 User's Manual (AC-F053-MC)

DELQA Field Maintenance Print Set (MP-02379)

DELQA Technical Description (EK-DELQA-TD-001)

Ethernet: A Local Area Network, Data Link Layer, and Physical Layer Specifications (AA-K759B-TK)

Ethernet Installation Guide (EK-ETHER-IN)

H4000 Ethernet Transceiver Technical Manual (H4000-TM)

Introduction to Local Area Networks (EB-22714-18)

MicoPDP-11 Systems Service Maintenance Guide (EK-MIC11-SG)

MicroVAX 11 System Maintenance Guide (AZ-GM3AA-MN)

Network Interconnect Exerciser Diagnostic (AC-T585A-MC)

XXDP+/SUPR User's Manual (AC-F348A-MC)

NOTE

When installed in a Micro-PDP11 or a MicroVAX, this equipment has been tested with a Class A computing device and has been found to comply with part 15 of FCC Rules. Operation in a residential area may cause unacceptable interference to radio and TV reception requiring the operator to take whatever steps are necessary to correct the interference.

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This chapter introduces the M7516 module, which is a DIGITAL Ethernet Local-Area-Network to Q-bus Adapter (DELQA). The sections are as follows.

Section 1.2	Ethernet Overview
Section 1.3	DELQA Overview
Section 1.4	Specification
Section 1.5	Interfaces
Section 1.6	Functional Description

1.2 ETHERNET OVERVIEW

1.2.1 General Description

Ethernet employs a branching-bus topology, with all nodes granted equal access rights. Using repeaters, the main bus can be extended up to 2.8 kilometers (1.74 miles) between the two furthest nodes of the network. Along this length, up to 1024 nodes can be tapped into the network.

Each node is a single addressable entity, comprising a controller and a transceiver. The transceiver is connected to the Ethernet cable by a cable tap. The cable that connects the transceiver to the controller can be up to 50 meters long. The transceiver itself is not always necessary; for example, the connection to the Ethernet may be made using a DELNI multiplexer.

Figure 1-1 shows an example of a large-scale Ethernet configuration.

Safety warnings are shown in Figures 1-3 and 1-4.

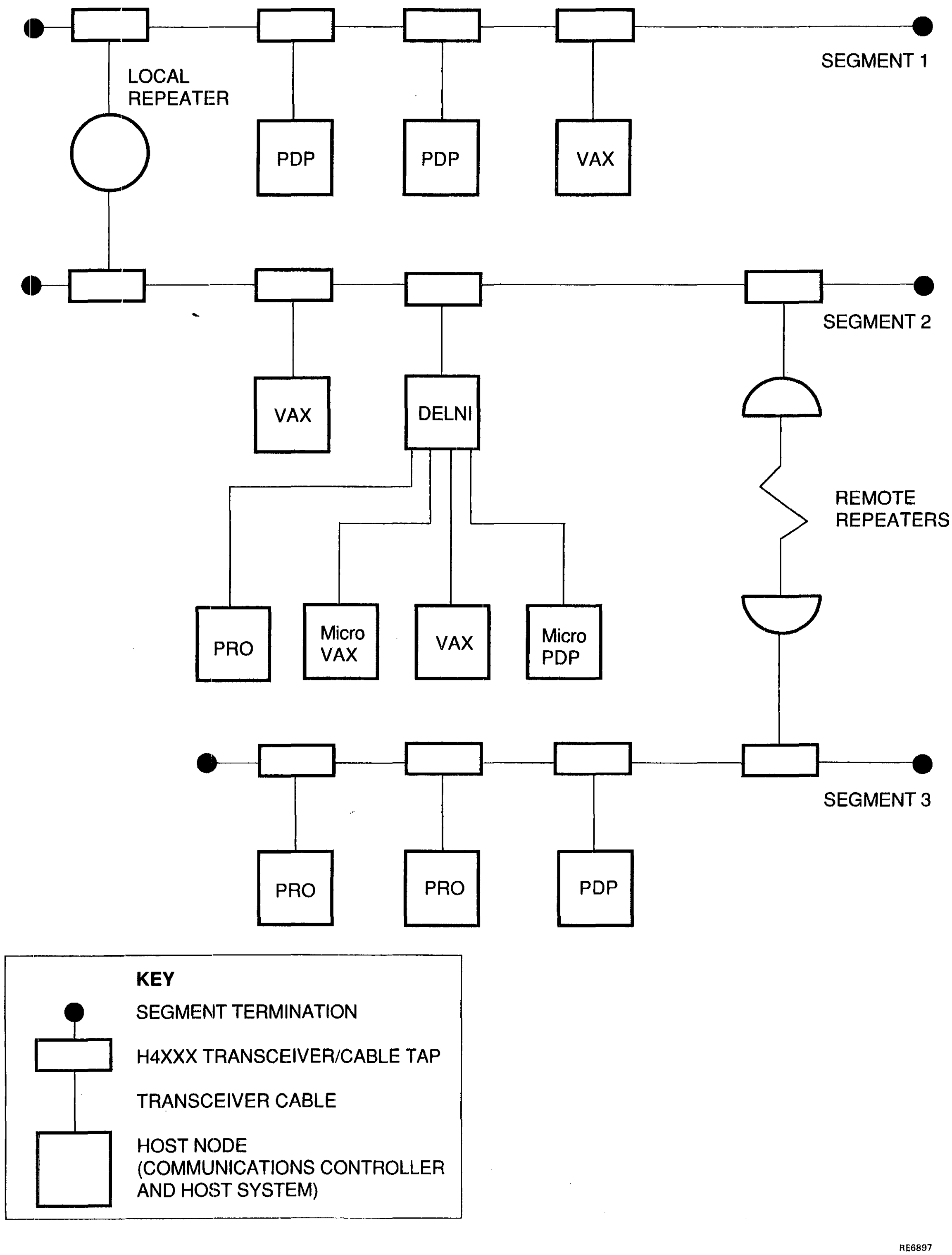


Figure 1–1 Typical Ethernet Configuration

WARNING

Ethernet installations may extend to thousands of meters and couple hundreds of separate items of equipment. To prevent hazardous voltages appearing on the installation, it is important that all the equipment be part of a common equipotential bonding system as defined in IEC publication 364-4-41 clauses 413.1.2 and 413.1.6. Where it is required o couple equipment outside of the main equipotential bonded area via ethernet, then optical repeaters or other such galvanically isolated measures must be employed. If in doubt please refer to Digital for advice.

VAROITUS

Ethernet-verkot voivat olla tuhansia metrejä pitkiä ja niihin voidaan liittää satoja erilaisia laitteita. Jotta verkkoon ei pääsisi syntymään vaarallisia jännitteitä, kaikkien laitteiden on ehdottomasti kuuluttava samaan potentiaalintasausjärjestelmään, jonka ominaisuudet on määritetty IEC:n julkaisussa 364-4-41, kohdissa 413.1.2 ja 413.1.6. Mikäli Ethernetiin halutaan liittäälaite, joka ei kuulu potentiaalintasausjärjestelmään, on käytettäväoptisia toistimia tai vastaavia galvaanisesti eristettyjä menetelmiä. Jos et ole varma käytettävästä menetelmästä, ota yhteys Digitaliin.

DANGER

Une installation Ethernet peut s'étendre sur des kilomètres et relier des centaines d'éléments. Afin d'éviter tout problème électrique, vérifiez la présence d'une mise à la terre commune ainsi qu'elle est définie par l'IEC (364.4.41, clauses 413.1.2 et 413.1.6). S'il s'avère nécessaire de relier par Ethernet des équipements non rattachés à une même terre, utilisez des répéteurs optiques ou autres matériels offrant la même qualité d'isolation. En cas de doute, prenez contact avec les Services techniques Digital.

VORSICHT

Ethernet-Netzwerke können sich über mehrere tausend Meter erstrecken und mehrere hundert einzelne Geräte miteinander verbinden. Zur Vermeidung von gefährlichen Spannungen im Netzwerk ist es unbedingt erforderlich, daß alle Geräte Teil einer gemeinsamen Erdungsschleife sind, wie in den IEC-Richtlinien 364-4-41, Abschnitte 413.1.2 und 413.1.6 angegeben. Wenn Geräte außerhalb der Erdungsschleife über Ethernet miteinander verbunden werden müssen, müssen optische Repeater oder andere galvanisch getrennte Mittel verwendet werden. Falls Sie Fragen haben, wenden Sie sich an Digital Equipment.

WAARSCHUWING

Ethernet-configuraties kunnen een afstand van verschillende kilometers overbruggen en honderden afzonderlijke apparaten met elkaar verbinden. Om te vermijden dat er zich gevaarlijke spanningen zouden voordoen op de configuratie, is het belangrijk dat alle apparatuur gebruik maakt van dezelfde voeding en dezelfde aarde, zoals gedefinieerd in de IEC-publikatie 364-4-41, bepalingen 413.1.2. en 413.1.6. Wanneer apparatuur die niet op eenzelfde equipotentiaal spanningsnet is aangesloten via Ethernet gekoppeld moet worden, moet men gebruik maken van optische repeaters of van andere galvanisch isolerende technieken. Bij twijfel gelieve u contact op te nemen met Digital.

ATTENZIONE

Le installazioni Ethernet possono estendersi per migliaia di metri e collegare diverse centinaia di elementi separati di apparecchiature. Per evitare il rischio di scariche elettriche al momento dell'installazione, è importante che tutte le apparecchiature siano collegate ad un comune sistema di massa come definito nella pubblicazione IEC 364-4-41, clausole 413.1.2 e 413.1.6. Laddove si richieda di collegare l'apparecchiatura fuori dalla principale area di massa via Ethernet, si devono utilizzare ripetitori su fibra ottica o qualsiasi altro strumento isolato gslvanicamente. Per qualsiasi informazione rivolgersi alla sede Digital più vicina.

ADVARSEL

Ethernetinstallasjoner kan strekke seg over flere tusen meter og ha tilkoblet flere hundre forskjellige utstyrsenheter. For å forhindre at det skal oppstå farlige spenninger på installasjonen, er det viktig at alt utstyret tilhører et felles ekvipotensialt forbindelsesystem, slik det er definert i IEC-publikasjon 364-4-41, paragrafene 413.1.2 og 413.1.6. Der hvor det er påkrevet å koble utstyr via Ethernet utenfor det ekvipotensiale hovedområdet, er det påbudt å benytte optiske linjeforsterkere (repeatere) eller tilsvarende galvanisk isolert materiale. Kontakt Digital hvis du er i tvil.

RE7124

Figure 1–2 Warnings

ATENCION

Las instalaciones basadas en Ethernet pueden cubrir áreas de varios centenares de metros e interconectar distintos módulos de un equipo. Para evitar que se den tensiones peligrosas en la instalación es necesario que todos los componentes se conecten a una masa única, de acuerdo con normas IEC 364-4-41 (§413.1.2 y §413.1.6). Cuando sea preciso utilizar Ethernet con componentes que no vayan conectados a dicha masa común se utilizarán repetidores ópticos u otros dispositivos de medida con aislamiento galvánico. En caso de duda consulte con Digital.

VARNING

Ethernet-installationer kan omfatta tusentals meter kabel som kopplar samman hundratals separata delar av en utrustning. För att skadliga spänningar ska undvikas är det viktigt att all utrustning har gemensam jord enligt vad som anges i IEC:s skrift 364-4-41, avsnitten 413.1.2 och 413.1.6. Där det är nödvändigt att ansluta utrustning med annan jordning via Ethernet, måste optiska kopplare användas eller andra åtgärder vidtas för att åstadkomma galvanisk isolering. Kontakta gärna Digital för ytterligare information.

AVISO

A instalação da Ethernet pode estender-se por milhares de metros e agrupar centenas de itens de equipamento.

Para evitar que voltagens perigosas surjam na instalação, é importante que todo o equipamento faça parte de um sistema eléctrico equipotencial comum, tal como definido na publicação 364-4-41 do IEC, cláusulas 413.1.2 e 413.1.6.

Onde fôr necessário ligar equipamento fora da área principal de ligação eléctrica equipotencial, através da Ethernet, deverão ser empregues repetidores ópticos ou outras soluções galvanicamente isoladas.

Em caso de dúvida, contacte a Digital.

ADVARSEL

Ethernet-installationer kan strække sig over tusindvis af meter og forbinde hundredevis af separate dele af udstyr. For at undgå farlig spænding i installationerne er det vigtigt, at alt udstyret er del af et fælles jordingspunkt som defineret i IEC publikation 364-4-41, klausulerne 413.1.2 og 413.1.6. Hvor det er nødvendigt at forbinde udstyr udenfor det større fælles jordingspunkt via Ethernet, skal der anvendes optisk kobling eller anden form for galvanisk isolering af udstyret. For yderligere oplysninger henvises til den lokale Digital afdeling.

א ז ה ר ה

התקנות ה-ETHERNET משתרעות לפעמים על פני אלפי מטרים, והן
עלולות לכלול כמה מאות פריטי ציוד נפרדים. כדי למנוע מתחים
חשמליים שעלולים להוות סכנה במתקן, מאד חשוב להקפיד שכל הציוד
יהווה חלק ממערכת חשמל משותפת הנמצאת באותו מבנה והמחברת בין
מרכיביה, השווים בכח ובפוטנציאל, כפי שהוגדר ב- IEC, דבר דפוס
364-4-41 סעיפים 413.1.2 ו- 413.1.6.
במקומות שבהם נדרש לחבר בין פריטי ציוד מחוץ למבנה הכולל את
מערכת החשמל הראשית המשותפת, באמצעות ETHERNET, אזי חייבים להשתמש
בציוד אופטי (OPTICAL REPEATERS, BRIDGES) או באמצעים אחרים
המבודדים רצף מתכתי.
במידה ויתעוררו ספקות, נא לפנות למשרד דיגיטל הקרוב.

注　意

イーサネットの設置は数千メータに及んだり、二、三百の設置項目（機器）に及ぶことがあります

設置に際する危険な電圧の発生を防ぐためには、ＩＥＣ公報３６４－４－４１の条項４１３．１．２、および４１３．１．６に定められているように、全ての機器が共通接地システムに接続されていることが重要です

共通接地システムに接続できない場所（他のビル等）にイーサネットを介して機器を設置する必要がある場合、オプチカルリピータ、または電気的に分離された手段を講じることが必要です

ご不明な点は当社にお問い合せ下さい

RE7125

Figure 1–3 Warnings

The principal characteristics of the Ethernet are:

Topology:	Branching bus
Medium:	Shielded coaxial cable
Transmission:	Manchester-encoded digital baseband signaling
Data Rate:	10 Megabits/second.
Access Control:	Carrier Sense, Multiple Access with Collision Detect (CSMA/CD).
Allocation:	64- to 1518-byte packet length (includes variable-length data field between 46 and 1500 bytes).

The maximum Ethernet configuration for a coaxial cable bus is as follows:

- Each segment of coaxial cable can be up to 500 meters long (1640.5 feet). Each segment must be terminated at both ends.
- Up to 100 nodes can be tapped into a cable segment. Each node must be between 2.5 meters (8.2 feet) and 1500 meters (4921.5 feet) from its nearest neighbors. Standard transceiver positions are usually marked at every 2.5 meters.
- A transceiver cable (from transceiver to node controller) can be up to 50 meters (164 feet) long.
- Repeaters are used to retransmit signals from one segment to another. A repeater uses a node position, and also contributes to the total node count, on both the segments that it connects. There can be up to two repeaters in the path between any two nodes.
- Repeaters can be placed at any position along a cable segment to extend the network bus up to a maximum of 2.8 kilometers (1.74 miles). This would comprise three segments of 500 meters, plus six transceiver cables of 50 meters, plus a 1 km fiber cable between remote repeaters.

The Ethernet configuration rules ensure the best network performance within physical channel limitations.

Figure 1-4 shows the limits of Ethernet connectivity.

1.2.2 Ethernet Layers

The Ethernet architecture is structured in two layers which correspond to the lowest layers in the International Standards Organization (ISO) model for Open Systems Interconnection (OSI).

The two layers have the following functions.

- The *physical layer* specifies the maximum number of nodes, their maximum separation, the data rate on the Ethernet bus, as well as the electrical and mechanical connections.
- The *data link layer* specifies the mechanism for access control (CSMA/CD), the procedure for multiaccess network control, and the format of transmission packets.

The physical layer and the data link layer together provide a *datagram* service for transmitting message packets between nodes. A datagram service cannot guarantee that a packet is received, because transmission and reception are the responsibility of higher levels in the network architecture; but it does guarantee that those packets that are received are correct.

The DELQA module handles all of the physical layer, and part of the data link layer functions. Host software handles the higher levels of protocol, as well as network management, error recovery, internetwork communication, and the user interface.

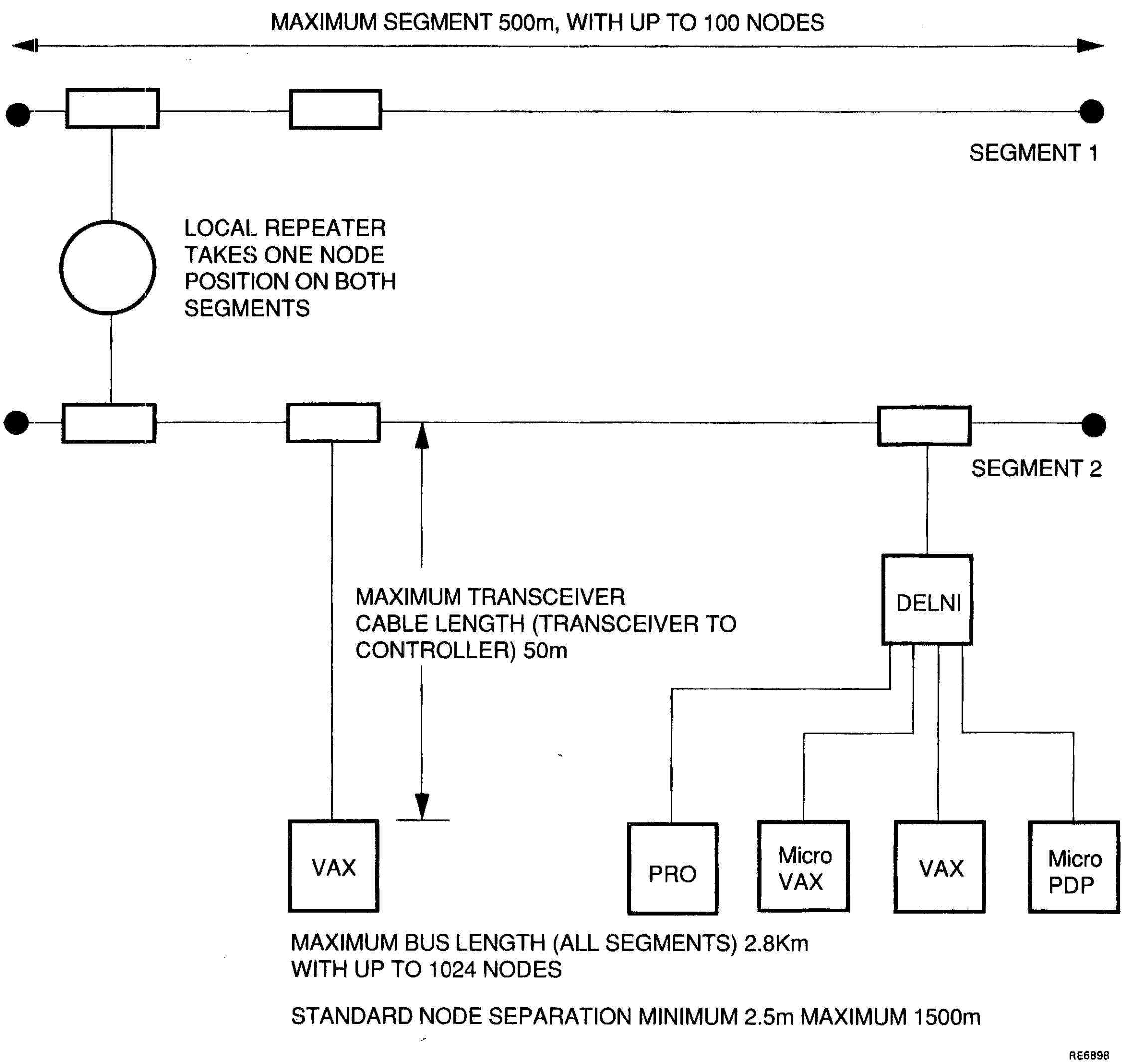


Figure 1–4 Ethernet Connectivity

1.2.3 Data Encapsulation

Data is transmitted over an Ethernet in packets (or frames) that have a specific format.

Figure 1-5 shows the format of an Ethernet packet. Table 1-1 gives the size of each field in an Ethernet packet.

Table 1–1 Field Sizes in an Ethernet Packet

Field	Bytes
Destination	6
Source	6
Type	2
Data	46 to 1500
CRC	4
Total packet	64 to 1518

A packet is preceded by a 64-bit preamble which is a pattern of alternating 1s and 0s for receiving node synchronization. The pattern ends with ...01011 rather than ...01010.

The fields in the packet are as follows.

1. The *destination field* contains the 48-bit address of the receiving node(s). The address is either physical or logical, and it may be any one of the following.
   - An individual node address (first address bit = 0)
   - A multicast address for a group of nodes (first address bit =1).
   - A broadcast address for all nodes (all address bits = 1).
2. The *source field* contains the 48-bit Ethernet physical address of the sending node.
3. The 16-bit *type field* determines how higher-level software interprets the data field.
4. The protocol *data field* itself must contain between 46 and 1500 bytes. If the data to be sent consists of less than 46 bytes, software must insert null bytes to fill the field.
5. The *frame check sequence* (FCS) contains a 32-bit Cyclic Redundancy Check (CRC) value. The DELQA module calculates this value, inserts it when a packet is transmitted, and checks it when a packet is received. The CRC is removed from a received packet before delivery to the host.

The *interframe spacing* (or interpacket gap) allows the physical channel to recover between packets. The minimum spacing is 9.6 microseconds.

Figure 1-3 shows the standard Ethernet packet format.

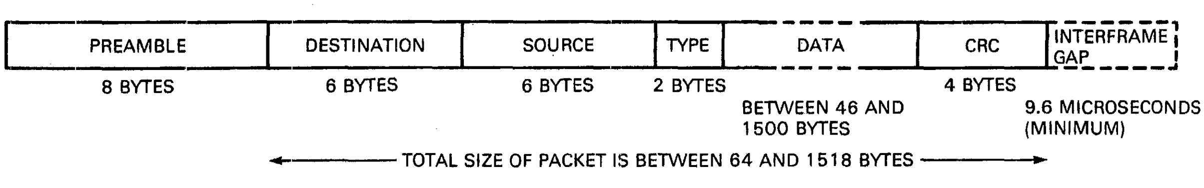


Figure 1–5 Ethernet Packet (Frame) Format

1.3 DELQA OVERVIEW

1.3.1 General Description

The Q-bus communications controller on the DELQA module interfaces both the MicroVAX and LSI-11 families of processors to an Ethernet Local Area Network (LAN). The controller is compatible with both 18- and 22-bit Q-bus backplanes.

The DELQA module conforms to the Ethernet specification (Version 2.0), and is compatible with the IEEE 802.3 *Specification for Carrier Sense with Multiple Access with Collision Detection.* In terms of Open Systems Interconnection, the module implements the functions of the physical layer and part of the data link layer.

The principal functions of the DELQA module are to:

- Transmit/receive data at 10 Mbits/s
- Perform packet serialization, formatting, Manchester encoding, and multiple retransmission
- Generate and check a 32-bit CRC for each packet
- Interface with the Ethernet transceiver
- Perform Direct Memory Access (DMA) transfers to and from host memory

DELQA (Normal Mode) also supports:

- Generation of MOP Remote console system ID message
- Processing MOP Remote console system ID requests
- Processing of Ethernet channel loopback protocol requests
- Processing of MOP Remote console BOOT requests
- Processing of IEEE 802.2 NULL LSAP XID message requests
- Processing of IEEE 802.2 NULL LSAP TEST message requests.
- Write the MOP Boot Password value
- Read the MOP Boot Password value
- Write the MOP System ID Parameters
- Read the MOP System ID Parameters
- Reset the MOP System ID Parameters
- Read the last MOP Remote Boot message received
- Read the datalink counters values
- Read and clear the values of the datalink counters
- Read the current DELQA Physical Ethernet Address
- On-Board (OBT) self-test:
  - Execution on Power-Up
  - Host software request bit
  - Completion status with error report
- Boot/diagnostic code support.

Switches are provided on the DELQA module for setting the Q-bus base address, the operating mode (Normal or DEQNA-lock), and various operating conditions (such as Remote Boot). Other functions and configurations are programmable from the host.

The DELQA module has an on-board self-test that is independent of the host. On-line and standalone diagnostics are also available. Three LEDs on the module indicate the test status of the module.

Figure 1-6 shows the functional block diagram of the DELQA.

1.3.2 Physical Description

The DELQA module is a dual-height module which plugs directly into the Q-bus backplane.

The DELQA module may be connected to the Ethernet physically and electrically using an H4*xxx* transceiver; or using a DELNI multiplexer and a transceiver. The connection is made through the cabinet kit and transceiver cable.

The cabinet kit consists of a bulkhead panel and cable which is manufactured as a single assembly.

1.3.3 Order Codes

The DELQA module consists of a base option (DELQA-M) and a cabinet kit (CK-DELQA-Y*x*). These options are ordered separately.

Table 1-2 lists the DELQA module options.

Table 1–2 DELQA Ordering Options

Base Option	Description	
DELQA-M	DELQA Module M7516	DELQA User's Guide (EK-DELQA-UG)
CK-DELQA-YB	BA23	12 inches (30.5 cm)
CK-DELQA-YA	BA123	21 inches (53.6 cm)
CK-DELQA-YF	H9642	36 inches (91.5 cm)
	Each kit is supplied with a module-to-bulkhead cable of the appropriate length, and a 15-pin bulkhead loopback connector (12-22196-02).	

1.3.4 Q-bus Addresses

Table 1-3 lists the Q-bus addresses that are available for use by a DELQA module.

Table 1–3 Module Addresses

Base Address	Unit	Module
17774440	DELQA 1	DELQA or DEQNA
17774460	DELQA 2	DELQA or DEQNA

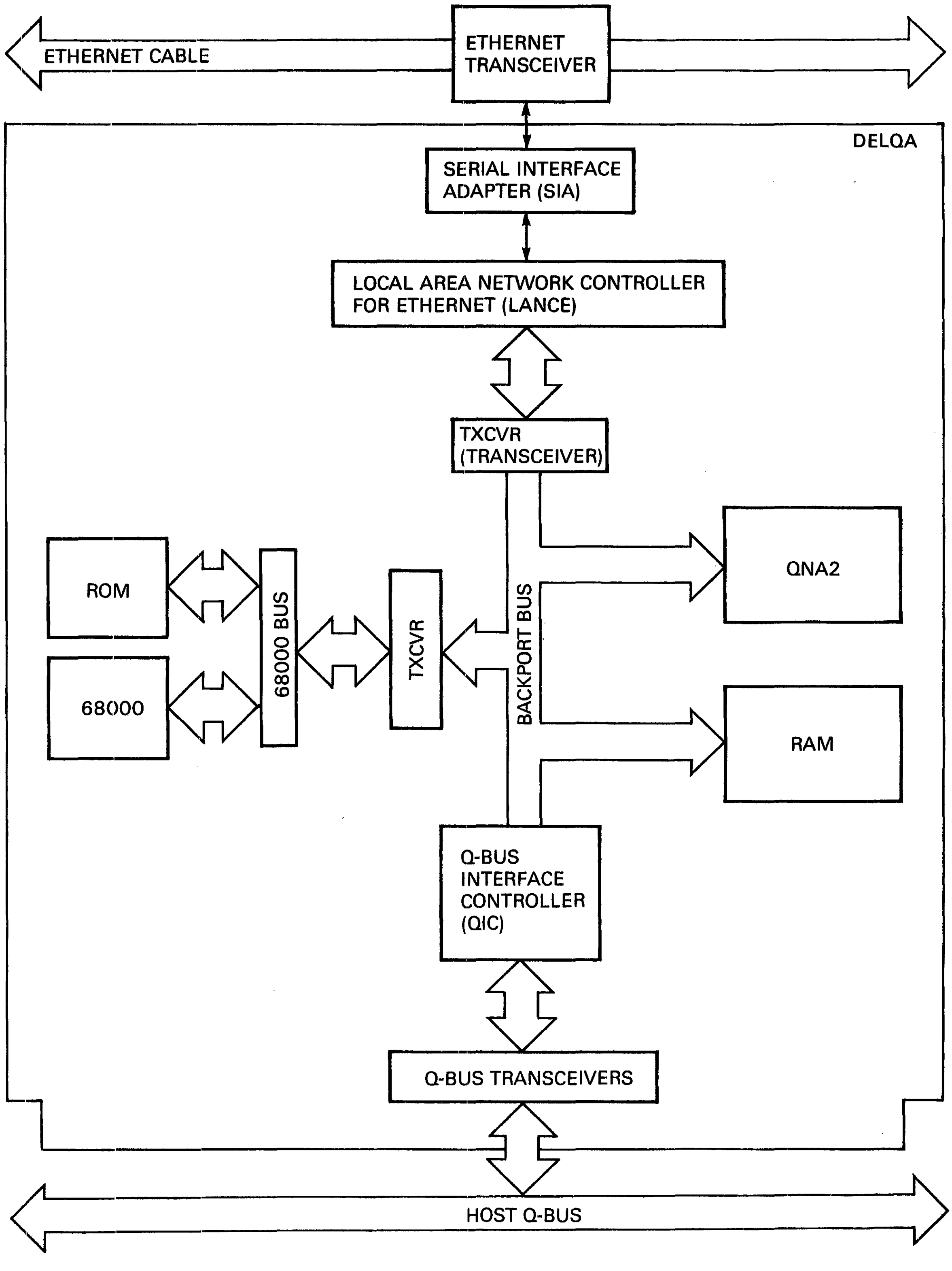


Figure 1–6 DELQA Functional Block Diagram

1.3.5 Ethernet Connection

A separate transceiver cable (order number BNE3X-*nn*), which must be ordered separately, connects the bulkhead to an Ethernet transceiver. This transceiver cable has a male 15-pin connector for fitting into the bulkhead.

NOTE

The signal connections of the DELQA cabinet kit comply with IEEE 802.3, and may be used with either the DELQA or DEQNA modules. DEQNA cabinet kits are not IEEE 802.3 compliant, and may only be used for Ethernet networks.

Table 1-4 lists the transceiver cable options. Table 1-5 lists the pin connections on the DELQA module and at the bulkhead.

Table 1-4 BNE3*x-nn* Transceiver Cable Options

Option	Material	Connector
BNE3A	PVC	Straight
BNE3B	PVC	Right angle
BNE3C	Teflon™	Straight
BNE3D	Teflon	Right angle
Part Number	**Length**	
-05/J2	16.4 ft (5 m)	
-10/J2	32.8 ft (10 m)	
-20/J2	65.6 ft (20 m)	
-40/J2	131.2 ft (40 m)	

™Teflon is a trademark of E.I. du Pont de Nemours & Company, Inc.

Table 1–5 DELQA Cabinet Kit Connections

	DELQA Signal Name	Module Plug	Bulkhead Connector	IEEE 802.3 Sheath
1	Power (+12 V)	1 (FUSE +)		
2	N.C.	KEY (Shield)		
3	Return (+12 V)	3		
4	Return (+12 V)	4	6	Voltage return
5	Ground	5	14	
6	Receive +	6	5	Receive +
7	Receive –	7	12	Receive –
8	Ground	8		
9	Ground	9		
10	Transmit +	10	3	Transmit +
11	Transmit –	11	10	Transmit –
12	Ground	12	4	Shield
13	Ground	13		
14	Collision +	14	2	Collision Presence +
15	Collision –	15	9	Collision Presence –
16	Ground	16		
17	N.C.	17		
18	N.C.	18	7	Control-Out A
19	N.C.	19	15	Control-Out B
20	FUSE OK	20 (FUSE –)	13	Voltage supply +12 V

1.4 SPECIFICATIONS

The DELQA module meets the Ethernet specification (Version 2.0) and is compatible with IEEE 802.3 *Specification for Carrier Sense with Multiple Access with Collision Detection.*

Table 1-6 gives the specifications of the DELQA module.

Table 1–6 DELQA Specifications

Physical	Dimension	Imperial	Metric	
	Height	5.19 inches	12.67 cm	
	Width	0.5 inches	1.27 cm	
	Length	8.94 inches	22.6 cm	
	Weight	12 ounces	0.34 kg	
Electrical	**Voltage**	**Tolerance**	**Typical current**	**Maximum current**
	+5.0 V	±5%	2.7 A	3.0 A
	+12.0 V	±5%	0.5 A	1.5 A
Q-bus loads	**AC**	**DC**		
	3.3	0.5		
Temperature	**Environment**	**Specification**		
	Storage	−40 to 66 degrees Celsius		
	Operation	5 to 50 degrees Celsius		

NOTE
BEFORE OPERATING with the DELQA module you must give it a reasonable time to stabilize in an environment within the operating range.

Airflow	Specification
	Airflow across the board must limit the outlet temperature to a maximum of 50 degrees Celsius, and the temperature rise across the board to 10 degrees Celsius.
	Under typical power dissipation, this can be achieved using a linear airflow of 1.2 meters/second.

NOTE
Do not subject any area of the board to a local ambient temperature above 70 degrees Celsius under any environmental conditions.

Relative Humidity	Environment	Specification
	Storage	10% to 95%, non-condensing

Table 1–6 (Cont.) DELQA Specifications

Relative Humidity	Environment	Specification
	Operation	10% to 95%, non-condensing
Altitude	**Environment**	**Specification**
	Storage	Maximum: 12.1 km (40,000 ft)
	Operation	Maximum: 2.4 km (8,000 ft)
	NOTE Derate the maximum operating temperature by 1.8 degrees Celsius for each 1000 meters (3281 feet) of altitude, unless constant cooling is provided.	

1.5 THE DELQA MODULE FUNCTIONAL DESCRIPTION

1.5.1 General Description

The DELQA module is a Q-bus communications option which enables higher-level software protocols, such as DECnet, to communicate over an Ethernet network.

The DELQA module conforms to the Ethernet Local Area Network Specification (Version 2.0), and is compatible with the IEEE Specification 802.3 for Local Area Networks.

The DELQA module transfers encapsulated data packets of 60 to 1514 bytes between buffers in host memory and an Ethernet transceiver. The DELQA module appends a 4-byte CRC to transmit packets, making the length of the packet on the Ethernet between 64 and 1518 bytes. The DELQA module strips the 4-byte CRC from received packets.

