DESQA TECHNICAL MANUAL

Order Number: EK-DESQA-TM-001

Prepared by Educational Services
of
Digital Equipment Corporation

August 1988

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

This document was prepared using VAX DOCUMENT, Version 1.0

CONTENTS

CHAPTER 2 Functional Description

CHAPTER 3 Programming

CHAPTER 4 Technical Description

CHAPTER 5 Maintenance

FIGURES

TABLES

PREFACE

The DESQA is a quad height module, designed to fit the BA200 series of enclosures. The module will provide an interface from the Q-Bus to one of two implementations of local area networks. The two are Thickwire (IEEE 802.3 10BASE5) and Thinwire (IEEE 802.3 10BASE2). Selection of the implementation supported is by a single jumper block on the module.
This manual describes how the DESQA communications module functions internally. It provides information for detailed system programming, field service support and for customer engineers

The chapters are as follows:

1. **CHAPTER 1** introduces the DESQA module and its interfaces.
2. **CHAPTER 2** is a functional description of the DESQA module.
3. **CHAPTER 3** describes how to program the DESQA module.
4. **CHAPTER 4** details the architecture and operations of the module.
5. **CHAPTER 5** maintenance
6. **APPENDIX A** summarises the details of the principal Integrated Circuits in the DESQA module.
7. **APPENDIX B** describes the Ethernet Network Management protocol.
8. **APPENDIX C** details the DESQA vector address and assignments.
9. **APPENDIX D** gives examples of host software programming of the DESQA.
10. **APPENDIX E** is a glossary.

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-CMINI-RM)

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

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

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

DECconnect System Installation and Verification Guide (EK-DECSY-VG)

DECconnect Planning and Configuration Guide (EK-DECSY-CG)

DESQA Installation Guide (EK-DESQA-IN)

FCC NOTICE: The equipment described in this manual generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A computing device pursuant to Subpart J Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interface when operated in a commercial environment. Operation of this equipment in a residential area may cause interference, in which case the user at his own expense may be required to take measures to correct the interference.

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This chapter introduces the M3127 module, which is a DIGITAL Ethernet Local-Area-Network switchable Q-Bus Adapter (DESQA). The sections are as follows:

Section 1.2 **ETHERNET OVERVIEW**
Section 1.3 **DESQA FUNCTIONAL OVERVIEW**
Section 1.4 **DESQA TECHNICAL OVERVIEW**
Section 1.5 **SPECIFICATION**

1.2 ETHERNET OVERVIEW

The DESQA can support either of two possible Ethernet cabling systems. These are the standard Ethernet which will be referred to as Thickwire Ethernet, and Thinwire Ethernet. The principal characteristics of the Ethernet are:

Topology:	Branching bus
Medium:	Shielded coaxial cable
Transmission:	Manchester-encoded digital baseband signalling
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).

1.2.1 Safety Warnings

See Figures 1-1 and 1-2.

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 forbindelsessystem, 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 1–1 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 והמחוברת 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 המבידדים רצף מתכתי.
במידה ויתעוררו בעיות, 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 1–2 Warnings

1.2.2 Thickwire Ethernet-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 kilometres (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 1-3 shows an example of a large-scale Ethernet configuration.

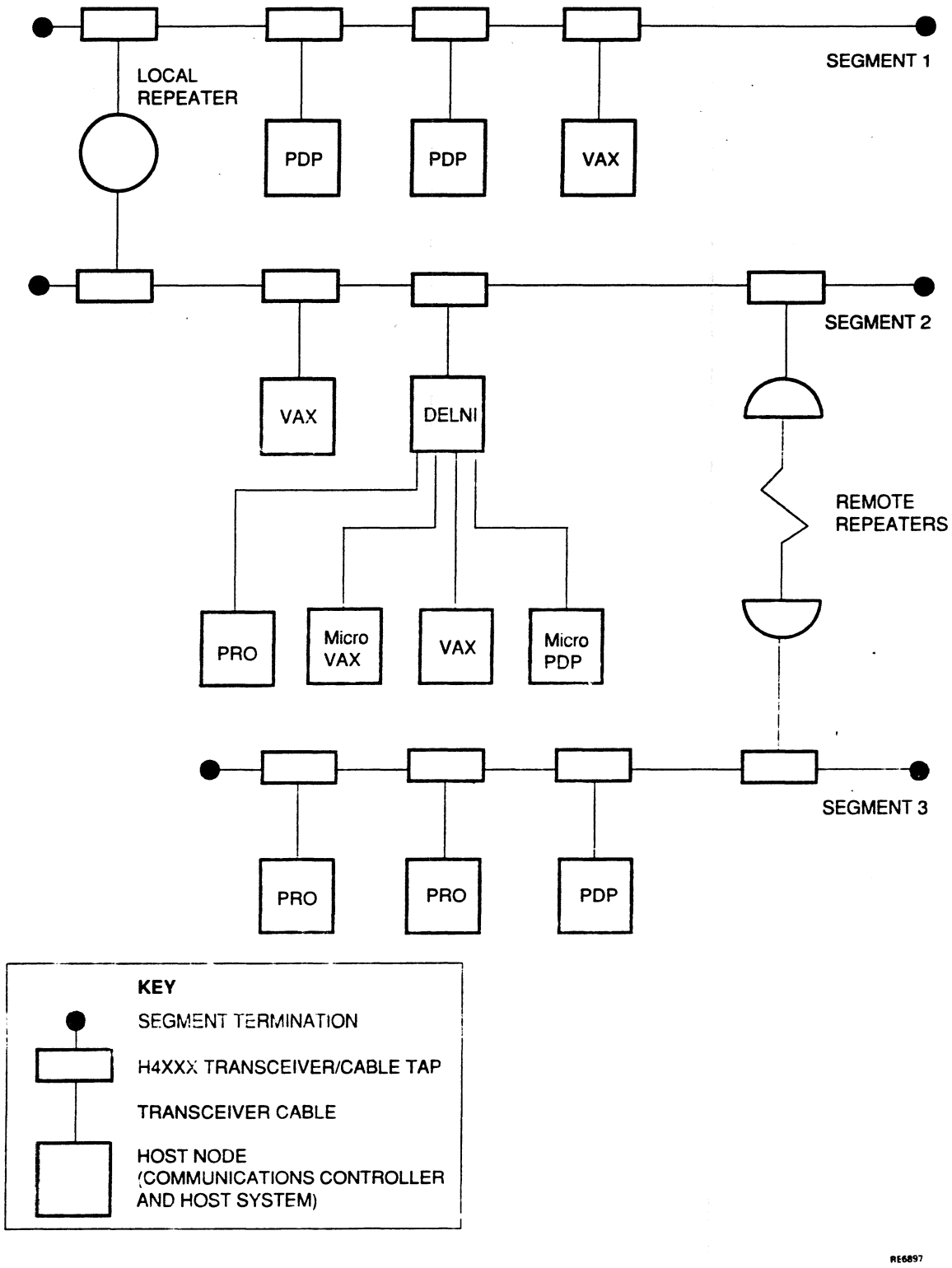


Figure 1–3 Typical Ethernet Configuration

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 neighbours. 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 kilometres (1.74 miles). This would comprise three segments of 500 meters, plus six transceiver cables of 50 meters, plus a 1 km fibre 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.

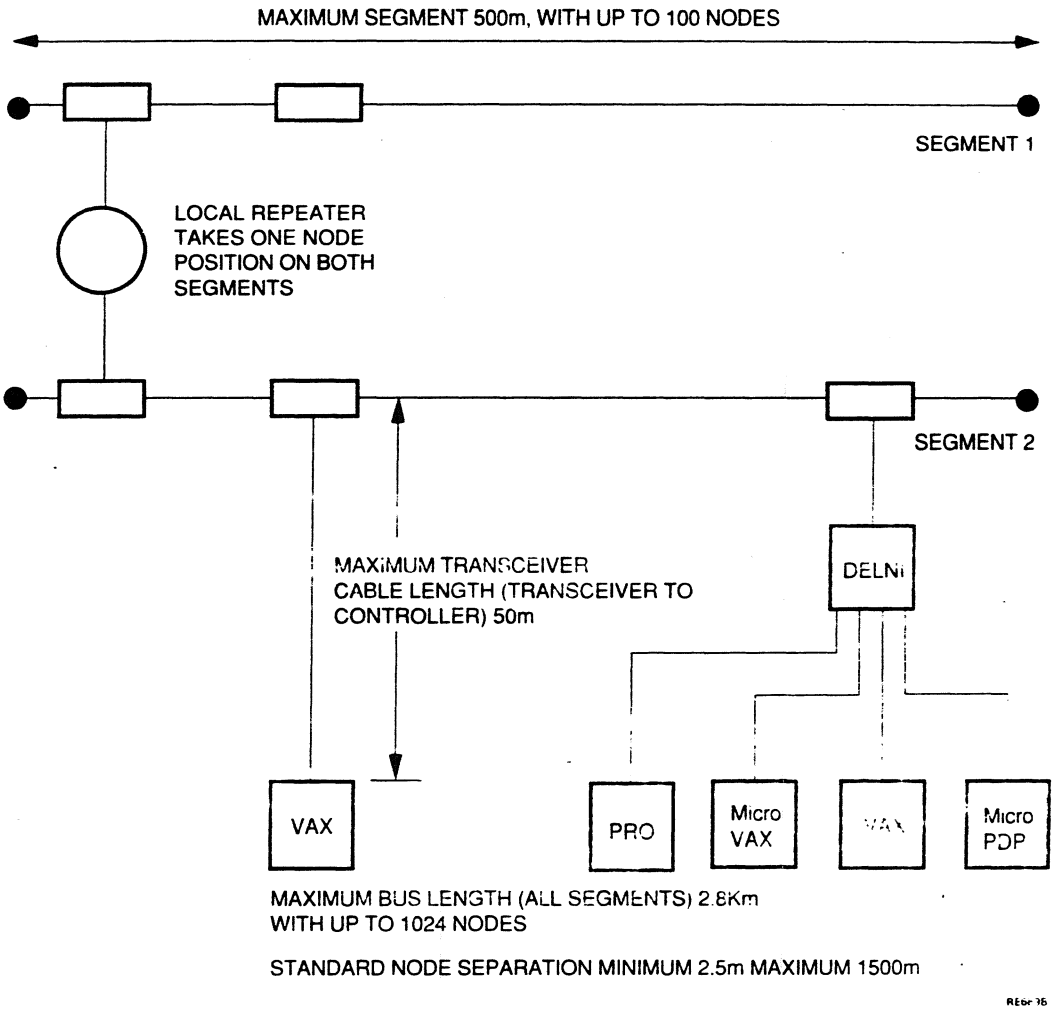


Figure 1–4 Ethernet Connectivity

1.2.3 Thinwire Ethernet-General Description

Thinwire Ethernet offers an alternative baseband cabling system that provides full Ethernet capability for personal computers, workstations, and low-end systems in offices and other local work areas. Thinwire Ethernet is a low-cost, flexible companion system to Thickwire Ethernet that utilises an industry standard cable to connect up to 30 stations in one 185 meter segment.

The DESQA is part of a cabling and hardware set that connects Q-Bus systems in BA200 series enclosures, to a standard Thinwire Ethernet cable segment. The maximum length of a Thinwire Ethernet segment is 185 meters. 30 devices may be connected to a single segment. A Thinwire Ethernet daisy chain segment consists of sections linked together by BNC T-connectors. The bottom of the "T" is inserted into a DESQA, which is inserted into the system enclosure. "T" connectors may be a minimum of 0.5 meters apart.

It is possible to have up to eight of these daisy chains by using a DEMPR to form a network of up to 232 stations. A local network interconnect (DELNI) may be used to concentrate up to eight DEMPR's.

For more information on Thinwire configuration using DEMPR, see the Digital Thinwire Ethernet Multiport Repeater (DEMPR Installation and User's Guide (EK-DEMPR-UG).

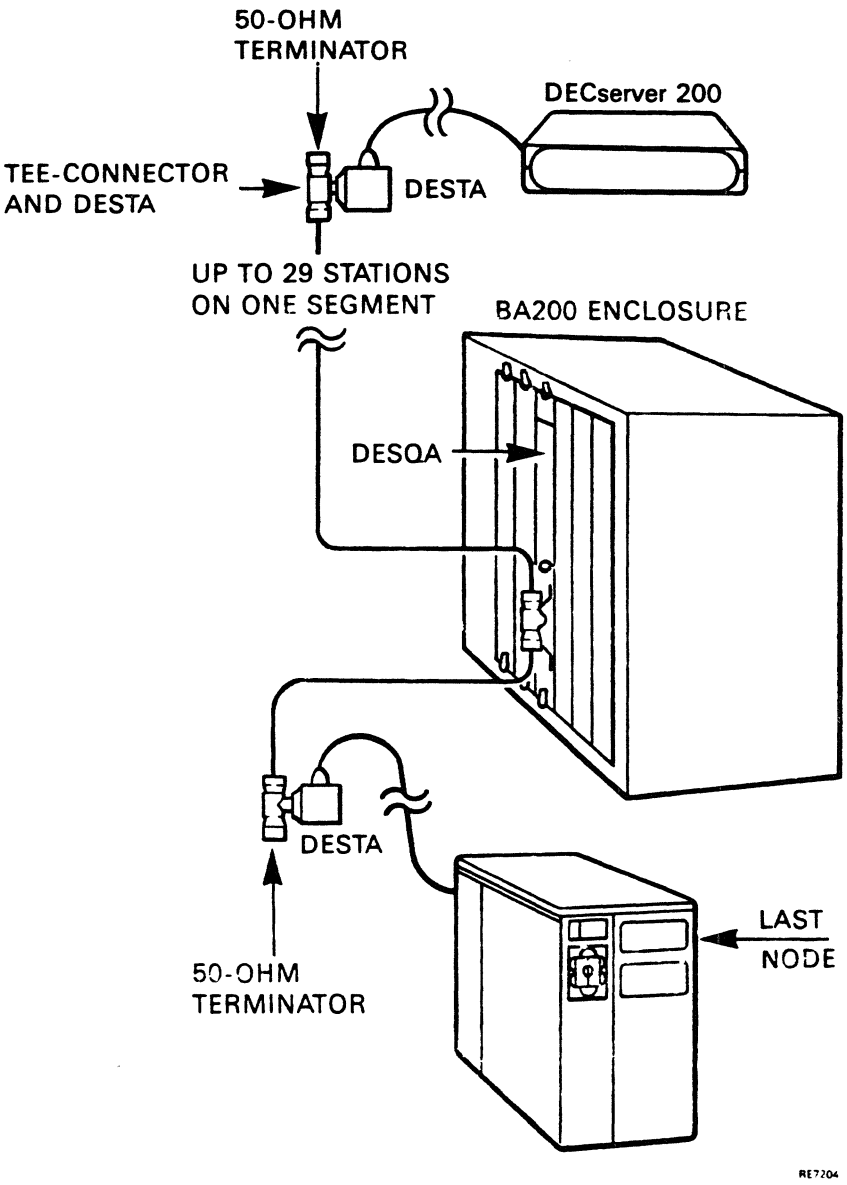


Figure 1–5 Typical Thinwire Ethernet DESQA/DESTA/DTE Daisy Chain.