Transmit packets are transferred from host memory to buffer memory (shared RAM) on the DELQA module board, and from there on to the Ethernet. Received messages follow the same path in the opposite direction.

The DELQA module is programmed from the Q-bus using eight word addresses in the I/O page, and can perform block-mode DMA to and from Q-bus memory. In addition to providing an Ethernet interface, DELQA supports some functions of the Maintenance Operations Protocol (MOP).

1.5.2 Operating Modes

The DELQA module operates in one of two modes.

DELQA (Normal mode) supports:

- The provision of DEQNA compatibility when using DIGITAL software drivers
- Generation of MOP Remote console request system ID requests
- Processing of MOP Remote console system ID requests
- Processing of Ethernet channel loopback protocol requests
- Processing of MOP Remote console BOOT request
- Processing of IEEE 802.2 NULL LSAP XID message requests
- Processing of IEEE 802.2 NULL LSAP TEST message requests.
- Write the MOP Boot Password value
- Read the MOP Boot Password value

- Write the MOP System ID Parameters
- Read the MOP System ID Parameters
- Reset the MOP System ID Parameters
- Read the last MOP Remote Boot message received
- Read the datalink counters values
- Read and clear the values of the datalink counters
- Read the current DELQA Physical Ethernet Address
- On-Board self-test:
  - Powerup execution in DELQA mode
  - Host software request bit
  - Completion status with error report
- Boot/diagnostic code support.
- In *DEQNA-lock mode*, the DELQA module provides functional compatibility with DEQNA modules when using some non-DIGITAL software drivers.

1.5.3 Host Programming

The DELQA module handles block-mode DMA automatically, once the buffers and control information have been set up by host software.

Host software is responsible for:

- Initializing the communications data area in host memory
- Writing a setup packet with information to initialize the DELQA
- Handling interrupts generated by DELQA, particularly on completion of each transmit or receive DMA transfer.

1.5.4 Module Components

Figure 1-7 shows the major components of the DELQA module.

- Processor subsystem
- LANCE/SIA subsystem
- QIC
- QNA2
- Memory subsystem

The module comprises five major functional subsystems, plus interconnecting buses.

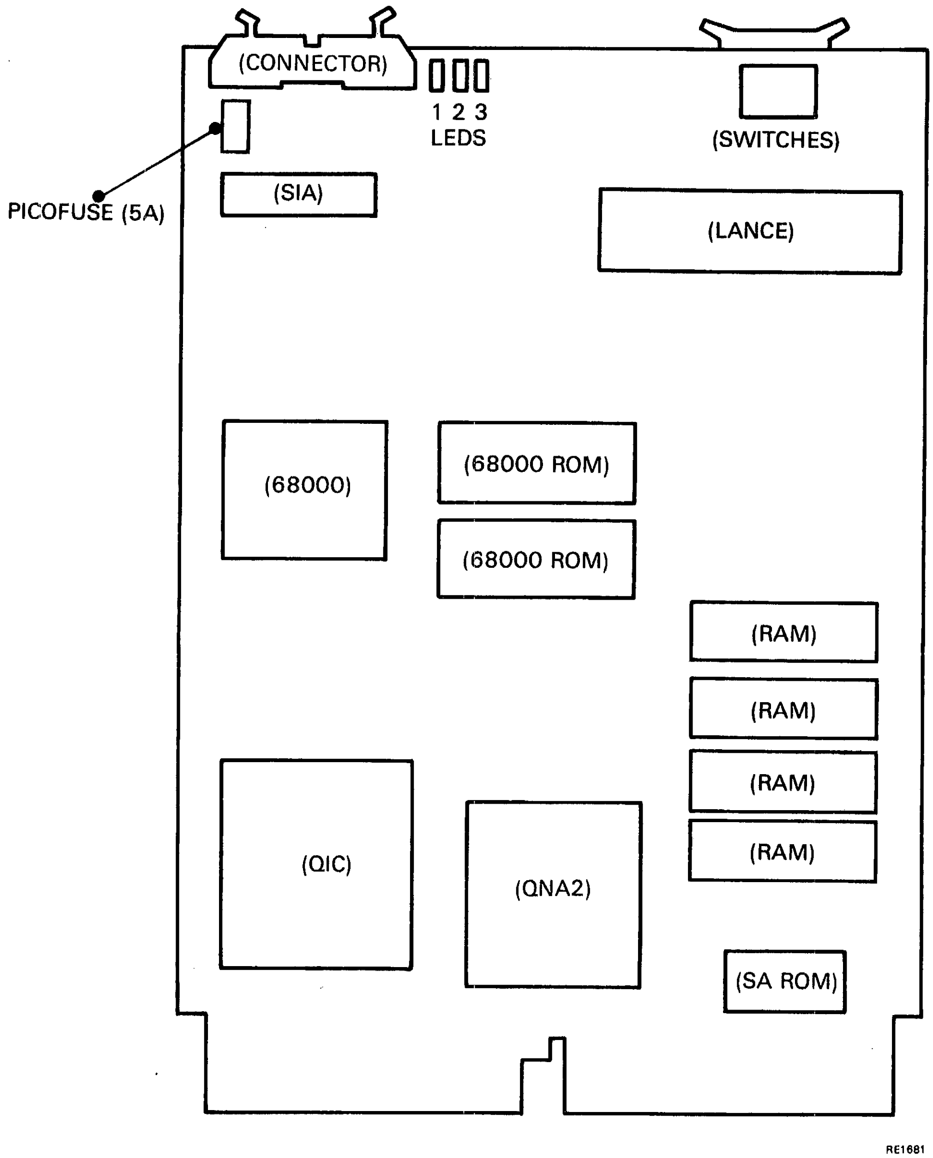


Figure 1–7 DELQA Module Board Layout

1.5.5 Processor Subsystem

The M68000 CPU plus firmware is responsible for:

- Self-test and initialization upon power-up and reset. Self-test verifies the integrity of each of the five functional subsystems, plus the integrity of the cable signal path from the DELQA module to the Ethernet transceiver.
- Managing all module configuration and control functions in accordance with the CSR and Setup packet parameters.
- Managing all data transfer functions, including initiation of DMA transfers to/from host memory by giving the appropriate instructions to the QIC.
- Network management support including MOP and IEEE 802.2 functions.

1.5.6 LANCE/SIA Subsystem

The LANCE/SIA chipset is responsible for:

- CSMA/CD network access, including collision handling
- Packet serialization/deserialization
- CRC generation/checking
- Framing
- Data encapsulation
- Buffer management
- On-board DMA
- Error reporting
- Address detection can be programmed to detect a specific physical address, or logical addresses.

1.5.7 QIC Subsystem

The QIC is a general-purpose Q-bus interface controller which provides:

- Q-bus slave control logic
- I/O page addressing
- DMA arbitration and control:
  - — On Q-bus side, both control-DMA (four words mixed writes and reads), and data DMA, 22-bit addressing
  - — On backport side, 16-bit DMA address register/counter
- Interrupt control
- Q-Bus NXM timeout detection
- The ability to initiate host reboot.

1.5.8 QNA2 Subsystem

The QNA2 arbitrates requests (R/W access, and DMA) for the backport bus, which is shared between the QIC, 68000, and LANCE. The QNA2 implements the following control functions.

- Arbitration of access rights for the QIC, 68000 microprocessor and LANCE
- Read/Write control logic for memory accesses
- Read/Write control logic for QIC registers, LANCE registers
- LANCE and QIC control
- 68000 control

The QNA2 also contains the module's Control and Status Register (CSR).

1.5.9 Memory Subsystem

The memory subsystem contains:

- 32K bytes of static RAM for packet buffering, and scratch area for 68000
- 32K bytes of Firmware ROM, which also includes the self-test diagnostic code. The firmware ROM also contains 4.0K bytes of PDP-11 boot/diagnostic code for execution by the host system (MicroVAX systems provide equivalent boot/diagnostic code in their own host system ROM.
- A unique Station Address (SA) ROM for each DELQA module

1.5.10 Q-bus Interfaces

The DELQA module is connected directly to the Q-bus backplane. The interface consists of slave and master logic.

- *Slave logic* gives the host access to the DELQA port registers.
- *Master logic* performs the QIC functions to:
  - — Address host memory
  - — Transfer data
  - — Fetch descriptors
  - — Store status
  - — Increment addresses and word counts.

1.5.11 Q-bus Timers

The DELQA module has two on-board timers that operate automatically.

- The *holdoff timer* causes the DELQA module to wait for approximately 6.4 microseconds before requesting the bus again, thus allowing other DMA devices to acquire the bus.
- The *bus time-out timer* causes the bus cycle to abort with a Nonexistent Memory (NXM) interrupt if the bus slave fails to respond within approximately 10 microseconds.

1.6 MODULE INTEGRITY

The DELQA module has built-in self-test routines, provides support for Maintenance Operation Protocol (MOP) network functions, and is supported by host system diagnostics.

1.6.1 Self-Test

In Normal mode, the DELQA module executes a comprehensive self-test on powerup. This takes approximately five seconds.

The firmware ROM on the DELQA module contains 4.0K bytes of PDP-11 boot/diagnostic code. If the module is controlled by a PDP-11 host, the host can execute this code in order to increase fault coverage. This enables the DELQA module to determine that the DELQA module is operating correctly, before it attempts to access the Ethernet.

For MicroVAX systems a similar test is found on the host CPU boot ROM.

1.6.2 Maintenance Operations Protocol (MOP)

In Normal mode, the DELQA module is capable of implementing the following MOP functions (a subset of DECnet operations) on behalf of a Remote Console (another node on the Ethernet) without host intervention. These are:

- Loopback to network
- Transmit system ID
- Respond to request system ID
- Respond to remote trigger request

A Trigger Instruction (remote console request to reboot) can be executed if the local host system contains the appropriate BOOT ROM for loading the boot code from the DELQA module.

1.6.3 IEEE 802.2 Link-layer Service Access Point (LSAP) Messages

In Normal mode, the DELQA module processes the following Link-layer Service Access Point (LSAP) messages when used on an IEEE 802.3 local area network. These are:

- NULL TEST (Loopback)
- NULL XID (Transmit ID).

1.6.4 Host System Diagnostics

Host systems provide diagnostic tests for Q-bus modules and for testing communications modules as part of a network.

The module tests are processor-specific. PDP-11 hosts support the Field Functional Diagnostic (ZQNA??) and the DEC/X11 Exerciser. MicroVAX hosts provide the MicroVAX Diagnostic Monitor (MDM).

The network tests are:

- DECnet Network Control Program (NCP)
- Network Interconnect Exerciser (NIE).

NOTE

The current Ethernet diagnostics are compatible with both the DELQA module and the DEQNA module. Early versions only support the DEQNA module and cannot be used with the DELQA module.

CHAPTER 2
INSTALLATION

2.1 SCOPE

This chapter describes how to install a DELQA module in a MicroVAX or MicroPDP-11 system. The sections are:

Section 2.2	Unpacking and Inspection
Section 2.3	Checking Installation Requirements
Section 2.4	Configuration and Installing the Module
Section 2.5	Post-Installation Checks
Section 2.6	Diagnostic Acceptance Tests

In each section the sequence of actions is numbered. Do the procedures in the order described.

WARNING
The procedures described in this chapter involve the removal of the system covers, and should be performed only by trained personnel.

ADVARSEL!
Ifølge de procedurer, som er beskrevet i dette kapitel, skal systemets beskyttelsesplader fjernes; dette bør kun udføres af personer der ved hvordan dette skal gøres.

WAARSCHUWING
Bij de procedures die in dit hoofdstuk worden beschreven dienen bepaalde delen van de systeemomhulling te worden verwijderd; dit mag uitsluitend worden gedaan door opgeleid personeel.

VAROITUS!
Tässä luvussa kuvatut toimenpiteet liittyvät järjestelmän suojakansien irrottamiseen. Ainoastaan koulutettu henkilökunta saa suorittaa nämä toimenpiteet.

AVIS!

Ce chapitre décrit les interventions qui demandent que les couvercles extérieurs des appareils soient enlevés. Ces travaux devraient être mis en main uniquement par des techniciens expérimentés.

VORSICHT!

Bei der Ausfuhrung der in diesem Kapitel beschriebenen Anweisungen mussen die Systemabdeckungen entfernt werden. Dies sollte nur von geschultem Personal ausgefuhrt werden.

אזהרה

הפעולות המתוארות בפרק זה, כרוכות בהסרת המכסים
של המערכת ויבוצעו אך ורק על ידי אדם מוסמך.

ATTENZIONE

La procedura descritta in questo capitolo comporta la rimozione delle coperture e deve essere eseguita solo da personale specializzato.

注 意

本章では，本体カバーの取り外し等について
述べてあります。作業は，必ず専門の担当者
によっておこなって下さい。

ADVARSEL

I dette kapitlet beskrives bl. a. hvordan man fjerner dekslene rundt systemet. Dette arbeidet må bare utføres av fagfolk.

AVISO

Os procedimentos descritos neste capítulo respeitam à forma como se retiram as protecções do sistema. Dada a sua especificidade, recomendamos que seja executado por pessoal especializado.

!ATENCION!
Los procedimientos descritos en este capitulo incluyen el desmontaje de las cubiertas del sistema y debe ser realizado solamente por personal entrenado.

VARNING
I detta kapitel beskrivs hur systemkaapan tas bort. Detta faar endast utfoeras av utbildad personal.

2.2 UNPACKING AND INSPECTION

NOTE
Static electricity can damage the DELQA module. Always wear an anti-static wrist strap connected to an active ground and use a grounded work surface whenever you work on a system with covers removed, or handle system modules.

When unpacking the DELQA kit, please check the contents of the shipping containers for damaged or missing parts.

1. Before opening the shipping containers, look for external damage such as dents, holes, or crushed corners.
2. Check the contents of each container, using the shipping list.
3. When unpacking the individual packages from the containers, inspect every DELQA part for shipping damage. Check carefully for cracks, breaks, and loose components. If you find that an item is damaged, do not open its package or you may void the warranty.
4. Report any damage or shortages to the shipper. If reshipment is likely, retain the shipping containers and packing materials.

Table 2-1 lists the parts that must be ordered separately with each DELQA.

Table 2–1 DELQA Installation Parts List

Base Option	Description	
DELQA-M	DELQA Module (M7516) DELQA User Guide (EK-DELQA-UG)	
Cabinet Kit	**Cabinet**	**Cable Length**
CK-DELQA-YB	BA23	12 inches (30.5 cm)
CK-DELQA-YA	BA123	21 inches (53.6 cm)
CK-DELQA-YF	H9642	36 inches (91.5 cm)
	Each kit is supplied with a module-to-bulkhead cable of the appropriate length, and a 15-pin bulkhead loopback connector (12-22196-02).	

2.3 CHECKING INSTALLATION REQUIREMENTS

This section describes what is required in the host system before you install the DELQA module. Do the procedures in the numbered order.

1. Verify that the correct host BOOT ROMs are installed if a down line load boot from the Ethernet is required in the host CPU.

   In a PDP-11 system, the host CPU must be able to load the extended bootstrap code from the DELQA BOOT ROM on the CPU module. An appropriate BOOT ROM is also required to enable the DELQA to initiate a system reboot when a Remote Boot request is received over the network.

2. Check Backplane Expansion Space/Power Requirements.

   The DELQA requires one dual Q-bus module slot.

   There are several factors to consider when configuring modules on the Q-bus. These include:

   — Backplane and I/O expansion space

   — Power requirements

   — Allocation of module base addresses

   — The physical priority of each module.

Check the power limits for the total system.

The total current drawn and the total power used by all modules must not exceed the bus and power loading limits for the system.

Refer to the host system manual for details.

The table below shows how much a DELQA module loads the Q-bus, and the typical and maximum current and power requirements of the module.

The module has a +5 volt power supply requirement. The module routes the +12 volt power supply requirement to the H4xxx transceiver.

Q-bus Load		Current				Power	
AC	DC	Typical		Maximum		Typical	Maximum
3.3	0.5	+5 V	+12 V	+5 V	+12 V	+5 V	+12 V
		2.7 A	0.5 A	3.0 A	1.5 A	19.5 W	33.0 W

NOTE
At powerup, the surge current into the transceiver is sufficiently high to current-limit some power supplies.

2.3.1 Fuses

NOTE
A 1.5 A fuse of the correct type must always be fitted in the bulkhead module.

Two fuses provide protection for the DELQA module and associated equipment:

- A 1.5 A/250 V slo-blo™ 1.25 inch by 0.25 inch glass fuse (order number 90-07213) protects the transceiver and its associated external wiring. The fuse may be replaced with another fuse of the same type, a Littlefuse™ type 31301.5, a BEL FUSE™ type 3SB1.5, or an equivalent. The 1.25 inch (3.8 cm) fuse holder (order number 12-22255-03) is located in the bulkhead assembly (not on the DELQA board).
- A 5.0 A/125 V axial lead picofuse (order number 12-05747-00) protects the DELQA module and internal wiring. The picofuse is fitted on the DELQA board near the bulkhead cable connector; it looks like a resistor and is soldered to the board in the same way. It should be replaced only by trained personnel.

2.3.2 Backplane Positioning

The DELQA is a dual-height module and may be positioned in either a Q/CD or Q/Q backplane slot. If it is installed in a Q/Q slot, with no other adjacent module, an M9047 grant continuity card is required in the vacant slot.

2.4 INSTALLING THE MODULE

This section describes the procedures for preparing the host and DELQA for installation. Do the steps in the order described. The DELQA module is configured as a DMA device in the same way as a DEQNA module. The first DEQNA/DELQA vector in the host system is fixed at 120. Subsequent DEQNA/DELQA modules are assigned a floating vector with a rank of 47. You must configure them at system start-up using the auto-configuration routines for floating vectors. These vectors are written by host software.

Table 2-2 shows the reserved addresses and vectors. Refer to Appendix A and the host system manual for further information.

Table 2–2 Module Address and Vectors

Base Address	Vector Address	Slot	Module
17774440	120 (fixed)	DELQA 1	DELQA or DEQNA
17774460	Floating (rank 47)	DELQA 2	DELQA or DEQNA

™ slo-blo is a trademark of S.B. Fuses
™ Littlefuse is a trademark of Littlefuse Inc, Illinois U.S.A.
™ BEL FUSE is a trademark of Belfuse Inc, New Jersey U.S.A.

2.4.1 Switch Settings

The switches must be set for compatibility with the host system configuration.

NOTE
Static electricity can damage the DELQA module. Always wear an anti-static wrist strap connected to an active ground and use a grounded work surface whenever you work on a system with covers removed, or handle a DELQA module.

The DELQA contains five switches, S1 to S5; however, only three of these switches are used to configure the DELQA. The remaining switches are reserved for DIGITAL.

S1 - UNIT SELECT Switch

This switch selects the module's I/O page base address. The following table decribes how this switch selects the base Q-bus address of the DELQA.

(S1) Position	DELQA Base Address
Closed	17774440 (shipped default)
Open	17774460

S2 - RESERVED Switch

S3 and S4 - MODE (S3) AND OPTION (S4) Switches

S5 - Reserved

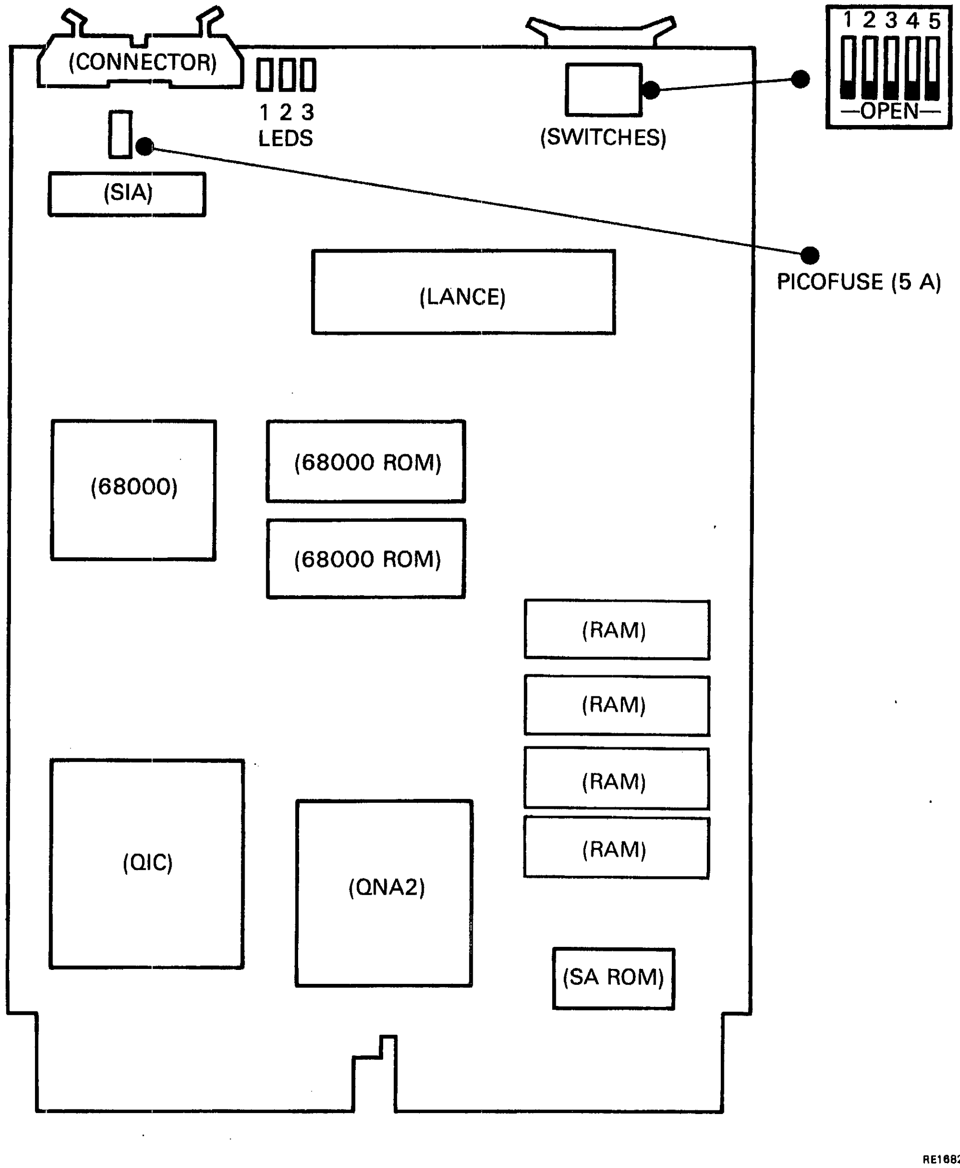


Figure 2–1 DELQA Switches in Default Position and LEDs

The mode switch defines two possible modes of operation for the DELQA. The preferred mode is the "Normal mode" which indicates that the DELQA is operating as a DELQA. All current DIGITAL software for the DEQNA may be used with confidence for the DELQA when the DELQA is switched to operate in Normal mode. "DEQNA-lock mode" should only be required for use with some non-DIGITAL software drivers to achieve compatibility with DEQNA programming features. The sanity timer enabling, on power-up, is controlled by the option switch when the DELQA is operated in DEQNA-lock mode. The following table defines the functions which may be selected through various combinations of S3 (mode switch) and S4 (option switch).

SWITCH DEFINITIONS

Mode	(S3) Position	Option	(S4) Position
Normal	Closed	Remote Boot DISABLED	Closed
		Remote Boot ENABLED	Open
DEQNA-lock	Open	Sanity Timer DISABLED	Closed
		Sanity Timer ENABLED	Open

NOTE
With S3 and S4 OPEN, a Host boot will occur every 4 minutes if the sanity timer is not reset by the host software.

NOTE
Please see Section 1.5.2 for a summary definition of DEQNA-lock mode and Normal mode.

S5 - RESERVED Switch

DEFAULT SWITCH SETTINGS (ALL CLOSED)

Switch	Position	Definition
Switch 1:	Closed	17774440 (Base Address)
Switch 2:	Closed	Reserved
Switch 3:	Closed	Normal mode
Switch 4:	Closed	MOP Remote Boot Disabled
Switch 5:	Closed	Reserved

2.4.2 Ethernet Address

The unique physical address of the DELQA module within Ethernet is stored in the Station Address ROM on the DELQA board. A record of this address is printed on a sticker on the handle of the board.

This address should be given to the network manager for configuration.

If it is essential to replace an existing DELQA module in a network while retaining the same physical Ethernet address, it is possible to swap the Station Address ROM to a new board. (Be sure to swap the stickers at the same time.) However, the risk of damage to both board and ROM means that this is not a recommended procedure.

If the Ethernet address is changed, the only software change required involves updating the physical Ethernet address of the node at those host systems that use the node for downline loading over the Ethernet.

2.4.3 Inserting in System Backplane Slot

This section describes the procedure for fitting the module board into the BA23 system backplane. If you have a BA123 cabinet, refer to your *System User's Guide* for the physical details of accessing the system backplane and installing the module.

Figures 2-2 and 2-3 show how the module fits into the backplane and is connected to the bulkhead by the cabinet kit.

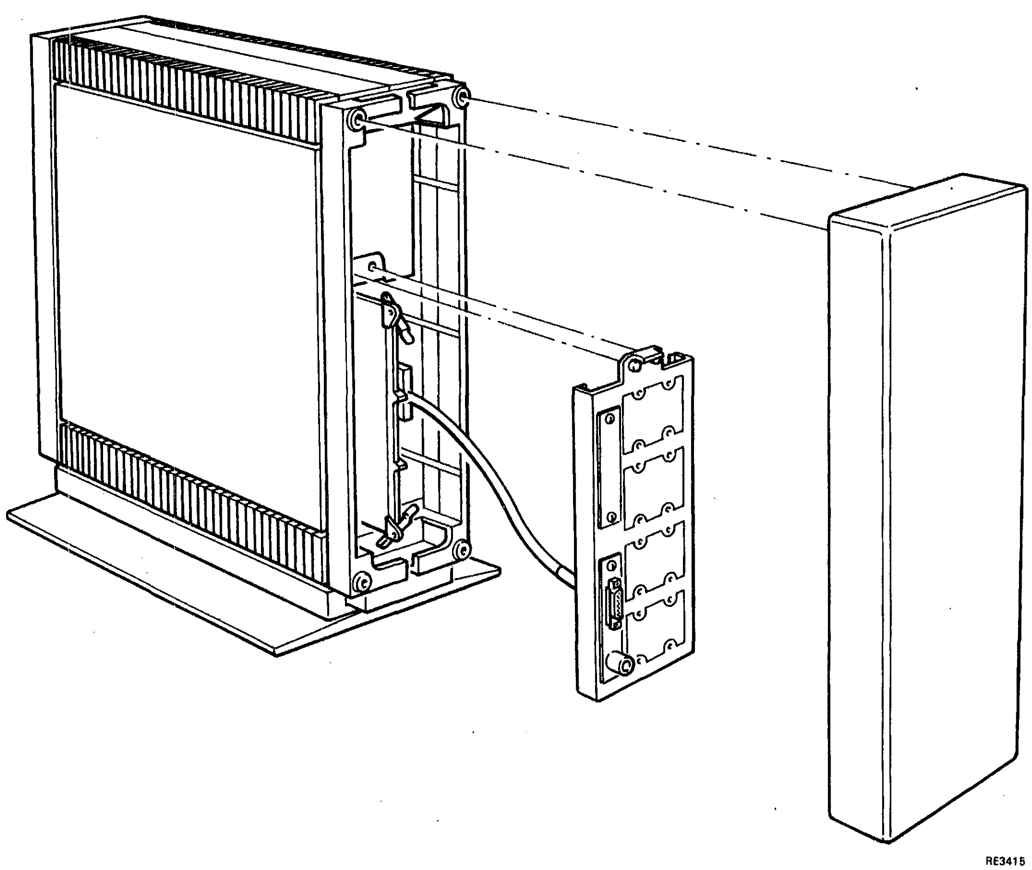


Figure 2–2 Rear Panel, Bulkhead, Blanking Panel, and Modules

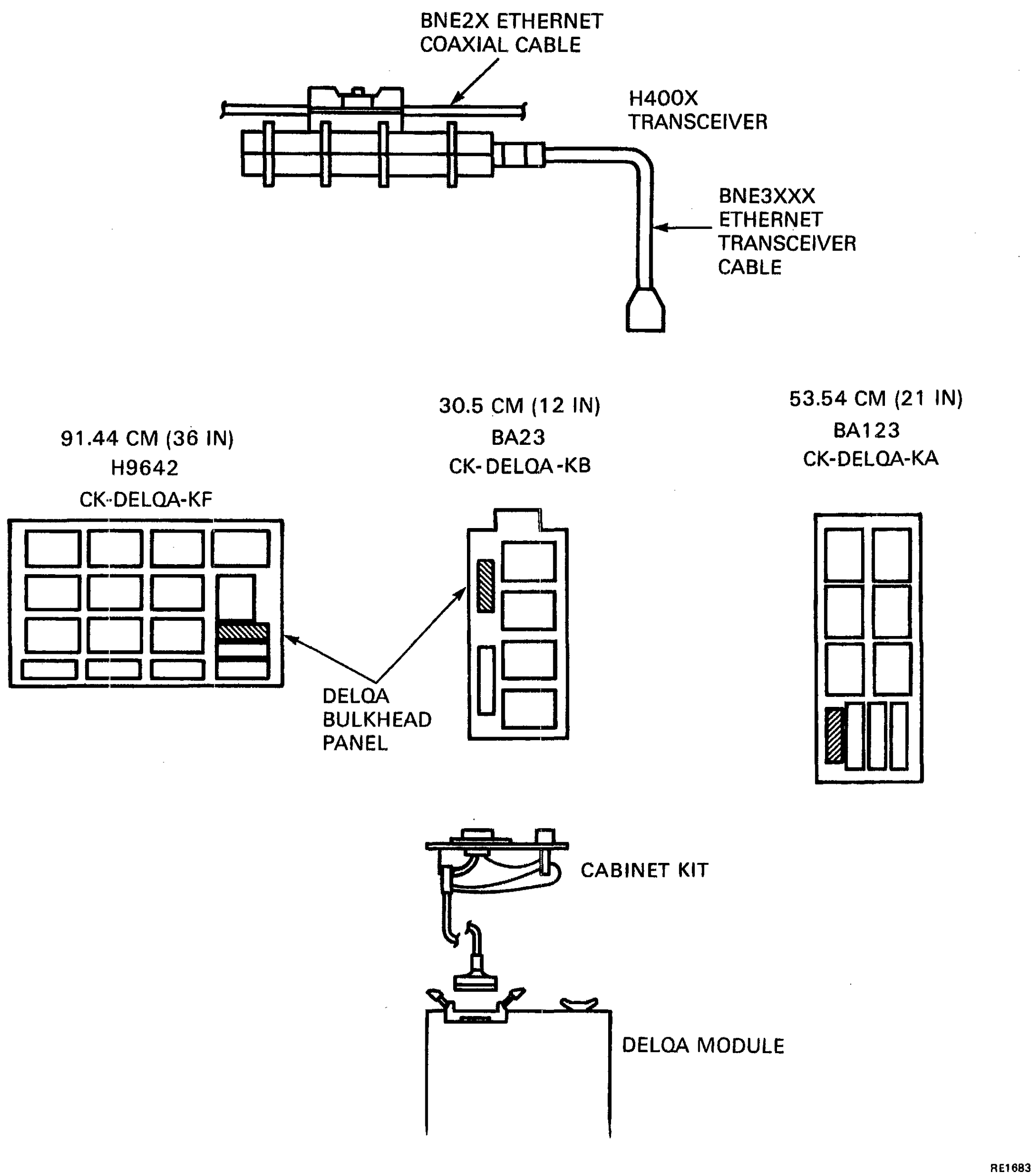


Figure 2–3 DELQA Cabinet Kits

1. Turn the system power off, and unplug the AC power line from the wall socket.
2. Remove the rear plastic cover by holding each end and pulling the cover towards you.
3. Open the bulkhead panel (also called: the system I/O panel; or distribution panel) by loosening the two screws at the end opposite the hinge, and swinging it open.
4. Relocate modules and grant continuity cards as necessary to free the appropriate backplane slot(s) for the DELQA module(s).
5. Slide the DELQA module(s) into the appropriate backplane card slot(s) in the Q-bus backplane, using the card guides to locate the module as it is pushed home to connect with the system backplane.

2.4.4 Cabinet Kit

Figure 2-2 (Rear Panel, Bulkhead, Blanking Panel, and Modules) shows how the module is connected to the bulkhead by a cabinet kit and how the transceiver cable is attached to the other side of the bulkhead assembly. The center section of Figure 2-3 indicates the appropriate panel location for the DELQA in each cabinet type.

6. Remove the blanking panel from the appropriate location in the bulkhead panel by unscrewing the two retaining screws. Save the two screws.
7. Insert the cabinet kit into the panel location, and secure the assembly using the two screws saved from the blanking panel.
8. Feed the cable from the cabinet kit to the module, and connect it to the module as indicated by the label **THIS SIDE UP**.
9. Connect a loopback connector to the bulkhead connector.
10. Turn the system power on and check for correct self-test LED indication (see Table 2-3).

If there is a problem in starting the host system, refer to the maintenance section of this guide.

Table 2–3 Module LED Sequences

			Normal Mode — Power-Up Sequence
LED1	**LED2**	**LED3**	**Definition**
ON	ON	ON	
ON	–	–	Executing internal logic self-test
ON	ON	–	Self-test executing external loopback test
ON	ON	ON	Ready (to execute citizenship tests and/or normal functions) or module self-test
			DEQNA-lock Mode — Power-Up Sequence
ON	ON	ON	All LEDs come on and stay on

If an Ethernet boot is initiated in either Normal or DEQNA-lock mode, or if software initiates a citizenship test, the following additional LED states are used.

LED1	LED2	LED3	Definition
ON	ON	ON	Ready (to execute citizenship tests and/or normal functions) or module self-test
–	ON	ON	Executing citizenship tests
–	–	ON	Internal loopback citizenship tests completed successfully
–	–	–	External loopback citizenship tests completed successfully

These sequences of LEDs should take less than 10 seconds. If the LEDs flash, this indicator error is discussed in Appendix D.

NOTE
LED states all ON, or all OFF at the end of the self-test indicate successful completion, depending on the boot mode.

11. Measure the power supply voltages for the system at the +5 V and +12 V testpoints. They should be within the tolerances defined in the host system manual.
12. Turn the system power off, and unplug the ac power line from the wall socket.
13. Disconnect the loopback connector.
14. Ensure that all cables are clear of panels and doors, and cannot be trapped or damaged by sharp edges.
15. Close the bulkhead panel and tighten the two screws at the end of the panel opposite the hinge.

2.5 DIAGNOSTIC ACCEPTANCE TESTS

This section describes customer-runnable diagnostics.

For further details of service-mode diagnostics, refer to Appendix B.

A MicroVAX II host and a MicroPDP-11 host have different diagnostic tests.

For further details, refer to Appendix B, and to the appropriate host system manual.

NOTE
Both MicroPDP-11 and MicroVAX II diagnostics distinguish DELQA modules from DEQNA modules in order to run tests specific to each type of module. MicroVAX I diagnostics do not support DELQA. The current Ethernet diagnostics are compatible with both DELQA and DEQNA. Early versions only support the DEQNA and cannot be used with the DELQA.

2.5.1 Installation Tests on MicroPDP-11 Systems

To verify that the MicroPDP-11 system and the DELQA module are functioning correctly:

1. Switch on the system
2. Boot the MicroPDP-11 Customer Diagnostic media. Refer to your MicroPDP-11 System Manual for further information.
3. Type I at the main menu to allow the diagnostics to identify the new module, and add it to the configuration file.

NOTE
Look at the list of devices displayed, and make sure that the new module is included. If it is not included, repeat the installation sequence, and make sure that the module switches have been set correctly.

4. Type T at the main menu to run the system tests. These should complete without error; if an error occurs, call DIGITAL Field Service.

A MicroPDP-11 Maintenance Kit is available, and may be ordered from your local DIGITAL office. This kit allows trained personnel to run individual diagnostic programs under the XXDP+ diagnostic monitor, and to configure and run DECX11 system test programs. The XXDP+ functional diagnostic is ZQNA??.BIN. See Appendix B for further details of ZQNA??.BIN.

2.5.2 Testing in MicroVAX II Systems

1. Switch on the system power.
2. Boot the MicroVAX Maintenance System media. Refer to your MicroVAX II System Manual for further information.
3. Type 2 at the main menu to show the system configuration and devices.

NOTE
Look at the list of devices displayed, and make sure that the new module is included. If it is not included, repeat the installation sequence, and make sure that the module switches have been set correctly.

4. Type 1 at the main menu to run the system tests. These should complete without error; if an error occurs, call DIGITAL Field Service.

2.6 CONNECTION TO ETHERNET

When diagnostics have shown an error-free system, connect the DELQA to an H4xxx transceiver or DELNI with a BNE3xxx cable.

CHAPTER 3
PROGRAMMING

3.1 SCOPE

This chapter contains information about programming the DELQA module. The sections are as follows.

Section 3.2	Overview
Section 3.3	Register Definitions
Section 3.4	Buffer Descriptor Definitions
Section 3.5	Data Transfer
Section 3.6	Configuration and Control
Section 3.7	Maintenance Operations Protocol (MOP): Module Support

3.2 OVERVIEW

The host software must provide routines to handle three basic types of operation on the module.

- Module initialization
- Configuration and control operations (addressing capabilities, loopback modes, and so on.)
- Data transfer (transmit/receive) operations

This section provides the definitions and procedures which the host software uses to implement these three basic types of operations with the DELQA.

As an introduction to these operations the following section provides an overview of the data transfer mechanism between the host and the DELQA.

Communication between the host and the DELQA is accomplished through buffer descriptors organized as list structures in host memory.

There is one descriptor associated with each data buffer, and there are separate descriptor lists for transmit and receive, Tx BDL and Rx BDL.

The information in each descriptor includes:

- The address of the data buffer
- The length of the data buffer
- Status information associated with the buffer.

The location of the descriptor lists is specified by the host writing the list address to the Tx BDL or Rx BDL I/O page register.

The transmit/receive protocol is described below.

3.2.1 Transmit—Host to Ethernet Data Transmission

The host builds a list of one or more transmit descriptors, and then writes the address of the start of the list to the DELQA Tx BDL register. Note that the host must always terminate the list with an invalid entry.

In response to the Tx BDL address register write, the DELQA takes the following action.

1. Starting at the address provided by the host, the DELQA reads the descriptor into RAM (by performing a Write Read Read Read (WRRR) control-DMA, where the Write operation is to set all the bits of word 1 to 1). All the bits of word 1 are reserved for the DELQA.
2. Bit 15 of the second descriptor word (the Valid bit) is examined to check that the descriptor is a valid one. If it is invalid, the DELQA sets XL in the Control and Status Register (CSR), and takes no further action.
3. If the descriptor is valid, and the DELQA has a transmit data buffer available, then a data-DMA transaction is performed to copy host data into the DELQA RAM, using the host buffer address and byte count supplied in the descriptor.
4. The next host descriptor is read, and steps 2 and 3 are repeated. If the descriptor is invalid, then the DELQA assumes it has reached the end of the list, sets XL in the CSR, and takes no further action.
5. When the packet has been transmitted on to the Ethernet, the DELQA writes the transmit status back to the host using a control-DMA WW (write-write) operation.
6. The DELQA sets the XI-bit in the CSR and then interrupts the host to indicate completion of transmission.

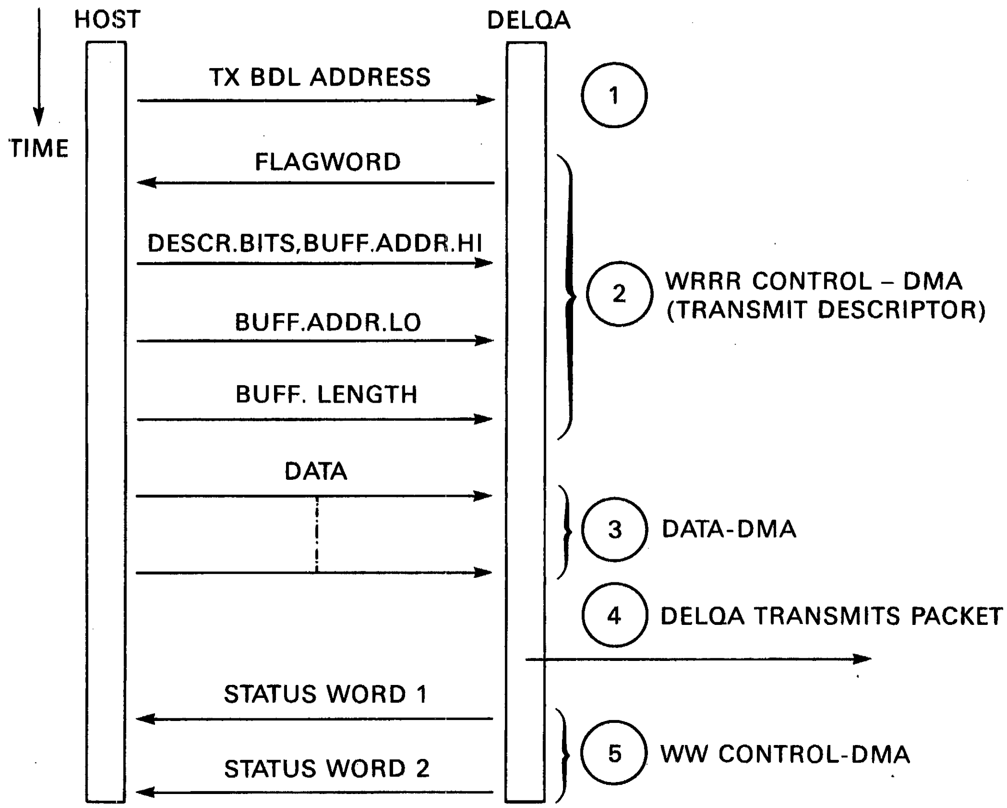
The host should respond to this interrupt by reading the CSR to determine the reason for the interrupt. The host should then clear the reason bit, by writing a 1 to it. See Figure 3-1.

3.2.2 Receive--Data Reception from Ethernet to Host

In order to receive any data from the Ethernet, the host must build a list of one or more receive descriptors, and then write the address of the start of the list to the DELQA Rx BDL register. The host must always terminate the list with an invalid entry.

In response to an Rx BDL address register write, the DELQA does the following.

1. Starting at the address provided by the host, the DELQA reads the descriptor into RAM (by performing a Write Read Read Read (WRRR) control-DMA, where the Write operation is to set all the bits of word 1 to 1). All the bits of word 1 are reserved for the DELQA.
2. Bit 15 of the second descriptor word (the Valid bit) is examined to check that the descriptor is a valid one. If it is invalid, the DELQA sets RL (Receive List Invalid) in the CSR, and takes no further action.
3. If the descriptor is valid, and the DELQA has received any packets from the network, then a data-DMA transaction is performed to copy the packet from the DELQA RAM to the host. Data is copied using the host address and byte count supplied in the descriptor.
4. The next host descriptor is read, and steps 2 and 3 are repeated. If the descriptor is invalid, then the DELQA assumes it has reached the end of the list, sets RL in CSR, and takes no further action.
5. When the data-DMA operation to the host has completed, the DELQA writes the status associated with the received packet back to the host using a control-DMA Write-Write (WW) operation.
6. The DELQA sets the RI-bit in the CSR, and interrupts the host to indicate that a receive operation has completed.



1. HOST WRITES TX BDL ADDRESS TO DELQA
2. DELQA FETCHES HOST DESCRIPTOR (WRITE-READ-READ-READ CONTROL-DMA)
3. DELQA DMAs DATA BUFFER FROM HOST
4. DELQA TRANSMITS PACKET
5. DELQA WRITES TRANSMIT STATUS TO HOST (WRITE-WRITE CONTROL-DMA)
6. THE DELQA WILL CONTINUE TO FETCH AND PROCESS HOST DESCRIPTORS UNTIL IT FINDS A DESCRIPTOR WITH THE VALID BIT CLEAR

RE4622

Figure 3–1 Transmit Sequence (No Chaining)

3.3 REGISTER DEFINITIONS

3.3.1 Control and Status Transfers

This section describes how the host uses the DELQA's hardware registers.

3.3.2 Control and Status Registers

Each DELQA is assigned a block of eight word-locations in the Q-bus I/O page. These locations are word-addressable only, and the DELQA acts as a bus slave to support access by the host software to the DELQA registers.

The accessible registers are:

- **Control and Status Register (CSR)**

  This is a one-word read/write control register.

- **Vector Address Register (VAR)**

  This is a one-word read/write control register.

- **Receive Buffer Descriptor List (BDL) Start Address Register**

  Transmit Buffer Descriptor List (BDL) Start Address Register

  These are two-word write-only registers that are maintained by the host software, and point to the Buffer Descriptor Lists (BDLs) in host memory.

- **Station Address ROM**

  This is a set of six read-only memory bytes (the lower bytes of the first six words in the DELQA space). The Station Address (SA) ROM contains the 48-bit physical address of the DELQA module in the Ethernet LAN.

Four of the I/O page addresses are write-only data registers, used to pass the start addresses of the BDLs for transmit and receive buffers. Two are read/write control registers.

The DELQA can act as bus master to the Q-bus, in order to implement DMA transfers (either block-mode or non-block-mode) between RAM on-board the DELQA and BDLs in host memory.

The registers are assigned to fixed blocks, so that more than one DELQA module can be mixed with other DELQA or DEQNA modules in the same host configuration, as shown in Figure 3-2 (Host I/O Page Map) and listed in Table 3-1.

Table 3–1 DELQA Unit I/O Base Addresses

S1	Base Address	Unit	Module
CLOSED	17774440	DELQA 1	DELQA or DEQNA
OPEN	17774460	DELQA 2	DELQA or DEQNA

3.3.2.1 Control and Status Register (CSR) Definitions

The Control and Status Register (CSR) is a read/write register that contains control and status information for the DELQA.

Figure 3-3 shows the CSR bits, and Table 3-2 summarizes the bit definitions in Normal mode. More complete bit definitions follow this table.

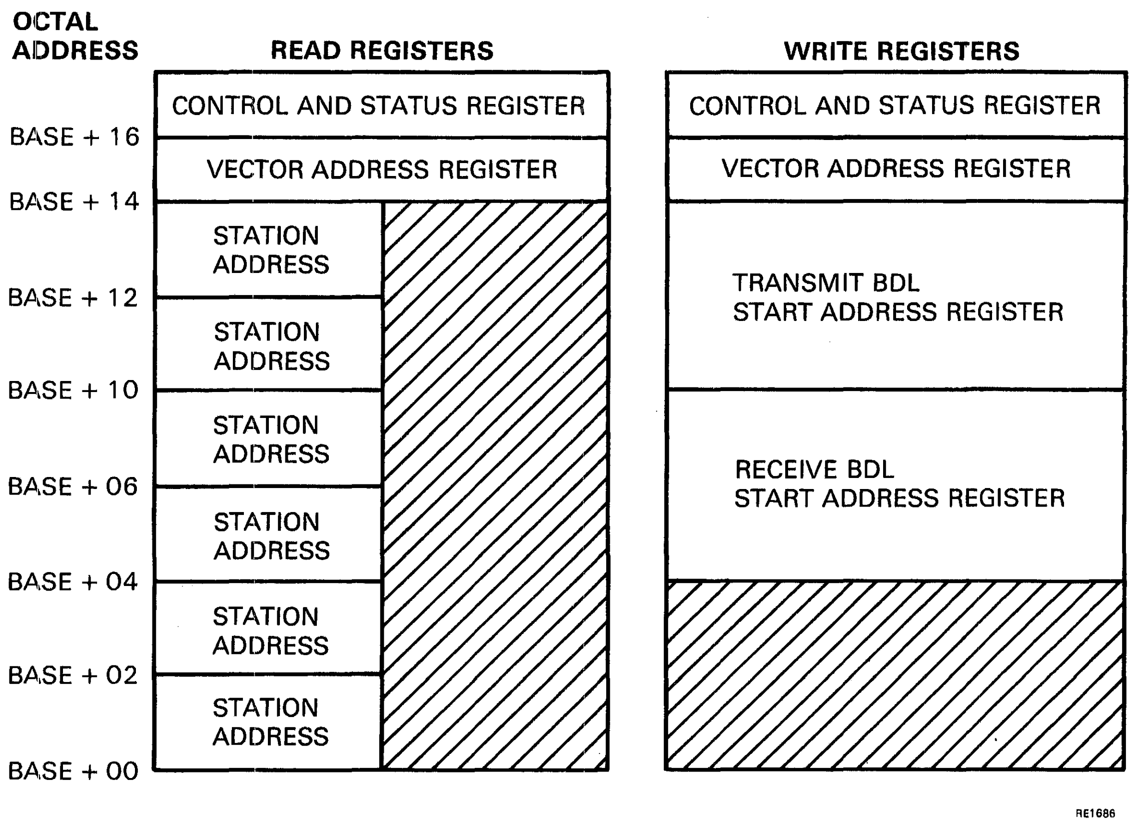


Figure 3–2 Host I/O Page Map

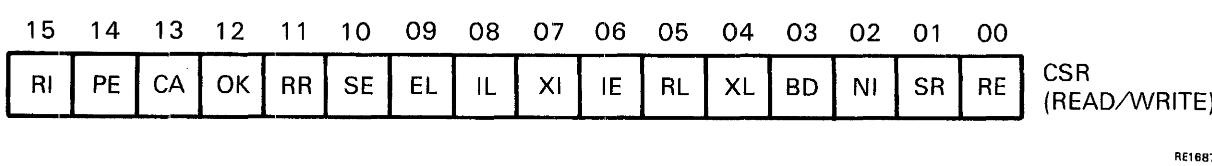


Figure 3–3 Control and Status Register (CSR)

Table 3–2 Control and Status Register (CSR) Normal Mode Usage

Bit		Description	State after Software Reset or Self-test	State after Powerup Reset
CSR00	R/W	Receiver Enable	0 (Clear)	0 (Clear) RE
CSR01	R/W	Software Reset	0 (Clear)	0 (Clear) SR
CSR02	R	Nonexistent-Memory Timeout Interrupt	0 (Clear)	0 (Clear) NXM
CSR03	R/W	Boot/Diagnostic ROM Load	0 (Clear)	0 (Clear) BD
CSR04	R	Transmit List Invalid/Empty	1 (Set)	1 (Set) XL
CSR05	R	Receive List Invalid/Empty	1 (Set)	1 (Set) RL
CSR06	R/W	Interrupt Enable	0 (Clear)	0 (Clear) IE
CSR07	R/W1	Transmit Interrupt Request	0 (Clear)	0 (Clear) XI
CSR08	R/W **	Internal Loopback	0 (Clear)	0 (Clear) IL
CSR09	R/W	External Loopback	0 (Clear)	0 (Clear) EL
CSR10	R/W	Sanity Timer Enable	0 (Clear)	0 (Clear) SE
CSR11	rr	Reserved: set to zero	0 (Clear)	0 (Clear)
CSR12	R	Ethernet Transceiver Power OK	No change	No change OK
CSR13	R	Carrier from Receiver Enabled	0 (Clear)	0 (Clear) CA
CSR14	R	Parity Error in Memory	0 (Clear)	0 (Clear) PE
CSR15	R/W1	Receive Interrupt Request	0 (Clear)	0 (Clear) RI

Key:

R—Read-only
W—Write-only
R/W—Read or Write
R/W1—Read or Write-one-to-clear (writing zero has no effect)
**—Active low
rr—Reserved bit with no access defined

The CSR bits are used as follows.

(CSR00) Receiver Enable (RE) **Read/Write**

When set: Enables the host to receive datagrams from the DELQA.

When cleared: Disables reception of datagrams from the DELQA.

Reset: Both software reset (CSR01) and power-up reset clear CSR00 and disable datagram reception.

This bit is set or cleared by the host only.

(CSR01) Software Reset)SR(**Read/Write**

When first set, and then cleared: The DELQA initiates a software reset.

This bit is set or cleared by the host only.

1
The DELQA may still be active for up to 100 micro-seconds after the software reset bit is cleared

2
Setting this bit does not reset the device, it must be first set, then immediately cleared (see 3.6.3).

(CSR02) Nonexistent-Memory Timeout (NXM) **Read-only**

When set: CSR02 is set if the DELQA times out while trying to access host memory.

When reset: Software reset and power-up reset clear CSR02.

This bit is set by the DELQA only, and cleared by the host. Note that the host must write a 1 to CSR07 in order to clear this bit.

(CSR03) Boot/Diagnostic ROM Load (BD) (PDP-11 only) **Read/Write**

When set and then cleared: The DELQA copies the boot/diagnostic code from its on-board B/D ROM across to 4K words of receive packet buffers in the host. The host should wait 150 milliseconds before clearing CSR03, and a further 150 milliseconds after that before executing the B/D code.

The host must have a PDP-11 CPU and the appropriate boot ROM, and the host software must follow the correct sequence of commands both before and after BD load; this includes clearing CSR00 (disable reception). See Section 3.6, Configuration and Control Procedures, for more details.

Reset: Both software reset (CSR01) and power-up reset clear CSR03.

This bit is set or cleared by the host only.

(CSR04) Transmit List Invalid/Empty (XL) **Read-only**

When set: The DELQA sets this bit to indicate to the host that it has encountered an invalid transmit descriptor (a transmit descriptor with the Valid bit clear). (The DELQA always interprets an invalid descriptor as marking the end of a list.)

The DELQA also sets this bit if it detects an NXM timeout, see CSR02.

When clear: This bit is cleared by the action of the host writing the high-order word of the transmit buffer descriptor list address to the Tx BDL I/O page register. This event indicates to the DELQA that the host has a list of transmit descriptors that it wishes to be processed.

Reset: Both software reset and power-up reset cause the DELQA to set this bit (that is, the list is considered invalid on reset).

This bit is always set by the DELQA, and cleared by the host (by writing the Rx BDL address).

(CSR05) Receive List Invalid/Empty (BL) **Read-only**

When set: The DELQA sets this bit to indicate to the host that it has encountered an invalid receive descriptor (a receive descriptor with the Valid bit clear). (The DELQA always interprets an invalid descriptor as marking the end of a list.)

The DELQA also sets this bit if it detects an NXM timeout, see CSR02.

When clear: This bit is cleared by the action of the host writing the high-order word of the receive buffer descriptor list address to the Rx BDL I/O page register. This event indicates to the DELQA that the host has a list of receive buffers into which the DELQA may deliver packets.

Reset: Both software reset and power-up reset cause the DELQA to set this bit (that is, the list is considered invalid on reset).

This bit is always set by the DELQA, and cleared by the host (by writing the Tx BDL address).

(CSR06) Interrupt Enable (IE) **Read/Write**

When set: This bit is set by the host to enable the DELQA to generate interrupts. Interrupts are generated under the following conditions.

1. The DELQA has completed a transmit operation.
2. The DELQA has completed a receive operation.
3. The DELQA has detected an NXM timeout.

The host should read the CSR to determine the reason for the interrupt (XI, RI, NI).

When clear: Interrupts to the host are disabled. (Interrupt bits XI, RI NI may still get set, but no interrupts will be generated).

When reset: This bit is set or cleared by the host only.

(CSR07) Transmit Interrupt Request (XI) **Read/Write-One-To-Clear**

When set: Indicates that the DELQA has transmitted at least one packet, and has written the transmit status to the status words of the corresponding buffer descriptor(s) in host memory. If CSR06 is also set, the DELQA will issue an interrupt to the host.

When reset: This bit is set by the DELQA and cleared by the host. Note that the host must write a 1 to CSR07 in order to clear it.

(CSR08) Internal Loopback (IL) **Read/Write**

(CSR09) External Loopback (EL) **Read/Write**

These bits are used to select the various DELQA loopback modes, but also have certain other functions.

Loopback modes:

CSR08	CSR09	Loopback Mode
0	0	Internal
0	1	Internal Extended
1	1	External

Note that CSR00 must be cleared for all loopback modes

Other functions:

CSR03	CSR08	CSR09	Function
0	1	0	Non-loopback operation
0	X	1	Read SA ROM checksum
1	X	1	B/D ROM Load

Where X = don't care

When reset: the host may set or clear both CSR08 and CSR09.

(CSR10) Sanity Timer Enable **Read/Write**

When set: The DELQA will enable the sanity timer after the host has transmitted the next setup packet. (Note that the setup packet is used to define the timeout period – see Section 3.6.6 for details). Once the sanity timer is enabled, any transmit activity by the host, such as datagram transmission, setup packet transmission, or loopback packets will reset the timeout counter.

When cleared: The DELQA will disable the sanity timer after the host has transmitted the next setup packet.

NOTE
Setting or clearing this bit by itself has no effect on the operation of the sanity timer. The host must remember to use the combination of CSR10 and setup packets to manage the sanity timer. The sanity timer will only be enabled or disabled based on the state of CSR10 after a setup packet is transmitted by the host.

When Reset: Both software reset and power-up reset clear CSR10 and disable the sanity timer. Note, however that power-up in DEQNA-lock mode, with switch S4 open on the DELQA module, will, by itself, cause the sanity timer to be enabled with the default (four-minute) timeout period (no setup packet is required).

(CSR11) Reserved: set to zero

(CSR12) Ethernet Transceiver Power (OK) **Read-only**

When set: Power is reaching the bulkhead connector.

When cleared: Either the fuse on the bulkhead assembly has blown, or there is no power to the bulkhead.

Reset: CSR12 is not affected by either software reset (CSR01) or power-up reset.

(CSR13) Carrier from Receiver Enabled (CA) **Read-only**

When set: In normal transmission or external loopback mode (CSR08 clear), CSR13 indicates that the DELQA is receiving a carrier signal from the Ethernet.

When cleared: There is no activity currently on the Ethernet, or internal or extended loopback mode is selected (CSR08 set).

CSR13 can be sampled to poll activity on the Ethernet.

Reset: Both software reset (CSR01) and power-up reset clear CSR13, because they set internal loopback mode (CSR08).

(CSR14) Parity Error in Host Memory (PE) **Read-only**

When set: Q-bus parity error during access to the host memory. This error is fatal, and the DELQA halts operation.

When cleared: Parity in the last host memory access was normal.

Reset: Both software reset (CSR01) and power-up reset clear CSR14.

In DEQNA-lock mode, CSR14 is reserved

(CSR15) Receive Interrupt Request (RI) **Read/Write-one-to-clear**

When set: Indicates that the DELQA has delivered at least one packet to host memory, and has written receive status to the status words of the corresponding buffer descriptor(s). If CSR06 is also set, the DELQA will issue an interrupt to the host.

When reset: Both software reset and power-up reset clear CSR14.

This bit is set by the DELQA and cleared by the host. Note that the host must write a 1 to CSR15 in order to clear it.

3.3.3 Vector Addresses

3.3.3.1 Vector Address Register (VAR) Definitions

The Vector Address Register (VAR) is a read/write register. The host system initializes VAR<09:02> with the address of the vector to the DELQA interrupt service routine. In Normal mode, VAR<15:10> are used for extra control and status information. In DEQNA-lock mode, only VAR00 is used for extra status information.

NOTE
The host software should disable interrupts (by clearing CSR06 or issuing a software reset) before writing to the Vector Address Register (VAR). Use a read/modify/write sequence to amend the VAR, and only attempt one operation (change vector; change mode; request self-test) at a time.

Figure 3-4 shows the VAR bits, and Table 3-3 summarizes the bit definitions for Normal mode operations. More complete bit definitions follow the table.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MS	OS	RS	S3	S2	S1	INTERRUPT VECTOR								RR	ID

VAR (READ/WRITE)

RE1688

Figure 3–4 Vector Address Register (VAR)

Table 3-3 Vector Address Register (VAR)

Bit	Normal Mode Access	Description	After Power-Up Reset	SR1	Selftest
VAR00	R/W	Identity Test Bit	0 (Clear)	No ch	No ch
VAR01	rr	Reserved			
VAR<09:02>	R/W	Interrupt Vector	Undefined	No ch	No ch
VAR<12:10>	R	Self-Test Status	1 (Set)	No ch	No ch
VAR13	R/W	Request Self-Test	1 (Set)	Clear	Clear
VAR14	R	Option Switch (S4) Setting	Reflect S4	No ch	No ch
VAR15	R/W	Mode Select	Reflect S3	No ch	No ch

Key:

R—Read-only
W—Write-only
R/W—Read or Write
rr—Reserved bit with no access defined
No ch—NO Change

The VAR bits record the DELQA status, as follows.

(VAR00) **Identity Test Bit** **Read/Write**

When set: VAR00 provides an identity test to distinguish a DELQA module operating in DEQNA-lock mode from a native-mode DEQNA module. To implement the test, the host software should:

1. Write VAR00=1
2. Immediately read VAR00

If VAR00=1, the module is a DELQA

If VAR00=0, the module is a DEQNA

3. Write VAR00=0

When cleared: VAR00 is ready for the identity test.

Reset: Power-up reset clears VAR00, but software reset (CSR01) has no effect on its value.

(VAR01) **Reserved**

(VAR <09:02>) **Interrupt Vector** **Read/Write**

In calculating the host interrupt vector address, the DELQA firmware reads only VAR<09:02> and assumes that VAR<01:00>=0.

Reset: Software reset (CSR01) has no effect on the interrupt vector, which is written directly by the host software using the I/O port.

The interrupt vector is undefined after power-up reset, until a new value is written by the host.

(VAR <12:10>) **Self-Test Status (Normal mode only)** **Read-only**

VAR<12:10> always indicate the latest status of the module self-test.

VAR12	VAR11	VAR10	Meaning
1	1	1	ROM CRC test
1	1	0	RAM test
1	0	1	68000 test
1	0	0	QIC test
0	1	1	QNA2 test
0	1	0	SA ROM test
0	0	1	LANCE test
0	0	0	Self-test completed without error

Self-test can be initiated in Normal mode, by power-up reset, or by a host write to VAR bit 13.

In DEQNA-lock mode VAR<12:10> always reads as zero.

(VAR13) **Request Self-Test (Normal mode only)** **Read/Write**

When set: The module is executing self-test.

Self-test takes approximately five seconds, and the contents of all the DELQA Q-bus registers should be treated as invalid during the test, and for another five seconds afterwards. The registers should not be written during this period.

When cleared: The self-test has completed, and VAR<12:10> indicate whether the self-test completed successfully or failed during a functional test. External loopback failures may be caused by an unconnected tranceiver, ALL other self-test failures should be treated as fatal.

Reset: In Normal mode, power-up reset sets VAR13 to initiate the module self-test.

In DEQNA-lock mode, VAR13 always reads as zero, and cannot be set.

(VAR14) **Option Switch Setting (Normal mode only)** **Read-only**

Immediately after power-up this bit may be used to determine the state of option switch S4.

When set: Option switch S4 is closed.

When cleared: Option switch S4 is open.

Reset: Software reset (CSR01) has no effect on VAR14, but power-up resets VAR14 to reflect the setting of option switch S4.

In DEQNA-lock mode, VAR14 is always zero.

(VAR15) Mode Select (Normal mode only) Read/Write

When set: If mode switch S3 is closed (for Normal mode), VAR15=1 selects Normal mode.

When cleared: If mode switch S3 is closed (for Normal mode), VAR15=0 selects DEQNA-lock mode and the DELQA clears VAR<14:10> for DEQNA compatibility. Use of this setting is not recommended.

Reset: Software reset (CSR01) has no effect on VAR15, but power-up reset resets VAR15 to reflect the setting of mode switch S3.

In DEQNA-lock mode, (mode switch 3 open), VAR15 is always zero.

NOTE
Host software must delay for a minimum of 5 seconds after power-up, before accessing device registers. This delay is essential to allow self-test to complete.

3.3.4 BDL Start Address Registers (BDL SARs)

There are two sets of Start Address Registers for the Buffer Descriptor Lists (BDLs):

- Transmit BDL Start Address Register
- Receive BDL Start Address Register.

Both registers are written by the host, and must be initialized with a word-write instruction. Reserved bits should be written as zero. The low-order word address must be written first, followed by the high-order word address. This is because the DELQA starts transfers as soon as it receives the high-order address.

To set up the transmit list for the first DELQA, write register 17774450 before register 17774452.

The DELQA starts DMA transfers of Ethernet packets as soon as they are transferred to on-board shared RAM from the receiver. When the Transmit BDL Start Address Register is initialized, the module starts DMA transfers of outgoing messages to shared RAM.

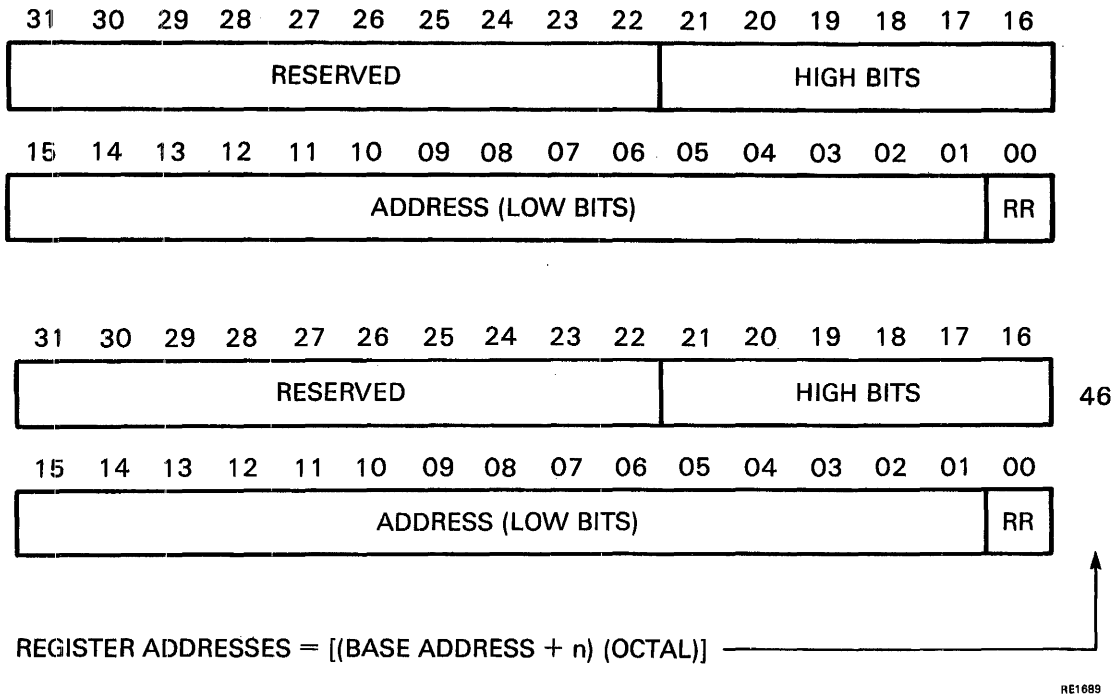


Figure 3–5 BDL Start Address Registers

3.3.5 Station Address Registers (SA ROM)

The lower bytes of each of the first six register locations contain the default Ethernet physical address of the DELQA module. The host accesses the 48-bit address by reading the register locations in ascending order. The host software is responsible for inserting the correct address in the source address field of each packet transmitted.

Two byte locations of the SA ROM include the checksum of the Ethernet physical address. The checksum is accessed by reading the first two bytes in reverse order, as follows.

1. Clear CSR00 (Receiver Enable) to disable Ethernet reception.
2. Clear CSR08 (Internal Loopback) and set CSR09 (External Loopback) to put the DELQA into external loopback mode.
3. Read the lower byte of word 1 in the DELQA I/O block.
4. Read the lower byte of word 0 in the DELQA I/O block.
5. To clear external loopback mode:

   Either set and then clear CSR01 (Software Reset)

   or write zero to EL (External Loopback).

3.4 HOST MEMORY DATA STRUCTURES

This section describes how Buffer Descriptor Lists (BDLs) are used to organize transmit and receive buffers.

When initialized, the DELQA has direct access to the host memory. The host memory should be set up with buffer space allocated for receive and transmit packets, and also for BDLs.

3.4.1 Receive and Transmit Buffers

The DELQA transfers packet data to and from receive and transmit buffers in the host memory. A buffer can contain an entire packet or part of a packet, but it cannot contain more than one packet.

The buffers that make up a message packet are referenced using a Buffer Descriptor List (BDL). Buffers contain only data; the status of each buffer is maintained in its buffer descriptor, and the sequence of buffers in the message is determined by the sequence of descriptors in the BDL. Only buffers that have the Valid bit set in their buffer descriptor can be used by the DELQA. The last entry in the BDL should have its Valid bit (bit 15) cleared to indicate termination of the BDL.

Transmit buffers may start and end on byte boundaries, but this is not recommended. Receive buffers must start and end on word boundaries. Word boundaries are recommended in both cases for faster processing.

3.4.2 Buffer Descriptor Lists (BDLs)

In the host database of BDLs there are two sections: the Transmit BDLs, and the Receive BDLs.

Each BDL is a forward-linked list of buffer descriptors. Contiguous descriptors are linked implicitly. Other descriptors can be linked explicitly by writing a chain address in the BDL.

Each descriptor identifies a single buffer by its starting address and length. The descriptor also contains space for the DELQA to supply status information associated with completed transmissions and receptions.