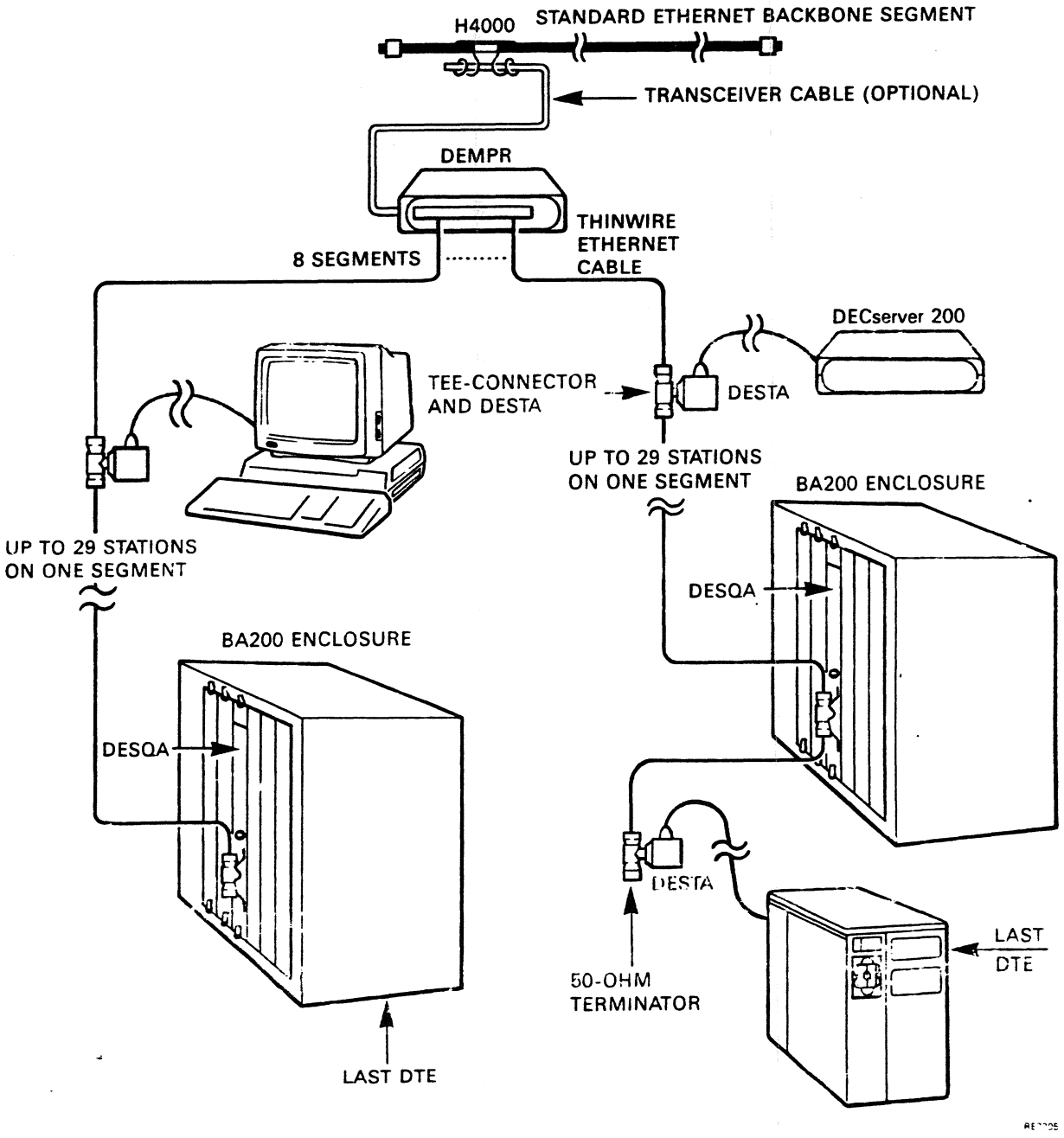


Figure 1–6 Typical Ethernet Backport to DEMPR to DESQA/DESTA/DTE Daisy Chain

1.2.4 Ethernet Layers

The Ethernet architecture is structured in two layers which correspond to the lowest layers in the International Standards Organisation (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 DESQA 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, inter-network communication, and the user interface.

1.2.5 Data Encapsulation

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

Figure 1-7 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 synchronisation. 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 multi-cast 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 DESQA 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 packet prior to 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 1-7 shows the standard Ethernet packet format.

NOTE

In an IEEE 802.3 packet, the type field is replaced with a 2 byte length field to indicate the number of data bytes contained in the data field.

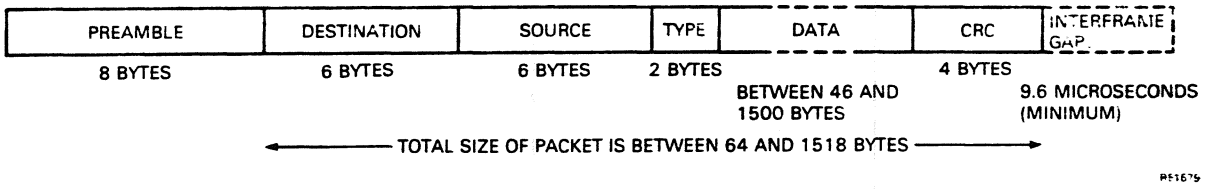


Figure 1–7 Ethernet Packet (Frame) Format

1.3 FUNCTIONAL OVERVIEW

The DESQA (DIGITAL Ethernet Switchable Q-Bus Adapter) is a quad height module having a recessed bulkhead which physically supports the interface connectors. This type of handle is required by the nature of the BA200 series enclosures, which eliminate the need for cabkits. The DESQA module is functionally equivalent to a DELQA with a cabkit for Thickwire networks combined with DESTA Thinwire transceiver for interfacing to Thinwire networks.

The DESQA provides an interface from the Q-Bus to either a Thinwire or Thickwire Ethernet (IEEE 802.3 Network). It is fully supported by both PDP-11 and MicroVAX families available in the BA200 enclosures. The choice of Thickwire or Thinwire is switch selectable.

1.3.1 Normal Mode and DEQNA-Lock Mode

The DESQA module operates in one of two switchable modes: Normal mode or DEQNA-lock mode.

In **Normal mode**, DESQA supports the following functions:

- Maintenance Operations Protocol (MOP) messages for Remote BOOT, Request ID, Transmit System ID and Loopback
- IEEE 802.3 Maintenance Messages for XID (Transmit ID) and TEST on NULL LSAP (Link-layer Service Access Point) access points
- Self-test on powerup and via host command
- Single Ethernet physical address (the first of any specified in a setup packet to replace the default held in Station Address ROM)
- Multiple Ethernet Multi-cast addresses

- All standard DEQNA functions, except multiple Ethernet physical addresses and the automatic enabling of the on-board sanity timer at powerup.

In **DEQNA-lock mode**, DESQA provides functional compatibility with DEQNA modules, but at the expense of losing some Normal mode functions. The functions supported are:

- Multiple Ethernet physical addresses
- Multiple Ethernet Multi-cast addresses
- Sanity timer (Switch enabled on powerup).

The operating mode is selected at powerup by the setting of the mode switch (S3) on-board the DESQA module.

If mode switch S3 is closed, the module operates in Normal mode; subsequently, host software can use the Vector Address Register (VAR15) to set the operating mode to either Normal mode or DEQNA-lock mode. If mode switch S3 is open, the operating mode is fixed as DEQNA-lock, and this cannot be altered by software. Use of DEQNA-lock mode is not recommended.

Host software can determine whether the module is a DESQA or a DEQNA by using bit 0 in the Vector Address Register.

1.3.2 Module Interfaces

The DESQA is a microprocessor-based device which provides all the logic necessary to connect to the Ethernet. It acts as a data communications controller, executing the physical layer protocol and part of the data link layer protocol (as defined by the seven-layer OSI model). The protocol functions include synchronisation, format conversions, encoding and decoding, formatting data packets and data link management.

The DESQA enables programs that execute higher levels of protocol, such as DECnet, to communicate with their peers over the Ethernet link.

The DESQA module performs all the channel access functions necessary to achieve maximum throughput with minimum intervention from the host processor, including:

- Block-mode Direct Memory Access (DMA) to host memory
- Control DMA, which uses Buffer Descriptor Lists (BDLs) in host memory to sequence transfers between data buffers in the host and in the DESQA.

The DESQA implements some of the Maintenance Operation Protocol (MOP) functions on request from a remote station without host intervention. These functions include host reboot, loopback operations, and system identification.

1.3.3 Module Operations

The DESQA module transfers data between buffers in host memory and the Ethernet transceiver. The main data flow operations are as follows:

1. Transfer of data between host memory buffers and shared RAM in the DESQA.
2. Conversion of data and status information between the host format and that used in the DESQA module.
3. Transfer of formatted data packets between shared RAM and the Ethernet connector.

The data is formatted in packets of between 60 and 1514 data bytes at a higher level of protocol. The DESQA module calculates and appends a four byte CRC (Cyclic Redundancy Check) to transmit packets, and strips the CRC from receive packets. Therefore, the full length of a packet on the Ethernet is between 64 and 1518 bytes.

Figure 1–8 shows how transmit and receive data is passed between the main functional components of the DESQA module.

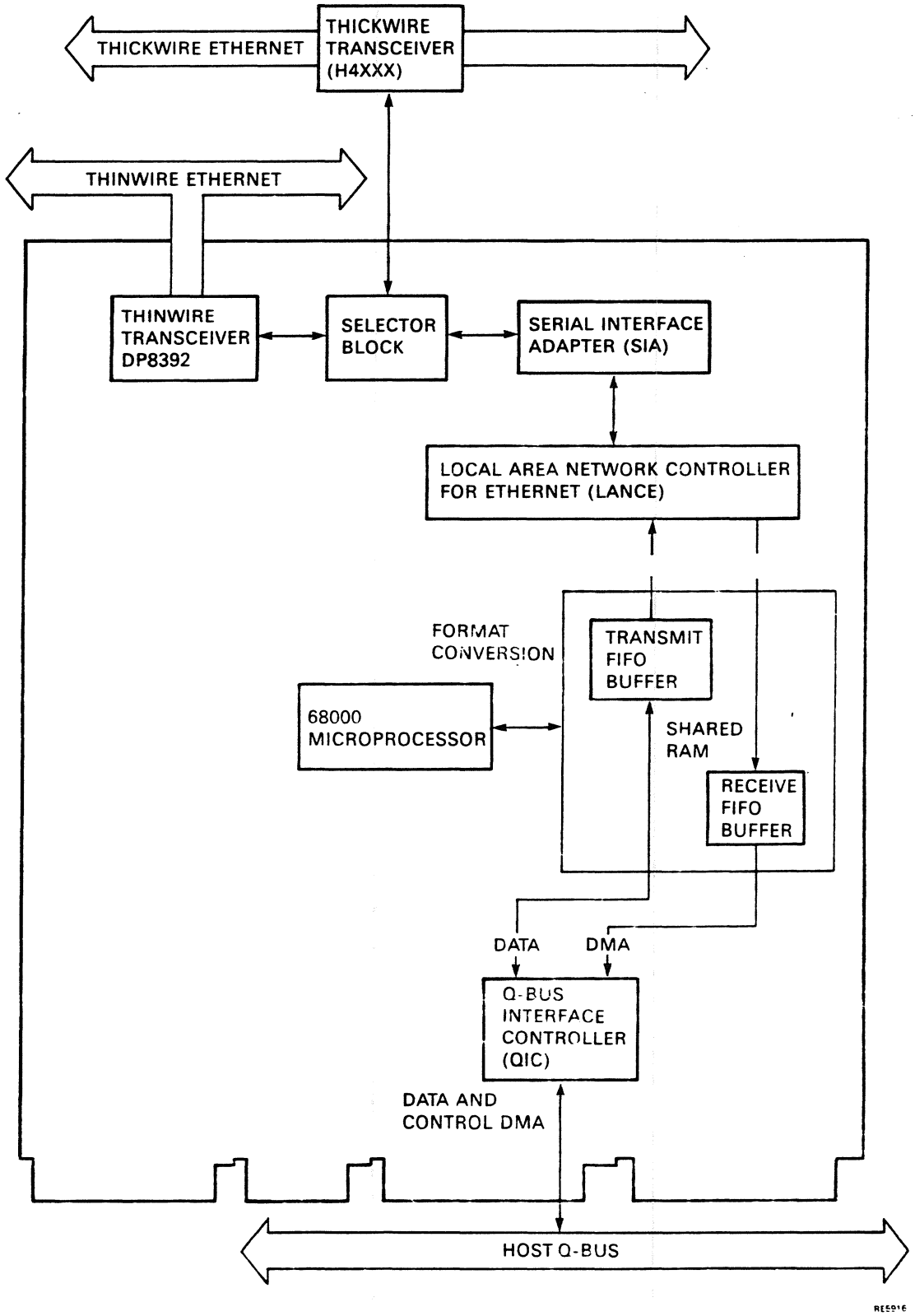


Figure 1–8 DESQA Functional Block Diagram

1.3.4 Protocol Functions

The Physical Channel Functions and the Data Link Functions are described in this section.

1.3.4.1 Physical Channel Functions

The DESQA module transmits and receives at 10 Mbit/s. It provides physical channel functions that are specific to Ethernet and necessary for the interface to the DIGITAL H4*xxx* Ethernet transceiver. DESQA executes the following functions:

During transmission

- Encodes serial data in Manchester format
- Recognises heartbeat and self-test collision detect signals from the DIGITAL H4*xxx* transceiver, verifying that the transceiver is monitoring collision detect signals.

During reception

- Senses transmission carrier from any Ethernet station
- Decodes the incoming serial bit-stream from its Manchester format.

1.3.4.2 Data Link Functions

The DESQA module provides the following Ethernet-specific functions at the data link layer:

During transmission

- Generates the 64-bit preamble for synchronisation
- Provides parallel-to-serial conversion of the frame
- Calculates the 32-bit CRC value and places it in the packet sequence field for transmission
- Attempts automatic, multiple re-transmissions upon receiving a collision detect signal.

During reception

- Checks the 32-bit CRC value in each incoming packet
- Performs address filtration, either to match an incoming message to the physical address of the module, or to accept messages broadcast to a group of stations
- Synchronises to the preamble, and removes it prior to processing
- Provides serial-to-parallel conversion of the frame.

The Ethernet provides a datagram service; that is, sending a packet does not guarantee that it will reach its destination. The software for the protocol layers above data link is responsible for checking that a packet has been received, and for recovering from conditions where it has not been received.

1.3.5 Q-Bus Interface

As bus master of the Q-Bus, the host can program the DESQA module by means of eight word-length registers in the I/O page of the Q-Bus memory map. The DESQA acts as bus slave to support access from the Q-Bus to these on-board locations.

Four of the I/O page addresses are write-only registers, used to pass the start addresses of the BDLs for transmit and receive buffers. Two are read/write registers: the Control and Status Register (CSR) and the Vector Address Register (VAR).

The lower bytes of each of the first six addresses are read-only registers, used to access the Station Address (SA) ROM. The SA ROM contains the 48-bit physical address of the DESQA module in the Ethernet LAN.

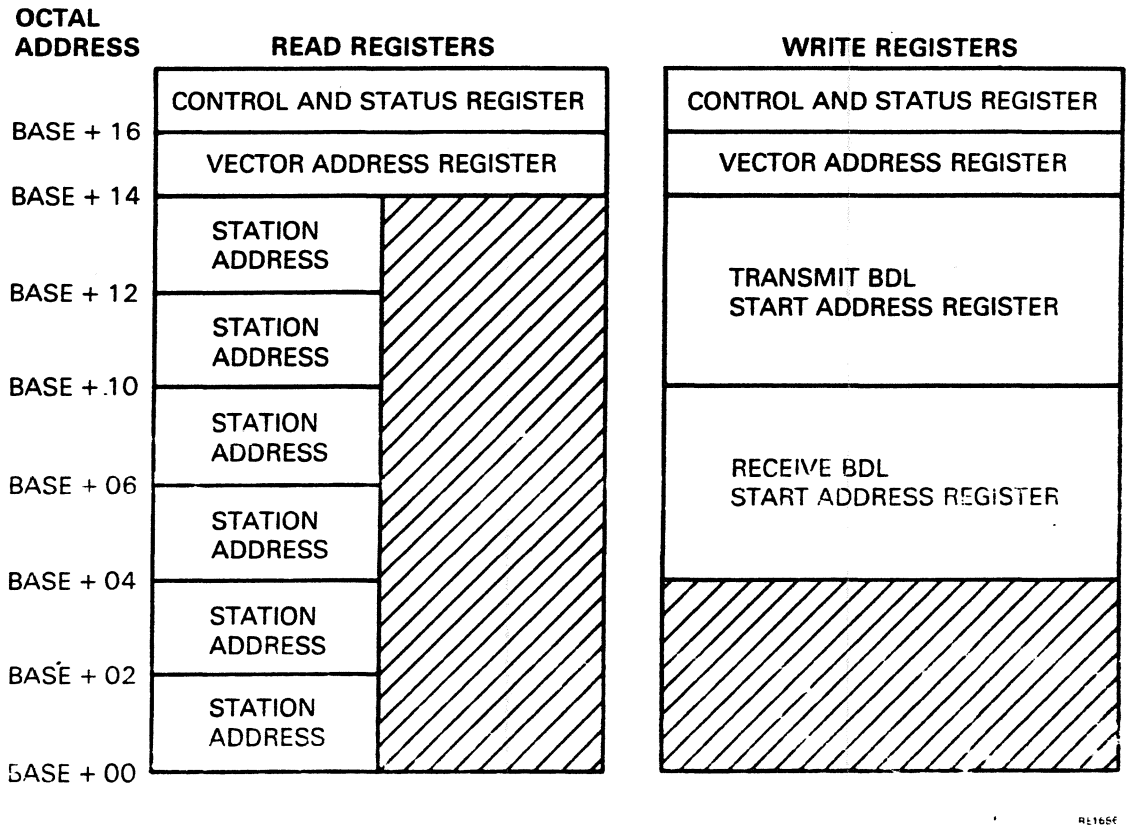


Figure 1–9 Host I/O Page Map

1.4 TECHNICAL OVERVIEW

The DESQA module comprises of one quad-height Q-Bus module. The DESQA module plugs into the Q-Bus backplane inside the BA200 enclosure.

1.4.1 Module Components

On the DESQA board there are six main integrated circuits to perform the main module operations (refer to Section 1.3.3). These operations are controlled by the on-board microprocessor. Data storage is provided by shared RAM.