The host memory may contain as many BDLs as seems necessary, each referring to a set of buffers in memory. To initiate transfers, the host software writes the start address of the next Transmit or Receive BDL to the Transmit BDL or Receive BDL Start Register in the DELQA I/O page.

Figure 3-6 shows the format of a buffer descriptor.

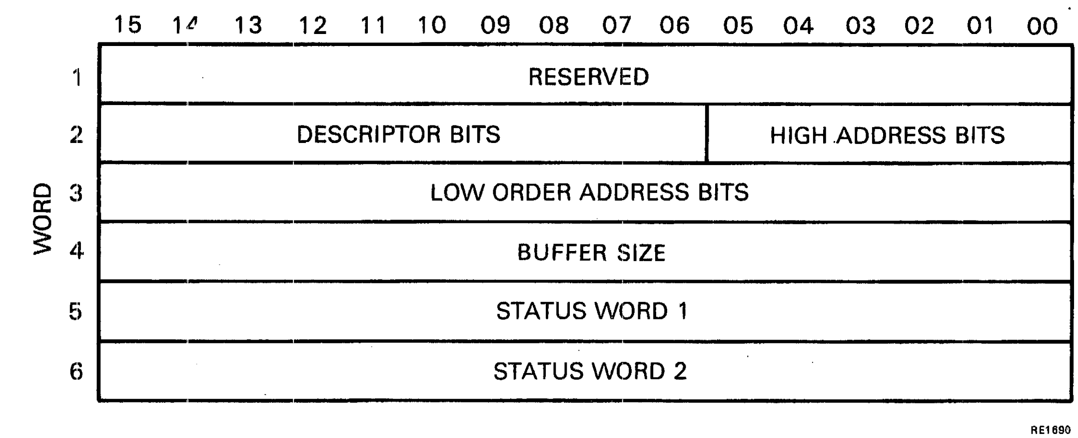


Figure 3–6 Buffer Descriptor Format

Each buffer descriptor in a BDL contains:

- A **reserved word:** reserved for DELQA use only.
- **Descriptor bits** that define the attributes of the buffer address: byte alignment; setup (optional); chaining (optional).
- **Buffer address** in the host memory.
- **Buffer size** in words, given as the two's-complement. (The word count does not include the two CRC words.)

The word count is always given as the number of aligned words for DMA transfer. So a one- or two-byte buffer has a word count of one, but a two-byte buffer that crosses a word boundary (that is starts on an odd address), has a word count of two. Therefore, a buffer that starts and ends on an odd-byte boundary must increase its word count by one.

The word count is taken from the byte count and the buffer alignment information in the H and L bits of the buffer address descriptor:

WORD COUNT = (BYTE COUNT + H + L)/2

NOTE
It is illegal for the host to specify a word count of zero

- Two **status words**. The status words may be omitted only when a buffer descriptor is forward-linked explicitly by a chain address to another buffer.

  When a complete packet has been transferred into or from the BDL, the DELQA firmware updates the last pair of status words with a record of any errors in reception or transmission.

To allow for multiple lists of descriptors, and to allow the DELQA to chain between them, the buffer address may be replaced by the start address of another BDL. The chain bit in the descriptor bits is set to indicate this.

3.4.3 Buffer Descriptor Bit Definitions

The buffer descriptor format is shown in Figure 3-6 and described in the following paragraphs.

3.4.3.1 Flag Word

Bit	Definition
F<15:00>	Reserved **Note:** The DEQNA module sets all of the bits in the flag word to 1 "during" the processing of a buffer descriptor. However, with DELQA the host software should not use these bits and their transition as an indication of the state of the descriptor. The host software should use the buffer descriptor status word 1 S1<15:14> bits to determine the buffer descriptor completion status.

3.4.3.2 Address Descriptor Bits

The host uses these bits to define the attributes of the associated buffer.

Bit	Definition
15	**V—Valid** **When set:** This bit indicates that this descriptor contains valid information (see the table below).
14	**C—Chain Address** **When set:** This bit indicates that the address contained in this descriptor is the address of another descriptor. This allows the DELQA to process multiple non-contiguous descriptor lists and explicitly "chain" the lists. Note that contiguous descriptors are implicitly chained (see the table below). Valid and Chain bit combinations:

Valid D15	Chain D14	Descriptor Use
1	0	This is a valid descriptor.
1	1	This descriptor contains the address of another descriptor.
0	0	This is an invalid descriptor (end of BDL).
0	1	Reserved

Bit	Definition
13	**E—End of Message** (Transmit Buffer Descriptor Only) This bit provides a mechanism for the host to chain together a number of buffers into a single packet. **When cleared:** This bit indicates that the associated buffer does not contain a complete packet. **When set:** This bit indicates that this buffer contains the last segment of the packet. (The DELQA will attempt to transmit the entire frame after this segment has been DMAd from the host).

Bit	Definition
12	**S—Setup** (Transmit Buffer Descriptor Only) **When set:** This bit indicates that the buffer contains a list of DELQA Ethernet destination addresses and control information.
11:08	**Reserved**
07	**L—Low Byte Termination** (Transmit Buffer Descriptor Only) **When set:** This bit indicates that this buffer ends on a byte boundary instead of a word boundary.
06	**H—High Byte Start** (Transmit Buffer Only) **When set:** This bit indicates that this buffer starts on a byte boundary instead of a word boundary. NOTE **When the transmit word count is 1, and the buffer starts on a byte boundary, the H bit must be set.**

3.4.3.3 Buffer Address
The high- and low-order address bits are either the 22-bit address of the buffer associated with this descriptor, or the address of another descriptor (see address descriptor bit 14, above).

3.4.3.4 Buffer Length (Word Count)
Buffer length is the two's complement value of the number of words in the buffer. [The word count is always measured in aligned words. Transmit buffer misalignment is not reflected in the word count, but the H and L descriptor bits denote this instead.

NOTE
It is not recommended that unaligned buffers be chained together, as this can degrade performance.

3.4.3.5 Status Words

Upon completion of a transmit or receive operation, the DELQA will update the two status words at the end of the buffer descriptor.

Status Word 1 bits 14 and 15 are used as a handshake between the DELQA and host. These bits are initialized by the host, and are updated by the DELQA to indicate that it has completed processing this descriptor and associated buffer. These bits should be initialized by the host as indicated below.

Bit	Definition

Transmit Status Word 1

15 **Lastnot** See the table below.

14 **Error or Used** See the table below.

Lastnot 15	Chain 14	Summary Status
1	0	Value initialized by the host.
1	1	This buffer has been used, but it is not the last segment of the packet; that is, a chained buffer.
0	0	This buffer contains the last segment of a packet, and has been transmitted with no errors.
0	1	This buffer contains the last segment of a packet, and has been transmitted with errors.

13 **Reserved**

12 **Loss**

When set: Indicates loss of carrier during transmission. Not valid for broadband applications.

11 **Reserved**

NOTE
In the DEQNA, Bits 11 and 12 have different functions for carrier status.

Bit	Definition
10	**STE (Sanity Timer Enabled)** The state of this bit is only valid in DEQNA-lock mode. **When set:** Indicates that the sanity timer was enabled via switch S4 at powerup.
09	**Abort** **When set:** Indicates that the transmission was aborted due to excessive collisions.
08	**Reserved**
07:04	**Count** The value in this field is an indication of the number of collisions that occurred before the transmission attempt associated with this status word. The only possible values are: 0 — No collisions, or packet aborted after 16 collisions (see Abort) 1 — One collision 2 — Between two and fifteen collisions Averaged over time, Count indicates the network loading.
03:00	**Reserved**

Transmit Status Word 2

Bit	Definition
15:10	**Reserved**
09:00	**TDR** TDR count for Time Domain Reflectometry.

Receive Status Word 1

Bit	Definition
15	**Lastnot** See the table below.

Bit	Definition		
14	**Error or Used** See the table below.		
	15	**14**	**Summary Status**
	1	0	Value initialized by the host.
	1	1	This buffer has been used, but it is not the last segment of the packet; that is, a chained buffer.
	0	0	This buffer contains the last segment of a packet, and has been transmitted with no errors.
	0	1	This buffer contains the last segment of a packet, and has been transmitted with errors.
13	**ESETUP** **When set:** Indicates a setup packet, external loopback packet, or internal-extended loopback packet.		
12	**Reserved**		
11	**Runt (Internal Loopback Failure)** **When set:** Indicates that the internal loopback operation was unsuccessful.		
10:08	**RBL** Received Byte Length bits 10 to 08. These bits are all set for a setup packet.		
07:03	**Reserved**		
02	**Frame** **When set:** Indicates a framing alignment error; that is, other than an integral number of bytes were received. This bit is only set if there was also a CRC error. See bit 01.		

Bit	Definition
01	**CRCERR** **When set:** Indicates that a CRC error has been detected in the current packet. The DELQA delivers all packets received with CRC errors.
00	**OVF (Overflow)** **When set:** Indicates that at least one packet has been discarded between the current and previous packet.

Receive Status Word 2

15:08	**RBL<07:00>** Receive Byte Length bits 07 to 00, duplicated from the lower byte.
07:00	**RBL<07:00>** Receive Byte Length bits 07 to 00. These bits and Receive Status Word 1 bits 10:08 (see above) form RBL<10:00>, the number of bytes transferred by the DELQA into the host receive buffer, less 60. Host software must add 60 to this value to get the length in bytes of the received packet, excluding the CRC (not transferred). Packet Length (bytes) = RBL<10:00> + 60 In the case of setup packets, and all types of loopback packets, the host need not add 60 to get the correct packet length.

3.5 DATA TRANSFER PROCEDURES

This section describes how the host software controls transmission and reception.

Data transfers are handled automatically by the DELQA, using data DMA. The host software controls transmissions by initializing and clearing buffer areas and associated control registers.

3.5.1 Transmit Packet

The host initiates transmission by first setting up a Transmit BDL, and then writing its address to the Transmit BDL Start Address register in the DELQA module.

The transmit buffers should be set up before attempting to set up the Transmit BDL. A transmit buffer can be up to 1514 bytes in length; this is the maximum number of bytes allowed in an Ethernet packet, excluding the four CRC bytes.

To complete the transmission, the DELQA executes the following steps.

1. Read the descriptor bits, and act on the buffer descriptor information as follows.
2. If the Valid bit is set, the DELQA accesses the start address and buffer length fields, reads the relevant buffer, transfers the contents to its on-board shared RAM, updates the status words, and continues to the next descriptor.

3. If the Valid bit is clear, the DELQA marks the end of the current BDL. The DELQA ceases to access the BDL and its associated buffers.

4. If the chain bit is set, the DELQA links to the BDL, via the start address indicated in the buffer address field, and continues to the next descriptor.

5. If the End-of-Message bit D<13> is set, the DELQA generates the preamble and CRC for the message, and transmits the complete message packet over the Ethernet. Then it updates the status words in the latest buffer descriptor with the outcome of the transmission. (If CSR06 Interrupt Enable is set, the DELQA also generates a transmit interrupt request to indicate that a message has been transmitted.)

To achieve acceptable transmission rates, the DELQA executes control DMA (to set up the next data DMA transfer), data DMA, and data transmissions in parallel. The host software reads the status or contents of buffers only after the DELQA has returned the transmission status to the status word bits.

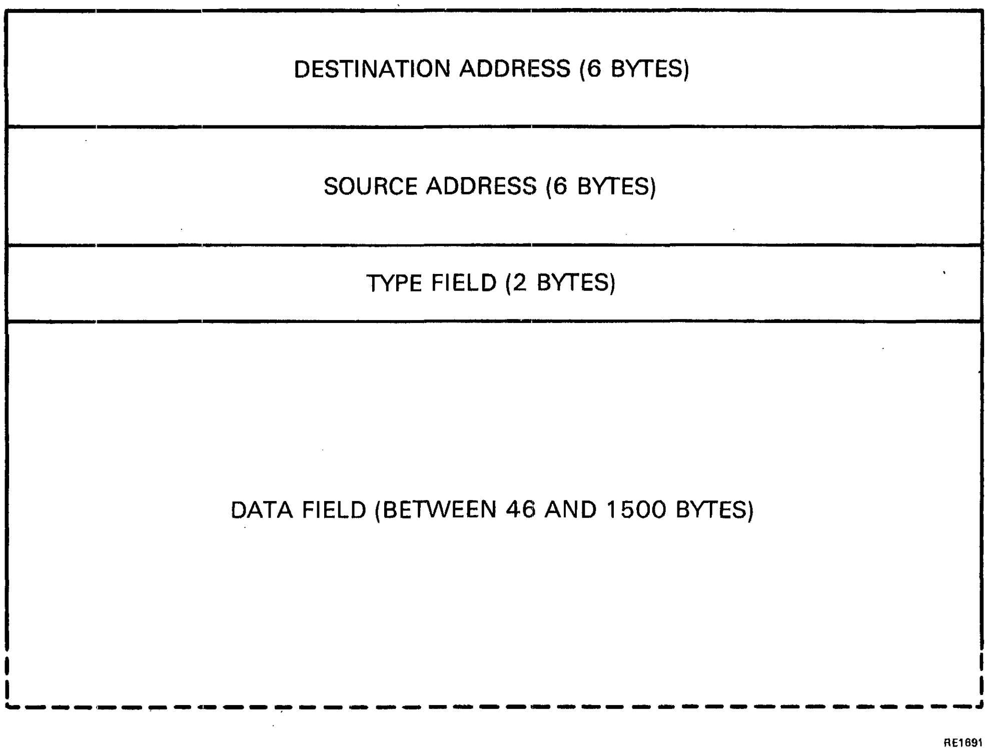


Figure 3–7 Ethernet Packet Format

3.5.2 Transmit Programming

The host software for packet transmission is responsible for the following actions:

1. Establish the location and contents of the transmit message buffers.

2. Initialize the start address and descriptor bits for each buffer descriptor in the Transmit BDL.

3. Write all the data fields within the transmit packet, including destination address (6 bytes), source address (6 bytes), type field (2 bytes), and data (between 46 and 1500 bytes).

   The DELQA hardware supplies the CRC automatically.

4. Clear the Valid bit in the last descriptor of the BDL.

5. Set the Valid bit in all the previous descriptors of the BDL.
6. Write the start address of the BDL to the BDL Start Address register on-board the DELQA, to initiate transmissions.

The host software should also provide an interrupt service routine to:

- Check the status and availability of transmit buffers
- Check CSR04 (Transmit List Invalid) to ensure that previous list processing has completed
- Check CSR02 (NXM) in case the interrupt was caused by a memory access error.
- Write 1 to clear CSR07 (Transmit Interrupt Request), if the bit is set.

3.5.3 Transmission Errors

In status word 1 of the last BDL entry for the transmitted message, the following status bits in the transmit buffer descriptor record transmission errors.

S1<09>	Abort	Excess collisions: there have been more than 15 attempts to transmit this packet. Check S1<12> in Transmit Status Word 1 in case the Ethernet circuit is faulty (see below).
S1<12>	Loss	Loss of carrier during transmission, usually due to a short circuit on the Ethernet. However, Loss does not abort transmission, because it may be set during a normal collision recovery.

The Time Domain Reflectometry (TDR) counter (S2<09:00>) is a 10 MHz counter which is enabled by the DELQA when a carrier signal is detected, and disabled when the carrier stops or a collision is detected. The contents of the TDR counter are valid only when Abort (S1<09>) is set, and may be used as a relative measure of the distance through the network between the module and the supposed fault or collision.

3.5.4 Receive Packet

The host initiates reception by first setting up a Receive BDL, and then writing its start address to the DELQA module.

To complete the receive process in response to activity on the Ethernet, the DELQA executes the following steps.

1. Read the descriptor bits, and act on the buffer status as follows.
   - If the Valid bit is cleared, it marks the end of the current BDL. The DELQA ceases to access the BDL and its associated buffers.
   - If the Chain bit is set, the DELQA links to the BDL whose start address is indicated in the buffer address field, and continues from step 2.
   - If the Valid bit is set, the DELQA accesses the start address and buffer length fields, reads the next part of the incoming message into the indicated buffer from its on-board shared RAM, and continues from step 2.
2. If the message ends, the DELQA terminates reception, and updates the status words in the last buffer descriptor used. (If CSR06 is set, the DELQA also generates a receive interrupt request to indicate that a message has been received.)

3.5.5 Receive Programming

The host software for packet reception is responsible for the following actions.

1. Establish the location and contents of the receive message buffers.

   Sufficient receive buffers should be allocated for at least one packet of the maximum expected length, in order to ensure that a receive interrupt request is generated before the next incoming message arrives.

   NOTE
   No interrupt is generated if there are not enough valid receive buffers in the Receive BDL to accommodate a complete packet.

2. Initialize the start address and descriptor bits for each buffer descriptor in the Receive BDL.
3. Initialize Status Word 2 of all the descriptors in the Receive BDL with unequal high and low bytes. (The DELQA makes the high and low bytes both equal to the received byte length, to indicate when the receive data is valid.)
4. Clear the Valid bit of the last BD in the BDL. Set the Valid bit in all BDs in the BDL except the last BD.
5. Set CSR00 (Receiver Enable) to enable Ethernet packet reception.
6. Write the start address of the BDL to the BDL Start Address register on-board the DELQA, to initiate reception.

The host software should provide an interrupt service routine to:

- Check the status and availability of receive buffers
- Check CSR05 (Receive List invalid) to ensure that previous list processing has completed.
- Write 1 to clear CSR15 (Receive Interrupt Request) if this bit is set.
- Check CSR02 (NXM) in case the interrupt was caused by a memory access error.

3.5.6 Receive Errors

In Status Word 1 of the last BDL entry for the received message, the following status bits in the receive buffer descriptor record reception errors:

S1<00>	OVF	Overflow: indicates that message data from the Ethernet has been lost between the current and the previous message.
S1<01>	CRCERR	CRC error: with the message truncated as a result. Affected packets are delivered, but the datalink error counters are updated.
S1<02>	Frame	Frame Alignment Error (some bytes incomplete): Frame is set only if CRCERR is set also.
S1<12>	Discard	Discard packet.
S1<13>	ESETUP	Looped Back Setup Mode packet or EL packet.

Receive Buffer Length, RBL<10:00> is the number of bytes in the current received packet, excluding the CRC. The value in RBL should be interpreted as follows:

- For **normal datagram reception**, RBL is 60 bytes smaller than the actual number of bytes received.
- For **other loopback modes**, RBL gives the correct value.
- For all looped setup packets the RBL bits <10:08> equal 1. The lower byte of the RBL gives the correct value.

3.6 CONFIGURATION AND CONTROL PROCEDURES

This section describes the host programming procedures for bootstrap loading (PDP-11 only), for DELQA setup, reset, and interrupt handling, and the sanity timer.

3.6.1 Boot/Diagnostic Load

NOTE
The on-board boot/diagnostic microcode is for use with modules connected to PDP-11 systems only.

The boot/diagnostic (BD) ROM on-board the DELQA contains PDP-11 code that can be loaded into the host memory and executed. This code is used for extended primary and DECnet bootstrap, and for the DELQA citizenship test.

The PDP-11 boot/diagnostic code can be loaded across the Q-bus in either Normal or DEQNA-lock mode, but the DELQA must be software reset before the boot/diagnostic code is loaded into the host memory.

All requests for this code from the DELQA must follow the correct sequence of commands. The operations listed below are the exact sequence implemented in existing DEQNA support code for use with PDP-11 CPU/system boot ROMs for CPU number KDJ-11/B. Timing values are indicated to be 150 milliseconds, but values as low as 100 milliseconds should be acceptable.

The BD loading sequence is as follows:

1. To reset the DELQA, set and then clear CSR01. This software reset:
   - Disables Receiver Enable by clearing CSR00
   - Enables internal loopback mode by clearing CSR08 (IL).
2. Build two Receive descriptor buffers, each of 2K bytes.
3. Load the pointer into the Receive BDL Start Address register.
4. Write the octal pattern 1010 to the CSR to:
   - Disable the receiver (CSR00 = 0)
   - Disable software reset (CSR01 = 0)
   - Request boot/diagnostic ROM code (CSR03 = 1)
   - Disable interrupts (CSR06 = 0)
   - Select internal extended loopback mode, by clearing IL (CSR08 = 0) and setting EL (CSR09 = 1)
   - Disable the sanity timer (CSR10 = 0).
5. Wait 150 milliseconds
6. Clear CSR03 (Boot/Diagnostic ROM Load)
7. Wait 150 milliseconds
8. Execute the boot/diagnostic code from the Receive descriptor buffers.

When the DELQA boot/diagnostic code begins executing, it requests the on-board self-test, and waits for it to complete before continuing. For compatibility with DEQNA/PDP-11 system boot ROMs, which do not wait for the DELQA ROM self-test to complete after powerup, the DELQA aborts the self-test when the boot/diagnostic sequence commences. In all cases, this sequence begins with a required software reset.

3.6.2 Setup

The setup packet is the only mechanism, other than the DELQA control registers (CSR and VAR), by which the host software can send commands, status, and control functions to the DELQA module.

The setup packet can be used to initialize the following functions within the DELQA module:

- Multicast address or promiscuous filtering for address recognition
- Timeout value for the sanity timer
- Up to 14 six-byte Ethernet addresses that the DELQA module is to recognize
- MOP configuration and control

3.6.2.1 Setup Packets

The setup packet is a special type of transmit packet. In setup mode, the transmit packet is not sent out on the Ethernet. Instead, it is stored as control information within the DELQA module.

Setup mode is entered by setting bit 12 of the address descriptor in a Transmit BD.

The setup packet is looped around internally, and placed in a receive buffer for verification and synchronization. Reception of messages from the Ethernet is blocked until the loopback of the setup packet is complete.

NOTE

Ethernet reception is disabled during processing of setup packets; excessive use may significantly affect performance.

3.6.2.2 Setup Information

The setup packet contains three main groups of information which the host software can issue to the DELQA.

1. **Target address information** contains the Ethernet physical and multicast addresses for which the DELQA is to receive messages.
2. **Control parameters** specify special reception modes (such as promiscuous or all multicast) and sanity timer timeout values.
3. **MOP information** is used to read and change MOP parameters.

Setup packets may contain either one or two of these groups of information. A combination of the specified length of the setup packet and the value of the first byte of the setup packet buffer indicates which groups of information are present.

Table 3-4 explains all the possible combinations of information groups.

Table 3–4 Setup Packet: Information Group Combinations

Information Groups (Maximum 2)	Packet Length in Bytes (Octal)	Value of Byte 1
Target addresses only	177 or less	
Target addresses and control parameters	200 to 377	Zero
Target addresses and MOP Element Blocks (MEBs)	400 exactly	Non-zero

More than one setup packet may be issued. Each setup packet overwrites completely the setup area up to the 200 byte offset, but the MOP area between the 200 byte and 256 byte offset is overwritten only if the MOP flag is set at the start of the packet. Therefore, the only useful setup packet lengths are 177, between 200 and 377, or 400 (octal) bytes.

The host should maintain a copy of the current setup data, in order to recreate the correct 14 addresses (which cannot be read back from the DELQA) whenever the setup information is modified. Since the DELQA can only have two types of setup packet information per setup packet, the DELQA will accumulate all setup packet information, unless respecified in a subsequent setup packet. Although setup packets may be repeatedly issued to the DELQA to modify parameters or to read internal values (that is, counters, system id parameters), only one setup packet should be outstanding to the DELQA at a time.

3.6.2.3 Setup Packet Buffer Descriptor

The DELQA recognizes setup packets of between 200 (octal) and 377 (octal) bytes as indicating that control information is present, as well as target addresses. The contents of the extra bytes between 200 (octal) offset and 377 (octal) offset can be arbitrary, because the control information itself is held in the buffer descriptor for the setup packet.

In the buffer descriptor, the lower bits of the buffer word count are used to specify special address filter modes (promiscuous or all multicast) for reception, and to reset the timeout value for the sanity timer.

Please refer to the equation in Section 3.4.2 in order to understand how the host specifies the descriptor's buffer BYTE size from a combination of the Descriptor's Word Count Field (Transmit Descriptor Word 4) and the Descriptor's Address Descriptor bits (Transmit Descriptor Word 2) D<07> "L" bit and D<06> "H" bit.

- Buffer size in bytes (byte count) = 200 (octal)

  This sets D<06:00> = 000000 (binary) which does not specify any control parameters in the size field of the buffer.

- Buffer size in bytes (byte count) = 201 (octal)

  This sets D<06:00> = 000001 (binary) which specifies that the DELQA should set All Multicast Enabled of the setup packet control parameters.

Table 3–5 Setup Packet Buffer Descriptor: Address Mode Bits

Bit	Meaning
Word 4: Buffer Word Count	
C<00>	All Multicast address filter
	Enables DELQA recognition of any multicast address
C<01>	Promiscuous address filter
	Enables recognition of any destination address
C<03:02>	
C<06:04>	Sanity timer timeout value
	Increases the timeout period of the sanity timer in factors of four.
	Code **Timeout** 000 0.25 seconds 001 1 second 010 4 seconds 011 16 seconds 100 1 minute 101 4 minutes (default) 110 16 minutes 111 64 minutes
C<15:07>	Word count of the buffer (which contains the entire setup packet), given as the two's complement.
C<10:8>	All 1s for all setup packets
C<7:00>	Size of setup packet buffer if less than 200
C<7:00>	= 0 if MOP specified in setup packet

3.6.2.4 Setup Packet Format

The first part of a setup packet defines the Ethernet addresses to which the DELQA should respond.

Figure 3-8 shows the setup packet format in bytes (octal). The columns are used to show how the DELQA can be programmed to recognize up to 14 six-byte Ethernet addresses. The low-order byte of the address is at the top of each column. The low-order bit of the low-order byte is the Multicast bit.

Each group of seven addresses is interleaved through 56 bytes of the setup packet. One of the addresses must be the physical address of the DELQA module. Any other specified addresses are multicast addresses. The broadcast Ethernet address (all 1s) may be optionally enabled.

Any columns not used should be set to the physical address (for better protection against mischievous Ethernet traffic). More than one physical address may be specified, but in Normal mode, only the first is used for receiving datagrams, and as the source address for system ID messages generated by the DELQA. In DEQNA-lock mode the specifications of multiple physical Ethernet addresses will cause the DELQA to filter all such physical Ethernet addresses for packet reception.

NOTE
Enabling more than one physical address is not recommended under normal circumstances. This may have a substantial impact on performance.

The MOP flag is located in the first byte of the setup packet (location 0). If the MOP flag is set to a non-zero value by the host software, the DELQA expects to find MOP information at the end of the setup packet, between offset 200 (octal) and 400 (octal). The setup packet itself must be exactly 400 (octal) bytes long.

Refer to Section 3.7 for information about MOP programming.

Table 3-6 lists the effects of power-up reset, software reset, and self-test on the parameters of the setup packet.

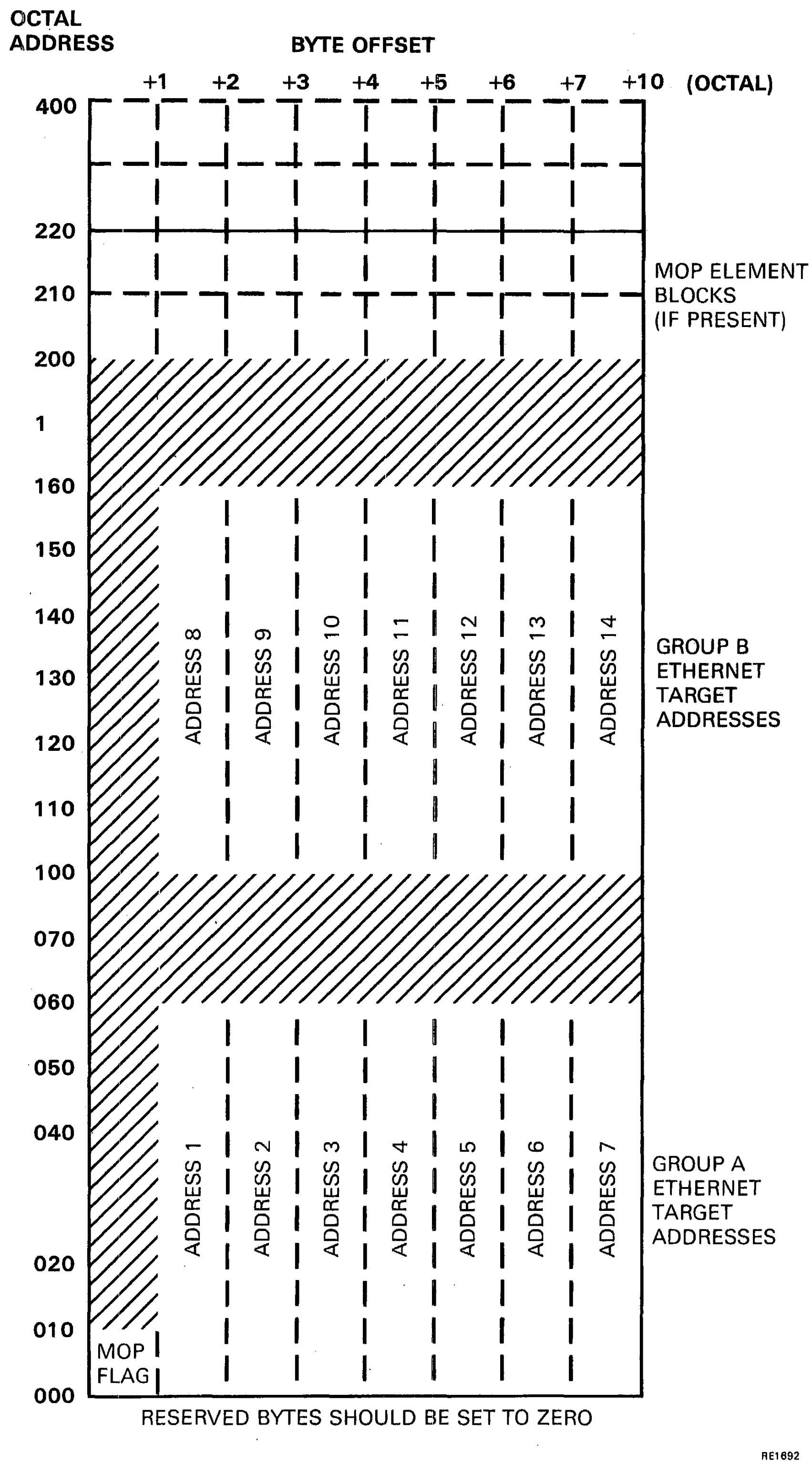


Figure 3–8 Setup Packet Format (Bytes)

Table 3–6 Effects of Reset on Setup Packet Data

Setup Packet Data	Value After Power-Up Reset or Self-Test	Value After Software Reset
Physical address	SA ROM address	N/A
Multicast addresses	Multicast disabled	N/A
Mode bit: All Multicast	Disabled	Disabled
Mode bit: Promiscuous	Disabled	Disabled
Mode bit: LED Value	LEDs set	N/A
Mode bit: Sanity timer	Reset to four minutes	N/A
MOP: Boot Password	Reset to zero default	N/A
MOP: Sys ID Data	Reset to defaults	N/A
MOP: Datalink counters	Counters cleared	N/A

3.6.3 Reset

The DELQA is reset during power-up, or by the host software using CSR01 (Software Reset). The module is affected differently by these two methods of reset.

Setting CSR01 (Software Reset) does not reset the DELQA hardware; instead it causes the DELQA to enter the DELQA reset state. The host software may verify that the DELQA is in the reset state by checking for a 10062 (octal) pattern in the CSR; this indicates that CSR<02,05,06,13> are set. The DELQA remains in the reset state until the host software clears CSR01 (Software Reset).

In reset state, the DELQA supports:

- Clear CSR01 (Software Reset) to exit the DELQA reset state
- Load Vector Address Register (VAR)
- If enabled by option switch S4:

  Either MOP remote boot
  Or Sanity timer timeouts.

Other attempts to write commands to the DELQA registers (such as setting interrupt enable, or loading transmit/receive lists) are not supported. The host software must clear the software reset bit separately, before writing other commands to the DELQA registers.

There is no timing requirement for CSR01 (Software Reset). After CSR01 has been cleared, it takes up to ten milliseconds for the DELQA to complete the initialization sequence.

Tables 3-2 and 3-3 summarize the effects of power-up reset, CSR01 (Software Reset), and VAR13 (Self-Test) on the Control and Status Register (CSR) and the Vector Address Register (VAR) respectively. Table 3-6 summarizes the effects on setup packet data.

3.6.4 Interrupt Handling

There are three interrupt conditions.

- Receive Interrupt Request (CSR15), when a complete packet has been received.
- Transmit Interrupt Request (CSR07), when a transmission is completed.
- Nonexistent Memory (CSR02), when a Q-bus or memory access error occurs. Setting CSR02 also sets CSR07 (Transmit Interrupt Request).

These conditions generate an interrupt only if CSR06 (Interrupt Enable) is set.

Interrupts are not queued. If multiple messages are handled by the DELQA before the Interrupt Request bit is cleared, there are no additional interrupts. The interrupt service should therefore scan the remaining descriptors in the current BDL to determine whether any other messages have been received.

3.6.5 Loopback

All loopback modes loop a data packet back through the on-board Ethernet controller (LANCE) to be read back into a Receive buffer.

There are four loopback modes, as follows.

- **Setup**: The setup packet does not reach the Ethernet, but it does pass through the LANCE before the contents are used to set up address and control data in the DELQA module. CSR00 (Receiver Enable) does not affect this mode of loopback.

  Setup mode is enabled by setting address descriptor bit D12 of the buffer descriptor for the setup data packet.

- **Internal**: Internal loopback only supports packet lengths of six bytes. The data packet does not reach the Ethernet, but it does pass through the LANCE on its way back to the host. The DELQA is initialized in this mode as a failsafe feature.

  Internal loopback is selected by setting CSR08 (IL) when CSR9 (EL) and CSR00 (Receiver Enable) are clear.

- **Internal extended**: Internal extended loopback mode can loopback all legal sizes of Ethernet packet, from 60 to 1514 bytes, excluding CRC. The data packet does not reach the Ethernet, but it does pass through the LANCE on its way back to the host.

  Internal loopback is selected by setting CSR08 (IL) and CSR9 (EL) when CSR00 (Receiver Enable) is clear.

- **External**: Extended loopback exercises the Ethernet serial interface (SIA) as well as the LANCE, and can loopback all legal sizes of Ethernet packet, from 60 to 1514 bytes, excluding CRC.

  A suitable external loopback connector must be connected either (as supplied) to the bulkhead assembly, or to the end of the transceiver cable, for the loopback test to be executed. The test should be run using first the minimum and then the maximum available Ethernet segment length.

  External loopback is selected by setting CSR09 (EL) when CSR08 (IL) and CSR00 (Receiver Enable) are clear. As with CSR03 (Boot/Diagnostic ROM Load), the DELQA must be disabled and the on-board transmit and receive buffers emptied before this function is invoked.