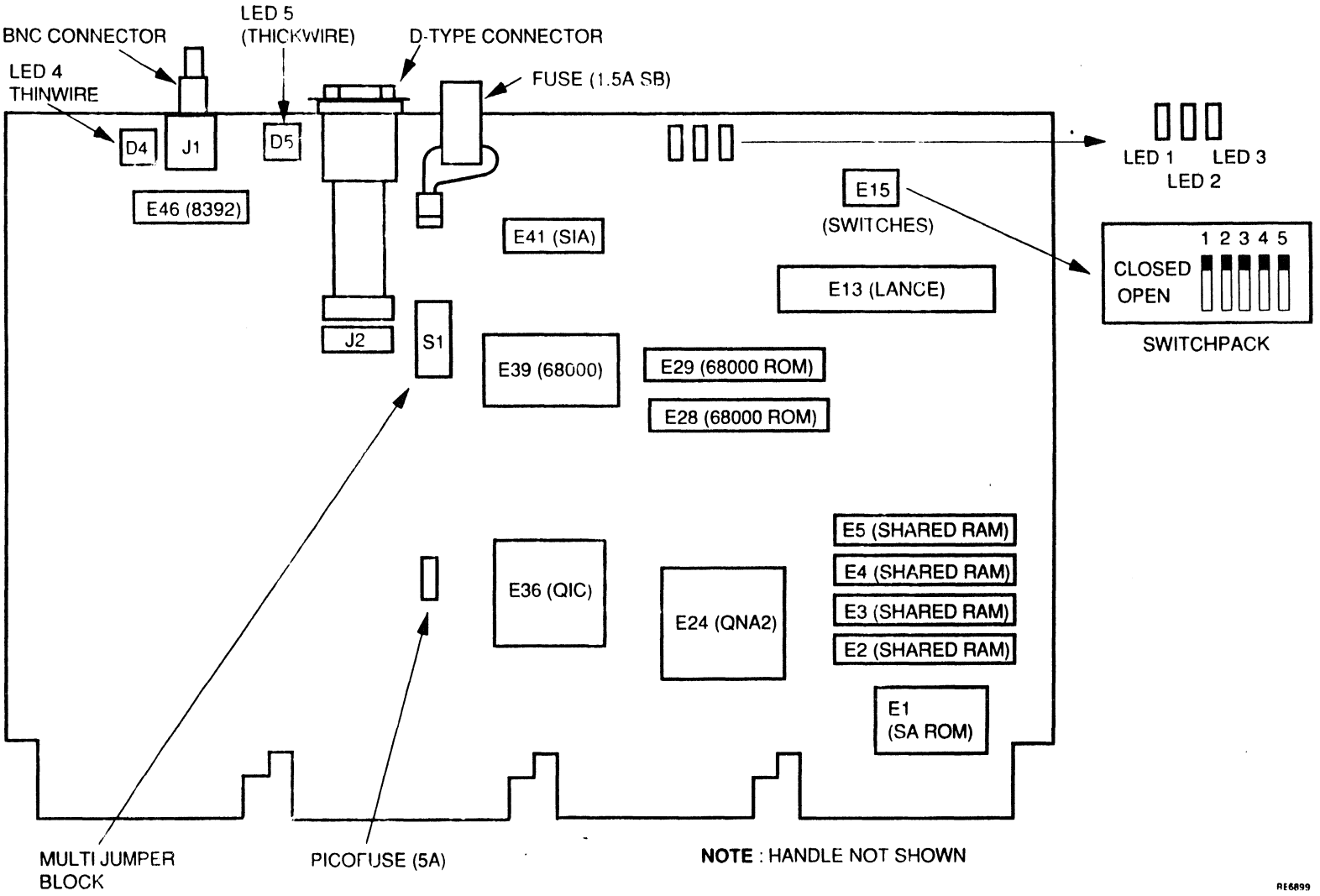


Figure 1–10 DESQA Physical Layout

Figure 1–11 shows the major components of the DESQA module and their interconnections.

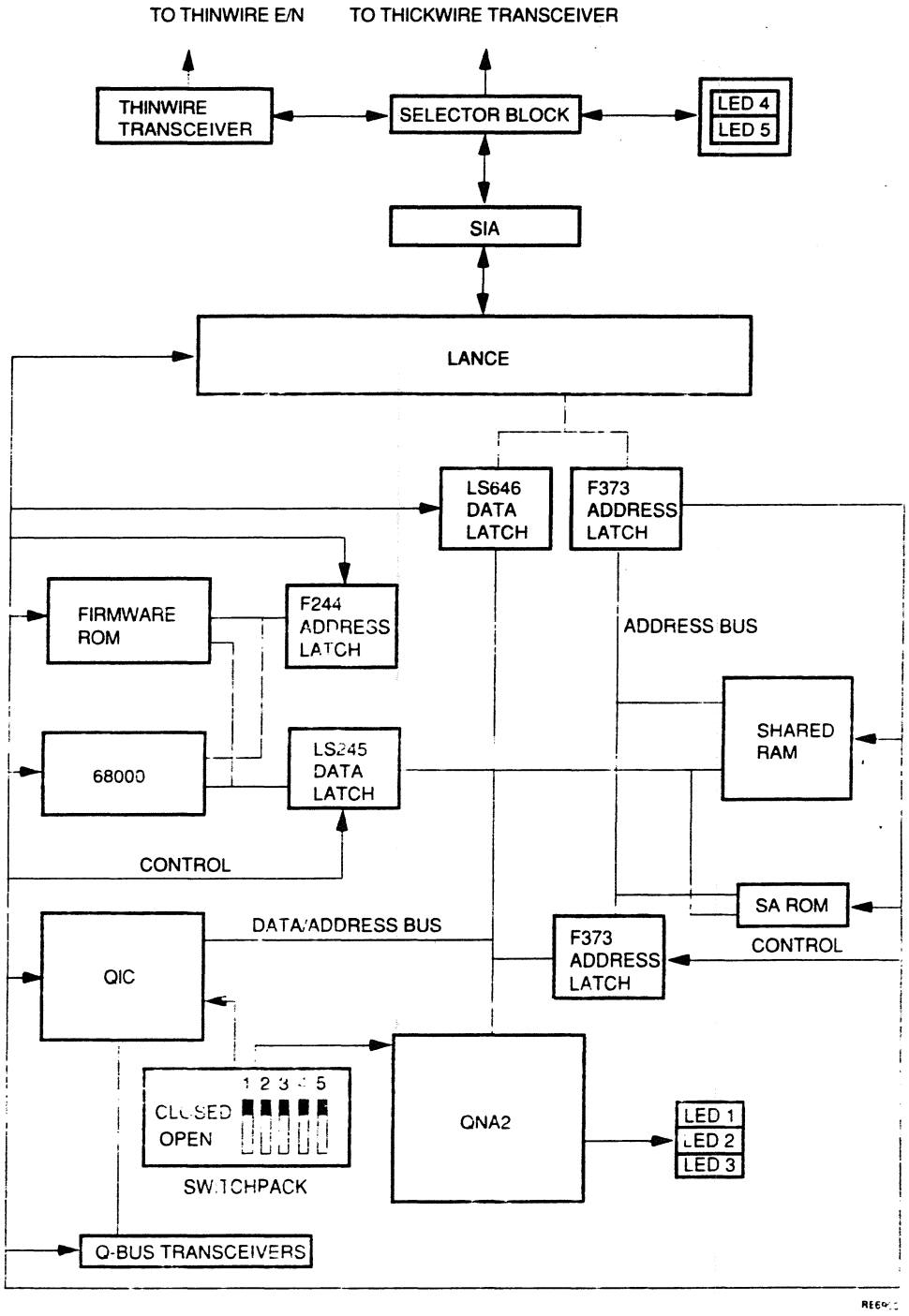


Figure 1–11 DESQA Hardware Block Diagram

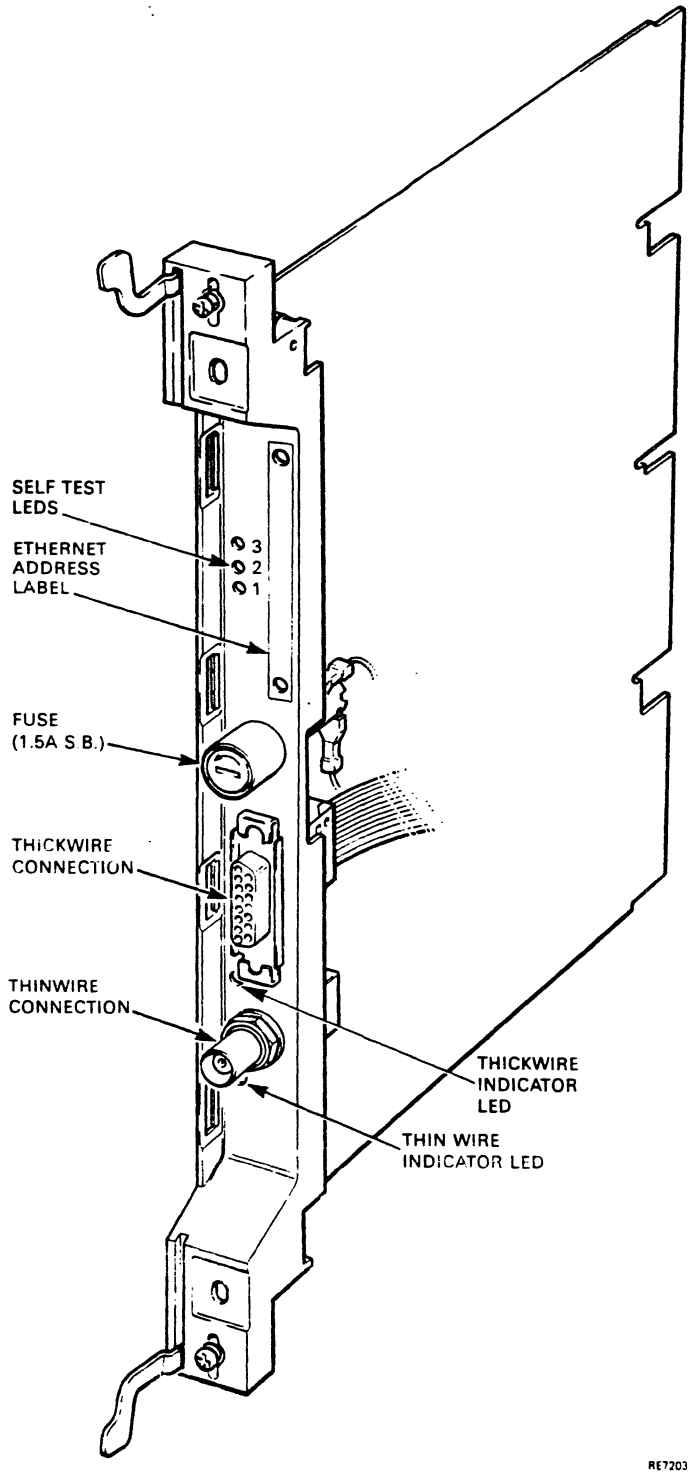


Figure 1–12 DESQA Handle

1.4.1.1 Integrated Circuits

The principal integrated circuits in the DESQA module are:

1. **Q-Bus Interface Controller (QIC)**

   This gate array provides a DMA path between the host system memory and the shared memory on the DESQA board, and between SA ROM and host memory. The QIC also allows the host system access to the internal registers of the DESQA.

2. **68000 Microprocessor**

   This is the 16-bit microprocessor which controls the DESQA and performs the necessary conversions of the frame data between host buffer format and Ethernet packet format.

3. **Q-Bus Network Arbitrator (QNA2)**

   This gate array provides access to the backport bus, arbitrating between the QIC, the 68000 microprocessor, and the LANCE.

4. **Local Area Network Controller for Ethernet (LANCE)**

   This is a VLSI circuit which performs bidirectional DMA between shared RAM and the Serial Interface Adapter (SIA).

5. **Serial Interface Adapter (SIA)**

   This is a VLSI circuit which links the LANCE to the transceiver. (H4xxx for Thickwire onboard DP8392 for Thinwire). The SIA provides Manchester encoding of data transmitted, and decoding of data received.

6. **Thinwire Transceiver (DP 8392)**

   This is a VLSI circuit which links the serial interface Adapter (SIA) to the Thinwire Ethernet BNC connector.

1.4.1.2 On-board Memory

The DESQA module contains both preprogrammed ROM and static RAM, as follows:

1. Firmware ROM contains 12K words of master control program for the module (executed by the 68000 microprocessor). In addition,the firmware ROM contains 4Kbytes of self test diagnostic code and 4 Kbytes of PDP-11 boot/diagnostic code for execution by a PDP-11 host. (MicroVAX systems provide equivalent boot/diagnostic code in their own host system ROM.)
2. Station Address (SA) ROM. (32 bytes) contains, in the first six bytes, the default physical address of the DESQA module. The SA ROM also contains a 2 byte checksum which is accessible to host software.
3. Shared RAM (16Kwords) provides the memory for data packet buffers, Buffer Descriptor Lists (BDLs) the Vector Address Register, the LANCE initialisation block (which includes the physical address of the module) and firmware data structures.

1.4.1.3 DC–to–DC Converter

This converts the +12Vdc from the backplane to the -9V required for the DP8392 transceiver.

1.4.2 Module Processing Operations

The operation of the DESQA can be broken down into processing modules, as follows:

1. The QIC transfers data between Q-Bus HOST memory and shared RAM, as follows:
   - Block-mode (or non-block-mode) data DMA (block size up to 16 words) to transfer data buffers directly to and from shared RAM
   - Control DMA (block size up to four words) to transfer control information and Buffer Descriptor List (BDL) entries. Control DMA does not use block-mode.
2. The 68000 microprocessor translates data and status information between the format recognised by the host system and that recognised by the LANCE.
3. The QNA2 provides access to the backport bus, arbitrating between (in order of priority): the QIC, the 68000 microprocessor, and the LANCE. It also provides all the gating and strobing signals necessary for these devices to access shared RAM.
4. The LANCE and the SIA operate together to transfer the translated packets between shared RAM and the Ethernet transceiver.

   If Thickwire mode is selected, the packets pass through the 15 way D-type connector on the handle and onto the transceiver cable. Alternatively, if Thinwire mode is selected, the packets pass through the Thinwire transceiver chip and BNC connector which connects via a "T" piece to the Thinwire cable.

5. The SIA uses Manchester coding to encode transmitted data and decode received data. The SIA detects and synchronises with incoming data, and passes it on to the LANCE. The LANCE checks the destination address, and, if it matches any of the module addresses, transfers the data direct by DMA into shared RAM.
6. The 68000 microprocessor controls the operation of the DESQA module, including initialisation of I/O and module self-test (implemented on powerup or initiated by host software).

1.4.3 Network Integrity Functions

The DESQA module provides the following features for network integrity:

- Self-test diagnostic execution on powerup or after host software initialisation
- Sanity timer to monitor host software
- Controller loopback
- Maintenance Operations Protocol (MOP) messages for Remote Monitor Console (RMC) operations: Remote BOOT, Request ID, Transmit System ID
- Maintenance Operations Protocol (MOP) messages for Ethernet Channel Test (ECT): Loopback
- Maintenance Operations Protocol (MOP): datalink counters, maintained and stored by DESQA
- IEEE 802.2 Maintenance Messages for XID (Transmit ID) and TEST on NULL Link-layer Service Access Points (LSAPs).

To assist in fault diagnosis and network management, the DESQA can also operate in promiscuous addressing mode. Effectively, this disregards the internal address filter logic. This allows the DESQA to accept all packets received from the network, and to verify the integrity of the received data by performing a 32-bit CRC check on each received packet. All transmissions (normal,loopback, and Setup) reset the sanity timer without affecting its status (enabled or disabled).

1.4.3.1 Self-test Diagnostics

In Normal mode, the DESQA executes a comprehensive self-test on powerup. This takes approximately five seconds to complete.

The firmware ROM on the DESQA contains 4 Kbytes of PDP-11 boot/diagnostic code. If the module is controlled by a PDP host, the host can execute this code in order to increase fault coverage. This enables the DESQA to determine that it is operating correctly, before it attempts to access the Ethernet.

1.4.3.2 Sanity Timer

The sanity timer acts as a check that the host software is operating effectively. If the host communications driver fails to reset the timer periodically, the timer initiates a system reboot at the host.

All transmissions (normal, loopback and setup), reset the sanity timer, without affecting it's status (enabled or disenabled).

1.4.3.3 Controller Loopback Modes

In internal loopback, the DESQA 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 powerup. The behaviour of the device differs according to its mode.

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

1.4.3.4 Remote Console Commands

When operating in Normal mode, the DESQA responds to the following MOP remote console commands:

- System Identification Request from another Ethernet station. The DESQA sends the current System ID parameters from shared RAM
- System ID Transmission is sent periodically to a Multi-cast Ethernet address
- Remote trigger command from another Ethernet station. The DESQA verifies the request, and then causes the host system to reboot by negating BDCOK on the Q-Bus interface in order to simulate a powerup restart.

1.4.3.5 Ethernet Channel Loopback

DESQA recognises Ethernet loopback test messages. The module checks to determine whether to forward or return the incoming message. Messages are received, decoded, and re-transmitted by DESQA independently of host software. Normal messages are passed through as usual.

Table 1–2 DESQA Bulkhead Thickwire D-Type Connections

Pin	Signal Name
6	Voltage common(+12V return)
13	Voltage supply (+12V)
14	Voltage shield
5	Receive +
12	Receive -
4	Receive shield
11	Transmit shield
3	Transmit +
10	Transmit -
1	Control-In shield
2	Collision Presence +
9	Collision Presence -
7	Control-Out A
15	Control-Out B
Shell	Protective Ground (Conductive Shell)

1.5 SPECIFICATIONS

The DESQA 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 1-3 gives the specifications of the DESQA module.