3.6.6 Sanity Timer

When DEQNA-lock mode is enabled by mode switch S3 open, the sanity timer is cleared and enabled automatically on power-up if switch S4 is open.

In either Normal or DEQNA-lock mode, the sanity timer is enabled when CSR10 is set and a setup packet is issued. When cleared and a setup packet is issued, CSR10 both disables and resets the sanity timer.

All transmissions (normal, loopback, and setup) reset the sanity timer without affecting its status (enabled or disabled). CSR10 is cleared at power-up reset and by CSR01 (Software Reset). The default timeout period is four minutes. Other limits between 0.25 seconds and 64 minutes can be programmed using a setup packet.

If the timer reaches its limit, BDCOK is negated on the Q-bus for approximately 3.6 microseconds, causing the host to reboot itself.

NOTE

In DEQNA-Lock mode the setting of switch S4 controls the operation of the Sanity Timer. In DEQNA-Lock mode the Sanity Timer is enabled by switch S4 open at powerup. If the Sanity Timer is enabled at powerup the DELQA will reboot it's Q-Bus host every four minutes, unless it is cleared by the host queuing a transmit to the DELQA.

The Sanity Timer can be disabled by clearing CSR10 and then sending a setup packet to the DELQA.

The Sanity Timer is never enabled at powerup in Normal Mode (Switch S3 closed)

3.7 MAINTENANCE OPERATIONS PROTOCOL (MOP): MODULE SUPPORT

This section describes how the host software can change the parameters that the DELQA uses when implementing the Maintenance Operations Protocol (MOP) functions as part of DECnet network management functions.

In Normal mode the DELQA implements the following MOP functions in response to remote console messages from other nodes on the Ethernet.

- Respond to request system ID
- Loopback reply to remote node

The DELQA also implements the following functions automatically and independently.

- Transmit system ID every 8 to 10 minutes.
- Maintain and store datalink counters as a record of transfers and errors.
- The DELQA can initiate a host system reboot in response to a Trigger instruction message from a remote console. This remote boot option must be selected explicitly by opening option switch S4 on the DELQA board.

The host software can read and amend the MOP implementation parameters by including special MOP Element Blocks (MEBs) in a setup packet.

The implementation of each of the MOP remote console functions and the format of the Ethernet messages are described in Chapter 4. The rest of this section describes how to use the MOP elements in a setup packet.

For further information, refer to the *DECnet Maintenance Operation Protocol (MOP) Functional Specification.*

Table 3–7 MOP Functions

Element Type	Function
0	MOP Termination
1	Read Ethernet Address
2	Reset System ID
3	Read Last MOP Boot
4	Read Boot Password
5	Write Boot Password
6	Read System ID
7	Write System ID
8	Read Counters
9	Read/Clear Counters

3.7.1 Internal Loopback

In internal loopback, the DELQA loops all messages through the module, and the host can neither send nor receive Ethernet messages.

Internal loopback may be entered either by the host command (set CSR08) or at device power-up. The behavior of the device differs according to its mode.

- In **Normal mode**, the characteristics of internal loopback depend on how loopback was initiated.
    a. From host command, no Ethernet access is possible.
    b. From device power-up, certain types of MOP message may be processed by the DELQA (that is, MOP boot if enabled by S4, Ethernet loop channel, and Request System ID).
- In **DEQNA-lock mode**, no Ethernet access is possible.

3.7.2 MOP Element Blocks (MEBs)

A MOP element is programmed by inserting a MOP Element Block (MEB) in a setup packet. Each MEB specifies a single MOP function for the DELQA to perform, and refers in turn to a MEB buffer which details the parameters for implementing the function.

NOTE

Although a given setup packet may contain from 0 to 10 MEB elements, each MEB may appear only once in a given setup packet. The terminating MEB type field of zero must always be the last MEB type. Omission of the terminating MEB type field of zero will cause the setup packet to fail to be properly processed. Although the DELQA loops backs all setup packets and sets the ESETUP bit in the receive descriptor of the looped setup packet, if

no terminating MEB type field of zero is found the DELQA will also set the Error or Used bit of the receive descriptor status word 1. The Error or Used bit, in the receive descriptor status word 1 will also be set if the buffer size specified for any MEB read operation is too small. See Figure 3-10 for required buffer sizes.

A MEB is fixed at six bytes in length and has the following fields.

Byte 1 –– MEB Buffer Type field (indicates MOP function)
Bytes 2, 3, 4 — MEB Buffer Base (MEBB) Address
Bytes 5, 6 — MEB Buffer Size

Although the format of each MEB buffer is specific to its type field, the following is true for all MEB buffers.

- Word orientation is used in all MEB definitions.
- Offsets from the MEB Base Address (MEBB) are defined in octal.

Figure 3-9 shows the relationship between the MEB in the setup packet and the corresponding MEB buffers. Figure 3-10 shows the format of the MEBs for MOP element types 1 to 9; the start addresses refer back to the MEB Base (MEBB) address shown in Figure 3-9.

3.7.3 MOP Element Type 0: MOP Termination

MOP element type 0 terminates a list of MOP elements in the setup packet.

3.7.4 MOP Element Type 1: Read Ethernet Address

MOP element type 1 provides a mechanism for the host software to verify the current Ethernet physical address.

This function permits the host software to verify that the DELQA has correctly loaded the physical address information from the setup packet. (The default Station Address (SA) in the DELQA ROM is usually read directly from the I/O port.)

Table 3–8 MOP Element Type 1

Offset	Bits	Description
MEBB+0	PA<15:00>	The low-order address bits <15:00> of the physical address. Written by the DELQA for a read function.
MEBB+2	PA<31:16>	The middle-order 16 address bits of the physical address. Written by the DELQA for a read function.
MEBB+4	PA<47:32>	The high-order 16 address bits of the physical address. Written by the DELQA for a read function.

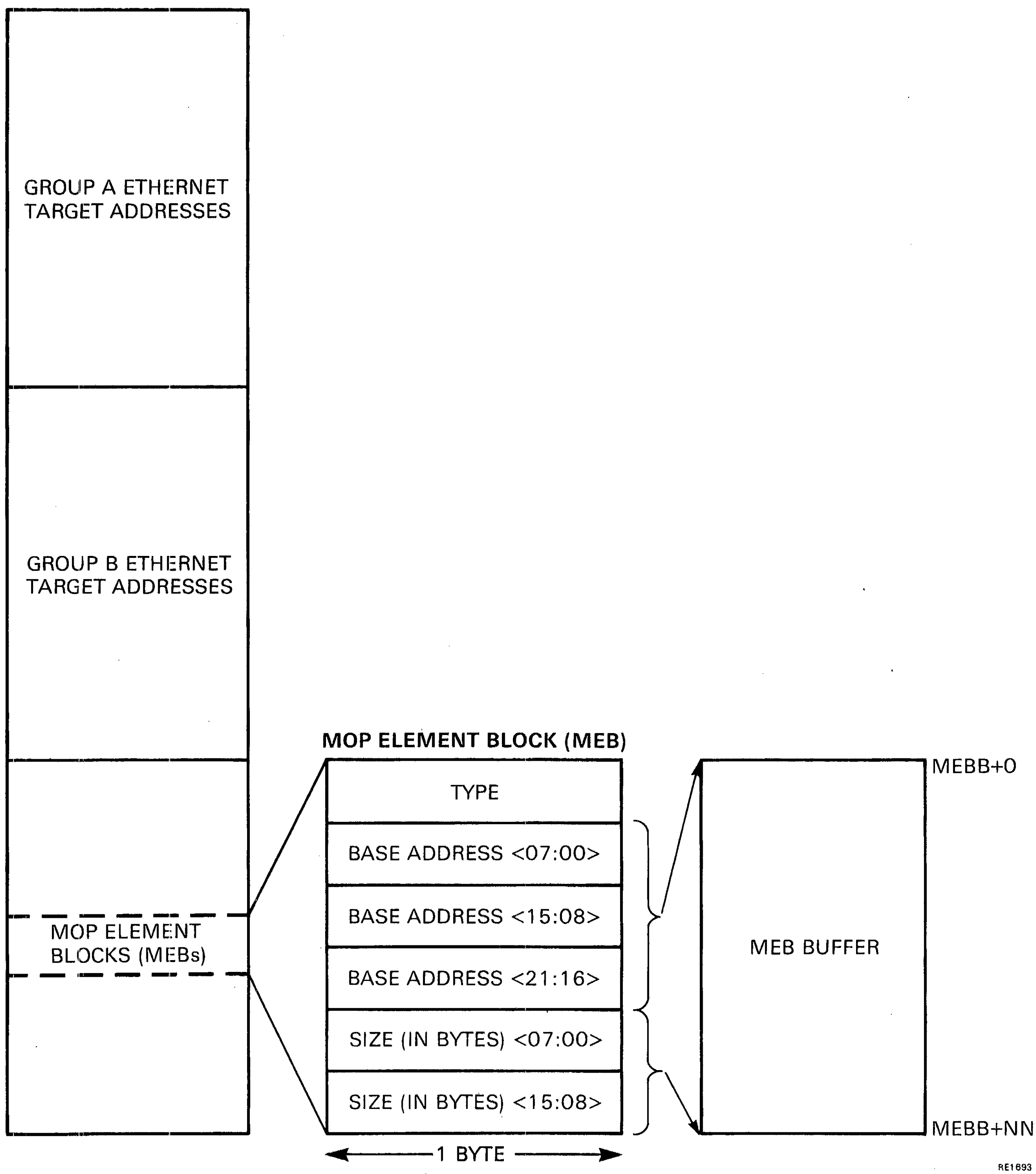


Figure 3–9 MOP Element Block Buffers in the Setup Packet

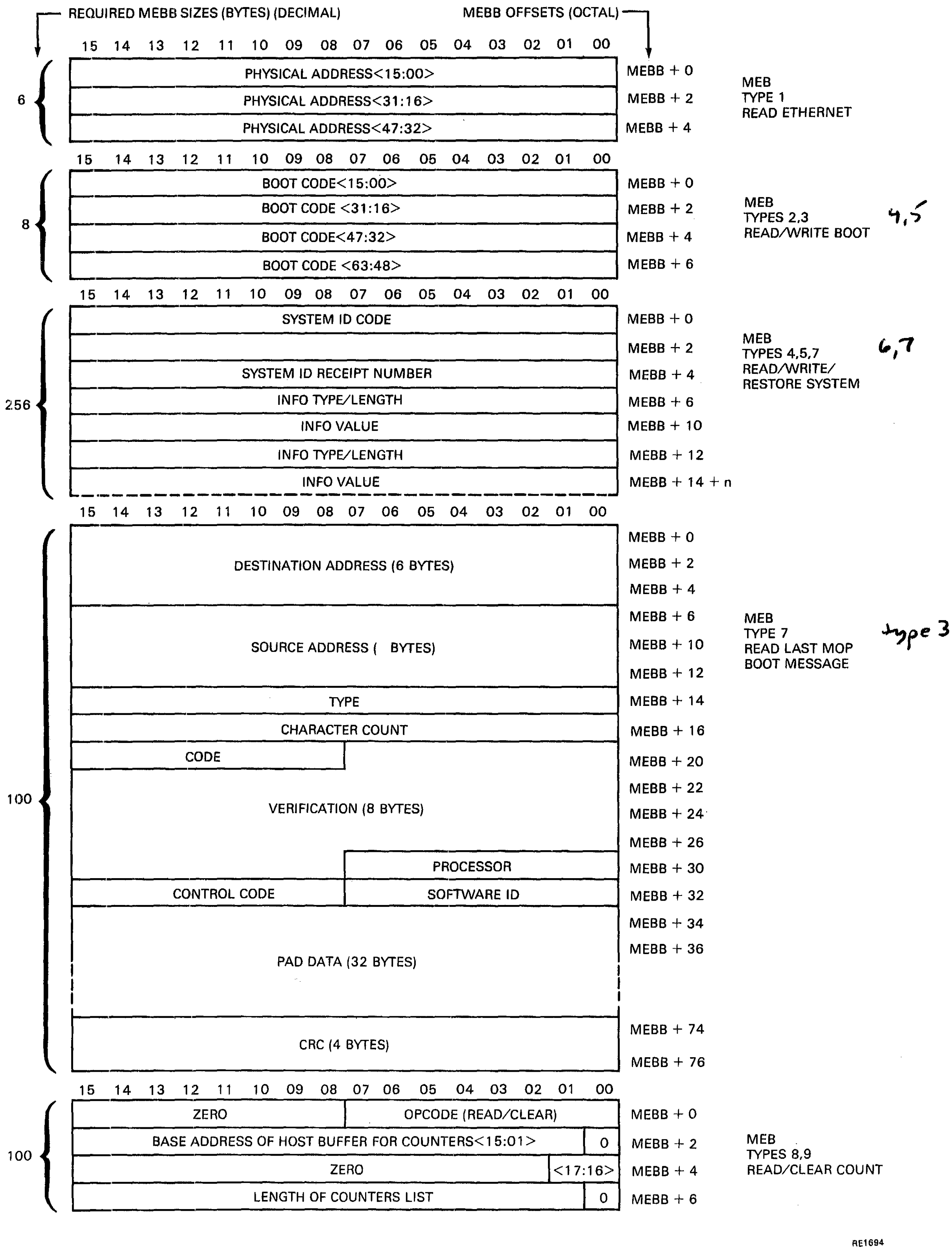


Figure 3–10 MOP Element Block Types 1 to 9

3.7.5 MOP Element Type 2: Reset System ID

MOP element type 2 resets the system ID to the default parameters stored on-board the DELQA. This default is then broadcast from the DELQA to the network automatically at power-up reset, and repeatedly at intervals until a modification occurs from a MOP element 5 (Write System ID).

MEB Type 2 has only a type field, and no associated MEBB specification.

3.7.6 MOP Element Type 3: Read Last MOP Boot

MOP element type 3 obtains a copy of the MOP remote boot message which caused the last local host reset. The only occasion when this function value returns a valid, non-zero MOP remote boot message is just following the execution of a SYSTEM PROCESSOR remote boot. In the case of a COMMUNICATION PROCESSOR remote boot, or a local power-up reset, this function returns a zero value.

3.7.7 MOP Element Types 4, 5: Read, Write Boot Password

The MOP element types 4 and 5 enable the host software to read and write the MOP boot verification password. This password is used only in Normal mode (mode switch S3) with remote boot enabled (option switch S4).

The boot password enables the DELQA to filter MOP remote boot messages. The default password is all 0s, which permits the DELQA to act on any MOP remote boot message.

The length of the password must be eight bytes.

Table 3–9 MOP Element Types 4, 5

Offset	Bits	Description
MEBB+0	PW<15:00>	Eight sequential bytes of the MOP remote boot password (least-significant hex-digit first)
MEBB+2	PW<13:16>	
MEBB+4	PW<47:32>	
MEBB+6	PW<63:48>	

3.7.8 MOP Element Type 6, 7: Read/Write System ID

MOP Element types 6 and 7 enable the host software to read and write the MOP system ID message.

The buffer specified for READ must be at least 256 (decimal) bytes.

The other fields may be broken down into individual Info units:

OTHER INFO TYPE   OTHER INFO LENGTH   OTHER INFO VALUE

The order in which Info units are arranged is not important, but the variable sizing of the units must be observed. The overall size of the MEB in the setup packet determines the number of Info elements specified.

Table 3-10 lists the Info types and Table 3-11 lists the Info value descriptions. For further details, refer to the *Maintenance Operations Protocol (MOP) Functional Specification.*

Table 3–10 MOP Element Types 6, 7

Offset	Bits	Description	
MEBB+0	IT<15:00>	Info type (binary value)	
		Type	**Information**
		0	Termination of other information type list
		1	Maintenance Version
		2	Functions
		3	Console User
		4	Reservation Timer
		5	Console Command Size
		6	Console Response Size
		7	Hardware Address
		8	System Time
		100	Communication Device
		101 to 199	Communication device related
		200	Software ID
		201 to 299	Software ID related
		300	System Processor
		301 to 399	System processor related
		400	Data Link
		401	Data Link Buffer Size
		402 to 499	Data link related
		Types 1, 2, 8, and 100 are required fields; types 3, 4, 5, and 6 are required for console messages.	
MEBB+2	IL<07:00>	Info Length in bytes of Info Value field (binary value)	

Table 3–10 (Cont.) MOP Element Types 6, 7

Offset	Bits	Description
MEBB+4	IV<15:08>	Info Value of
MEBB+6	IV<31:00>	up to 16 bytes
MEBB+10	IV<31:00>	
MEBB+12	IV<31:00>	
MEBB+14	IV<31:00>	
MEBB+16	IV<31:00>	
MEBB+20	IV<31:00>	
MEBB+22	IV<31:00>	
MEBB+24	IV<07:00>	

Table 3–11 Information Value Descriptions

Type	Bytes	Description
001	3	Maintenance Version Number (binary) Byte 1 Version number (lowest byte) Byte 2 ECO Byte 3 User ECO
002	2	Functions The bits indicate functions as follows: 0 Loop 1 Dump 2 Not supported by the DELQA 3 Multi-block loader (tertiary loader or system) 4 Boot 5 Console carrier 6 Data link counters 7 Console carrier reservation
003	6	Console User The system address of the system that has the console reserved. The mandatory field when the console carrier is available (Function bit 5). Invalid if the console carrier is not reserved (Function bit 7).
004	2	Reservation Timer The maximum value (in seconds) of the timer used to clear unused console reservations. The mandatory field when the console carrier is available (Function bit 5).
005	2	Console Command Size The maximum size of the console command buffer. The mandatory field when the console carrier is available (Function bit 5).
006	2	Console Response Size

Table 3–11 (Cont.) Information Value Descriptions

Type	Bytes	Description
		The maximum size of the console response buffer. The mandatory field when the console carrier is available (Function bit 5).
007	6	Hardware Address
		An address in the SA ROM on the DELQA (read-only)
008	10	System Time
		A segmented binary system time stamp.
100	1	Communication Device
		The hardware device type of the host channel in use (decimal for the DELQA). For DELQA Info Type=37 (for DEQNA Info Type=5). (Read-only)
101 to 199	16 (max)	Communication device related
		Information specific to the particular communication device.
200	17 (max)	Software ID
		The identification of the software the system is supposed to be running.
201 to 299	16 (max)	Software ID related
		Information specific to the particular software ID. Interpretation is specific to the receiving system; for example, a file specification may vary depending on the type of file server.
300	1	System Processor = **type**
		The type of system processor.
301 to 399	16 (max)	System processor related
		Information specific to the particular system processor.
400	1	Data Link
		The data link protocol; in this case, Ethernet.
401	2	Data Link Buffer Size
		The size of the data link buffer.
402 to 499	16 (max)	Data link related
		Information specific to the particular data link.

3.7.9 MOP Element Types 8, 9: Read, Read/Clear Counters

MOP element types 8 and 9 read the datalink counters which the DELQA maintains on-board in its shared RAM. A MOP element type 8 does not affect the state of the DELQA counters; a MOP element type 9 clears the counters after copying them to the MEB.

The buffer must be at least 100 (decimal) bytes long.

Counter values are unsigned integers. Counters latch at their maximum values to indicate overflow. For 16-bit counters, the order of bytes is:

Byte 1 — Lower 8 bits of counter
Byte 2 — Higher 8 bits of counter

For 32-bit counters, the order of words is:

Word 1 — Lower 16 bits of counter
Word 2 — Higher 16 bits of counter

The DELQA counters are in the contiguous format described in Table 3-12.

Table 3–12 MOP Elements Type 8, 9 MEBB Format

Count	Specification	
1	Seconds Since Last Zeroed	16 bits
	The number of seconds since the counters were last zeroed	
2	(Data) Bytes Received	32 bits
	The total number of data bytes received error free, excluding the data link protocol overhead	
3	(Data) Bytes (Sent) Transmitted	32 bits
	The total number of data bytes successfully transmitted, excluding the data link protocol overhead, and not counting data-link-generated retransmissions, but including transmissions in which the collision test signal failed to set	
4	Packets (Frames) Received	32 bits
	The total number of datagrams received error free.	
5	Packets (Frames Sent) Transmitted	32 bits
	The total number of datagrams successfully transmitted, including transmissions in which the collision test signal failed to set.	
6	Multicast Bytes Received	32 bits
	The total number of multicast data bytes received error free, excluding the data link protocol overhead	
7	Multicast Packets (Frames) Received	32 bits

Table 3–12 (Cont.) MOP Elements Type 8, 9 MEBB Format

Count	Specification	
	The total number of multicast datagrams received error free.	
8	Packets Transmitted: (Initially) Deferred	
	The total number of datagrams successfully transmitted on the first attempt after deferring, including transmissions in which the collision test signal failed to set.	
9	Packets Transmitted (single collision): 2 Attempts	32 bits
	The total number of datagrams successfully transmitted on two attempts, including transmissions in which the collision test signal failed to set.	
10	Packets (multiple collisions) Transmitted: 3+ Attempts	32 bits
	The total number of datagrams successfully transmitted on three or more attempts, including transmissions in which the collision test signal failed to set.	
11	Transmit Packets Aborted (Send failure)	16 bits
	The total number of datagrams aborted during transmission for one or more of the bitmapped errors.	
12	Transmit Packets Aborted (Send Failure) Bitmap	
	Bit <00> RTRY	Excessive Collisions: Retry error after 16 unsuccessful transmission attempts.
	Bit <01> LCAR	Loss of Carrier (Carrier check failed): Retry error (after 16 unsuccessful transmission attempts), loss of carrier flag, and non-zero TDR value on last attempt.
	Bit <02> = 0	Short Circuit (not supported on the DELQA).
	Bit <03> = 0	Open Circuit (not supported on the DELQA).
	Bit <04> MLEN	Data Block Too Long. The DELQA aborted the transmission because the datagram exceeded the maximum packet size.
	Bit <05> LCOL	Remote Failure to defer: late collision on the last transmission attempt.
	Bits <15:06> = 0	Undefined

Table 3–12 (Cont.) MOP Elements Type 8, 9 MEBB Format

Count	Specification	
13	Packets Received with Error (Receive Failure)	16 bits
	The total number of datagrams received with one or more errors logged in the bitmap, including only those datagrams that passed destination address comparison.	
14	Packets Received with Error (Receive Failure Bitmap)	
	Bit <00> CRC	Block Check Error: a datagram failed the CRC check.
	Bit <01> FRAM	Framing Error: a datagram failed the CRC check and did not contain an integral multiple of eight bits.
	Bit <02> MLEN	Message Length Error (Frame too long): a datagram was larger than 1518 bytes.
	Bits <15:03> = 0	Undefined
15	Reserved for Host Counter Word	16 bits
	The host software for the DNA Network Management layer should maintain the DECnet MOP counter for **Unrecognized frame destination error**. This indicates that a packet was received by the DELQA, passed Ethernet destination address filtering, but failed Ethernet protocol type filtering. The host software is responsible for Ethernet protocol type filtering.	
16	Receive Packet Lost: Internal Buffer Error (Data Overun)	16 bits
	The total number of times that an incoming packet was discarded due to lack of internal buffer space. Incoming packets must be error-free to be counted.	
17	Receive Packet Lost: Local Buffer Error (System Buffer Unavailable)	16 bits
	The total number of times that there was a problem with a receive list data buffer. This counter is incremented on one of more of the following occurrences.	
	Buffer Unavailable	A datagram was lost because there was no available buffer on the receive list.
	Buffer Too Small	A datagram was truncated because it was larger than the available buffer space on the receive list.
18	Reserved for Host Counter Word	16 bits
	The host software should maintain the DECnet MOP counter for **User Buffer unavailable**. This indicates that a packet was received by the DELQA, delivered to the device buffer pool in the host memory, but discarded due to insufficient user-level receive buffers. The host software manages user-level buffers.	
19	Multicast Bytes Transmitted	32 bits

Table 3–12 (Cont.) MOP Elements Type 8, 9 MEBB Format

Count	Specification	
	The total number of multicast data bytes successfully transmitted, excluding data link protocol overhead, and not counting the DELQA generated retransmissions, but including transmissions in which the collision test signal failed to set.	
20	Reserved	16 bits
21	Reserved	16 bits
22	Babble Counter	16 bits
	Counter for the total number of times the DELQA LANCE reported the babble condition on the channel.	

CHAPTER 4
MAINTENANCE

4.1 SCOPE

This chapter describes the maintenance activities for the DELQA. The sections are as follows.

Section 4.2	Maintenance philosophy
Section 4.3	Built-in diagnostics
Section 4.4	Maintenance Operations Protocol (MOP): Network Support
Section 4.5	IEEE 802.3 Network Support: Null link-layer Service Access Points
Section 4.6	Network diagnostics
Section 4.7	Module diagnostics

WARNING
The procedures described in this chapter involve the removal of the system covers, and should be performed only by trained personnel.

ADVARSEL!
Ifølge de procedurer, som er beskrevet i dette kapitel, skal systemets beskyttelsesplader fjernes; dette bør kun udføres af personer der ved hvordan dette skal gøres.

WAARSCHUWING
Bij de procedures die in dit hoofdstuk worden beschreven dienen bepaalde delen van de systeemomhulling te worden verwijderd; dit mag uitsluitend worden gedaan door opgeleid personeel.

VAROITUS!
Tässä luvussa kuvatut toimenpiteet liittyvät järjestelmän suojakansien irrottamiseen. Ainoastaan koulutettu henkilökunta saa suorittaa nämä toimenpiteet.

AVIS!

Ce chapitre décrit les interventions qui demandent que les couvercles extérieurs des appareils soient enlevés. Ces travaux devraient être mis en main uniquement par des techniciens expérimentés.

VORSICHT!

Bei der Ausfuhrung der in diesem Kapitel beschriebenen Anweisungen mussen die Systemabdeckungen entfernt werden. Dies sollte nur von geschultem Personal ausgefuhrt werden.

אזהרה

הפעולות המתוארות בפרק זה, כרוכות בהסרת המכסים של המערכת ויבוצעו אך ורק על ידי אדם מוסמך.

ATTENZIONE

La procedura descritta in questo capitolo comporta la rimozione delle coperture e deve essere eseguita solo da personale specializzato.

注意

本章では，本体カバーの取り外し等について述べてあります。作業は，必ず専門の担当者によっておこなって下さい。

ADVARSEL

I dette kapitlet beskrives bl. a. hvordan man fjerner dekslene rundt systemet. Dette arbeidet må bare utføres av fagfolk.

AVISO

Os procedimentos descritos neste capítulo respeitam à forma como se retiram as protecções do sistema. Dada a sua especificidade, recomendamos que seja executado por pessoal especializado.

!ATENCION!
Los procedimientos descritos en este capitulo incluyen el desmontaje de las cubiertas del sistema y debe ser realizado solamente por personal entrenado.

VARNING
I detta kapitel beskrivs hur systemkaapan tas bort. Detta faar endast utfoeras av utbildad personal.

4.2 MAINTENANCE PHILOSOPHY

4.2.1 Preventive Maintenance

There are no preventive maintenance procedures for the DELQA module. However, when the host system is serviced it is good practice to check the DELQA installation for loose connectors, damaged cables, and similar faults.

4.2.2 Corrective Maintenance

The DELQA module has been designed to enable diagnostics to determine a faulty Field Replaceable Unit (FRU) rapidly. Corrective maintenance in the field therefore consists of changing FRUs. Component replacement in the field is not intended and is not recommended.

The diagnostic tests are processor-specific.

- For PDP-11 host processors

  Network testing
  DECnet Network Control Program (NCP)
  Network Interconnect Exerciser (NIE) running under Diagnostic Runtime Services (DRS)

  Module testing
  Field functional diagnostic (ZQNA??) running under diagnostic runtime services (DRS)
  DEC/X11 Exerciser.

- For MicroVAX processors

  Network testing
  DECnet Network Control Program (NCP)
  Network Interconnect Exerciser (NIE) running under the MicroVAX Diagnostic Monitor (MDM)

  Module testing
  MicroVAX Diagnostic Monitor (MDM)

4.2.3 Field Replaceable Units (FRUs)

The Field Replaceable Units (FRUs) are:

- The DELQA module
- Bulkhead assembly fuse
- Cabinet kit
- Transceiver and transceiver cable (or bulkhead loopback connector).

Figure 4-1 shows the field replaceable units in the DELQA installation.

NOTE
Early versions of the DEQNA diagnostics are not compatible with the DELQA. For PDP-11 processors use ZQNAI or later. For MicroVAX processors use Diagnostic release 124 or later.

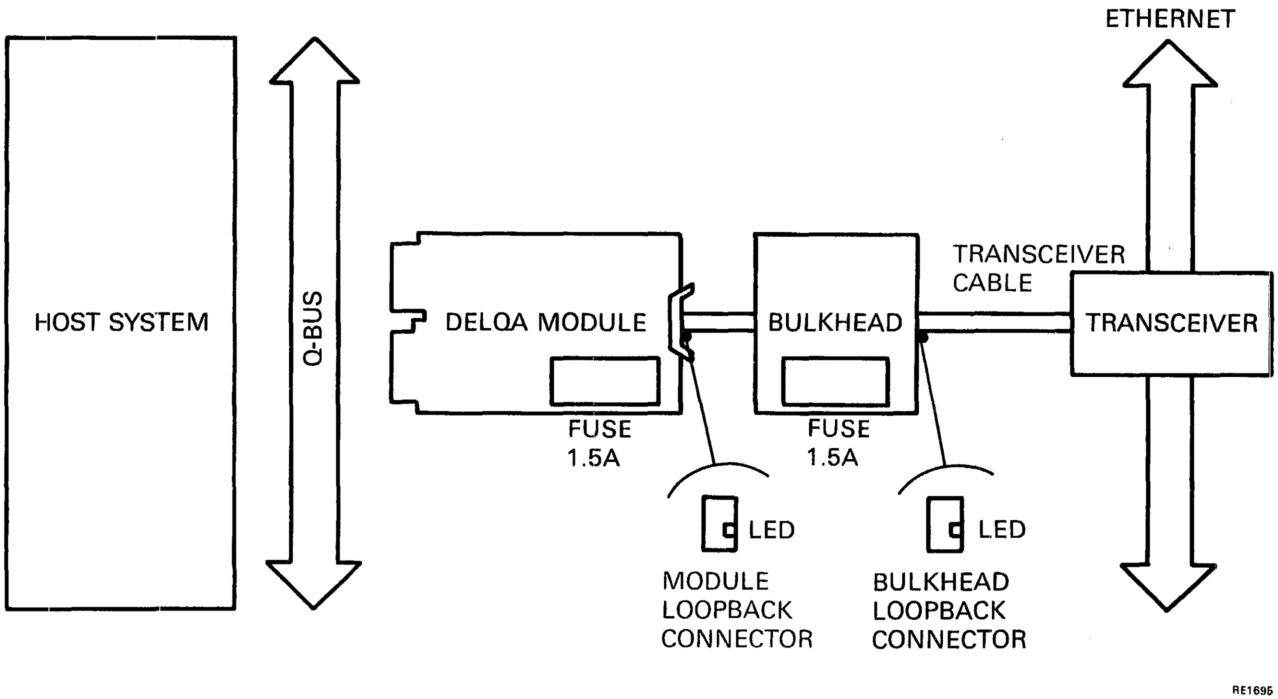


Figure 4–1 Field Replaceable Units (FRUs)

Refer to Chapter 2.3.1 for correct fuse details.

4.2.4 Diagnostic Procedure

The general strategy for identifying a fault is as follows:

1. Check the DELQA configuration to ensure that the system can identify the module correctly.
2. Run the module test(s) to test for faulty FRUs.
3. Run the network test(s) to test for faults in network configuration and/or operation.

4.3 SELF-TEST

The DELQA has a comprehensive self-test which is executed at powerup in Normal mode only. In addition, in Normal mode, the self-test can be requested by the host operating system software through the DELQA Q-bus registers. The test takes about five seconds to run and consists of the following sections.

1. The ROM-32 checksum test checks for corrupted ROM content.
2. The RAM test checks memory addressing and operation.
3. The 68000 microprocessor test checks for correct execution of 68000 instructions and handling of CPU exceptions.
4. The QIC function tests check QIC programming functions.
5. The QNA2 function tests check:
   - Access to the DELQA CSR
   - Sanity timer operation
   - Access to the SA ROM
   - QNA2 interrupts to the 68000 when the host accesses BDLs.
6. The SA ROM test verifies the checksum on the SA ROM.
7. The LANCE/SIA subsystem tests check:
   - The LANCE internal Control and Status Register
   - The LANCE subsystem address and data paths
   - The CRC generation circuitry (using correct and incorrect CRCs)
   - The notification of RETRY error following collisions on 16 successive attempts to transmit a packet
   - The broadcast, multicast, and physical address filtering
   - The internal loopback
   - The external loopback.

4.3.1 Extended Primary Bootstrap

A PDP-11 host can boot from a DELQA using a method similar to that for a mass-storage device. Part of the BD ROM on board the DELQA contains PDP-11 code for bootstrapping a PDP-11 from the network. This part of the boot/diagnostic BD ROM is made up of three sections.

- The Extended Primary bootstrap (EPB) code
- The DELQA citizenship (CQ) test code
- The DECnet secondary loader code.

The host primary boot code passes control to the EPB code (loaded from the BD ROM), which then loads and verifies the complete contents of the BD ROM into host memory. The EPB code then initiates the CQ test before allowing the DELQA to access the Ethernet.

4.3.2 Citizenship Test

The DELQA citizenship test (CQ) is a series of diagnostic test routines that determine whether the DELQA is operating correctly and can access the Ethernet, or is faulty and requires further diagnosis. Test results are indicated in part by the LEDs on the DELQA, and complete test reports are returned to host register R0.

The CQ test uses internal loopback, internal extended loopback, and external loopback modes, and requires the DELQA and an H4xxx transceiver (or equivalent); connection to the Ethernet is required. The sanity timer is enabled for testing but is not expected to time out. If the timer does time out it is an error. When all testing is complete, the sanity timer is turned off, unless switch S3 is open, in which case it is left on.

The CQ test is a free-standing subroutine and can be called by other software. For example, during network boot, CQ can determine if the node should be allowed to proceed from the initialized state to either a functional or a nonfunctional state.

4.3.2.1 Citizenship Test Descriptions

The citizenship tests are described in the list that follows. The corresponding error bit values that appear in host register R0 are also given.

T1: **Station Address Verification**
The default physical address is verified and copied from the Station Address (SA) ROM into a test packet for later use. If this test fails, testing continues until the final external loopback test or another test failure occurs. Possible errors are:

R0 Bit	Error Description
00	Station address is all zeros, or all ones, or is not a valid DELQA address

T2: **Device Interrupt Test**
A transmit descriptor is given to the UUT after interrupts are enabled. The UUT should generate a transmit interrupt. Possible errors are:

R0 Bit	Error Description
11	No interrupt occurred, or interrupt occurred prematurely, or wrong interrupt occurred

T3: **Setup Mode and Receive Processing Test**
A series of setup packets with a repeating test pattern for checking stuck-at faults is transmitted to the UUT. The patterns are varied so that each byte in the station address memory is tested with all patterns. Possible errors are:

R0 Bit	Error Description
12,01	Setup packet echoed data check
09,12,01	Setup packet operation timeout
14,12,01	Setup operation status check

T4: **Internal Loopback and Address Filter**
A setup packet is generated with all target addresses identical and based on a pattern of one walking bit. This packet is set up in the Unit Under Test (UUT). Then, two internal loopback packets are generated and transmitted for each address in the pattern. The first packet is addressed to the complemented target address, which is not in the pattern, and must be correctly transmitted and received as a runt. The second packet is addressed to a target address in the pattern, and must be correctly transmitted and received.

The test is repeated 48 times with a walking bit of one (other bits zero) advanced by one bit each time, and then with a walking bit of zero (other bits one). Possible errors are:

R0 Bit	Error Description
02	Transmit/receive data compare check.
11	Unexpected receive interrupt
09,02	True packet transmission and receive error
12,02	Setup packet echoed data check
14,02	False packet receive error

T5: **Internal Extended Loopback and Protocol**
The Unit Under Test (UUT) is put in internal extended loopback mode and packets of varying (increasing) length are circulated through the transmitter and receiver. The packets are made up of bit patterns designed to show stuck-at conditions and faults in the buffer and FIFO processing. The received packets are verified to be sure that the data was properly transferred. The packet length starts at the minimum Ethernet packet size and continues until beyond the maximum size. Possible errors are:

R0 Bit	Error Description
03	General packet transmit/receive data compare check
03	Long packet not detected
09,03	Test packet transmit or receive timeout
14,03	General operation status check
14,03	Long packet not detected via operation status

T6: **DMA to Q-bus Interface Processing**
An internal extended loopback packet with the station address is transmitted using a chained descriptor, with buffers, elements and high/low bytes. This packet is received and verified. Possible errors are:

R0 Bit	Error Description
04	Transmit (scatter/gather) data check
09,04	Transmit (special) and receive timeout
14,04	Receive or transmit operation status check

T7: **Transceiver Operational and Status**
A setup packet with the physical address of the Unit Under Test (UUT) is generated and looped back through the DELQA. The packet also turns off LED 2 and sets the sanity timer value reset to 1/4 second. CSR13 (Carrier from Receiver) is monitored to be sure it is cleared; or, if it is set, that it is cleared within approximately 100 microseconds. Possible errors are:

R0 Bit	Error Description
12	Setup packet echo data check
09,12	Setup packet operation timeout
14,12	Setup packet operation status check
15	CSR carrier bit on for too long

T8: **External Loopback and Ethernet Protocol**
This test is executed only if no other errors have been detected.

The physical address of the Unit Under Test (UUT) is assumed to be set up by T7. The next minimum-size Ethernet packet, addressed to the UUT with a data pattern of descending-integers, is transmitted and received using external loopback. Finally, the maximum-size Ethernet packet is generated and sent to the UUT. The maximum packet is addressed to the UUT and has a data pattern of descending integers. Both packets will test Ethernet protocol processing, and the maximum packet will test the transmit FIFO memory. Possible errors are:

R0 Bit	Error Description
15	External loopback not operational
05	External loopback transmitted/received packet data compare check
09,05	External loopback operation timeout
14,05	Extérnal loopback operation status check

4.3.2.2 Citizenship Test Results

The CQ test either executes successfully or fails, as follows.

a. CQ test successful: the value of host register R0 is zero and the DELQA is set up as follows.

- All three DELQA module LEDs are off. (See steps 2, 3, 15, and 16 from Sections 2.4.3 and 2.4.4, to gain access to these LEDs.)
- All 14 target addresses are set to the physical address from the station address ROM.
- The sanity timer is set to its default interval (four minutes) and disabled or enabled, according to the setting of option switch S3.
- Modes for promiscuous and all multicast address filtering are off.
- The DELQA has been reset.
  — Receive is disabled
  — Transmit is disabled.

b. CQ test fails: the LED indicators display the following error codes.

LED1	LED2	LED3	Definition
OFF	OFF	OFF	(Step 4) CQ test passed
OFF	OFF	ON	(Step 3) External loopback test failed
OFF	ON	ON	(Step 2) DELQA internal error
ON	ON	ON	(Step 1) Cannot upload the BD ROM contents, or the first setup packet prefill failed

The bits in register R0 indicate the test that failed. If bit 15 is the only bit set, the DELQA passed all the CQ tests except those which require a connected transceiver.

The error definitions are listed in Table 4-1.

Table 4–1 Citizenship Test: Error Bit Definitions

Bit	Error Definition and Source(s)
15	External loopback not operational (Tests 7 and 8)
	Ethernet not operational H4000 not operational (blown fuse, disconnected)
14	Operation completion status check (all tests)
	CSR status after final reset not nominal CSR status after transmit and/or receive not nominal Receive descriptor flags and status word 1 not nominal Received byte length check Transmit descriptor flags and status word 1 not nominal TDR value = 0
13	Sanity timer interrupt (general error)
	Power failed during test Unexpected sanity timer interrupt
12	Setup packet or target address echo check (all tests)
	Setup packet transmit timeout Transmit status not nominal Setup packet receive timeout Receive status not nominal Echoed data not identical to transmitted data Extra word at end of setup packet not nominal
11	Spurious or missing device interrupt (general error)
	Expected device interrupt not detected Device did not detect nonexistent memory (NXM) bus state 18-bit or 22-bit addressing failure Unexpected DELQA device interrupt

Table 4–1 (Cont.) Citizenship Test: Error Bit Definitions

Bit	Error Definition and Source(s)
10	Bus timeout or NXM interrupt (general error) I/O page not accessible for read or write Cannot read station address ROM Test code attempted to access nonexistent memory
09	Device operation timeout (all tests) Unit under test failed to complete a transmit and/or receive in time
08	Undefined
07	Self-test error
06	Final operation failed to clear device
05	Ethernet external loopback test check (Test 8) Ethernet protocol processing check Ethernet minimum valid length processing check Ethernet maximum valid length processing check
04	DMA-to-Q-bus interface processing check (Test 6) DMA odd/even length and address processing check Multi-element transmit descriptor processing check Chained transmit descriptor processing check
03	Internal extended loopback transmit buffer data check (Test 5) Ethernet protocol processing check Transmit buffer memory malfunction Packet size processing error (protocol error)
02	Station address compare test check (Test 4) Address filter logic passing all addresses Address filter logic not passing expected addresses
01	Station address/receive processing check (Test 3) Target address RAM malfunction Packets not properly stored in receive buffer Receive memory malfunction
00	Invalid Ethernet station address (Test 1) I/O page register read failure (see also bit 10) Unit under test is not a DEQNA (M7504) Station address ROM malfunction Invalid DELQA address

4.4 MAINTENANCE OPERATIONS PROTOCOL (MOP): NETWORK SUPPORT

In Normal mode the DELQA implements Maintenance Operations Protocol (MOP) functions in response to the following remote console messages from other nodes on the Ethernet.

- The request system ID message
  The DELQA responds by transmitting its current system ID message.
- Remote boot trigger instruction
  The DELQA may respond to a trigger instruction only if option switch S4 is open to enable remote boot. The instruction can only be implemented if the host system has the appropriate boot ROM.
- Loopback request message
  The DELQA will respond to a loopback request message.

The DELQA also transmits its current system ID parameters automatically every 8 to 10 minutes.

This section describes the Ethernet messages that initiate these functions, and how the functions are executed. Table 4-2 lists the message types.

For further information, refer to the DECnet maintenance operation protocol (MOP) functional specification.

Table 4–2 Maintenance Operation Protocol (MOP) Messages

Message
Request System ID
System ID
Remote Boot
Loopback Request

NOTE

The DELQA ROM firmware does not support the following MOP functions, which are required for remote boot operations using non-system boot ROM:

- **Program Request (outbound from the DELQA).**
- **Memory Load with Transfer address (inbound to the DELQA).**

4.4.1 MOP Remote Console Message: Request System ID

The DELQA responds to a remote console request for system ID by transmitting its current system ID parameters, which it holds in its on-board shared RAM. The processing of this request/response protocol returns the receipt value specified in the request, together with its system ID.

Figure 4-2 shows the format of the request system ID message, and Table 4-3 gives details of the contents.

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
PAD DATA	43 BYTES
CRC	4 BYTES

RE1698

Figure 4–2 Request ID Message Format

Table 4–3 Request ID Message Format

Field	Length	Description
DESTINATION ADDRESS	6	The Ethernet physical address of the DELQA
SOURCE ADDRESS	6	The Ethernet 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	Function code for request ID value = 05 hex
RESERVED	1	Value = 00 hex
RECEIPT NUMBER	2	Receipt number that identifies the request
PAD DATA	43	Pad characters (anything to pad the message out to 64 bytes)
CRC	4	Incoming block check character

4.4.2 MOP Remote Console Message: System ID

In Normal mode, the ROM-based firmware in the DELQA module sends a system ID message every 8 to 10 minutes to the remote console server multicast address. These messages contain the device type and Ethernet address of the host system. This information may be modified by including MOP element types 5 or 6 in a setup packet.

Certain DECnet and Network Interconnect Exerciser (NIE) utilities can map the nodes on an Ethernet network by listening for these messages.

Figure 4-3 shows the format of the system ID message, and Table 4-3 gives details of the contents.

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
FUNCTION: TYPE	2 BYTES
FUNCTION: LENGTH	1 BYTE
FUNCTION: VALUE 1	1 BYTE
FUNCTION: VALUE 2	1 BYTE
HARDWARE ADDRESS: TYPE 2 BYTES	2 BYTES
HARDWARE ADDRESS: LENGTH 1 BYTE	1 BYTE
HARDWARE ADDRESS: VALUE	6 BYTES
DEVICE: TYPE	2 BYTES
DEVICE: LENGTH	1 BYTE
DEVICE: VALUE	1 BYTE
PAD/PARAMETERS	146 BYTES
CRC	4 BYTES

RE1699

Figure 4–3 System ID Message Format

Table 4-4 System ID Message Format

Field	Bytes	Description
DESTINATION ADDRESS	6	The Ethernet physical address of the requesting station or the remote console service multicast address. Value = AB-00-00-02-00-00 hex. = (00AB) (0200) (0000)
SOURCE ADDRESS	6	The Ethernet physical address of the DELQA
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 hex to (05DA) DA-05 hex
CODE	1	Function code for system ID Value = 07 hex
PAD OF ZERO	1	Value = 00 hex
RECEIPT NUMBER	2	A receipt number to identify the request
MOP VERSION:		
TYPE	2	Value = (0001) 01-00 hex
LENGTH	1	Value = 03 hex
VERSION	1	Value = 03 hex
ECO	1	Value = 01 hex
USER ECO	1	Value = 00 hex
FUNCTION:		
TYPE	2	Value = (0002) 02-00 hex
LENGTH	1	Value = 02 hex
VALUE 1	1	Value = See functions bit mask in MOP element type S: write system ID
VALUE 2	1	Value = 00 hex

Table 4–4 (Cont.) System ID Message Format

Field	Bytes	Description
HARDWARE ADDRESS:		
TYPE	2	Value = (0007) 07-00 hex
LENGTH	1	Value = 06 hex
VALUE	6	Default the DELQA physical address from SA ROM
DEVICE:		
TYPE	2	Value = 37 decimal for the DELQA (Certain DELQAs can transmit 37 Octal from a PDP Host)
LENGTH	1	Value = 01 hex
VALUE	1	The DELQA device code. Value = 11 hex
PAD/PARAMETERS	146	The set of additional parameters supplied by host software through the setup packet. If not supplied, zeros are added by the DELQA
CRC	4	Outgoing block check character

4.4.3 MOP Remote Console Boot Message

In Normal mode with option switch S4 open to enable remote boot, the DELQA processes MOP remote console boot messages as follows.

1. Validate the boot verification code.
2. Force a host system reboot by negating BDCOK on the Q-bus.

If option switch S4 is closed to disable remote boot, and a MOP remote console boot message is received, the DELQA firmware delivers this message to the host software as it would with any normal datagram received. However, normal datagram service occurs only if CSR00 (Receiver Enable) is set and sufficient receive buffers are available to the DELQA in host memory.

The DELQA does not support any other modes of MOP remote boot processing.

The DELQA supports MOP remote boot on any system which supports either the DEQNA or DELQA in the host CPU boot ROM. This includes the following systems.

- PDP-11 systems which have system boot ROM support for the DEQNA. Either type of MOP boot (system processor or communications processor) may be used.
- MicroVAX II systems which have system boot ROM support for the DEQNA. Either type of MOP boot (communications processor or system processor) may be used.

Figure 4-4 shows the format of the boot ID message, and Table 4-5 gives details of the contents.

DESTINATION ADDRESS	6 BYTES
SOURCE ADDRESS	6 BYTES
TYPE	2 BYTES
CHARACTER COUNT	2 BYTES
CODE	1 BYTE
VERIFICATION	8 BYTES
PROCESSOR	1 BYTE
CONTROL	1 BYTE
SOFTWARE ID	1 BYTE
PAD DATA	32 BYTES
CRC	4 BYTES

RE1700

Figure 4–4 Boot ID Message Format

Table 4–5 Boot ID Message Format

Field	Length	Description
DESTINATION ADDRESS	6	The physical Ethernet address of the DELQA
SOURCE ADDRESS	6	The physical Ethernet 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 hex
CODE	1	The function code for the boot message. Value = 06 hex
VERIFICATION	8	The code to be compared against the verification code supplied by host software. Before boot loading, the DELQA checks that the codes match. If the host software has not supplied a verification code, or the code is zero, the DELQA accepts any value in the verification field of the boot message
PROCESSOR	1	Value = 00 hex: system boot Value = 01 hex: communication boot

Table 4–5 (Cont.) Boot ID Message Format

Field	Length	Description
CONTROL	1	Value = 00 hex: boot from 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

4.4.3.1 Processing a Remote Message

When a message with TYPE = remote Console and CODE = Boot is read into a DELQA buffer, the DELQA firmware processes it as follows:

1. If the DELQA option switch S4 is closed to disable remote boot the incoming boot trigger message is treated as a normal datagram and delivered to the host.
2. If there are any receive errors (for example, CRC error) with the boot message, the message is treated as a normal datagram and delivered to the host.
3. If the MOP protocol message fields for character count, processor, control, and software ID contain values within the expected limits, boot processing continues. Otherwise, boot processing stops, and the message is delivered to the Host.
4. The DELQA compares the verification code contained within the boot message with the value stored by the DELQA RAM. The stored value is either specified by a previous setup packet, or the default value of zero which permits all received MOP remote boots to be accepted.

   Host software must specify a setup packet with a nonzero boot code in order to use the filtering function provided by the MOP boot verification code.

   If the DELQA firmware cannot match the verification code, the message is delivered to the Host.
5. The DELQA decodes the processor field from the boot message to determine whether the communication processor or the system processor is to be booted.
   - If the system processor is to be booted, the DELQA negates BDCOK (causing a system power fail trap). The remote boot message is stored in its on-board shared RAM so that host firmware can poll the DELQA for the source of the boot initiator (by sending a setup packet containing MOP element type 7).

     Within the boot message there may be the Ethernet physical address of the initiator and other server-specific information. A subsequent program request may call for the correct program from the correct remote loading host.
   - If the communications processor is to be booted, the DELQA negates BDCOK (causing a system power fail trap), performs a hardware reset, and executes its internal self-test. The remote boot message is not stored.

4.4.4 Ethernet Channel Loopback Protocol Support

The DELQA firmware supports the loop maintenance protocol by recognizing and replying to loopback messages transmitted over the Ethernet, and by host diagnostics, including the Network Interconnect Exerciser (NIE) and DECnet Network Control Program (NCP).

This feature assists in diagnosing Ethernet connection faults, especially for host systems that have no mass storage device, and so require all operating and diagnostic software to be downloaded.

Loop messages have the following fields.

- **TYPE** = unique loopback value (0090) ETHERNET PROTOCOL
- **DESTINATION:**

  Either, DESTINATION = the DELQA physical Ethernet address.

  Or, DESTINATION = Multicast address for Ethernet loopback protocol. The DELQA does not check multicast addresses for the loopback type. Loopback messages received with multicast addresses are delivered to the host as normal datagram traffic.
- **FUNCTION:**

  Either, FUNCTION = Forward

  Forward messages are transmitted by the DELQA to the next node in the loop, but are not delivered to the local attached node.

  Or, FUNCTION = Reply

  Reply messages are not processed by the DELQA ROM firmware but they are delivered to the local attached node/host just as any other received datagrams.

When a message with TYPE = Loopback is read into a DELQA buffer, the DELQA firmware processes it as follows.

1. If the CRC is wrong, the message is treated as a normal datagram, and loopback processing stops.
2. If the destination address field contains either the physical address of the DELQA or the broadcast address, loopback processing continues.

   If the destination address field contains a multicast address the message is treated as a normal datagram, and loopback processing stops.
3. The function field is located by adding the value in the skip count field to the location of the skip count field plus 2.

   The action taken depends on the function value, as follows.

   Function value = 1:
   This indicates a loop reply to the DELQA. If the forward address field contains a physical address, the DELQA delivers the message to the local host using the receive buffers. Loopback processing then stops.

   Function value = 2:
   This indicates a message to be forwarded. If the forward address field contains a physical address, the DELQA does the following.

   a. Inserts the contents of the forward address field into the destination address field.
   b. Replaces the source address field with the physical address of the DELQA.
   c. Adds eight to the value of the skip count.
   d. Strips the four-byte CRC from the message.
   e. Transmits the resulting message, generating and appending a four-byte CRC.

If the function value is not 1 or 2, or the forward address field does not contain a physical address, loopback processing stops and the message is delivered to the Host as normal datagram traffic.

Figure 4-5 shows the MOP loopback message, and Table 4-6 gives details of the contents.

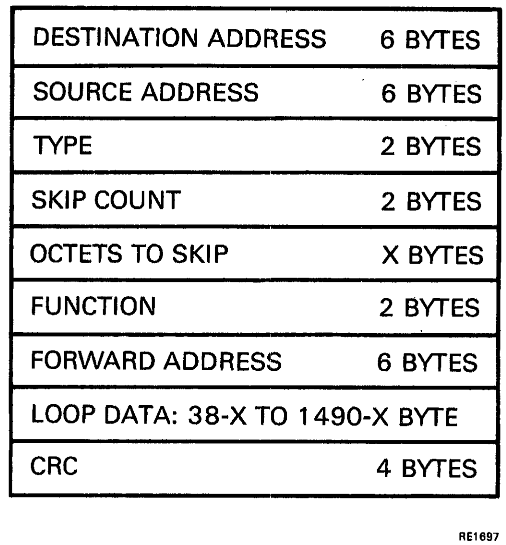


Figure 4–5 Loop Message Format

Table 4–6 Loop Message Format

Field	Length	Description
DESTINATION ADDRESS	6	Inbound: physical address of the DELQA, or the broadcast address. Outbound: forward address
SOURCE ADDRESS	6	Inbound: physical address of the loop-requesting station. Outbound: physical address of the DELQA
TYPE	2	Loop test message type. Value = (0090) 90-00 hex
SKIP COUNT	2	Inbound: offset to the function field. Outbound: offset plus 8
OCTETS TO SKIP	8n	Encapsulated loop header information. n = 0 to 186
FUNCTION	2	Value = (0001) 01-00 hex for reply. Value = (0002) 02-00 hex for forward
FORWARD ADDRESS	6	The physical address the inbound message is to be sent to. For a reply, this is the physical address of the DELQA
LOOP DATA	36 to 1490 −8n	Loop test data

Table 4–6 (Cont.) Loop Message Format

Field	Length	Description
CRC	4	Inbound: Block check character. Outbound: Block check character appended by the DELQA

4.5 IEEE 802.3 NETWORK SUPPORT: NULL LINK-LAYER SERVICE ACCESS POINTS

In NORMAL mode the DELQA implements IEEE 802.2 logical link control messages when they are received on a NULL Link-layer Service Access Point (LSAP) within an IEEE 802.3 standard local area network.

These messages can be used to interrogate and test many link layer service points per node. Therefore, IEEE 802.2 logical link control messages which are received on a non-NULL LSAP are passed on to the host system as normal datagrams.

For details of this message format and protocol, refer to the ANSI/IEEE Draft International Standard 802.2 Logical Link Control.

4.5.1 TEST Message

The IEEE 802.2 TEST message is similar in function to the loopback message on Ethernet networks.

4.5.2 XID (Transmit ID) Message

The IEEE 802.2 XID (Transmit ID) message is similar in function to the MOP remote console request system ID messages on an Ethernet network.

The DELQA does not broadcast IEEE 802.2 XID messages automatically as it does with MOP system IDs, since it is not required by the IEEE 802.2 protocols.

4.6 NETWORK DIAGNOSTICS

4.6.1 DECnet Network Control Program (NCP)

The DECnet Network Control program (NCP) provides a command-driven interface for executing loopback tests on the Ethernet, and for examining network and datalink counters.

Some of the relevant commands are:

- LOOP
- SHOW
- TELL
- TRIGGER.

The TRIGGER command may be used to initiate boot loading from the DELQA for PDP-11 host systems that have the appropriate boot ROM support.

The commands may be issued either from the local host system or, by using the TELL command, from a remote node. The functions are performed concurrently with other DECnet operations, and do not interfere with other Ethernet traffic (although there may be some degradation of throughput).

Refer to the DECnet System Manager's Guide for further information.

4.6.2 Network Interconnect Exerciser (NIE)

The Network Interconnect Exerciser (NIE) diagnostic program is used to determine the connectivity of nodes on the Ethernet; to determine the ability of nodes to communicate with each other; and to support node installation verification and problem isolation.

The NIE does not test the DELQA, but the communications link to which it is connected; therefore, the NIE assumes that the DELQA has successfully completed the citizenship test.

The NIE is used with XXDP+ and MicroVAX Diagnostic Monitor.

Refer to Appendix B for further information.

4.7 MODULE DIAGNOSTICS

The Field Replaceable Units (FRUs) that will be indicated by these diagnostics are:

- The DELQA or the DEQNA modules
- The cabinet kit
- The fuse.

4.7.1 MicroVAX Diagnostic Monitor (MDM)

The MicroVAX Diagnostic Monitor (MDM) offers a selection of menu-driven tests and utilities that may be run in verify or service modes. These are:

- Utilities for external loopback tests and NIE
- Service tests for external loopback
- Verify tests for:
  - Internal and internal extended loopback
  - Setup packet handling
  - Buffer Descriptor List (BDL) handling
  - DMA and interrupt handling
  - Transmit and receive circuitry and firmware
  - Address filtering
- Device exerciser for testing the DELQA simultaneously with other system devices

MDM prompts the operator when it needs to use an external loopback connector.

Refer to Appendix B for further information.

4.7.2 PDP-11 Field Functional Diagnostic (ZQNA??)

The field functional diagnostic program (ZQNA??) tests the DELQA in Q-bus systems. It attempts to isolate faults to the Field Replaceable Units (FRUs):

Tests are executed under the supervision of the Diagnostic Runtime Services (XXDP+), and controlled by an operator from a console (hard copy or video).

ZQNA?? is not an Ethernet network exerciser. The ZQNA?? verifies that the DELQA can execute Ethernet protocol, and that valid network traffic can be transmitted and received. The Network Interconnect Exerciser (NIE) provides a higher level of testing.

ZQNA?? tests the DELQA in all loopback modes: internal loopback and internal extended loopback modes, with or without an external loopback connector or transceiver connected.

External loopback mode is used with a connected transceiver or external loopback connector. Alternatively, external loopback mode can be used with a terminated transceiver that is not attached to a network cable. Executing ZQNA?? using external loopback mode in a system connected to a live Ethernet does not disrupt the Ethernet.

Refer to Appendix B for operating information.

4.7.3 PDP DEC/X11 Exerciser

The DELQA DEC/X11 Exerciser exercises one DELQA at maximum activity rates. It transmits and receives random-length packets (using either 18- or 22-bit physical address space). The DELQA transmits and receives the same packet.

For operating information, refer to Appendix B.

APPENDIX A
VECTOR ASSIGNMENTS

This appendix lists the rank of vector assignments for MicroPDP-11 systems.

The DELQA has a fixed I/O page address, as selected by the on-board switch S1, and uses a fixed vector of 120(octal) for the first DELQA and a floating vector assignment for the second DELQA. The floating vector assignments start at 300(octal), and are assigned by rank to the units on the host system. The rankings are shown in Table A-1; the highest ranks have the lowest numbers.

A.1 THE FLOATING VECTOR ASSIGNMENT

If a host node has a KXV11 and an RXV21 and a DELQA, the DELQA is allocated the third floating vector because it is third in rank.

A device may use both fixed and floating vectors and addresses, and the assigned rank may be different for a floating address and a floating vector.

The DELQA module is configured as a DMA device, in the same way as a DEQNA module. The first DEQNA/DELQA vector is fixed by host system software at 120(octal). Subsequent DEQNA/DELQA modules are assigned a floating vector with a rank of 47(octal), and should be configured at system start-up using the auto-configuration routines for floating vectors.

A.2 FLOATING VECTORS

The DELQA uses one 16-bit word for a vector address.

The vector assignment rules are as follows:

- Each device occupies vector address space equal to **Size** in words.

  For example, the DLV11-J occupies 16 words of vector space. If its vector was 300(octal), the next available vector would be at 340(octal).

- There are no gaps, except those needed to align an octal modulus.

Table A–1 Floating Vector Address Assignments

Rank	Device	Size (Decimal)	Modulus (Octal)
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 to DLV11-E	4	10
15	DJ11	4	10
16	DH11	4	10
17	VT40	8	10
17	VSV11	8	10
18	LPS11	12	10
19	DQ11	4	10

†KL11 or DL11 as console has a fixed vector.

Table A–1 (Cont.) Floating Vector Address Assignments

Rank	Device	Size (Decimal)	Modulus (Octal)
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/DMR11	4	10
27	DZ11/DZS11/DZV11, DZ32	4	10
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§
35	TS11, TU80	2	4§
36	LPA11-K	4	10
37	IP11/IP300	2	4§
38	KW11-C	4	10
39	RX11/RX211 RXV11/RXV21	2	4§
40	DR11-W	2	4
41	DR11-B	2	4§
42	DMP11	4	10
43	DPV11	4	10
44	ML11	2	4‡

§The first device of this type has a fixed vector. Any extra devices have a floating vector.

‡ML11 is a MASSBUS device which can connect to UNIBUS using a bus adapter.

Table A-1 (Cont.) Floating Vector Address Assignments

Rank	Device	Size (Decimal)	Modulus (Octal)
45	ISB11	4	10
46	DMV11	4	10
47	DEUNA DEQNA/DELQA	2	4§
48	KDA50/RQDX3	2	4§
49	DMF32	16	4
50	KMS11	6	10
51	PCL11-B	4	10
52	VS100	2	4
53	TU81	2	4
54	KMV11	4	10
55	KCT32	4	10
56	IEX	4	10
57	DHV11/DHU11	4	10
58	DMZ32/CPI32 (async)	12	4
59	CPI32 (sync)	12	4
60	QNA	12	4
61	QVSS	4	10
62	VS31	2	4
63	LNV11	2	4
64	QPSS	2	4
65	QTA	2	4
66	DSV11	2	4

§The first device of this type has a fixed vector. Any extra devices have a floating vector.

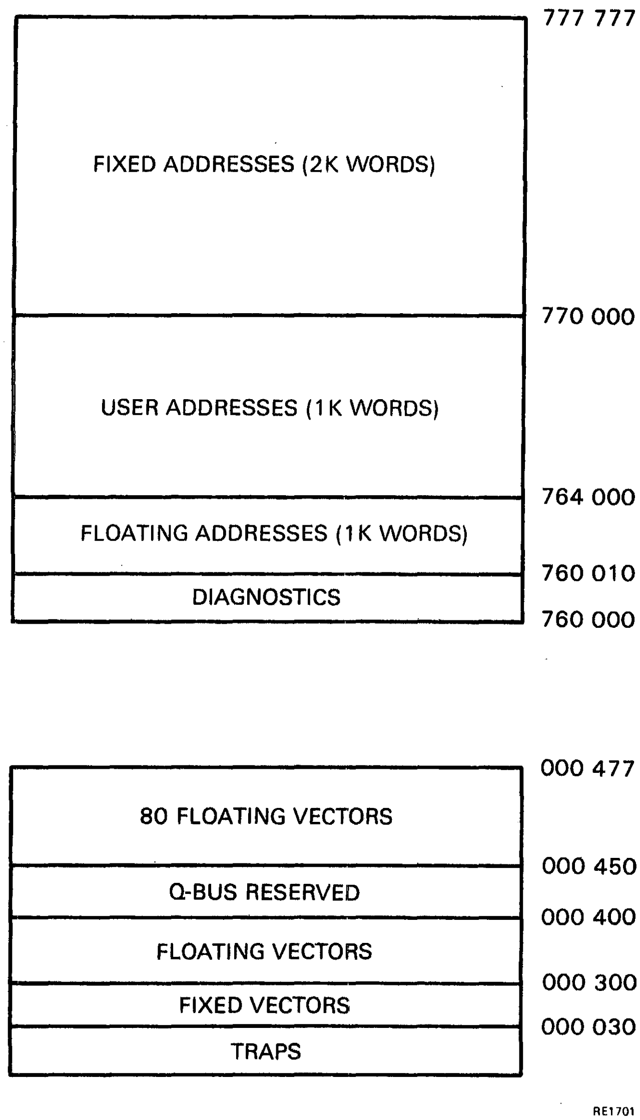


Figure A–1 Q-bus Address Map

APPENDIX B
DIAGNOSTICS

B.1 SCOPE

This appendix outlines the diagnostic tests available for troubleshooting the DELQA module. For further information, refer to the appropriate test handbook and/or system manual.

The sections are:

Section B.2	**Operating Environments**
Section B.3	**Network Interconnect Exerciser**
Section B.4	**PDP-11 Functional Diagnostic**
Section B.5	**MicroVAX Diagnostic Monitor (MDM)**
Section B.6	**DEC/X11 Exerciser**

B.2 OPERATING ENVIRONMENTS

B.2.1 PDP-11 Diagnostic Runtime Services (DRS)

The Diagnostic Runtime Services (DRS) are implemented by a program called XXDP+. To start this program, use the following procedure.

1. Boot XXDP+
2. Give the date
3. Type: **R NAME**

   where **NAME** is the name of the BIN or BIC file for this program; for example, CVNIABO for the PDP-11 Network Interconnect Exerciser (NIE).
4. Type: **START**
5. DRS prompts: **CHANGE HW (L) ?**

   Respond with **Yes** (unless the hardware information has been preloaded using the setup utility) and answer all the hardware questions that follow:

   # OF UNITS (D) ?

   Respond with the number of units to be tested (there is no default). At least one device must be specified for the program to run. To abort testing the device, type 0.

   DRS requests the following information for each device:

   BASE ADDRESS OF DELQA/DEQNA?

   Respond with the address of the I/O page register assigned for one of the DELQA devices; refer to Chapter 2 for details.

INTERRUPT VECTOR ADDRESS?

Respond with the DELQA interrupt vector address; refer to Appendix A.

WHAT IS THE PRIORITY LEVEL?

Respond with the DELQA interrupt priority level: 4

6. DRS prompts: **CHANGE SW (L) ?**

   Respond with N(o).

For a complete description of DRS, refer to the *XXDP+ User's Manual.*

B.2.1.1 DRS Commands

There are 11 DRS commands. The system can recognize a command by its first three characters; for example, you can type STA instead of START.

Table B-1 lists the DRS commands.

Table B–1 Diagnostic Runtime Services (DRS) Commands

Command	Description
ADD	Activate a unit for testing (all units are considered active at START time).
CONTINUE	Continue at the test that was interrupted (after CTRL/C).
DISPLAY	Print a list of all device information.
DROP	Deactivate a unit.
EXIT	Return to the XXDP+ monitor (XXDP+ operation only).
FLAGS	Print the state of all flags.
PRINT	Print statistical information (if implemented by the diagnostic).
PROCEED	Continue from an error halt.
RESTART	Start the diagnostic without initializing.
START	Start the diagnostic from an initial state.
ZFLAGS	Clear all flags.

B.2.1.2 DRS Switches

To modify supervisor operation, several switches can be appended to each DRS command. The system will recognize a switch by its first three characters. For example, you can type /TES:1–5 instead of /TESTS:1–5.

The switches can be used in combination, for example:

Example B-1 DRS Switch Combinations

START/TESTS:1–5/PASS:1000/EOP:100

executes tests 1 to 5, tests all units 1000 times, and prints the end-of-pass messages only after every 100 passes.

Table B-2 lists the DRS switches that can be used with each command, with a brief description of each. Table B-3 indicates the commands to which each switch applies.

Table B–2 Command Switches

Switch	Description
/EOP:*ddddd*	Report End-of-Pass message, and pass count and total errors, only after every *ddddd* passes.
/FLAGS:flag	Set the specified flag(s).
/PASS:*ddddd*	Execute *ddddd* passes, where *ddddd* = 1 to 65535 decimal.
/TESTS:*list*	Execute only the tests specified by *list* (a string of test numbers). For example: *START/TESTS:1:5:7–10* runs tests 1, 5, 7, 8, 9, and 10, and no others.
/UNITS:*list*	Test/ADD/DROP only those units (0 to 63) specified by list. For example: *START/UNITS:0:5:10–12* tests units 0, 5, 10, 11, and 12.

Table B–3 Switch Application

Commands			Switches		
	Tests	Pass	Flags	EOP	Units
ADD					X
CONTINUE		X	X	X	
DISPLAY					X
DROP					X
EXIT	(none)				
FLAGS	(none)				
PRINT	(none)				
PROCEED			X		
RESTART	X	X	X	X	X
START	X	X	X	X	X
ZFLAGS	(none)				

B.2.1.3 DRS Flags

Commands are used with the /FLAGS switch to set up certain operational parameters, such as "loop on error". The flags remain as specified by the last /FLAGS switch. All flags are cleared:

1. At startup, and remain cleared until explicitly set with the /FLAGS switch
2. After a START command, unless set with the /FLAGS switch with the ZFLAGS command
3. With the ZFLAGS command.

Flags can be specified in combinations. For example:

Example B-2 DRS Flag Combinations

/FLAGS:LOE:IER:BOE

causes the program to loop on error, inhibit error reports, and sound the bell on error.