Table 1–3 DESQA Specifications

Physical	Dimension	Imperial	Metric	
	Height	12.13 inches	30.8 cm	
	Width	0.95 inches	2.4 cm	
	Length	9.45 inches	24 cm	
	Weight	20.3 ounces	0.58 kg	
Electrical	**Voltage**	**Tolerance**	**Typical current**	**Maximum current**
	+5.0 V	±5%	2.15 A	2.7 A
	+12.0 V	±5%	0.21 A	1.5 A
Q-Bus loads	**AC**	**DC**		
	3.3	0.5		
Temperature	**Environment**	**Specification**		
to 66C(151F)\)	Storage	-40C(-40F)		
	Operation	5C(41F) to 50C(122F)		

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

Airflow	Specification
	Airflow across the board must limit the outlet temperature to a maximum of 50C(122F), and the temperature rise across the board to 19C(50F).
	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 70C(158F) under any environmental conditions.

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

CHAPTER 2
FUNCTIONAL DESCRIPTION

2.1 SCOPE

This chapter describes the main operations of the DESQA module, and its system port interface with the Q-Bus. The sections are as follows:

2.2 MODULE OPERATIONS

The DESQA module transfers data between buffers in host memory and either the Thickwire Ethernet transceiver or the Thinwire Ethernet. The main operations are as follows:

1. Transfer of data between host memory buffers and shared RAM in the DESQA.
2. Conversion of data and status information between host format and that used in the DESQA module.
3. Arbitration of access requests on the backport bus.
4. Transfer of encapsulated data packets between shared RAM and the Ethernet connector.
5. DESQA module control.

The data is brought together at a higher level of protocol to form packets of between 60 and 1514 data bytes. The DESQA module calculates and appends a four byte CRC to transmit packets, and strips the CRC from receive packets. Therefore, the full length of a packet on the Ethernet is between 64 and 1518 bytes.

2.2.1 Q-Bus Transfers

The QIC is a dual-ported device: one set of data and address lines is connected to the Q-Bus; the other interfaces with the backport bus in the DESQA module. In the data DMA path, each port is double buffered, and each port is controlled by sequencer logic in the QIC. Figure 2–1 shows the Q-Bus interface functions.

The QIC supports DMA between the Q-Bus and shared RAM:

1. Block-mode or (non-block-mode) data DMA.
2. Non-block-mode control DMA.

Table 1–3 (Cont.) DESQA 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 1C(1.8F) for each 1000 meters (3281 feet) of altitude, unless constant cooling is provided.	

These two types are described in the next sections.

2.2.1.1 Data DMA

The QIC can execute up to 16 words of block-mode DMA between Q-Bus system memory and shared RAM each time it acquires control of the Q-Bus. The transfer must use either read only instructions, or write only instructions.

When executing DMA transfers to block-mode memory, the QIC relinquishes the Q-Bus whenever:

1. The transfer reaches a 16 word boundary.
2. Another device is requesting the Q-Bus when the transfer reaches a 7 word boundary. This causes the QIC to relinquish the bus at the 8 word boundary.

2.2.1.2 Control DMA

The QIC can also handle up to four words of control DMA, using mixed read and write operations on the Q-Bus to access BDLs. This is known as control DMA; it is used to transfer status information as well as to access descriptors, as follows:

1. Buffer Descriptor Fetch. This operation acquires the Q-Bus, executes one non-block-mode write and three non-block-mode read instructions, then relinquishes the Q-Bus.
2. Store status. This operation acquires the backport bus, executes two non-block-mode write instructions, then relinquishes the Q-Bus.

2.2.1.3 Non-block-mode DMA

When executing transfers to non-block-mode memory, the QIC relinquishes the Q-Bus whenever it reaches a four-word boundary.

2.2.1.4 Data and Status Conversion

The 68000 microprocessor handles:

1. Formatting of buffers for transmission.
2. Reformatting of packets received from the Ethernet.
3. Conversion of status information for both transmit and receive messages.

Figure 2–4 shows the data transfers involved in formatting the message data.

2.2.2 Backport Bus Arbitration

The Q-Bus Network Arbitration (QNA2) controls access to the backport bus on the DESQA module, arbitrating between access requests from the QIC, the 68000 microprocessor, and the LANCE. The QNA2 provides all the gating and the strobing signals necessary for these devices to access the shared RAM.

Figure 2–3 shows the control and gating signals that the QNA2 generates to control the backport bus.

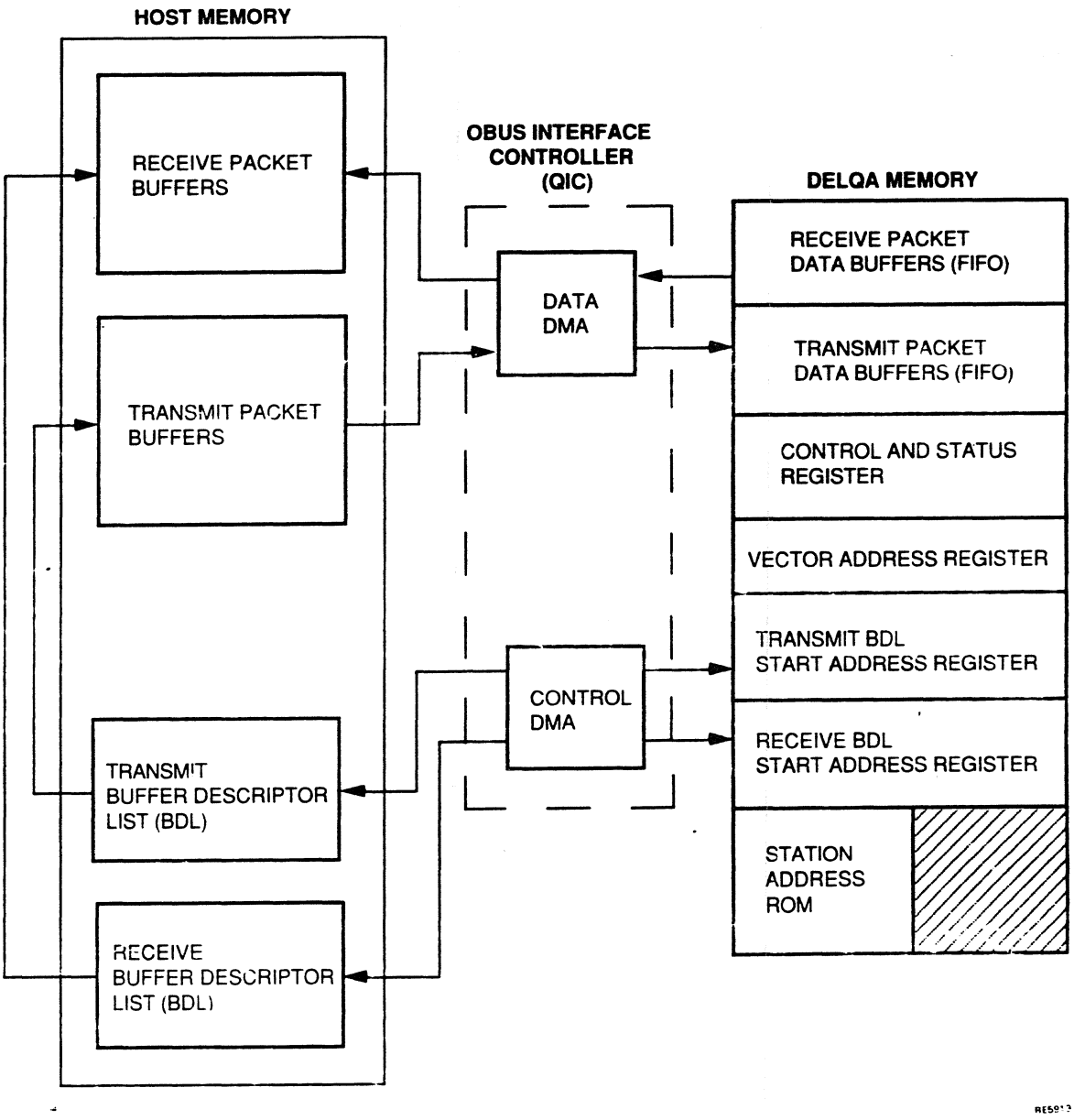


Figure 2–1 Q-Bus Data Structures

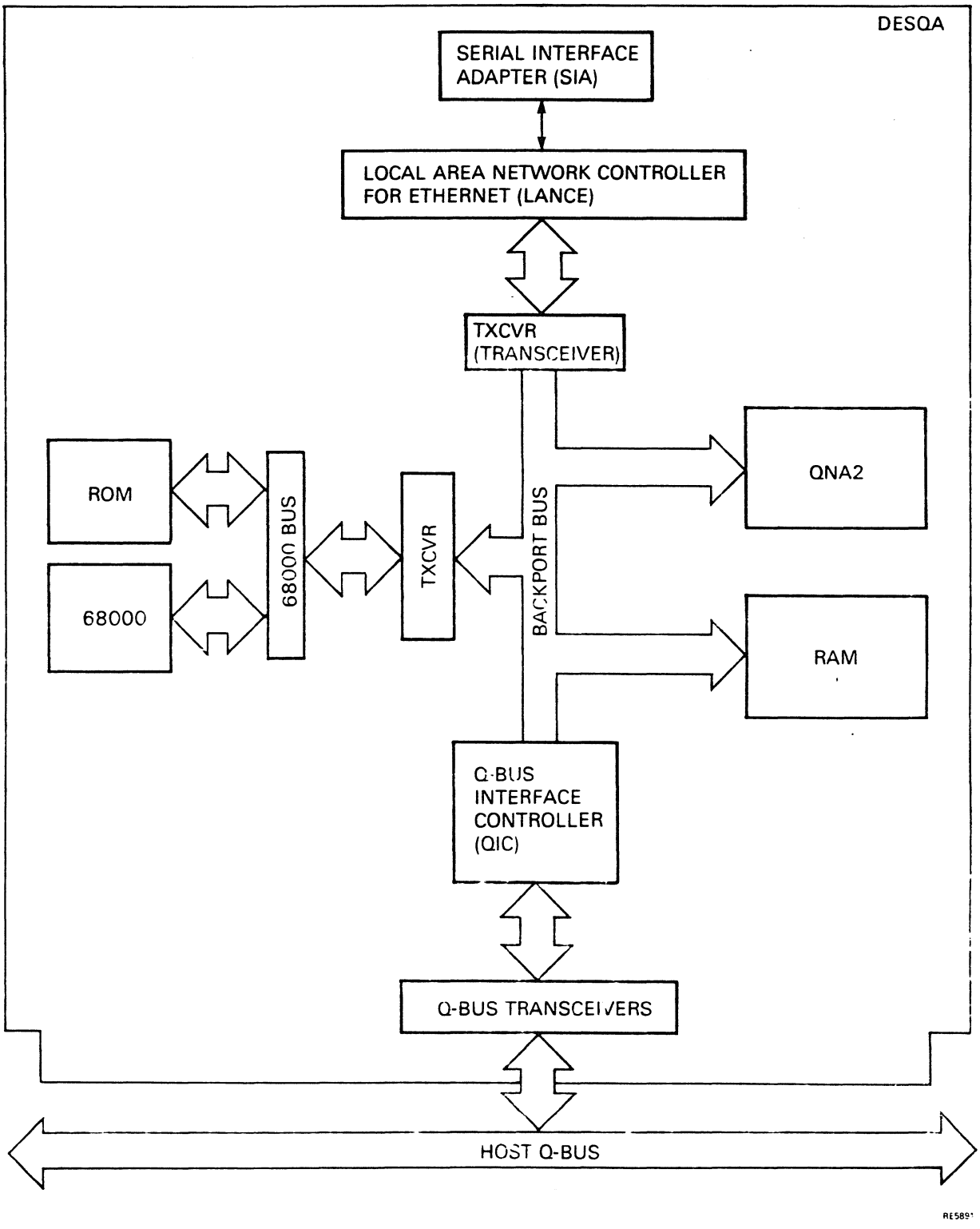


Figure 2–2 DESQA Backport Bus Structure

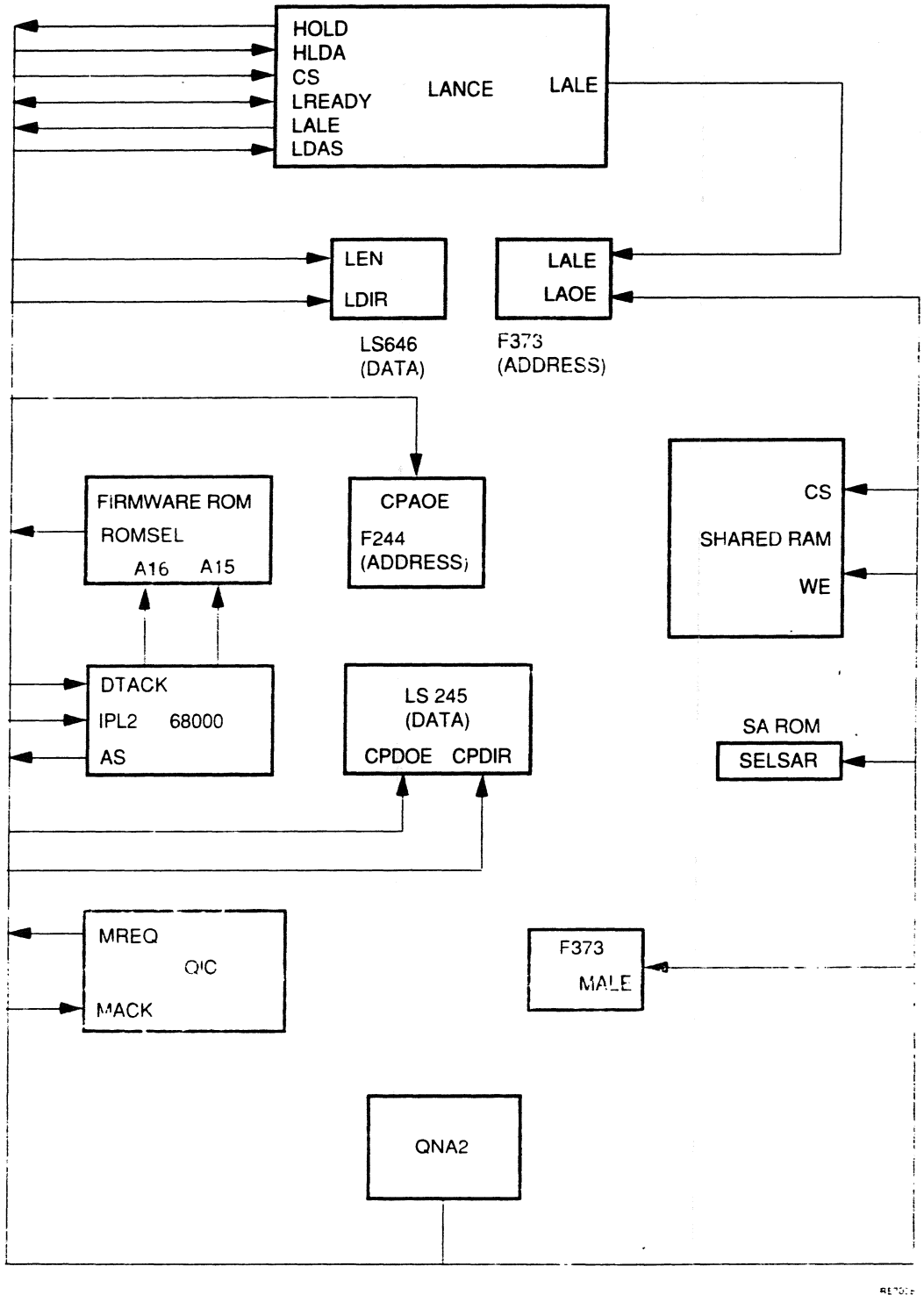


Figure 2–3 Backport Control Functions

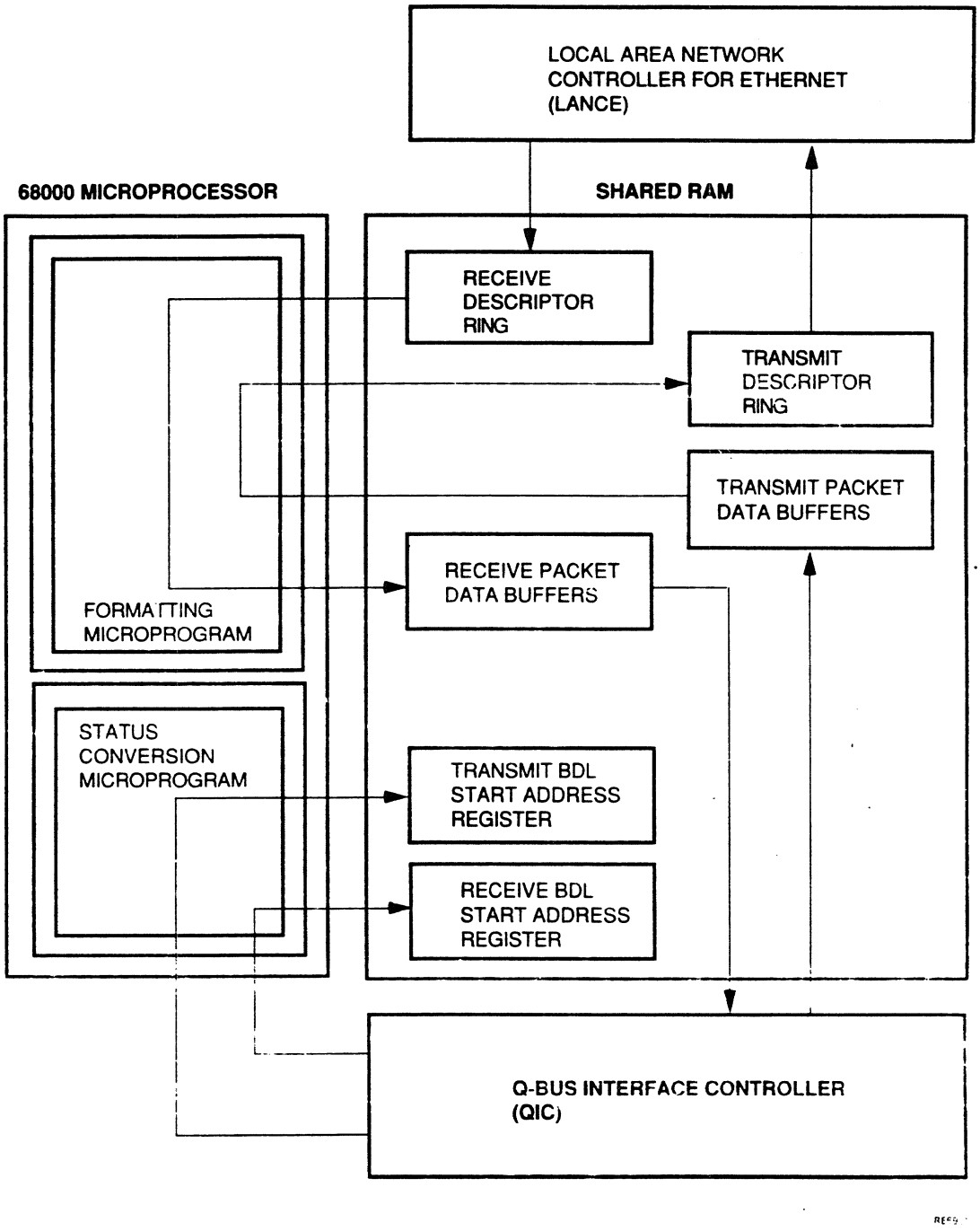


Figure 2–4 DESQA Data Path

2.2.3 Ethernet Transfers

Ethernet transfers are handled by the LANCE (Local Area Network Controller for Ethernet) in conjunction with the SIA (Serial Interface Adapter).

2.2.3.1 Transmission

When the 68000 has translated a data frame into a packet suitable for transmission, the LANCE transfers it by DMA from the shared RAM to the SIA. The SIA Manchester encodes the data, and sends the packet using either the internal Thinwire Ethernet transceiver, or the external Thickwire Ethernet Transceiver.

2.2.4 Ethernet Transfers

Ethernet transfers are handled by the LANCE (Local Area Network Controller for Ethernet) in conjunction with the SIA (Serial Interface Adapter).

2.2.4.1 Transmission

When the 68000 has translated a data frame into a packet suitable for transmission, the LANCE transfers it by DMA from the shared RAM to the SIA. The SIA Manchester encodes the data, and sends the packet using either the internal Thinwire Ethernet transceiver, or the external Thickwire Ethernet Transceiver.

2.2.4.2 Reception

When the SIA detects activity on the Ethernet, it synchronises with the preamble of the incoming message. The SIA then decodes the data and transfers the packet to the LANCE. The LANCE filters the destination address. If the address on the incoming packet matches one of the addresses in module memory, the LANCE transfers the data by DMA to the shared RAM.

2.2.5 Master Module Control

The 68000 microprocessor executes the master control program for the DESQA module. The master control program is held as firmware in the ROM associated with the microprocessor on the 68000 bus, and executes the following functions:

1. Conversion of control and status information.
2. Initialisation of QIC and LANCE on powerup.
3. DESQA self-test on powerup or on initiation by host.
4. Interrupt control.

The QNA2 contains the CSR (Control and Status Register), and the DESQA IR (Interrupt Register) for the DESQA module.

2.3 SYSTEM PORT INTERFACE

The DESQA is accessed through twelve registers which are mapped into the Input/Output page of Q-Bus address space in the host system. When it receives the valid address of one of the registers from the bus master, the DESQA responds as a bus slave.

2.3.1 Port Registers

The address mapping implemented by the QIC enables the host system to access the following internal registers in the DESQA:

1. **Control and Status Register (CSR).**

   This is a one-word, read/write register held in the QNA2.

2. **Vector Address Register (VAR).**

   This is a one-word, read/write register held in the shared RAM.

3. **Receive BDL Start Address Register.**

   Transmit BDL Start Address Register (BDL SARs).

   These are two-word, write-only registers that are maintained by the host in shared RAM.

4. **Station Address ROM.**

   This is a set of six read only memory bytes (the lower bytes of the first six words in the DESQA space).

All the registers are word addressable only. The 68000 microprocessor can access these registers directly from the backport bus.

2.3.2 Master Module Control

The 68000 microprocessor executes the master control program for the DESQA module. The master control program is held as firmware in the ROM associated with the microprocessor on the 68000 bus, and executes the following functions:

1. Conversion of control and status information.
2. Initialisation of QIC and LANCE on powerup.
3. DESQA self-test on powerup or initiated by host.
4. Interrupt control.

The QNA2 contains the CSR (Control and Status Register), and the DESQA IR (Interrupt Register) for the DESQA module.

2.4 CONFIGURING THE DESQA MODULE

The DESQA can support either of two possible baseband Ethernet cabling systems, standard or Thinwire Ethernet. You set a jumper block on the module to match the cabling system. The DESQA is shipped from the factory in Thinwire cable mode.

The jumper block has three rows of pins (Figure 2–5).

- To select Thinwire, you cover the middle row of pins and the pins labelled Thinwire.
- To select Thickwire, you cover the middle pins and the pins labelled Thickwire.

You use a switchpack (Figure 2–5) to select the CSR address and other operating features of the DESQA. The switchpack has five switches, S1 through S5. Table 2–1 lists their functions. The DESQA is shipped from the factory with all five switches closed.

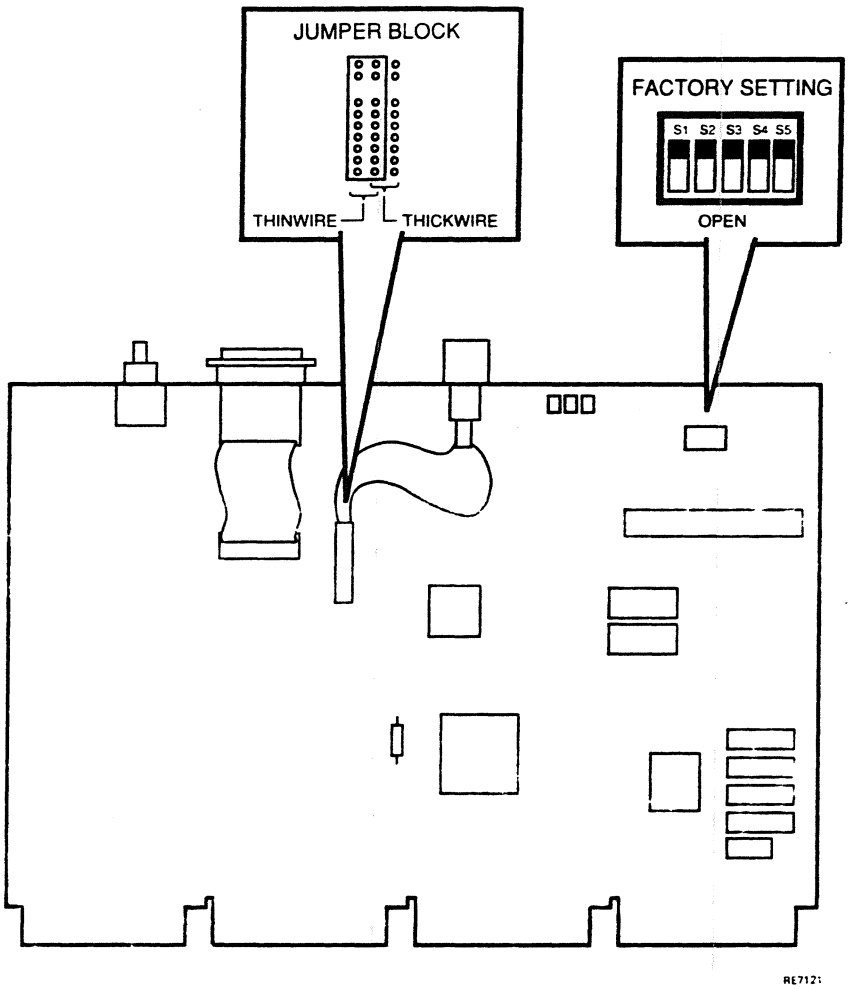


Figure 2–5 DESQA-SA Module Layout

Table 2–1 DESQA Switches

Switch	Setting	Function
S1	closed	Selects CSR address 17774440. (factory)
	open	Selects CSR address 17774460. (for second DESQA)
S2	–	Reserved.
S3	closed	Selects normal mode. (factory)
	open	Selects DEQNA lock mode.
S4	closed, and S3 closed	Remote boot disabled. (factory)
	open, and S3 closed	Remote boot enabled.
	open, and S3 open	Sanity timer automatically enabled at power-up.
	closed, and S3 open	Sanity timer not enabled at power-up.
S5	–	Reserved.

The DESQA can operate in DEQNA lock mode or normal mode, which you select by using S3. Table 2–2 lists the differences between DEQNA lock mode and normal mode.

Note that S4 is an option switch, whose function depends on the position of S3. S4 enables a sanity timer that monitors the host for inactivity. All transmissions (normal, loopback, or setup) reset the sanity timer without affecting its status (enabled or disabled). If a hardware or software problem occurs, the timer reaches its limit, causing the host to reboot itself. You should enable the sanity timer *only* for specific applications.

NOTE

If you enable the sanity timer in DEQNA mode and download software or diagnostics, the sanity timer may time out before the load is complete. This time-out causes the system to reboot.

The DESQA interrupt vector of 120 is written into a read/write register by software. If you install a second DESQA, its interrupt vector floats, with a rank of 47.

Table 2–2 DESQA Mode Differences

Support	DEQNA Lock Mode	Normal Mode
All DEQNA functions	yes	yes
Maintenance operation protocol (MOP) functions (including remote boot)	no	yes
Self-test support	no	yes
Boot/diagnostic code support	yes	yes
Sanity timer	yes	no

CHAPTER 3
PROGRAMMING

3.1 SCOPE

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

3.2 OVERVIEW

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

- Module initialisation
- 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 DESQA.

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

Communication between the host and the DESQA is accomplished through buffer descriptors organised 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 DESQA 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 DESQA takes the following action.

1. 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 DESQA sets XL in the Control and Status Register (CSR), and takes no further action.
2. If the descriptor is valid, and the DESQA has a transmit data buffer available, then a data-DMA transaction is performed to copy host data into the DESQA RAM, using the host buffer address and byte count supplied in the descriptor.
3. The next host descriptor is read, and steps 2 and 3 are repeated. If the descriptor is invalid, then the DESQA assumes it has reached the end of the list, sets XL in the CSR, and takes no further action.
4. When the packet has been transmitted on to the Ethernet, the DESQA writes the transmit status back to the host using a control-DMA WW (write-write) operation.
5. The DESQA 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.

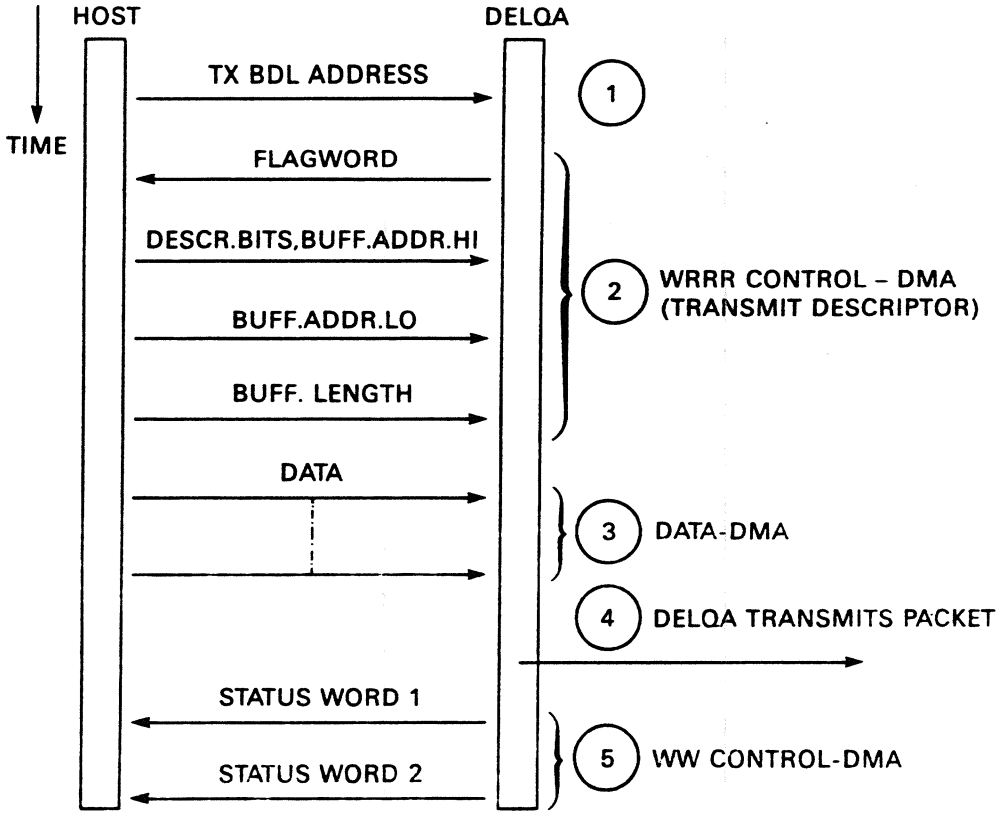
3.2.2 Receive—Data Reception from Ethernet to Host

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 DESQA Rx BDL register. The host must always terminate the list with an invalid entry.

If the host wishes to receive packets for an address other than that in the SA ROM, then the host must specify this address using a setup packet (see Section 3.6).

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

1. Starting at the address provided by the host, the DESQA 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 DESQA.
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 DESQA sets RL (Receive List Invalid) in the CSR, and takes no further action.
3. If the descriptor is valid, and the DESQA has received any packets from the network, then a data-DMA transaction is performed to copy the packet from the DESQA 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 DESQA 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 DESQA writes the status associated with the received packet back to the host using a control-DMA Write-Write (WW) operation.
6. The DESQA 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

RE4*.2

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 DESQA's hardware registers.

3.3.2 Control and Status Registers

Each DESQA is assigned a block of eight word-locations in the Q-Bus I/O page. These locations are word-addressable only, and the DESQA acts as a bus slave to support access by the host software to the DESQA 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 DESQA space). The Station Address (SA) ROM contains the 48-bit physical address of the DESQA 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 DESQA 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 DESQA and BDLs in host memory.

The registers are assigned to fixed blocks, so that more than one DESQA module can be mixed with other DESQA 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 DESQA Unit I/O Base Addresses

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

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 DESQA.