The flags are listed and described in Table B-4.

Table B–4 Flags Application

Flag	Effect
ADR	Execute the autodrop code.
BOE	Sound the bell on error.
EVL	Execute evaluation (on diagnostics supporting evaluation).
HOE	Halt on error—return control to DRS command mode.
IBE	Inhibit all error reports except first level (first level contains error type, number, PC, test and unit).
IDR	Inhibit the program from dropping units.
IER	Inhibit all error reports.
ISR	Inhibit statistical reports (applies only to diagnostics which support statistical reporting).
IXE	Inhibit extended error reports called by PRINTX macros.
LOE	Loop on error. One error occurrence will cause the test to loop until the operator takes the program out of the loop.
LOT	Loop on test.
PNT	Print the test number as the test executes.
PRI	Direct messages to the line printer.
UAM	Unattended mode (no manual intervention).

B.2.2 MicroVAX Diagnostic Monitor (MDM)

The MicroVAX Diagnostic Monitor (MDM) is a bootable, menu-driven, maintenance and diagnostics system which runs the MicroVAX Diagnostic Monitor (MDM) as part of its optional Service version. For DELQA testing, MDM includes:

1. External loopback utilities tests, including NIE utilities
2. Functional tests
3. Exerciser tests.

The installation version of MDM is supplied as standard with MicroVAX systems. It provides two levels of testing.

1. Verification test for the configuration. A console display lists the devices found. If an installed device is missing from the list, its configuration details (address and vector assignments) and physical connections should be checked.
2. System-level functional and exerciser tests for all devices that are currently configured. Displays on the console, and LED indicators on the device itself, show the current test status of each device.

All tests are accessed from the main MDM menu display, using subsidiary menus to initiate Installation and Service tests.

Section B.5 describes the DELQA MDM tests in more detail.

Refer to the *MicroVAX Diagnostic Monitor* for further information about MDM.

B.3 NETWORK INTERCONNECT EXERCISER (NIE)

This is an overview of the Network Interconnect Exerciser (NIE) program for the DELQA. For more information refer to the appropriate Diagnostic, listing.

B.4 INTRODUCTION

The NIE diagnostic program is used to determine the connectivity of nodes on the Ethernet. It determines the ability of nodes to communicate with each other, and supports node installation, verification and problem isolation.

The NIE does not test the device (DELQA), but the communications link to which it is connected; therefore, the NIE assumes that the DELQA has passed device-specific diagnostics. If any hardware errors occur during execution, the NIE reports the error by message to the operator. Unless command to halt on error (see Section B.7.1.2), the NIE resumes testing where it left off after reporting the error. However, note that the NIE does not test the DELQA to its performance limits, diagnose problems, provide comprehensive hardware testing, nor identify a failed FRU.

The NIE runs under control of either the PDP-11 Diagnostic Runtime Services (DRS) software or MDM; therefore, it cannot run concurrently with any operating system, nor can anyone else use the system while the NIE is running. In addition, overall performance of the Ethernet can be degraded by running the NIE.

B.5 OPERATING MODES

The NIE is command-driven; that is, it executes commands given by the user. Commands are described in Paragraph Section B.7 In addition to entering commands, the user can select one of two operating modes: unattended or operator directed.

B.5.1 Unattended Mode

This mode allows Ethernet testing without operator interaction. The tests share a table comprising the physical addresses of the nodes to be tested (Node Table), and use default test parameters that cannot be modified by the operator. The unattended mode:

1. Runs internal loop test
2. Runs external loop test
3. Builds node table
4. Runs direct loop message test
5. Runs pattern test
6. Runs multiple message activity test

B.5.1.1 Build Node Table

The build subroutine is called to collect the physical addresses of the Ethernet nodes. It begins by transmitting a Request ID message on the Ethernet, to find a node to test. As the other nodes respond with their IDs, the NIE collects the IDs and adds the nodes to the node table, to include them in the tests.

B.5.1.2 Direct Loop Message Test

This test checks the ability of a node to respond to a loopback request. (See Paragraph Section B.7.2, RUN TEST command, DIRECT test.) A node has a maximum of 8 seconds to respond; three attempts are made to contact each node.

B.5.1.3 Pattern Test

This test sends six different loop direct messages to each node in the node table. (See Paragraph Section B.7.2, RUN TEST command, PATTERN test.)

B.5.1.4 Multiple Message Activity Test

This test uses the direct loop maintenance feature to create a large volume of Ethernet traffic. Loopback requests are sent to a subset (for example, 10) of the available nodes. All nodes in the subset are expected to respond, but data integrity is checked for only one of the responses (to save overhead). Upon successful completion, testing continues, checking the response from a different node each time. After all the nodes in the subset have been tested, testing continues with a different subset. This test is expected to cause multiple collisions and can affect overall Ethernet performance.

B.5.2 Operator Directed Mode

The commands available in this mode are listed below and described in Paragraph Section B.7.2.

HELP
BUILD
CLEAR MESSAGE
CLEAR NODE
CLEAR SUMMARY
IDENTIFY
MESSAGE
NODE
RUN DIRECT
RUN LOOPPAIR
RUN PATTERN
RUN ALL
RUN RESP
SAVE
UNSAVE
SHOW COUNTERS
SHOW MESSAGES
SHOW NODES
SUMMARY
EXIT

B.6 SYSTEM REQUIREMENTS

The following hardware is the minimum required to run the CVNIA NIE program.

- LSI-11 processor
- 28 Kwords memory
- Event line enabled or real-time clock
- Console terminal
- Any XXDP+ supported load media
- DELQA Ethernet to Q-Bus Adapter (minimum of 1, maximum of 2; tested individually)

The NIE uses XXDP+ as the program loading system and the PDP-11 Diagnostic Runtime Services (DRS) for the program environment.

B.7 COMMAND DESCRIPTION

B.7.1 DRS Commands

The 11 DRS commands are listed in Table C-1, with a brief description of each. The system will recognize a command by its first three characters; for example, you can type STA instead of START.

Table B–5 DRS Commands

Command	Description
START	Start the diagnostic from an initial state.
RESTART	Start the diagnostic without initializing.
CONTINUE	Continue at test that was interrupted (after <CTRL>C).
PROCEED	Continue from an error halt.
EXIT	Return to XXDP+ monitor (XXDP+ operation only).
ADD	Activate a unit for testing (all units are considered active at START time).
DROP	Deactivate a unit.
PRINT	Print statistical information (if implemented by the diagnostic).
DISPLAY	Type a list of all device information.
FLAGS	Type the state of all flags (see Section B.7.1.2)
ZFLAGS	Clear all flags (see Section B.7.1.2)

B.7.1.1 Switches

Several switches can be appended to DRS commands, to modify supervisor operation. The switches are defined in Table C-2, with a brief description of each. (Note: ddddd = 1 to 65535 decimal.) The switches can be used in combination. For example:

START/TESTS:1–5/PASS:1000/EOP:100

will cause tests 1 through 5 to execute; all units will be tested 1000 times; and the end of pass messages will be printed only after every 100 passes. The system will recognize a switch by its first three characters. For example, you can type /TES:1–5 instead of /TESTS:1–5. Table B–7 lists the switches that can be used with each command.

Table B–6 DRS Command Switches

Switch	Description
/EOP:ddddd	Report End of Pass message only after every ddddd passes.
/FLAGS:flag	Set specified flag(s) (see Section B.7.1.2)
/PASS:ddddd	Execute ddddd passes.
/TESTS:list	Execute only the tests specified by list (a string of test numbers). For example:
	START/TESTS:1:5:7–10
	will run tests 1, 5, 7, 8, 9, and 10. No other tests will be run.
/UNITS:list	START/ADD/DROP only those units (0–63) specified by list. For example:
	START/UNITS:0:5:10–12
	will test units 0, 5, 10, 11, and 12

Table B–7 Switch Application

Commands	Tests	Switches Pass	Flags	EOP	Units
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
EXIT	(none)				
ADD					X
DROP					X
PRINT	(none)				
DISPLAY					X
FLAGS	(none)				
ZFLAGS	(none)				

B.7.1.2 Flags

Flags are used to set-up certain operational parameters, such as looping on error. All flags are cleared:

1. at startup and remain cleared until explicitly set with the /FLAGS switch
2. after a START command unless set with the /FLAGS switch
3. with the ZFLAGS command.

No other commands, without a /FLAGS switch, affect the state of the flags; they remain as specified by the last /FLAGS switch. The flags are listed and described in Table C-4.

Flags can be specified in combinations. For example:

/FLAGS:LOE:IER:BOE

causes the program to loop on error, inhibit error reports, and sound the bell on error.

Table B-8 DRS Command Flags

Flag	Effect
ADR	Execute autodrop code.
BOE	Sound bell on error.
EVL	Execute evaluation (on diagnostics which have evaluation support).
HOE	Halt on error — control is returned to DRS command mode.
IBE†	Inhibit all error reports except first level (first level contains error type, number, PC, test and unit).
IDR	Inhibit program dropping of units.
IER†	Inhibit all error reports.
ISR	Inhibit statistical reports (applies only to diagnostics which support statistical reporting).
IXE†	Inhibit extended error reports (those called by PRINTX macros).
LOE	Loop on error.
LOT	Loop on test.
PNT	Print test number as test executes.
PRI	Direct messages to line printer.
UAM	Unattended mode (no manual intervention).

†Error messages are described in Paragraph Section B.8.1.

B.7.2 NIE Commands

NIE commands are typed in response to the prompt:

NIE> (A) ?

The commands are interpreted from left to right; and you need type only enough characters to uniquely specify a command. Command descriptions and examples follow.

Table B–9 NIE Commands

Command	Description
HELP or **?**	Types a brief description of NIE commands. Example: NIE> (A) ? **H** or NIE> (A) ? **?**
BUILD	This command is used to build the node table. It causes the exerciser to listen for system ID messages (broadcast by all nodes every 10 minutes). All such identifying nodes are added to the node table. The command stops if no new nodes have been added for 10 minutes or 40 minutes have elapsed. The average time for this command should be 15 to 25 minutes. It is possible to miss a transmission within the 10 minute period. Therefore, if no nodes appear in the table after a BUILD, wait 4 or 5 minutes and retry the BUILD. Example: NIE> (A) ? **BU**
CLEAR MESSAGE	This command resets message parameters to the default values.
CLEAR NODE/ADR	This command clears the specified node from the node table. The node can be specified by either its 12-digit (hex) physical address or its logical name (from the node table). To find the logical name associated with an address, execute the SHOW NODE command. Example: Clear a node using its Ethernet address: NIE> (A) ? **CL N/AA-00-04-FF-FF-F0** Clear a node using its logical name: NIE> (A) ? **CL N/N3**
CLEAR NODE/ALL	This command clears the node table. Example: Clear all nodes:

Table B-9 (Cont.) NIE Commands

Command	Description
	NIE> (A) ? **CL N/ALL** A cleared node can be restored to the node table with the UNSAVE command.
CLEAR SUMMARY	This command clears the summary table.
IDENTIFY ADR	Sends a Request ID message to the node specified by ADR. The returned system ID parameters are typed. Example: NIE> (A) ? **ID AA-00-04-FF-FF-F0**
MESSAGE/TYPE= /SIZE=n/COPIES=m	This command allows the operator to select the current message parameters. Any or all parameters can be changed. The defaultparameters are: /TYPE=ALPHA/SIZE=512/COPIES=1. The size of the message buffer is between 46 and 512 bytes. The number of copies of each message sent to each node can be between 1 and 255 copies. The message types are listed inTable C-5. Examples: Change type: NIE> (A) ? **M/T=ZERO** Change size: NIE> (A) ? **M/S=256** Change both size and type: NIE> (A) ? **M/S=512/T=ALPHA**

Table B–9 (Cont.) NIE Commands

Command	Description

NIE Test Message Types

Type	Content
ALPHA	!"#$%'()*+,-./0123456789;:=?ABCDEFG etc.
ONES	All ones (11111111).
ZEROS	All zeros (00000000).
1ALT	Alternating ones and zeros (10101010).
0ALT	Alternating zeros and ones (01010101).
CCITT	International Telegraph and Telephone Consultation Committee pseudo-random test pattern.
OPERATOR SELECTED	Operator selected pattern of less than 72, characters using 0–9, A–Z, and spaces (not used in PATTERN)

NODE ADR/TYPE

This command allows the operator to enter nodes into the node table. Nodes are specified by their 12-digit(hex) Ethernet physical address; and can be further specified (by /TYPE) to be either target or assist (default = target). Before changing a node's type, the node must first be cleared from the node table (see CLEAR command).

Examples:

Enter target node:

NIE> (A) ? **N AA-00-04-FF-FF-F0**

or

NIE> (A)? **N AA-00-04-FF-FF-F0/T**

Enter assist node:

NIE> (A) ? **N AA-00-04-FF-FF-F0/A**

Change a target node to an assist node:

NIE> (A) ? **CL N/AA-00-04-FF-FF-F0**

NIE> (A) ? **N AA-00-04-FF-FF-F0/A**

Table B-9 (Cont.) NIE Commands

Command	Description
RUN <TEST>/PASS=nn	Causes the specified test to execute for nn passes (default PASS = 1). If nn = 0, the test will run indefinitely. Prior to running the test(s), the NODE command should be used to enter the node addresses (taken from the node table) to be tested. The LOOPPAIR test requires node pairs, specified as target and assist nodes. Each test uses the currently selected values for message type, size, and copies. The tests are as follows.

DIRECT—This test sends a loop direct message to all of the nodes in the node table, waits for a response, checks returned data integrity, and reports any errors to the operator. The message to the target node comprises encapsulated forward and reply messages. The response from the target node comprises the same reply message. (See Figure C-1.)

LOOPPAIR —This test sends transmit, receive, and full assisted loopback messages, comprising encapsulated forward and reply messages, to the node pairs in the node table. (See Figures C-2, C-3, and C-4.) In each case, the test waits for a response and checks the data.

PATTERN — This test sends six different loop direct messages to each node in the node table. Each of six message types (ALPHA, ONES, ZEROS, 1ALT, 0ALT, CCITT—see Table C-5) is sent to each node. Returned data is checked for errors.

ALL — This two-part test performs the most extensive check of the network. It sends a loop direct message to each node in the node table. If this is successful, the exerciser builds an array of node pairs and sends a full assisted loopback message to each pair in the array. Table C-6 shows a sample array of pairs for a node table with seven nodes.

Node Pair Array					
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					

RESP—The RESPONDER test is a section of code that provides loop-server functions, such as: forwarding messages, answering console ID requests, and transmitting a system ID every 8 to 9 minutes. This must be run to use the DELQA as a loop assist or target node on the Ethernet. The other tests ignore forwarding requests, and will not transmit console IDs.

Examples:

Table B–9 (Cont.) NIE Commands

Command	Description
	Run the DIRECT test for one pass:
	NIE> (A) ? **R D**
	Run the DIRECT test for 5 passes:
	NIE> (A) ? **R D/P=5**
	Run the DIRECT test for infinite passes:
	NIE> (A) ? **R D/P=0**
	Run the LOOPPAIR test:
	NIE> (A) ? **R L**
	Run the RESPONDER test:
	NIE> (A) ? **R R**

NOTE

The only way to end a large or infinite number of passes is to type <CTRL>C. However, be careful: type RESTART in response to DSR> (after the <CTRL>C), to return to the NIE> prompt and preserve the counters. If you type START in response to DSR> after the <CTRL>C, you will destroy all summary statistics and counters.

Table B–9 (Cont.) NIE Commands

Command	Description
SAVE	This command saves the contents of the node table. Both the PDP NIE and the VAX NIE save the contents internally, not to a disk file. Example: NIE> (A) ? **SAV**
UNSAVE	This command restores the contents of the node table from the internally saved table. Example: NIE> (A) ? **UNS**
SHOW COUNTERS	Types the contents of the host node DEUNA internal counters. The counters are described elsewhere in this manual (see <REFERENCE>(CX??)) Example: NIE> (A) ? **SH C**
SHOW MESSAGE	Types the current message parameters for size, type, and copies. Example: NIE> (A) ? **SH M**
SHOW NODES	Types the contents of the node table. Example: NIE> (A) ? **SH N**
SUMMARY	This command types the summary table. The NIE maintains the following information about nodes to which it has sent messages: **RECEIVES NOT COMPLETE** **RECEIVES COMPLETE** **LENGTH ERRORS** **DATA COMPARE ERRORS** **BYTES COMPARED** **BYTES TRANSFERRED** BYTES COMPARED represents data minus the loop-server protocol overhead; therefore, it will be less than BYTES TRANSFERRED which represents data plus loop-server protocol overhead. Example: NIE> (A) ? **SUMM**

Table B–9 (Cont.) NIE Commands

Command	Description
EXIT	Returns control to the diagnostic supervisor (either VDS or DRS). The DRS RESTART and CONTINUE commands cannot be used if the EXIT command was used. Used to leave the NIE. Example: NIE> (A) ? **EXIT**

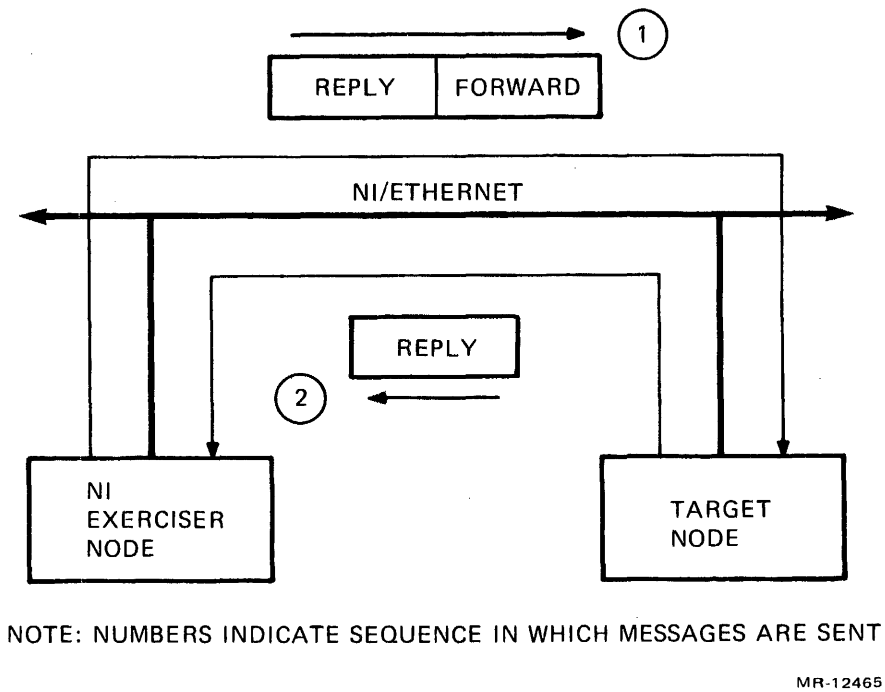


Figure B–1 Loop Direct Messages Test Path

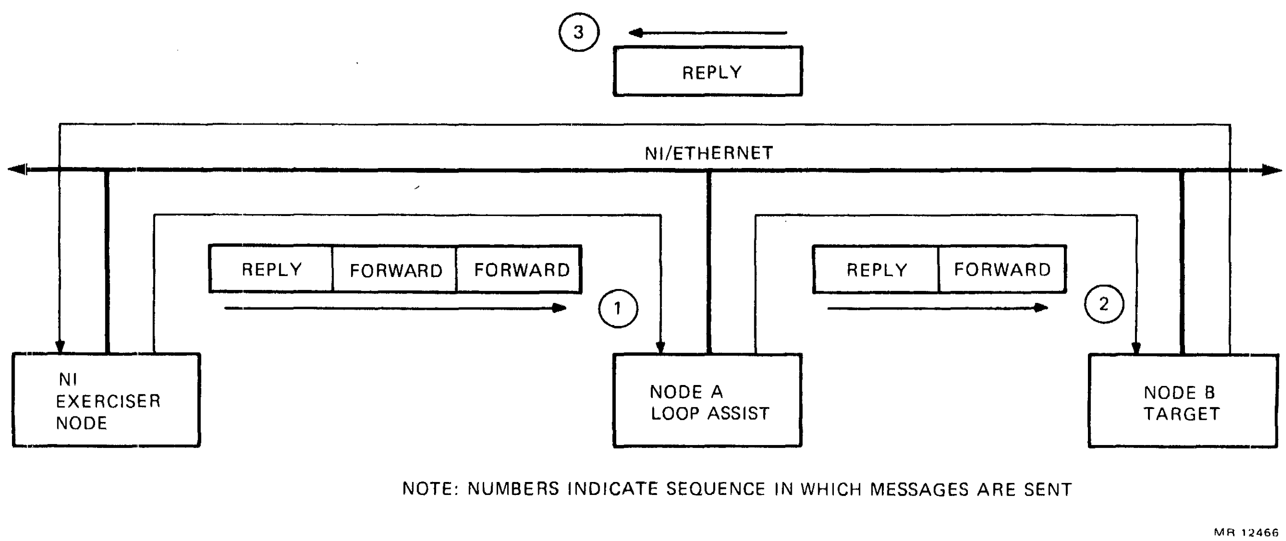


Figure B–2 Transmit Assist Loopback Message Test Path

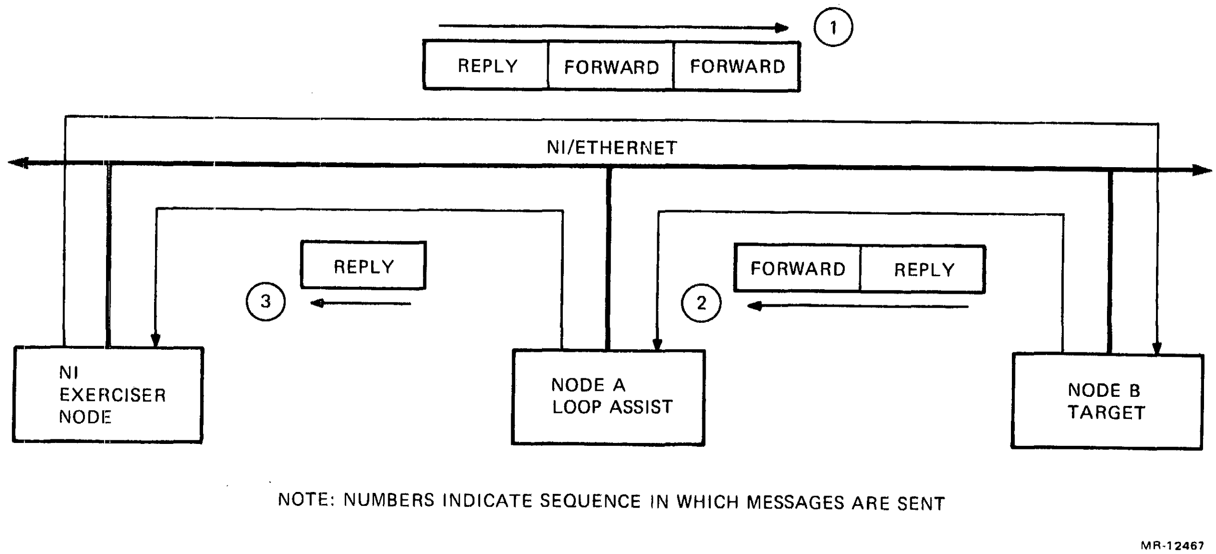


Figure B–3 Full Assist Loopback Message Test Path

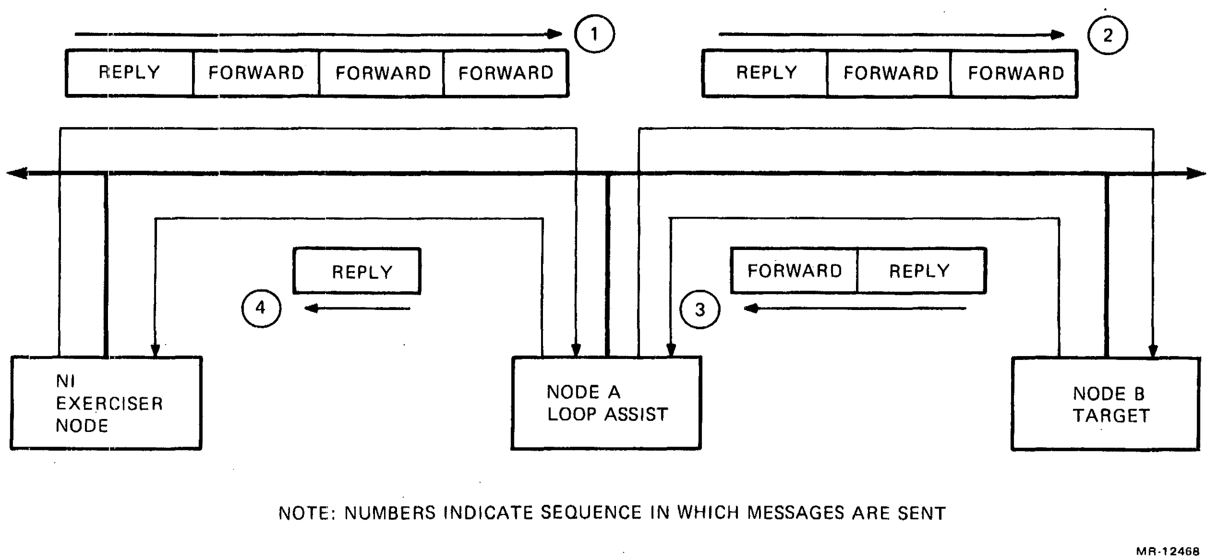


Figure B–4 Receive Assist Loopback Message Test Path

B.8 ERRORS

B.8.1 Error Messages

The three levels of error messages that a diagnostic can issue are: general, basic, and extended.

B.8.1.1 General

General error messages are always typed unless the IER flag (XXPD+) is set. The format is as follows:

NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:xxxxxx

ERROR MESSAGE

where:

NAME	= diagnostic name
TYPE	= error type (system fatal, device fatal, hard or soft)
NUMBER	= error number
UNIT NUMBER	= 0 through n (n is last unit in PTABLE; that is, device information table)
TST NUMBER	= test and subtest where error occurred
PC:xxxxxx	= address of error message call

B.8.1.2 Basic

Basic error messages contain some additional information about the error. These are always typed unless the IER or IBR flag (XXPD+) is set. These messages are typed after the associated general error message.

B.8.1.3 Extended

Extended error messages contain supplementary error information, such as register contents or good/bad data. These are always typed unless the IER, IBR, or IXR flag is set. These messages are typed after the associated general error message and any associated basic error messages.

Examples:

Lost packet error during LOOPPAIR testing:

CVNIA HRD ERR 00028 ON UNIT 00 TST 001 SUB 000 PC:064442

TIMEOUT OCCURRED - LOOP MESSAGE TYPE - RECEIVE ASSIST

FAILING TARGET NODE ADDRESS: AA-00-03-00-00-00

FAILING ASSIST NODE ADDRESS: AA-00-03-00-00-02

Lost packet error during PATTERN testing:

```
CVNIA HRD ERR 00028 ON UNIT 00 TST 001 SUB 000 PC:63730

TIMEOUT OCCURRED BEFORE LOOPBACK REPLY
FAILING NODE ADDRESS: AA-00-03-00-00-00
DATA PATTERN: ONES
```

B.8.2 Other Error Messages

Error Message	Description
?ILL CMD-BAD SYNTAX	A command with an illegal character was typed; retype the command.
?INCOMPLETE	A required part of a command was omitted.
?NUMBER TOO BIG	The numeric string value in the command line was larger than 65535 (177777 octal).
?BAD RADIX	An 8 or 9 was typed when an octal string was expected.

B.9 PDP-11 FUNCTIONAL DIAGNOSTIC (ZQNA??)

The Field Functional Diagnostic Program (ZQNA??) tests the DELQA in Q-bus systems. It attempts to isolate faults to the following Field Replaceable Units (FRUs):

- DELQA or DEQNA only
- Cabinet kit
- Fuse

Refer to the *XXDP+ User Manual* for further information.

NOTE
Early versions of ZQNA?? only work with the DEQNA. ZQNAI? is the first version to be compatible with both DEQNA and DELQA.

B.9.1 ZQNA?? Environment

Tests are executed under the supervision of the Diagnostic Runtime Services (XXDP+), and controlled by an operator from a console (hard copy or video).

B.9.2 ZQNA?? Test Descriptions

The setup tests are described below.

- **Basic operation tests:**
  1. Device self-test
  2. Station address verification
  3. BD code verification
- **Internal Extended Loopback Test**
  4. Rx/Tx descriptor mechanism verification
  5. Q-bus NXM test
  6. Q-bus DMA interface (transmit) test
  7. Q-bus DMA interface (receive) test
  8. Internal extended loopback path stuck-at test
  9. Extended memory test

- **Setup mode loopback tests**
  10. Setup mode loopback test
- **Internal loopback tests**
  11. Ethernet Address recognition test
  12. Ethernet Address recognition mode test
  13. Overflow status check
- **External loopback tests**
  14. External lopback path verification test
  15. Carrier bit test
  16. Sanity timer test

B.9.3 ZQNA?? Error Reports

A diagnostic can issue general and specific types of error messages.

General error messages are always printed unless the IBE and/or IER flag is set. The information shown is:

NAME = Diagnostic name

ER_TYPE = Error type (all errors are hard errors)

ER_NO = Error number

UNIT_NO = 0

TEST_NO = Test and subtest where error occurred

PC_ADDR = Program counter contents.

General error messages may include two sublevels:

- *Basic error messages* are printed after the associated general error message, and contain some additional information about the error. They are always printed unless one or more DRS error flags (IBE, IXE, IER) are set.
- *Extended error messages* are printed after the associated general error message and any associated basic error messages. Extended error messages contain additional error information, such as register contents or good/bad data. They are always printed unless either the IXE or IER flag (or both) is set. The format of a typical extended error message is shown in Figure B-2.

Specific error messages are defined as needed. The following are possible error messages.

Device fatal error messages:

CSR REGISTER FAILED TO RESPOND

NO INTERRUPT FROM DELQA

Return status messages:

TRANSMIT STATUS ERROR

RECEIVE STATUS ERROR

CSR STATUS ERROR

B.10 MICROVAX DIAGNOSTIC MONITOR (MDM)

MDM diagnostics are based on a functional testing approach where the objective is to gain the maximum coverage and to isolate DELQA faults down to the Field Replaceable Unit (FRU).

The recommended test strategy for identifying Field Replaceable Units (FRUs) that are causing a fault is as follows.

1. Check the MDM configuration listing for the correct DELQA details.
2. Run the Verify tests (FUNCTIONAL and EXERCISER) to check DELQA functions.
3. Run the Field Service Functional tests and the utility tests if yet more advanced and detailed fault-finding is essential to identify a faulty FRU.

B.10.1 MDM Environment

MDM test commands require loopback connectors in the following cases.

- Tests in functional mode and exerciser mode, including TEST CABLES (Utility 1) require one of:
  - Bulkhead loopback connector (order number 12-22196-02)
  - Loopback connector attached to the DELQA board (order number 70-21489-01)
  - Cable and Ethernet transceiver to provide external loopback.
- TESTLOOP (Utility 2) and ECHOSERVER (Utility 3) enable two MicroVAX systems to send packets to each other.

The Verify tests (FUNCTIONAL and EXERCISER) do not require any loopback connections.

The operator is prompted whenever a loopback cable/connector is required.

B.10.2 MDM Service Test Descriptions

B.10.2.1 Verify Mode Tests

MDM Verify mode is designed for use by untrained personnel. It prohibits operator intervention during testing.

The sequence of tests is as follows.

TEST1 Initialize controller Test bulkhead assembly fuse

(If Normal mode) Invoke self-test

Possible failure: nonexistent memory interrupt (NXM)

TEST2 Send several setup packets

Test BDL handling

Test interrupt ability

Test DMA

TEST3 Loop packets in internal loopback mode

Test address decoding logic

TEST4 Loop packets back in internal extended loopback mode with different data types (zeros, ones, alternating, CCITT, and so on)

Test internal RAM Tests long packet logic

TEST5 Loop packets in internal extended loopback mode

Use chained descriptors and multiple buffers

TEST6 The device is placed in different station address filtering modes and packets are looped through using internal loopback. Depending upon the mode and the station address, the packet may or may not be received.

Test promiscuous filtering

Test multicast filtering

Test normal filtering

TEST7 (*Normal mode only*) Verify the operation of Extended setup packet by looping several back through the DELQA.

Verify correct processing of the MOP element blocks

B.10.2.2 Field Service Functional Tests

Field service tests are designed for operators experienced in the testing and debugging of DIGITAL equipment. The operator may be asked to make minor physical alterations, including mounting an external loopback connector.

This diagnostic has one field service test, FIELD_TEST, which uses external loopback to test the bulkhead loopback connector or the DELQA-cable-transceiver loop. (Internal loopback is tested in the MODE routines.)

TEST1 Send packets using external loopback mode

B.10.2.3 Field Service Exerciser

The exerciser is designed to stress the MicroVAX system, creating a simulation of normal system operation. By executing several exercisers together, on the same MicroVAX system, it is expected that any marginality in the system design and operation can be isolated and corrected. The EXERCISER does not require the operator to modify the system.

TEST1 Performs the setup test

Internal loopback test

Internal extended loopback test

Allocate and deallocate buffers

B.10.3 MDM Utilities

The utilities are designed for the sophisticated troubleshooter, who needs to configure the system for specific tests. These utilities, using already verified circuits, produce test scaffolds to track down failures in the back panel, cables, or the terminals attached to the DELQA.

Utility 1 **TESTCABLES**

Repeatedly prompts the user to connect a loopback connector/cable at any point in the communications path in order to isolate an error to a particular segment.

Utility 2 **TESTLOOP**

Sends packets to an echo server and expects to receive those packets, error free, back over the Ethernet.

Utility 3 **ECHOSERVER**

Acts as an echo server for the TESTLOOP utility.

Utility 4 **NIE**

B.11 DEC/X11 EXERCISER

The DEC/X11 Exerciser exercises one DELQA at maximum activity rates. It transmits and receives random-length packets (using either 18- or 22-bit physical address space). The DELQA transmits and receives the same packet.

One pass of the exerciser consists of 1000 iterations of transmitting a packet, receiving a packet, and comparing the contents of the transmit packet to the receive packet. Packet length is random for each iteration. The transmit and receive status words and CSR status are all checked for correct contents.

The DELQA is dropped from further testing if any of the following occurs.

- The DELQA does not reset properly.
- The CSR and/or the receive and/or transmit status word(s) are in error.
- A hard error occurs.
- A transmit and/or receive interrupt is not generated.
- The transceiver is disconnected while in external loopback mode.

Internal extended loopback mode is the default mode of operation.