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

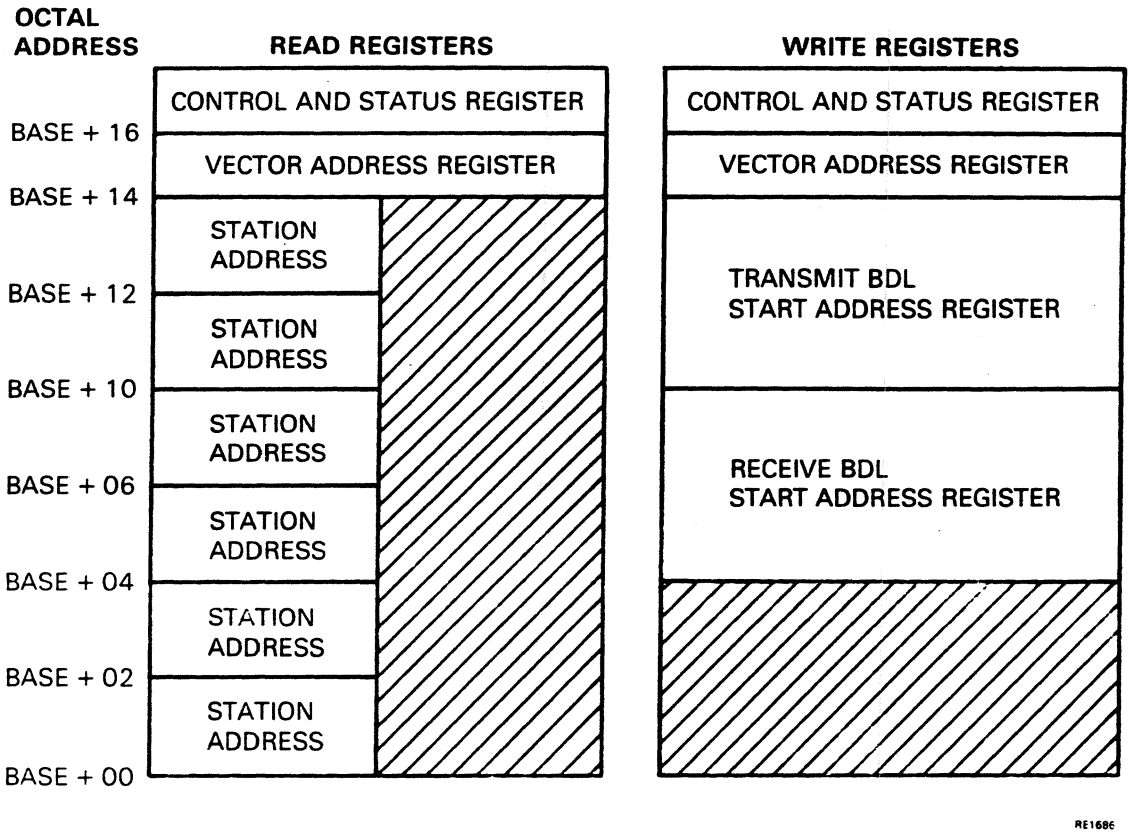


Figure 3–2 Host I/O Page Map

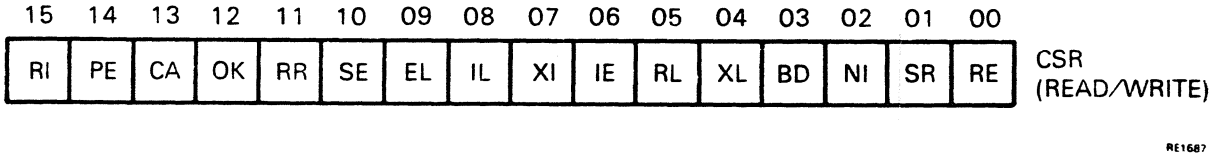


Figure 3–3 Control and Status Register (CSR)

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

Bit	Register Access	Description	State after Software Reset or Self-Test	State after Power-Up 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
CSR13	R	Carrier from Receiver Enabled	0 (Clear)	0 (Clear) CA

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

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

Bit	Register Access	Description	State after Software Reset or Self-Test	State after Power-Up Reset
		NOTE **If Thickwire mode is selected then a short circuit on the Thickwire Ethernet cable can be detected by CSR bit 13. If, however, Thinwire mode is selected, it will not be possible to detect a short circuit on the Thinwire Ethernet.**		
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 DESQA.

When cleared: Disables reception of datagrams from the DESQA.

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.

NOTE.1
The DESQA may still be active for up to 100µs after the clearing of the software reset bit.

NOTE.2
Setting this bit does not reset the device. It must be first set then immediately cleared. See Section 3.6.3

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

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

This bit is set or cleared by the host only.

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

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

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

This bit is set by the DESQA 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 DESQA 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 DESQA 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 DESQA always interprets an invalid descriptor as marking the end of a list.)

The DESQA 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 DESQA 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 DESQA to set this bit (that is, the list is considered invalid on reset).

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

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

When set: The DESQA 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 DESQA always interprets an invalid descriptor as marking the end of a list.)

The DESQA 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 DESQA that the host has a list of receive buffers into which the DESQA may deliver packets.

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

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

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

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

1. The DESQA has completed a transmit operation
2. The DESQA has completed a receive operation.
3. The DESQA 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 DESQA 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 DESQA will issue an interrupt to the host.

When reset: This bit is set by the DESQA 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 DESQA 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 DESQA 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 DESQA 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 DESQA 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 DESQA 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 DESQA 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 DESQA 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 DESQA will issue an interrupt to the host.

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

This bit is set by the DESQA 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 initialises VAR<09:02> with the address of the vector to the DESQA 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 summarises the bit definitions for Normal mode operations. More complete bit definitions follow the table.

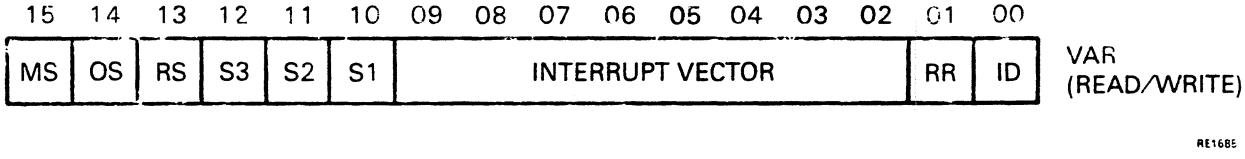


Figure 3–4 Vector Address Register (VAR)

Table 3–3 Vector Address Register (VAR)

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

Key:

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

(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

The self-test can be initiated in Normal mode, by power-up reset, or by a host write to VAR bit 13 after the self-test completes.

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 DESQA 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 due to an unconnected transceiver. 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.

The VAR bits record the DESQA status, as follows.

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

When set: VAR00 provides an identity test to distinguish a DESQA 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 DESQA

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 DESQA 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.

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 DESQA 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, in order 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 initialised 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 DESQA starts transfers as soon as it receives the high-order address.

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

The DESQA 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 initialised, the module starts DMA transfers of outgoing messages to shared RAM

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 DESQA 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 DESQA into external loopback mode.
3. Read the lower byte of word 1 in the DESQA I/O block.

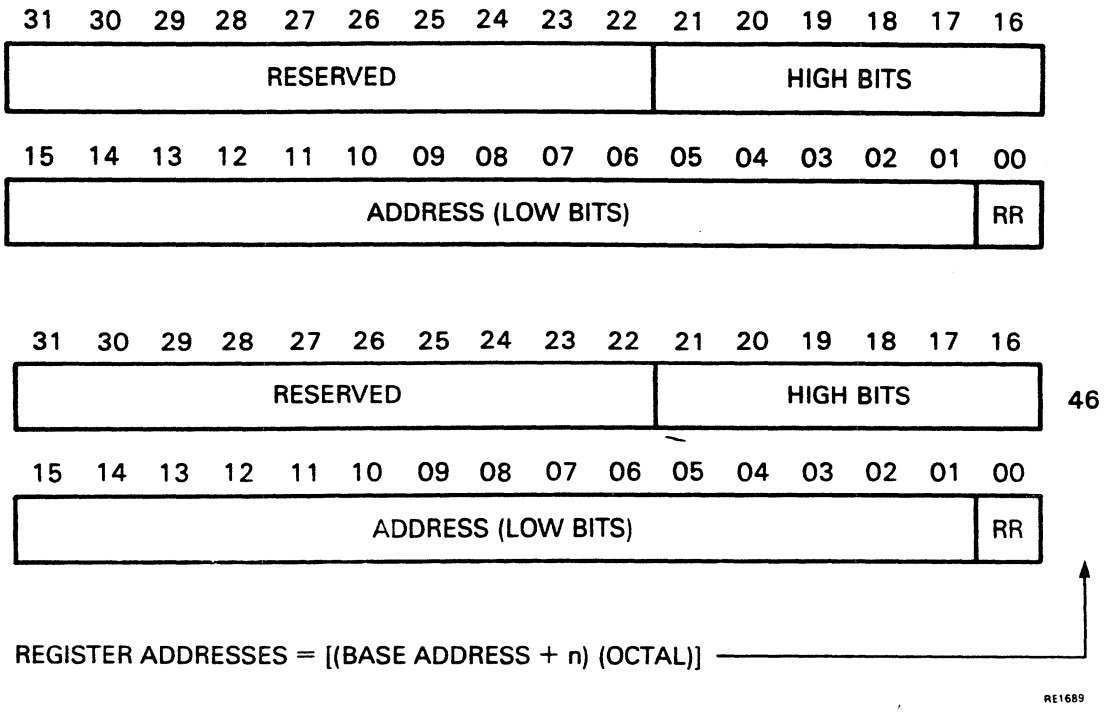


Figure 3–5 BDL Start Address Registers

4. Read the lower byte of word 0 in the DESQA 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 organise transmit and receive buffers.

When initialised, the DESQA 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 DESQA 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 DESQA. 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 DESQA 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 DESQA I/O page.

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

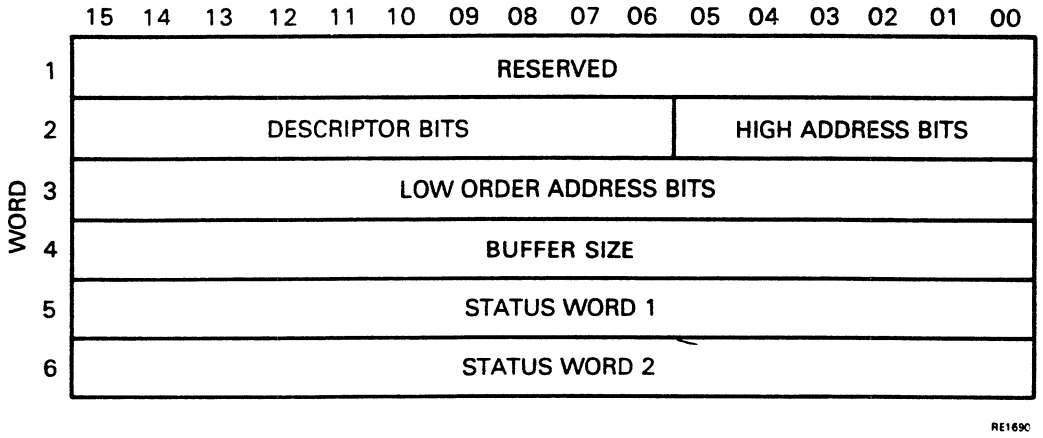


Figure 3–6 Buffer Descriptor Format

Each buffer descriptor in a BDL contains:

- A **reserved word:** reserved for DESQA 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

- 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 DESQA 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 DESQA 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 Reserved Word

Table 3–4 Reserved 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 DESQA 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 DESQA 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

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.

Bit	Definition
	When set: This bit indicates that this buffer contains the last segment of the packet. (The DESQA will attempt to transmit the entire frame after this segment has been DMAd from the host).
12	**S—Setup** (Transmit Buffer Descriptor Only) **When set:** This bit indicates that the buffer contains a list of DESQA 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 misaligned is not reflected in the word count, instead the H and L descriptor bits are used to denote this.

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 DESQA 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 DESQA and host. These bits are initialised by the host, and are updated by the DESQA to indicate that it has completed processing this descriptor and associated buffer. These bits should be initialised by the host as indicated below.

Table 3–5 Status Words

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 initialised 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.

11 **Reserved**

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

Table 3–5 (Cont.) Status Words

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

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

Table 3–5 (Cont.) Status Words

Bit	Definition

Receive Status Word 1

15 **Lastnot** See the table below.

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

15	14	Summary Status
1	0	Value initialised 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**

Table 3–5 (Cont.) Status Words

Bit	Definition
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.
01	**CRCERR** **When set:** Indicates that a CRC error has been detected in the current packet. The DESQA 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 DESQA 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 DESQA, using data DMA. The host software controls transmissions by initialising 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 DESQA 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 DESQA 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 DESQA 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 DESQA marks the end of the current BDL. The DESQA ceases to access the BDL and its associated buffers.
4. If the chain bit is set, the DESQA 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 DESQA 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 DESQA also generates a transmit interrupt request to indicate that a message has been transmitted.)

To achieve acceptable transmission rates, the DESQA 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 DESQA has returned the transmission status to the status word bits.

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. Initialise 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 DESQA 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 DESQA, 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

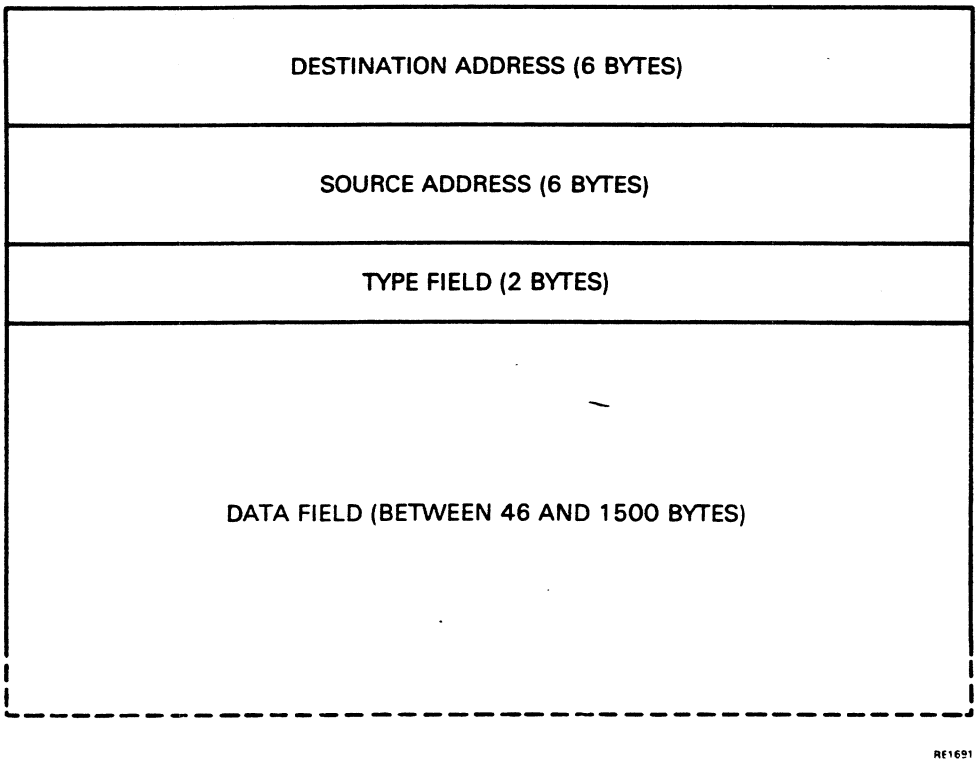


Figure 3–7 Ethernet Packet Format

- 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 DESQA 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 DESQA module.

To complete the receive process in response to activity on the Ethernet, the DESQA 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 DESQA ceases to access the BDL and its associated buffers.
   - If the Chain bit is set, the DESQA 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 DESQA 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 DESQA terminates reception, and updates the status words in the last buffer descriptor used. (If CSR06 is set, the DESQA 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. Initialise the start address and descriptor bits for each buffer descriptor in the Receive BDL.
3. Initialise Status Word 2 of all the descriptors in the Receive BDL with unequal high and low bytes. (The DESQA 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 the 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 DESQA, 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 DESQA 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 DESQA 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 DESQA citizenship test.

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

All requests for this code from the DESQA 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 DESQA, 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 DESQA boot/diagnostic code begins executing, it requests the on-board self-test, and waits for it to complete before continuing. The DESQA aborts the self-test when the boot/diagnostic sequence commences. This is for compatibility with the DEQNA/PDP-11 system boot Roms which do not wait for the DESQA ROM self-test to complete after power-up. 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 DESQA control registers (CSR and VAR), by which the host software can send commands, status, and control functions to the DESQA module.

The setup packet can be used to initialise the following functions within the DESQA module.

- Multi-cast address or promiscuous filtering for address recognition
- Timeout value for the sanity timer
- Up to 14 six-byte Ethernet addresses that the DESQA module is to recognise
- 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 DESQA module.

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

The setup packet is looped around internally, and placed in a receive buffer for verification and synchronisation. 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;their 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 DESQA.

1. **Target address information** contains the Ethernet physical and multi-cast addresses for which the DESQA is to receive messages.
2. **Control parameters** specify special reception modes (such as promiscuous or all multi-cast) 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 3-6 explains all the possible combinations of information groups.

Table 3–6 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 DESQA) whenever the setup information is modified. Since the DESQA can only have two types of setup packet information per setup packet, the DESQA will accumulate all setup packet information, unless respecified in a subsequent setup packet. Although setup packets may be repeatedly issued to the DESQA to modify parameters or to read internal values (that is, counters, system id parameters), only one setup packet should be outstanding to the DESQA at a time.

3.6.2.3 Setup Packet Buffer Descriptor

The DESQA recognises 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 multi-cast) 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 DESQA should set All Multi-cast Enabled of the setup packet control parameters.

Table 3–7 Setup Packet Buffer Descriptor: Address Mode Bits

Bit	Meaning
Word 4: Buffer Word Count	
C<00>	All Multi-cast address filter Enables DESQA recognition of any multi-cast 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 DESQA should respond.

Figure 3-8 shows the setup packet format in bytes (octal). The columns are used to show how the DESQA can be programmed to recognise 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 Multi-cast 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 DESQA module. Any other specified addresses are multi-cast 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 DESQA. In DEQNA-lock mode the specifications of multiple physical Ethernet addresses will cause the DESQA to filter all such physical Ethernet addresses for packet reception.

NOTE

Enabling more than one physical address is not recommended under normal circumstances as 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 DESQA 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 3-8 lists the effects of power-up reset, software reset, and self-test on the parameters of the setup packet.

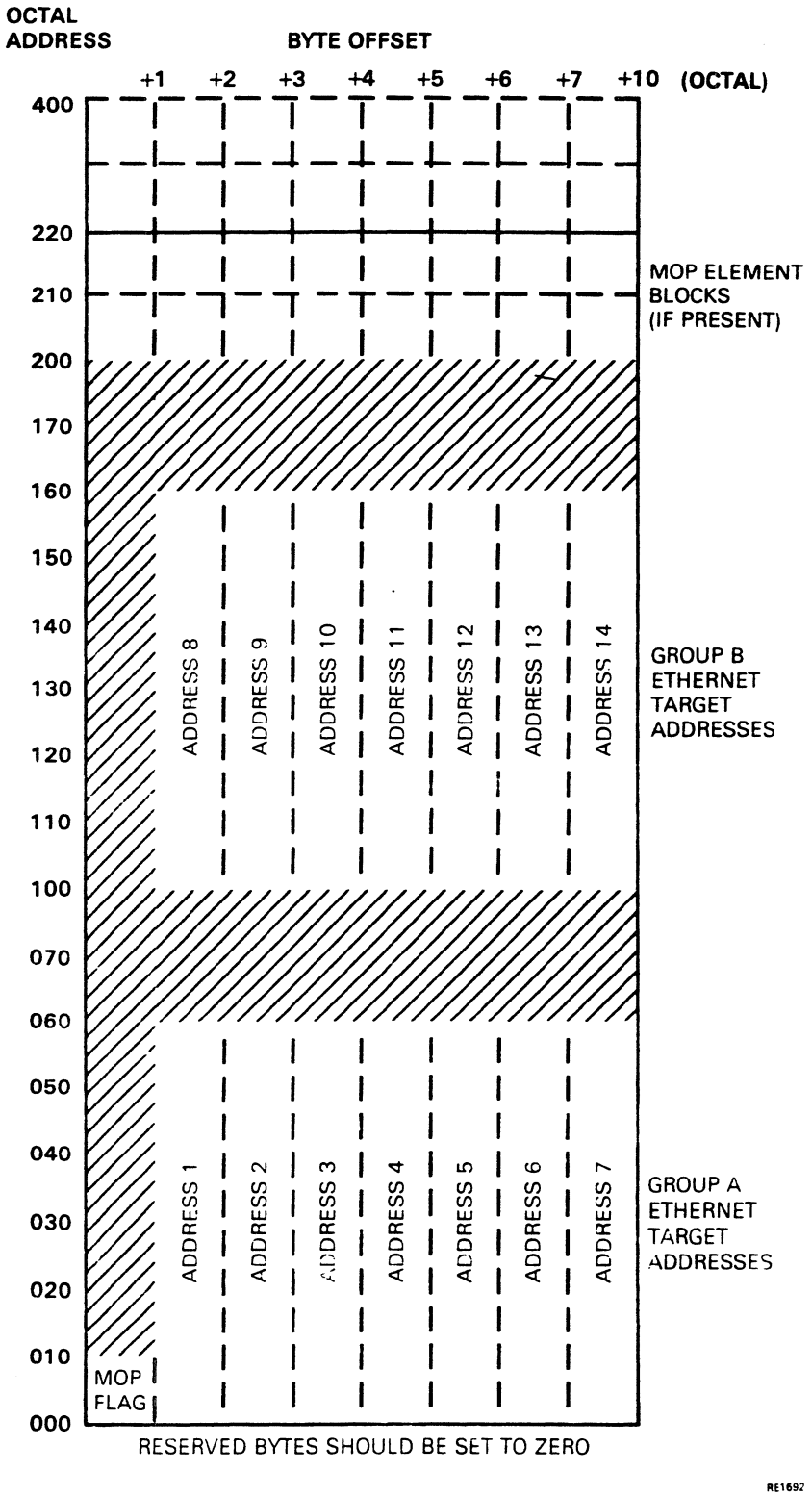


Figure 3–8 Setup Packet Format (Bytes)

Table 3–8 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
Multi-cast addresses	Multi-cast disabled	N/A
Mode bit: All Multi-cast	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 DESQA 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 DESQA hardware; instead it causes the DESQA to enter the DESQA reset state. The host software may verify that the DESQA 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 DESQA remains in the reset state until the host software clears CSR01 (Software Reset).

In reset state, the DESQA supports:

- Clear CSR01 (Software Reset) to exit the DESQA 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 DESQA 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 DESQA registers.

After CSR01 has been cleared, it takes up to 10 milliseconds for the DESQA to complete the initialisation sequence.

Tables 3-2 and 3-3 summarise 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 3-8 summarises the effects on setup packet data.

NOTE
The DESQA can remain active for up to 100μs after the Software Reset bit is cleared.

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 DESQA 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 DESQA 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 DESQA is initialised 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 DESQA 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 set to open, the sanity timer is cleared and enabled automatically on power-up if option 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.

3.7 MAINTENANCE OPERATIONS PROTOCOL (MOP): MODULE SUPPORT

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

In Normal mode the DESQA 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 DESQA 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 DESQA board.

The DESQA 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 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 3–9 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 DESQA 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 behaviour 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 DESQA (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 DESQA 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 DESQA 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 DESQA 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 DESQA has correctly loaded the physical address information from the setup packet. (The default Station Address (SA) in the DESQA ROM is usually read directly from the I/O port.)

Table 3–10 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 DESQA for a read function.
MEBB+2	PA<31:16>	The middle-order 16 address bits of the physical address. Written by the DESQA for a read function.
MEBB+4	PA<47:32>	The high-order 16 address bits of the physical address. Written by the DESQA for a read function.

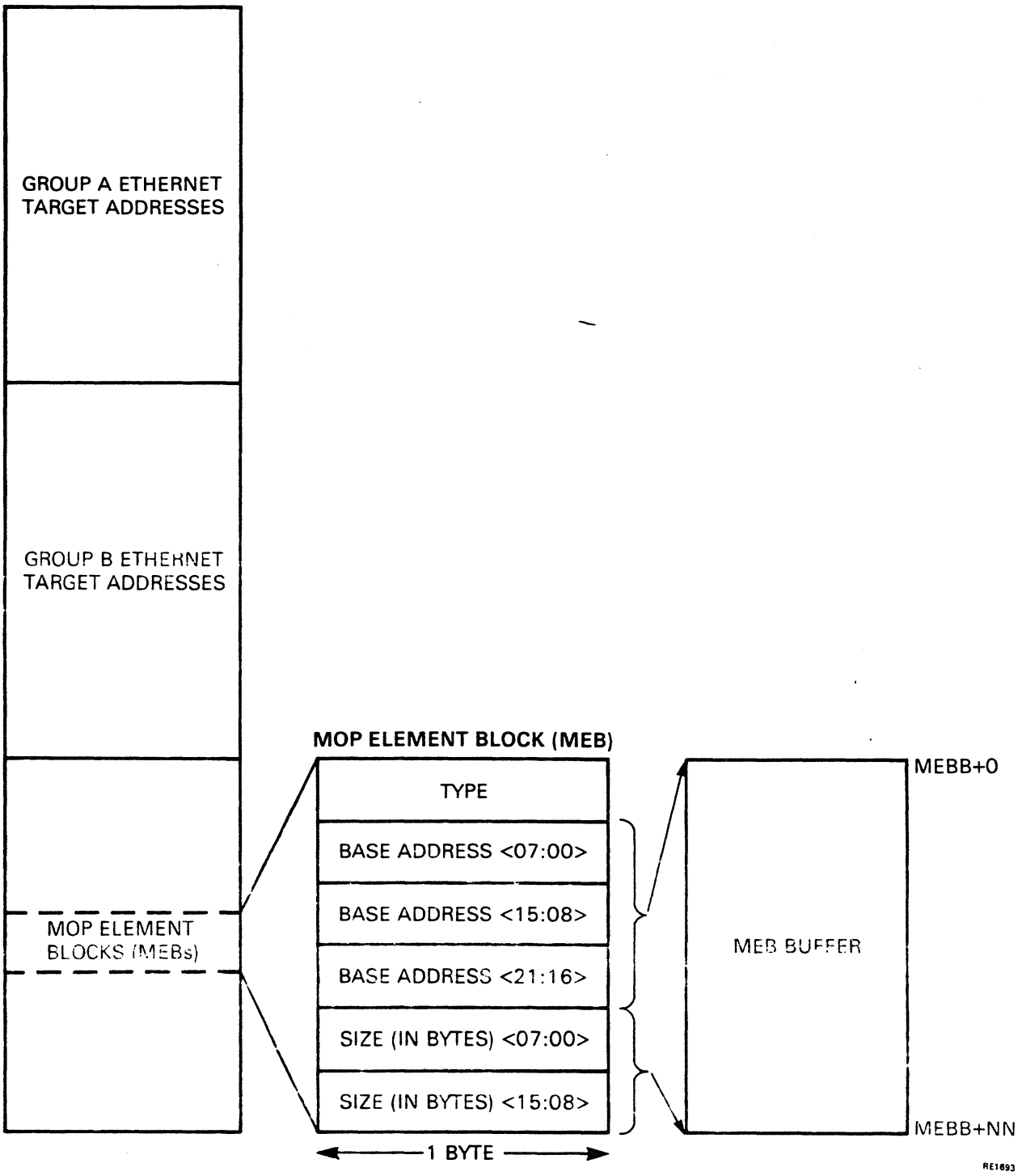


Figure 3–9 MOP Element Block Buffers in the Setup Packet

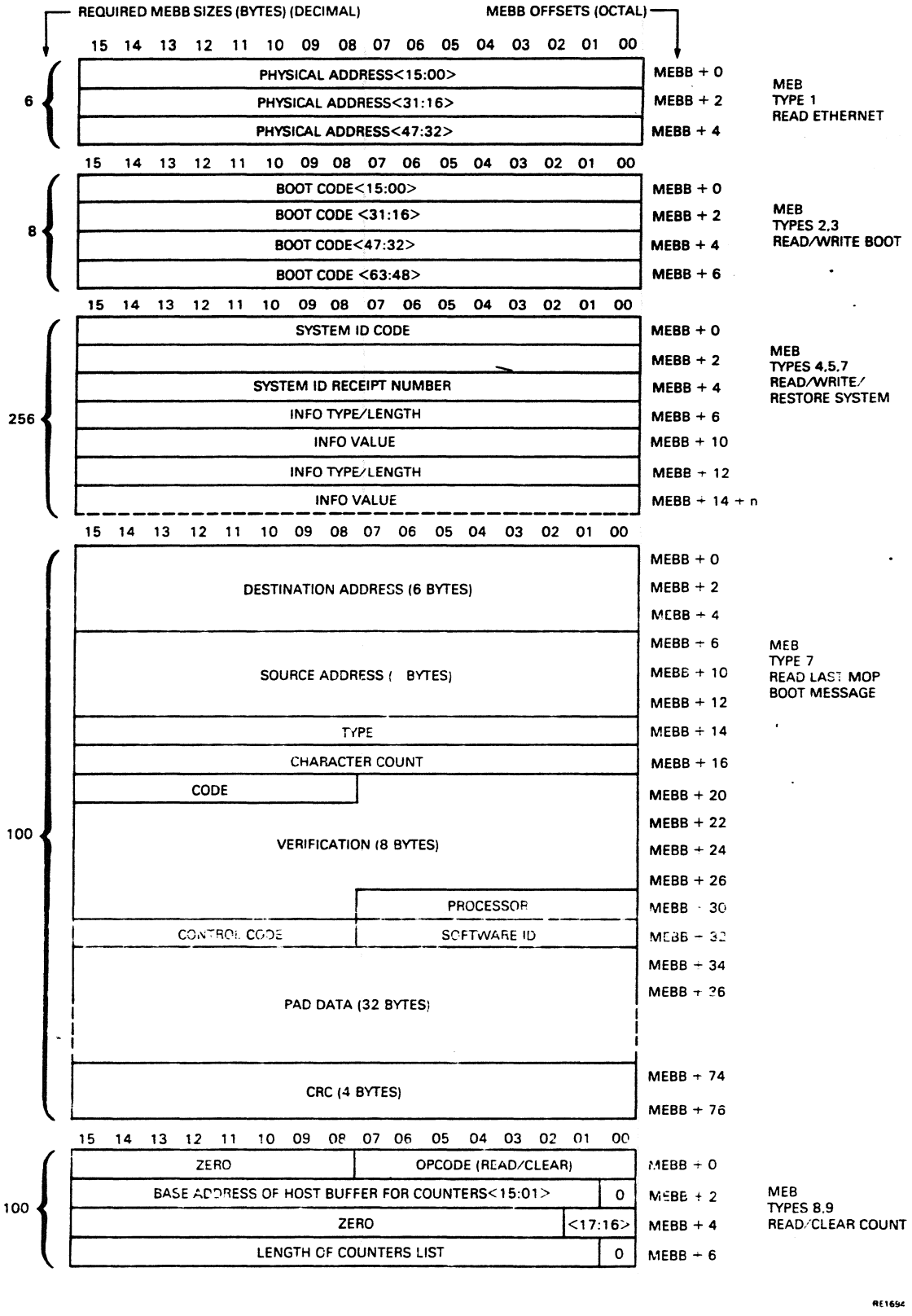


Figure 3–10 MOP Element Block Types 1 to 9

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 DESQA. This default is then broadcast from the DESQA 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-closed) with remote boot enabled (option switch S4-open).

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

The length of the password must be eight bytes.

Table 3–11 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 3-12 lists the Info types and Table 3-13 lists the Info value descriptions. For further details, refer to the *Maintenance Operations Protocol (MOP) Functional Specification.*

Table 3–12 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 3–12 (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 3–13 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 DESQA 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).

Table 3–13 (Cont.) Information Value Descriptions

Type	Bytes	Description
006	2	Console Response Size 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 DESQA (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 DESQA). For DESQA 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.

Table 3–13 (Cont.) Information Value Descriptions

Type	Bytes	Description
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 DESQA maintains on-board in its shared RAM. A MOP element type 8 does not affect the state of the DESQA 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 DESQA counters are in the contiguous format described in Table 3-14.

Table 3–14 MOP Elements Type 8, 9 MEBB Format

Count	Specification	
1	Seconds Since Last Zeroed	16 bits in datalink counter
	The number of seconds since the counters were last zeroed.	
2	(Data) Bytes Received	32 bits in datalink counter
	The total number of data bytes received error free, excluding the data link protocol overhead.	
3	(Data) Bytes (Sent) Transmitted	32 bits in datalink counter
	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 in datalink counter
	The total number of datagrams received error free.	
5	Packets (Frames Sent) Transmitted	32 bits in datalink counter

Table 3–14 (Cont.) MOP Elements Type 8, 9 MEBB Format

Count	Specification
	The total number of datagrams successfully transmitted, including transmissions in which the collision test signal failed to set.
6	Multi-cast Bytes Received — 32 bits in datalink counter
	The total number of multi-cast data bytes received error free, excluding the data link protocol overhead.
7	Multi-cast Packets (Frames) Received — 32 bits in datalink counter
	The total number of multi-cast datagrams received error free.
8	Packets Transmitted: (Initially) Deferred — 32 bits in datalink counter
	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 in datalink counter
	The total number of datagrams successfully transmitted on two attempts, including transmissions in which the collision test signal failed to set.
10	Packets Transmitted (multiple collisions): 3+ Attempts — 32 bits in datalink counter
	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 — 16 bits in datalink counter
	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 DESQA).
	Bit <03> = 0 — Open Circuit (not supported on the DESQA).
	Bit <04> MLEN — Data Block Too Long. The DESQA aborted the transmission because the datagram exceeded the maximum packet size.

Table 3–14 (Cont.) MOP Elements Type 8, 9 MEBB Format

Count	Specification	
	Bit <05> LCOL	Remote Failure to defer: late collision on the last transmission attempt.
	Bits <15:06> = 0	Undefined
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 **Unrecognised frame destination error**. This indicates that a packet was received by the DESQA, 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 Overrun)	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.

CHAPTER 4
TECHNICAL DESCRIPTION

4.1 SCOPE

This chapter describes how the components of the DESQA module work together on the backport bus. The sections are as follows:

Section 4.2 **ARCHITECTURE**
Section 4.4 **Q-BUS INTERFACE**
Section 4.5 **ETHERNET INTERFACE**
Section 4.6 **BACKPORT BUS**
Section 4.7 **MODULE CONTROL**

4.2 ARCHITECTURE

4.2.1 Overview

The DESQA module is built around a backport bus which passes data between the QIC, shared RAM, and the LANCE. A 68000 microprocessor handles formatting, module control, and initialisation.

The QIC handles the interface with the Q-Bus protocols, and the LANCE handles the interface with the Ethernet protocols. Shared RAM is used as an intermediate buffer store.

The QNA2 arbitrates between the 68000, the lance and the QIC, for access to the backport bus. The QNA2 provides all the control signals for successful data transfer. The QNA2 contains the DESQA control and status register (CSR) and internal interrupt register.

Firmware ROM contains both the master control program for the DESQA module and 4Kbytes of PDP-11 boot/diagnostic code for the Q-Bus system. The firmware ROM can be addressed only by the 68000 microprocessor.

The Station Address (SA) ROM is preprogrammed with the 48-bit default physical address of the DESQA module in the Ethernet LAN.

Figure 4–1 shows the principal control signal paths through the DESQA module.