You must set the sanity timer switches S4 to enable the timer before executing the sanity timer test. When sanity timer testing is complete, restore the switches to the positions they occupied before the test.

B.11.1 Environment

It is assumed that, prior to running this exerciser, both the DELQA citizenship (CQ) test and field functional test have been successfully run. The default parameters are:

Device address: 17774440

Interrupt Vector: 700

BR level: 4

Number of devices: 1

The hardware switch settings are:

- Mode switch S3: OPEN to enable DEQNA-lock mode
- Option switch S4: OPEN to enable the sanity timer at power up.

To run the exerciser in external loopback, the DELQA under test must be connected to the transceiver, or the external loopback connector must be connected.

The options for Software Register 1 (SR1) bits 0 and 1 are described in Table B-7.

Table B–10 DELQA DEC/X11 Exerciser Software Register Bits

BIT 1	BIT 0	
Bit Value		Description
0	0	Exerciser runs in internal extended loopback mode (default). The transceiver is not needed.
0	1	Exerciser runs in external loopback mode. The transceiver or external loopback connector is required.
1	0	Print error messages.
1	1	Do not print error messages.

B.11.2 Command Descriptions

To set external loopback mode, type the following commands:

MOD QNAA0 16 <RETURN>

1 <RETURN>

To test a DELQA in the second slot (address 17774460), type the following commands after the exerciser has been loaded:

MDD QNAA0 6 <RETURN>

174460 <LINE FEED>

704 <RETURN>

For additional information, refer to the *DEC/X11 User Manual*

APPENDIX C
PROGRAMMING EXAMPLES FOR PDP-11 SYSTEMS

This appendix contains programming examples written in MACRO-11 for the DELQA. They are presented only as a general guide for the prospective user, and not as the best or only method of driving the DELQA. These programs are not guaranteed or supported by Digital Equipment Corporation.

Programming examples are provided for the following.

1. Data definitions
2. Resetting the DELQA
3. Configuring the DELQA
4. A simple interrupt handler
5. Data transmission
6. Data reception
7. Executing on-board diagnostics

C.1 DATA DEFINITIONS

The following data definitions are used throughout the sample programs. Note that all numbers are octal unless otherwise specified.

Table C–1 Data Definitions for Sample Programs

bit15	=	100000
bit14	=	040000
bit13	=	020000
bit12	=	010000
bit11	=	004000
bit10	=	002000
bit09	=	001000
bit08	=	000400
bit07	=	000200
bit06	=	000100
bit05	=	000040
bit04	=	000020
bit03	=	000010
bit02	=	000004
bit01	=	000002
bit00	=	000001

The following sample programs refer to a DELQA installed at I/O page base address 17774440, and the following definitions of I/O port registers apply throughout:

```
lqarll: .word   174444          ; Rx BDL low-order address bits
lqarlh: .word   174446          ; Rx BDL high-order address bits
lqatll: .word   174450          ; Tx BDL low-order address bits
lqatlh: .word   174452          ; Tx BDL high-order address bits
lqavar: .word   174454          ; Vector Address Register
```

```
;
; VAR bit definitions
;
        ms =    bit15           ; Mode Select
        os =    bit14           ; Option Switch
        rs =    bit13           ; Request to execute self-test
        s3 =    bit12           ;
        s2 =    bit11           ; Self-test status
        s1 =    bit10           ;
;
lqacsr: .word   174456          ; Control and Status Register
;
; CSR bit definitions
;
        ri =    bit15           ; Receive Interrupt
        el =    bit09           ; External Loopback
        il =    bit08           ; Internal Loopback
        xi =    bit07           ; Transmit Interrupt
        ie =    bit06           ; Interrupt Enable
        sr =    bit01           ; Software Reset
        re =    bit00           ; Receiver Enable
```

The following define the octal offsets and bit values of individual fields of a buffer descriptor:

```
;
; Flag word. [ Note : This field is reserved ]
;
bflw    =       0
;
; Address descriptor bits.
;
bdes    =       2
h       =       bit06           ; High byte only beginning
l       =       bit07           ; Low byte only termination
s       =       bit12           ; indicates that this is a ...
                                ; ...Setup packet
e       =       bit13           ; End of message
c       =       bit14           ; Chain address
v       =       bit15           ; Valid bit
invalid =       0               ; descriptor is not Valid
;
; 6 High Order buffer address bits.
;
bpah    =       2
;
; 16 Low Order buffer address bits.
;
bpad    =       4
;
; Twos complement of buffer size in words.
;
bsiz    =       6
;
; Status word #1.
;
bsw1    =       10
unused  =       100000      ; unused status word
lastok  =       0           ; this descriptor contains the last...
                            ; ...or only segment of a packet...
                            ; ...with no errors.
;
; Bits defined for Receive status word #1
;
esetup  =       bit13       ; indicates a looped back Setup or...
                            ; ...External Loopback packet
rblh    =       bit08!bit09!bit10 ; high order 3 bits of...
```

```
                                ; ...receive byte length of packet
;
; Status word #2.
;
bsw2    =       12
;
; Bits defined for Receive status word #2
;
rbll    =       377             ; low order 8 bits of receive...
                                ; ...byte length of packet
```

C.2 RESETTING THE DELQA

The DELQA undergoes a software reset when CSR bit 1 is cleared from 1 to 0.

```
;
; *-------------------------------------------------------------*
; *                                                             *
; *         A routine to Software Reset the DELQA.              *
; *                                                             *
; *-------------------------------------------------------------*
;
lqares: bis     #sr,@lqacsr      ; set SR in CSR
        bic     #sr,@lqacsr      ; clear SR to generate sw reset
        rts     pc               ; return to caller
```

C.3 CONFIGURING THE DELQA

The DELQA may be configured using SETUP packets for various modes of reception of Ethernet packets. The following routines demonstrate a method of assembling, transmitting and receiving a SETUP packet.

```
;
; *-------------------------------------------------------------*
; *                                                             *
; *   Routine to enable the DELQA to accept any legal packet    *
; *   from the Ethernet whose destination address is in the     *
; *   following list.                                           *
; *   All address digits are octal.                             *
; *                                                             *
; *      100-001-002-003-004-005      physical  address #1      *
; *      100-002-002-003-004-005      physical  address #2      *
; *      101-001-002-003-004-005      multicast address #1      *
; *      101-002-002-003-004-005      multicast address #2      *
; *      377-377-377-377-377-377      broadcast address         *
; *                                                             *
; *-------------------------------------------------------------*
;
setupa: mov     #stpads,r1        ; r1 := addr of SETUP addr list
        mov     #nstpad,r2        ; r2 := number of addrs in list
        clr     r3                ; r3 := SETUP control byte
        jsr     pc,lqastp         ; assemble, transmit, receive...
                                  ; ... the SETUP packet
        rts     pc                ; return to caller
;
; Table of Ethernet addresses.
;
stpads: .byte   100,001,002,003,004,005
        .byte   100,002,002,003,004,005
        .byte   101,001,002,003,004,005
        .byte   101,002,002,003,004,005
        .byte   377,377,377,377,377,377
nstpad  =       <.-stpads>/6
```

```
;
; *--------------------------------------------------------------*
; *                                                              *
; * Subprogram : LQASTP                                          *
; *                                                              *
; * This subprogram generates and transmits a SETUP packet to    *
; * the DELQA using a list of addresses (physical, multicast,    *
; * broadcast) supplied by the caller. The caller may also       *
; * specify control information concerning device filtering      *
; * modes (e.g. all multicast, promiscuous) and Sanity Timer     *
; * timeout values.                                              *
; *                                                              *
; * Note that if the caller specifies less than 14 Ethernet      *
; * addresses, this code pads the Setup packet with duplicate    *
; * addresses until there are 14 addresses in the packet.        *
; *                                                              *
; * Parameters :                                                 *
; *                                                              *
; *  R1 <-- table of 6 byte filter addresses.                    *
; *  R2 <-- number of addresses in table.                        *
; *  R3 <-- control byte value defined as follows :              *
; *                                                              *
; *         bit  #0   - Enable[1]/Disable[0] All Multicast Mode. *
; *         bit  #1   - Enable[1]/Disable[0] Promiscuous Mode.   *
; *         bits #2-3 - Specify which of the three DELQA LEDs to *
; *                     switch off.                              *
; *         bits #4-6 - Specify factor of 4 times 1/4 second for *
; *                     Sanity Timer timeout value.              *
; *                                                              *
; *--------------------------------------------------------------*
;
lqastp: mov     r1,-(sp)        ;
        mov     r2,-(sp)        ;
        mov     r3,-(sp)        ; save registers
        mov     r4,-(sp)        ;
        mov     r5,-(sp)        ;
;
        mov     r1,stplst       ; save addr of callers addr table
        mov     r3,ctlbyt       ; save control byte
        mov     r2,numsad       ; save number of addrs in table
        mov     #stpkt,r3       ; R3 := start of setup packet
        mov     #2,r0           ; FOR 2 addr blocks in setup pkt DO
                                ;
10$:    mov     #6,r5           ;   FOR 6 bytes in each address DO
                                ;     BEGIN
20$:    mov     r1,adrbyt       ;     save addr byte pointer
        clrb    (r3)+           ;     zero first byte in column
;
        mov     #7,r4           ;     FOR 7 addrs in each block DO
                                ;       BEGIN
30$:    movb    (r1),(r3)+      ;       get next addr byte from tbl
        dec     r2              ;       decrement address count
        ble     40$             ;       IF NOT end_of_table THEN
        add     #6,r1           ;         skip to next addr in table
40$:    sob     r4,30$          ;       END [ for ]
;
        mov     adrbyt,r1       ;     restore addr table pointer
        inc     r1              ;     skip to next addr byte in tbl
        mov     numsad,r2       ;     restore addr table entry count
        sob     r5,20$          ;     END [ for ]
;
; Zero the unused bytes from :
;
;       <1> offset 160 to 177 octal
;       <2> offset  60 to  77 octal.
```

```
;
        mov     #stpkt+60,r4  ;    r4 := addr of 1st unused area
        mov     #10,r3        ;    FOR x = 1 to 10 (octal) words DO
                              ;      BEGIN
70$:    mov     #0,100(r4)    ;      word[160+x] := 0
        clr     (r4)+         ;      word[60+x]  := 0
        sob     r3,70$        ;      END [ for ]
;
; Insert the next 7 addresses at offset 100 (octal) into packet.
;
        mov     #stpkt+100,r3 ;    r3 := addr of second addr block
        mov     stplst,r1     ;    goto start of callers addr tbl
        sub     #7,r2         ;    IF > 7 addrs in table THEN DO
        ble     90$           ;
        add     #7*6,r1       ;      jump to the 8th addr in table
90$:    mov     r2,numsad     ;      remember number of addrs left
        sob     r0,10$        ;    END [ for ]
;
        mov     ctlbyt,r0     ; restore control byte
;
; The byte length seen by the DELQA must be 200 (octal)
; plus the 7-bit control information.
;
        add     #200,r0       ; r0 := effective SETUP pkt length
        mov     r0,r4         ; save byte count
        ash     #-1,r4        ; divide by 2 for word count
;
; Two cases arise here :
;
;    <1> Even byte length SETUP packet.
;        Use address descriptor flags V(valid), E(end of message)
;        and S(setup).
;
;    <2> Odd byte length SETUP packet.
;        Use V,E and S and also L(low byte only termination).
;
; NOTE : In the case of Setup packets, the Address Descriptor
;        bits H and L are not used to determine the transmit
;        buffer alignment as they are with other packets. Instead
;        the H and L bits are used solely to calculate the
;        logical length of the Setup packet for the purpose of
;        determining the control information from the packet
;        length modulo 200 octal.
;
        mov     #v!e!s,sttxdl+bdes ; assume an even length packet
        bit     #1,r0              ; IF odd length setup pkt THEN DO
        beq     95$                ;   BEGIN
        inc     r4                 ;   incrmt word count for odd byte
        bis     #1,sttxdl+bdes     ;   set addr descriptor L bit
                                   ;   END
                                   ;
95$:    neg     r4                 ; get 2s complement word count
        mov     r4,sttxdl+bsiz     ; save it in TxBDL field
;
        bic     #ie,@lqacsr        ; disable interrupts via CSR IE
;
; Validate the Receive and Transmit buffer descriptor lists.
;
        mov     #strxdl,@lqarll ; write Rx BDL addr to the DELQA
        clr     @lqarlh         ; and validate it
        mov     #sttxdl,@lqatll ; write Tx BDL addr to the DELQA
        clr     @lqatlh         ; validate it, start transmission
;
        jsr     pc,wtrixi       ; Wait for XI and RI to be ...
                                ; ...asserted in CSR
```

```
;
; RI and XI must be reset to '0' by writing '1' to them.
;
        bis     #ri!xi,@lqacsr  ; reset XI and RI to '0' in CSR
;
; Verify that there were no transmit errors.
;
        mov     sttxdl+bsw1,r1  ; r1 := transmit status word 1
        bic     #^C<bit15!bit14> ; zero all except LASTNOT...
                                ;  ...and ERROR/USED
        cmp     #lastok,r1      ; IF NOT transmit error THEN DO
        beq     100$            ;    check for receive error
;
; Transmit error handling code should be placed here.
;
        br      endstp          ;    quit
;
; Verify that there were no receive errors.
;
100$:   mov     strxdl+bsw1,r1  ; r1 := receive status word 1
        bic     #^C<bit15!bit14> ; zero all except LASTNOT...
                                ;  ...and ERROR/USED
        cmp     #lastok,r1      ; IF NOT receive error THEN DO
        beq     endstp          ;    quit
;
; Receive error handling code should be placed here.
;
endstp: mov     (sp)+,r5        ;
        mov     (sp)+,r4        ;
        mov     (sp)+,r3        ; restore callers registers
        mov     (sp)+,r2        ;
        mov     (sp)+,r1        ;
;
        rts     pc              ; return to caller
;
; Define Buffer Descriptor Lists.
;
; Transmit Buffer Descriptor List.
;
sttxdl: .word   unused          ; flag word
        .word                   ; reserved for addr descrptr bits
        .word   stpkt           ; addr of assembled SETUP packet
        .word                   ; reserved for packet length
        .word   unused          ; status word #1
        .word   unused          ; status word #2
;
; Follow with an invalid descriptor.
;
        .word   unused          ; flag word
        .word   invalid         ; INVALID descriptor
        .word   0,0             ; dummy buffer addr and word count
;
; Receive Buffer Descriptor List.
;
strxdl: .word   unused          ; flag word
        .word   v               ; this descriptor is VALID
        .word   recbuf          ; addr of SETUP packet Rx buffer
        .word   -rcbfln/2       ; 2s comp of Rx buffer word length
        .word   unused          ; status word #1
        .word   unused          ; status word #2
;
; Follow with an invalid descriptor.
;
        .word   unused          ; flag word
        .word   invalid         ; this descriptor is INVALID
```

```
        .word   0,0             ; dummy buffer addr and word count
;
; Reserve storage for the assembled SETUP packet.
;
stpkt:  .blkb   377
;
        .even
;
; Reserve storage for the Receive buffer.
;
recbuf: .blkb   377
rcbfln  =       .-recbuf
;
        .even
;
; Temporary work storage.
;
ctlbyt: .word                   ; SETUP control byte
numsad: .word                   ; # entries in callers addr table
adrbyt: .word                   ; addr of current addr byte in...
                                ; ...callers address table
stplst: .word                   ; address of callers address table
```

```
;
; *-----------------------------------------------------------*
; *                                                           *
; * Subprogram WTRIXI                                         *
; *                                                           *
; * This subprogram waits for RI and XI to be asserted in     *
; * the CSR.                                                  *
; *                                                           *
; *-----------------------------------------------------------*
;
wtrixi: mov     r1,-(sp)        ; save r1
;
; Wait for RI and XI to be set. Note that a real application
; program would need a timeout check.
;
                                ; REPEAT
10$:    mov     @lqacsr,r1      ;   r1 := (CSR)
        bic     #^C<ri!xi>,r1   ;   zero all bits except RI and XI
        cmp     #ri!xi,r1
        bne     10$             ; UNTIL RI and XI are set to '1'
        mov     (sp)+,r1        ; restore r1
        rts     pc              ; return to caller
```

C.4 A SIMPLE INTERRUPT HANDLER

```
;
; *--------------------------------------------------------------*
; *                                                              *
; *     Subprogram LQAINT                                        *
; *                                                              *
; *     This subprogram handles DELQA Transmit and Receive       *
; *     interrupts.                                              *
; *                                                              *
; *--------------------------------------------------------------*
;
lqaint: mov     r1,-(sp)          ; save work register
        bit     #xi,@lqacsr       ; IF XI is not set to '1' THEN DO
        beq     ck.rxi            ;   go check for Rx Interrupt
                                  ; ELSE DO
        movb    #1,txdone         ;   set TXDONE flag
;
; Reset XI to '0' by writing a '1' to it.
; Must be careful to avoid accidentally resetting RI also.
;
        mov     @lqacsr,r1        ;   r1 := (CSR)
        bic     #ri,r1            ;   RI := 0 to avoid writing...
                                  ;   ... '1' to it
        mov     r1,@lqacsr        ;   reset XI to '0' by writing...
                                  ;   ... '1' to it
;
ck.rxi: bit     #ri,@lqacsr       ; IF RI is not set to '1' THEN DO
        beq     endint            ;   return
                                  ; ELSE DO
        movb    #1,rxdone         ;   set RXDONE flag
;
; Reset RI to '0' by writing a '1' to it.
; Must be careful to avoid accidentally resetting XI also.
;
        mov     @lqacsr,r1        ;   r1 := (CSR)
        bic     #xi,r1            ;   XI := 0 to avoid writing...
                                  ;   ... '1' to it
        mov     r1,@lqacsr        ;   reset RI to '0' by writing...
                                  ;   ... '1' to it
endint: mov     (sp)+,r1          ;   restore r1
        rti                       ; return to caller
;
txdone: .byte                     ; storage for interrupt flag byte
rxdone: .byte                     ; storage for interrupt flag byte
```

C.5 DATA TRANSMISSION

The steps required to initiate a transmission are:

- Write the 16 LOW-order bits of the Transmit BDL address to the I/O page low-order Tx BDL register.
- Write the 6 HIGH-order bits of the Transmit BDL address to the I/O page high-order Tx BDL register.

The completion of the transmission may be detected in one of two ways:

- Polling the XI bit in the CSR with DELQA interrupts disabled.
- Enabling the DELQA to interrupt the host when transmission is complete.

```
;
; *-------------------------------------------------------------*
; *                                                             *
; *      Routine to transmit a packet and poll the CSR to       *
; *      detect transmission completion.                        *
; *                                                             *
; *-------------------------------------------------------------*
;
txpktl: bic     #ie!el,@lqacsr      ; disable DELQA interrupts...
                                    ; ... and loopback modes
        bis     #il,@lqacsr         ; disable Internal Loopback
        mov     #tx.bdl,@lqatll     ; write addr of Tx BDL to DELQA
        clr     @lqatlh             ; start the transmission
;
; Wait for XI to be set. Note that a real application program
; would need a timeout check.
;
                                    ; REPEAT
10$:    bit     #xi,@lqacsr         ;   test XI bit in CSR
        beq     10$                 ; UNTIL XI = '1'
;
; Reset XI to '0' by writing a '1' to it.
; Must be careful to avoid accidentally resetting RI also.
;
        mov     @lqacsr,r1          ; r1 := (CSR)
        bic     #ri,r1              ; RI := '0' to avoid writing...
                                    ; ... a '1' to it
        mov     r1,@lqacsr          ; reset XI to '0' by writing...
                                    ; ... a '1' to it
;
; Check the transmit status to verify that no transmit errors have
; occurred.
;
        mov     tx.bdl+bsw1,r1      ; r1 := transmit status word 1
        bic     #^C<bit15!bit14>    ; zero all except LASTNOT...
                                    ; ...and ERROR/USED
        cmp     #lastok,r1          ; IF NOT transmit error THEN DO
        beq     20$                 ;   continue
                                    ; ELSE DO
;
; Error handling code should be placed here.
;
20$:    rts     pc                  ; return to caller
;
; Transmit Buffer Descriptor List.
;
tx.bdl: .word   unused              ; flag word
        .word   v!e                 ; Valid desc and End of Msg
        .word   tx.buf              ; address of transmit buffer
        .word   -tx.len/2           ; 2s comp buffer word length
        .word   unused              ; status word #1
        .word   unused              ; status word #2
;
```

```
; Follow with an invalid descriptor.
;
        .word   unused              ; flag word
        .word   invalid             ; this descriptor is INVALID
        .word   0,0                 ; dummy buffer address, length
;
; Reserve storage for maximum length transmit buffer.
;
tx.buf: .blkb   1514.               ; 1514 (Decimal) bytes
tx.len  =       .-tx.buf
```

```
;
; *-------------------------------------------------------------*
; *                                                             *
; * Routine to transmit a packet and wait for the DELQA to      *
; * interrupt the host to signify that the transmission is      *
; * complete.                                                   *
; *                                                             *
; *-------------------------------------------------------------*
;
txpkt2:
lqavec  =       400                 ; DELQA host interrupt vector
pri07   =       340                 ; Processor Status of ...
                                    ; ...interrupt handler - IPL = 7
        bic     #ie!el,@lqacsr      ; disable DELQA interrupts...
                                    ; ...and External Loopback
        bis     #il,@lqacsr         ; disable Internal and ...
                                    ; ...Internal Extended Loopback
;
; We must write the address of the DELQA interrupt vector to the
; VAR without altering the Mode Select bit (bit15) in that
; register.
;
        mov     @lqavar,r1          ; r1 := (VAR)
        bic     #^C<ms>,r1          ; zero all except MS (bit 15)
        bis     #lqavec,r1          ; OR in addr of interrupt vector
        mov     r1,@lqavar          ; and write it back to the VAR
;
        mov     #lqaint,lqavec      ; load interrupt vector with...
                                    ; ...address of interrupt handler
        mov     #pri07,lqavec+2     ; establish IPL of handler = 7
;
        clrb    txdone              ; init TXDONE interrupt flag
        bis     #ie,@lqacsr         ; enable DELQA interrupts
        mov     #tx.bdl,@lqatll     ; write addr of Tx BDL to DELQA
        clr     @lqatlh             ; start the transmission
;
; Wait for the interrupt flag TXDONE to be set to '1' to signify
; completion of transmission. Note that a real application program
; would need a timeout check.
;
                                    ; REPEAT
10$:    tstb    txdone              ;   test TXDONE flag
        beq     10$                 ; UNTIL TXDONE is set to '1'
;
; Check the transmit status to verify that no transmit errors have
; occurred.
;
        mov     tx.bdl+bsw1,r1      ; r1 := transmit status word 1
        bic     #^C<bit15!bit14>    ; zero all except LASTNOT...
                                    ; ...and ERROR/USED
        cmp     #lastok,r1          ; IF NOT transmit error THEN DO
        beq     20$                 ;   continue
                                    ; ELSE DO
;
; Error handling code should be placed here.
;
20$:    rts     pc                  ; return to caller
;
; Transmit Buffer Descriptor List.
;
tx.bdl: .word   unused              ; flag word
        .word   v!e                 ; Valid desc and End of Msg
        .word   tx.buf              ; address of transmit buffer
        .word   -tx.len/2           ; 2s comp buffer word length
        .word   unused              ; status word #1
```

```
        .word   unused          ; status word #2
;
; Follow with an invalid descriptor.
;
        .word   unused          ; flag word
        .word   invalid         ; this descriptor is INVALID
        .word   0,0             ; dummy buffer address, length
;
; Reserve storage for maximum length transmit buffer.
;
tx.buf: .blkb   1514.           ; 1514 (Decimal) bytes
tx.len  =       .-tx.buf
```

C.6 DATA RECEPTION

The steps required to initiate a reception are:

- Set RE to 1 in CSR. This is not necessary if the DELQA is receiving a packet which was looped by Internal, External, Internal Extended, or Setup loopback modes.
- Write the 16 LOW-order bits of the Receive BDL address to the I/O page low-order Rx BDL register.
- Write the 6 HIGH-order bits of the Receive BDL address to the I/O page high-order Rx BDL register.

The completion of the reception may be detected in one of two ways.

- Polling the RI bit in the CSR with DELQA interrupts disabled.
- Enabling the DELQA to interrupt the host when reception is complete.

```
;
; *--------------------------------------------------------------*
; *                                                              *
; * Routine to wait for a packet from the Ethernet.              *
; * Polling is used to determine when a packet has been          *
; * received.                                                    *
; *                                                              *
; *--------------------------------------------------------------*
;
rxpktl: bic     #ie!el,@lqacsr      ; disable DELQA interrupts...
                                    ; ...and loopback modes
        bis     #il!re,@lqacsr      ; disable Internal Loopback...
                                    ; ...and enable receiver
        mov     #rx.bdl,@lqarll     ; write addr of Rx BDL to DELQA
        clr     @lqarlh             ; initiate a packet reception
;
; Wait for RI to be set. Note that a real application program
; would need a timeout check.
;
                                    ; REPEAT
10$:    bit     #ri,@lqacsr         ;   test RI bit in CSR
        beq     10$                 ; UNTIL RI = '1'
;
;
; Reset RI to '0' by writing a '1' to it.
; Must be careful to avoid accidentally resetting XI also.
;
        mov     @lqacsr,r1          ; r1 := (CSR)
        bic     #xi,r1              ; XI := '0' to avoid writing...
                                    ; ... a '1' to it
        mov     r1,@lqacsr          ; reset RI to '0' by writing...
                                    ; ... a '1' to it
;
; Check the receive status to verify that no receive errors have
; occurred.
;
        mov     rx.bdl+bsw1,r1      ; r1 := receive status word 1
        bic     #^C<bit15!bit14>    ; zero all except LASTNOT...
                                    ; ...and ERROR/USED
        cmp     #lastok,r1          ; IF NOT receive error THEN DO
        beq     20$                 ;   continue
                                    ; ELSE DO
;
; Error handling code should be placed here.
;
20$:
;
; Calculate length in bytes of received packet.
; NOTE : This assumes that the packet received is NOT a loopback
;        packet. Therefore, the receive length as determined from
```

```
;        the receive status words is 60 decimal less than the
;        actual packet length.
;        Additional code is required to handle the loopback cases.
;
        mov     rx.bdl+bsw1,r1      ; r1 := receive status word 1
        bic     #^C<rblh>,r1        ; zero all bits except RBLH
        mov     rx.bdl+bsw2,r2      ; r2 := receive status word 2
        bic     #^C<rbll>,r2        ; zero all bits except RBLL
        add     r1,r2               ; r2 := pkt length - 60 decimal
        add     #60.,r2             ; r2 := correct packet length
;
        rts     pc                  ; return to caller
;
; Receive Buffer Descriptor List.
;
rx.bdl: .word   unused              ; flag word
        .word   v                   ; Valid descriptor
        .word   rx.buf              ; address of receive buffer
        .word   -rx.len/2           ; 2s comp word length of buffer
        .word   unused              ; status word #1
        .word   unused              ; status word #2
;
; Follow with an invalid descriptor.
;
        .word   unused              ; flag word
        .word   invalid             ; this descriptor is INVALID
        .word   0,0                 ; dummy buffer address, length
;
; Reserve storage for maximum length receive buffer.
;
rx.buf: .blkb   1514.               ; 1514 (Decimal) bytes
rx.len  =       .-rx.buf
```

```
;
; *-------------------------------------------------------------*
; *                                                             *
; * Routine to wait for a packet from the Ethernet.             *
; * This routine waits for a Receive interrupt from the DELQA to *
; * to determine that a packet has been received.               *
; *                                                             *
; *-------------------------------------------------------------*
;
lqavec  =       400             ; DELQA host interrupt vector
pri07   =       340             ; Processor Status of ...
                                ; ...interrupt handler - IPL = 7
rxpkt2: bic     #ie!el,@lqacsr  ; disable DELQA interrupts...
                                ; ...and loopback modes
        bis     #il!re,@lqacsr  ; disable Internal Loopback...
                                ; ...and enable receiver
;
; We must write the address of the DELQA interrupt vector to the
; VAR without altering the Mode Select bit (bit15) in that
; register.
;
        mov     @lqavar,r1      ; r1 := (VAR)
        bic     #^C<ms>,r1      ; zero all except MS (bit 15)
        bis     #lqavec,r1      ; OR in addr of interrupt vector
        mov     r1,@lqavar      ; and write it back to the VAR
;
        mov     #lqaint,lqavec  ; load interrupt vector with...
                                ; ...address of interrupt handler
        mov     #pri07,lqavec+2 ; establish IPL of handler = 7
;
        clrb    rxdone          ; init RXDONE interrupt flag
        bis     #ie,@lqacsr     ; enable DELQA interrupts
        mov     #rx.bdl,@lqarll ; write addr of Rx BDL to DELQA
        clr     @lqarlh         ; initiate the reception
;
; Wait for the interrupt flag RXDONE to be set to '1' to signify
; completion of reception. Note that a real application program
; would need a timeout check.
;
                                ; REPEAT
10$:    tstb    rxdone          ;   test RXDONE flag
        beq     10$             ; UNTIL RXDONE is set to '1'
;
; Check the receive status to verify that no receive errors have
; occurred.
;
        mov     rx.bdl+bswl,r1  ; r1 := receive status word 1
        bic     #^C<bit15!bit14> ; zero all except LASTNOT...
                                ;  ...and ERROR/USED
        cmp     #lastok,r1      ; IF NOT receive error THEN DO
        beq     20$             ;    continue
                                ; ELSE DO
;
; Error handling code should be placed here.
;
20$:
;
; Calculate length in bytes of received packet.
; NOTE : This assumes that the packet received is NOT a loopback
;        packet. Therefore, the receive length as determined from
;        the receive status words is 60 decimal less than the
;        actual packet length.
;        Additional code is required to handle the loopback cases.
;
        mov     rx.bdl+bswl,r1  ; r1 := receive status word 1
```

```
        bic     #^C<rblh>,r1      ; zero all bits except RBLH
        mov     rx.bdl+bsw2,r2    ; r2 := receive status word 2
        bic     #^C<rbll>,r2      ; zero all bits except RBLL
        add     r1,r2             ; r2 := pkt length - 60 decimal
        add     #60.,r2           ; r2 := correct packet length
;
        rts     pc                ; return to caller
;
; Receive Buffer Descriptor List.
;
rx.bdl: .word   unused              ; flag word
        .word   v                   ; Valid descriptor
        .word   rx.buf              ; address of receive buffer
        .word   -rx.len/2           ; 2s comp word length of buffer
        .word   unused              ; status word #1
        .word   unused              ; status word #2
;
; Follow with an invalid descriptor.
;
        .word   unused              ; flag word
        .word   invalid             ; this descriptor is INVALID
        .word   0,0                 ; dummy buffer address, length
;
; Reserve storage for maximum length receive buffer.
;
rx.buf: .blkb   1514.               ; 1514 (Decimal) bytes
rx.len  =       .-rx.buf
```

C.7 EXECUTING ON-BOARD DIAGNOSTICS

The DELQA firmware contains self-test code which may be executed by setting the RS (Request to execute Self-test) bit to 1 in the VAR. RS is cleared to 0 by the firmware when the self-test has completed execution and the self-test status bits are updated in the VAR.

```
;
; *---------------------------------------------------------------*
; *                                                               *
; *    Routine to start Self Test running and wait for the        *
; *    results.                                                   *
; *                                                               *
; *---------------------------------------------------------------*
;
stast:  bis     #rs,@lqavar         ; request Self Test execution
;
; Wait for RS to be set to '0' again in the VAR. Note that a real
; application program would need a timeout check.
;
                                    ; REPEAT
10$:    bit     #rs,@lqavar         ;  test RS bit
        bne     10$                 ; UNTIL RS = '0'
;
; The Self Test status is stored in VAR<12:10>.
;
        mov     @lqavar,r1          ; r1 := (VAR)
        bic     #^C<s3!s2!s1>,r1    ; clear all but Self Test...
                                    ; ... status bits
        beq     20$                 ; Self Test found no fault
;
; Self Test has discovered a fault in the DELQA.
; Error handling/reporting code should be placed here.
;
20$:    rts     pc                  ; return to caller
```

C.8 BUFFER DESCRIPTOR MANAGEMENT ALGORITHM

The algorithm is described using pseudo code for the Transmit BDL, however this algorithm must be used for both Transmit and Receive BDLs.

- The host device driver manages each transmit and receive BDL as a "ring" or contiguous list of descriptors. Each BDL must have at all times at least one invalid buffer descriptor to terminate the BDL.

  The host loads new BDs onto this BDL while the processing other BDs on the same BDL.
- The host "locks" its own access to the BDLs by performing all host BD loading and unloading at the same Device Interrupt Priority Level (IPL). Thus host unloading never interrupts host loading.
- The host only specifies buffers in the receive BDL greater to or equal to 1518 bytes.
- The host always sets the Buffer Descriptor's VALID bit D<15> after the remaining descriptors fields are all set up.
- The host always clears the Buffer Descriptor's VALID bit D<15> after processing a completed descriptor within the interrupt service routine.

- **BD RECOVERY ALGORITHM PART 1: LOAD RECOVERY** After the host sets the VALID bit in a Buffer Descriptor (BD) with new data for the DELQA to transmit, it is recommended that the host use the following algorithm with the BD just issued to the DELQA:

```
if  (DELQA CSR bit XL = 1)
then begin

       if  (BD bit LASTNOT S<15> = 1) and
           (BD bit ERROR   S<14> = 0)
       then begin

              Load address pointer of BD into DELQA
              Transmit BDL Start Address Register.

            end

     end (* END OF LOAD BD ALGORITHM *)
```

- **BD RECOVERY ALGORITHM PART 2: UNLOAD RECOVERY** In the host device driver's interrupt service routine the following algorithm is highly recommended:

```
WRITE ONE TO CLEAR DELQA CSR BIT XI

start_bd_unload:

  BD = NEXT BD TO BE RETURNED BY DELQA ON TRANSMIT BDL

  if (BD VALID BIT D<15>)) = 1)
  then begin

         if  (BD bit LASTNOT S<15> = 1) and
             (BD bit ERROR   S<14> = 0)
         then begin

                if  (DELQA CSR bit XL = 1)
                then begin

                       if  (BD bit LASTNOT S<15> = 1)
                            and
                           (BD bit ERROR   S<14> = 0)
                       then begin

                                Load address pointer of
                                BD  into DELQA Transmit
                                BDL     Start     Address
                                Register.

                            end
                     end
              end
         else begin

                process completed BD and then  loop back
                to  "start_bd_unload:"  to  get  next BD
                which  will be  returned from  the DELQA
                on the transmit BDL.

              end

      end (* END OF UNLOAD BD ALGORITHM *)
```

Index

A

B

C

D

E

F

H

I

L

M

N

O

P

Q

R

S