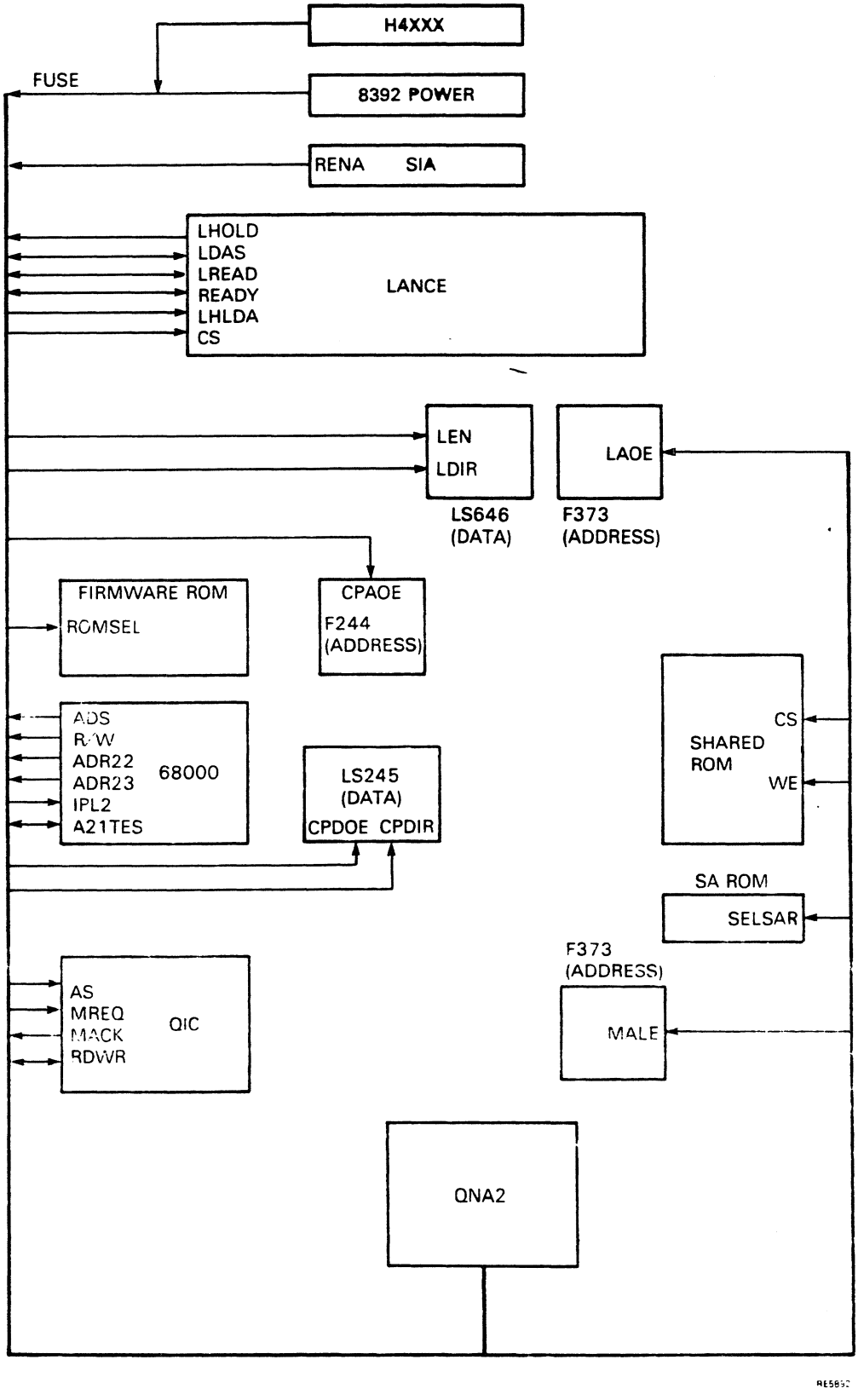


Figure 4–1 QNA2 Control Signals

Table 3–14 (Cont.) MOP Elements Type 8, 9 MEBB Format

Count	Specification	
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 DESQA, 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	Multi-cast Bytes Transmitted	32 bits
	The total number of multi-cast data bytes successfully transmitted, excluding data link protocol overhead, and not counting the DESQA 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 DESQA LANCE reported the babble condition on the channel.	

4.2.2 Selection Switches

The DESQA has five manual selection switches: S1 to S5. These are part of a switchpack on the DESQA board.

The mode switch defines two possible modes of operation for the DESQA. The preferred mode is the "Normal mode" which indicates that the DESQA is operating as a DESQA. All current DIGITAL software for the DEQNA may be used with confidence for the DESQA when the DESQA 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 DESQA 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).

Table 4–1 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

4.3 CHECKING LED TEST PATTERNS

The DESQA module has a power-up self-test that includes an external loopback test. In order for the external loopback test to pass, you must set up the DESQA as follows:

- **In Thinwire Ethernet mode:** Connect a tee connector to the BNC connector on the DESQA handle. Connect either two terminators or a valid Thinwire network to the tee connector.
- **In standard Ethernet mode:** Connect a D-type loopback connector, a H4080 loopback connector, or a valid Ethernet network to the 15-pin D subminiature connector on the DESQA handle.

The self-test is only available when the DESQA is in normal mode. The host operating system software can request the self-test through the DESQA Q-Bus register.

Table 4–2 Power-Up LED Patterns

LED			
1	**2**	**3**	**Meaning**
Normal Mode			
•	o	o	Running the internal logic self-test.
•	•	o	The self-test is running an external loopback test.
•	•	•	Ready to run citizenship tests and/or normal functions, or run the module self-test.
DEQNA Mode			
•	•	•	LEDs turn on and stay on.

• = on. o = off.

REMOTE BOOTS:
If a remote boot is initiated, make sure the LED sequences in Table 4–3 also appear.

Table 4–3 Remote Boot LED Patterns (Normal or DEQNA Mode)

LED			
1	**2**	**3**	**Meaning**
o	•	•	Running citizenship tests.
o	o	•	Internal loopback citizenship tests completed successfully.
o	o	o	External loopback citizenship tests completed successfully.

• = on. o = off.

Confirm that all three LEDs are off, indicating that all citizenship tests completed successfully.

NOTE
If any LEDs remain flashing, an error has occurred.

Whether the DESQA is in normal mode or DEQNA mode, make sure all three LEDs turn on within 10 seconds, indicating a successful power-up.

4.3.1 Clocks

The clock signals in the DESQA module run from a 40MHz oscillator located at the Ethernet end of the board.

The main clock signal is divided by two to produce two antiphase 20MHz clock signals. One of these is fed into the X1 input of the SIA, where it is used to generate the 10MHz Transmit Clock signal to the LANCE. The other clock is fed to the QIC and the QNA2, so that these two chips can operate synchronously at 20MHz.

The QNA2, as bus controller, generates the 10MHz clock that drives the 68000 microprocessor.

Figure 4–2 shows the main clock signals, and the switchpack.

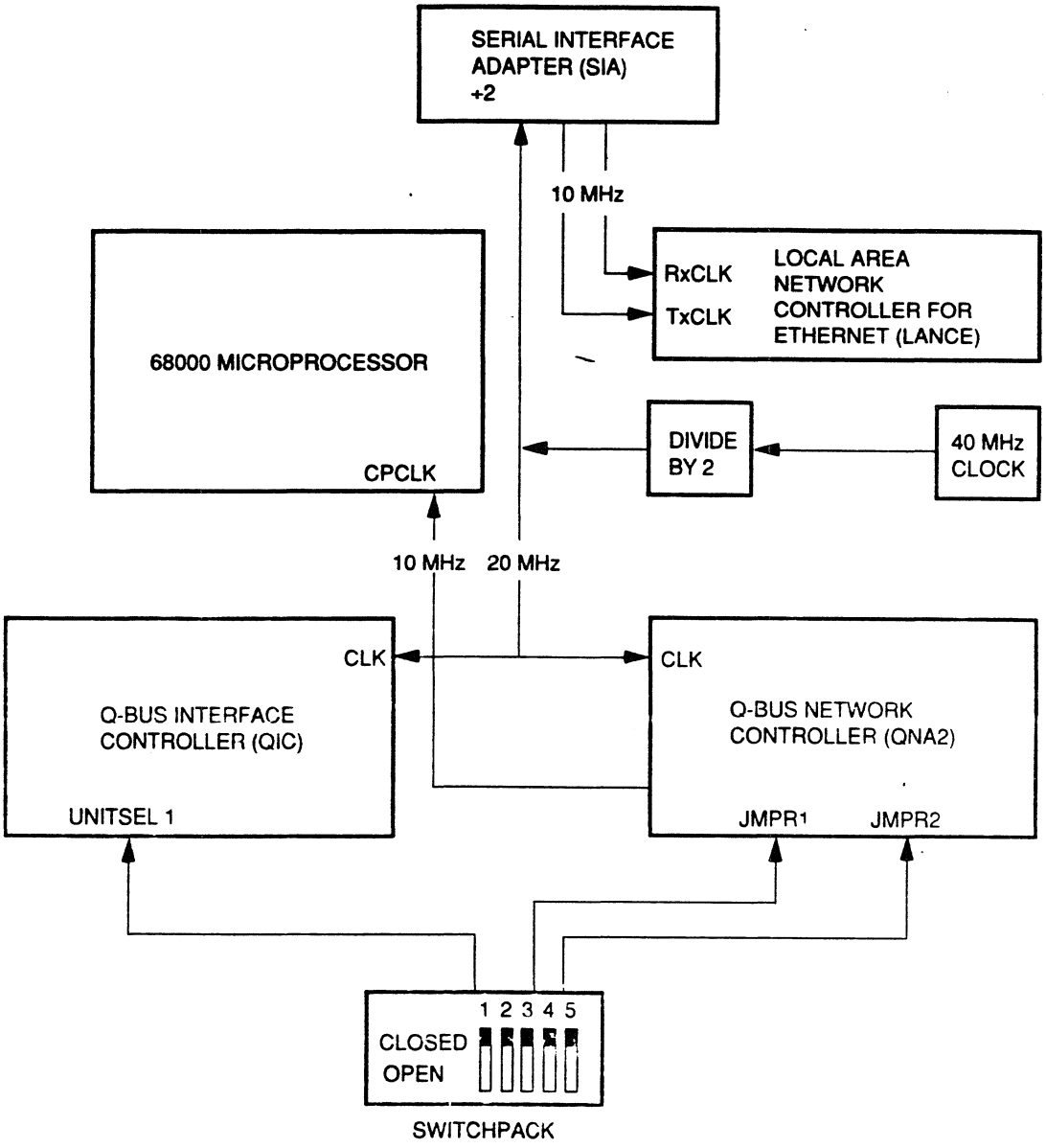


Figure 4–2 DESQA Clocks And the Main Selection Switches

4.3.2 POWER LOADING

Table 4–4 DESQA and transceiver power requirements

Function	Typical Voltage	Typical Current	Maximum Current	Backplane Pins
TTL level	+ 5V +/-0.25V	2.15A +/-10%	2.7A	BV1, AA2, BA2
TTL ground				AJ1, AM1, AT1 BJ1, BM1, AC2, BC2
Transceiver power	+12V +/-0.60V	0.21A	1.5A (Note)	BD2
Transceiver ground				BT1, CT1, CC2, DT1, DC2.

NOTE

At powerup, the surge current of the transceiver can cause certain power supplies to current limit or even to fail. These failures can also be caused during powered-up connect. There is no surge protection circuitry on the DESQA module. However the +12V supply to the 15-pin D-Type connector is limited by a 5A pico fuse on the module, and a handle mounted 1.5A/250V slo-blo fuse. A fuse of the correct type must always be fitted to the bulkhead module.

4.3.2.1 Fuses

- A 1.5A/250 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 by another fuse of the same type, a Littlefuse™ type 31301.5, a BEL FUSE™ type 3SB1.5, or an equivalent. The 1.25 inch (3.8cm) fuse holder (order number 12-22255-03) is located in the DESQA bulkhead.
- A 5.0 A/125 V axial lead pico fuse (order number 12-05747-00) protects the DESQA module and internal wiring. The pico fuse is fitted on the DESQA board. It looks like a resistor and is soldered in the same way. It should be replaced only by trained personnel.

™ slo-blo is a Trademark of S.B. Fuses

™ Littlefuse is a Trademark of Littlefuse Inc., Illinois, USA

™ BEL FUSE is a Trademark of Belfuse Inc., New Jersey, USA

4.4 Q-BUS INTERFACE

4.4.1 Overview

The QIC (Q-Bus Interface Controller) handles the Q-Bus interface protocols. To provide a complete Q-Bus interface, four octal bus transceivers and two quad transceivers are connected between the Q-Bus and the QIC. The QIC shares a 20MHz clock input with the QNA2.

The QIC provides complete Q-Bus slave control logic for host access to the DESQA I/O page registers.

On the Q-Bus side, the QIC supports control DMA and data DMA, using block-mode to achieve the highest possible speeds. The QIC generates two control signals to change the direction of the DC021 transceivers:

1. DC012IN controls the direction of transfer of the three DC021 transceivers connected to the Q-Bus data/address lines, BDAL<21:0>.
2. TSACK (Transmit DMA Selection Acknowledge) controls the direction of transfer for the fourth DC021 transceiver, which carries the bus control signals that enable the QIC to act as Q-Bus master for DMA.

On the backport side, the QIC supports DMA transfers to and from shared RAM along 16 data/address lines. The QIC is controlled through a series of registers that are accessed by the 68000 microprocessor.

Figure 4–3 shows the Q-Bus interface. Table 4–5 summarises the signals and pinouts.

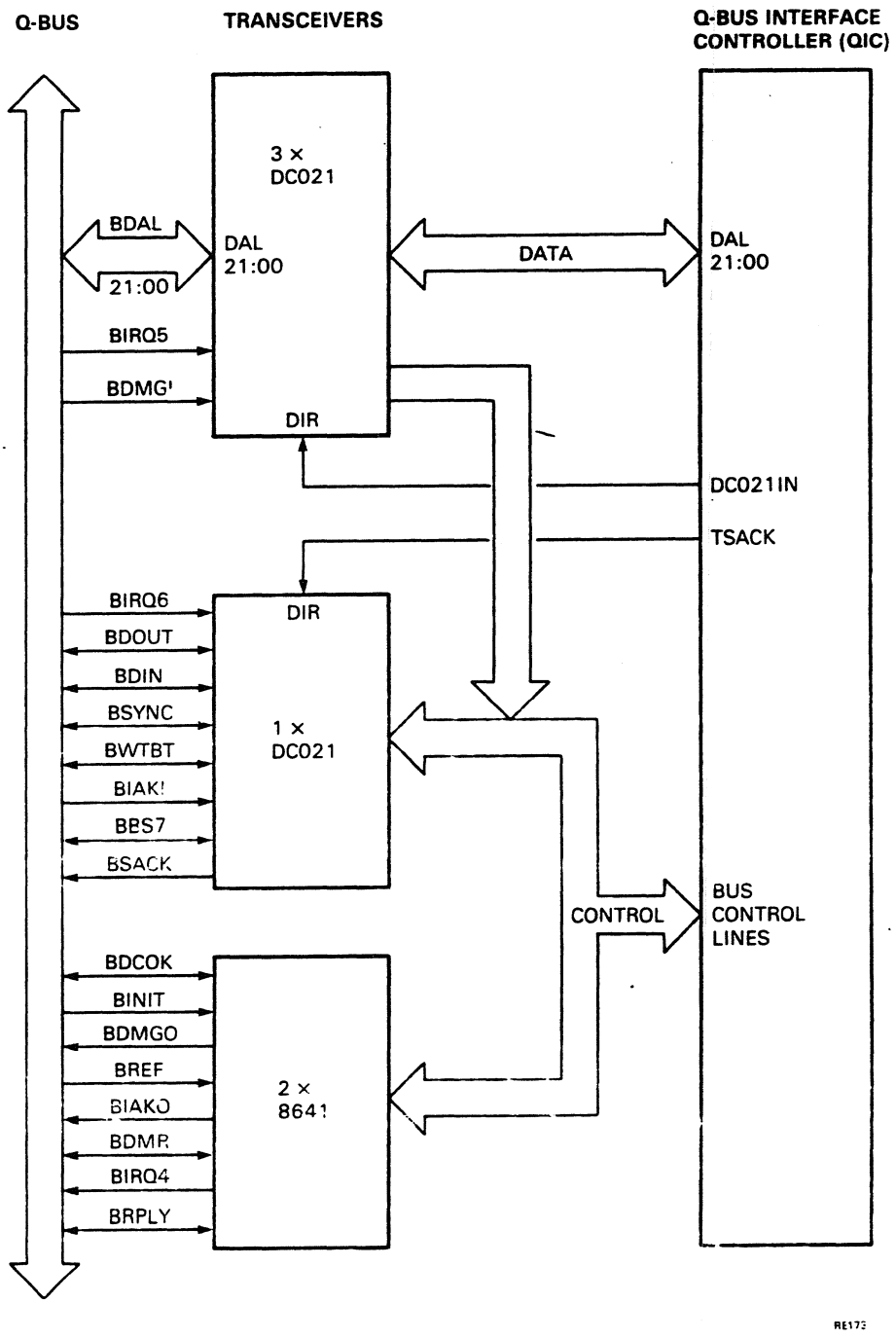


Figure 4–3 Q-Bus Interface (hardware)

Table 4–5 Q-Bus Connector Pins and Signals

Pin	Signal	Description	Pin	Signal	Description
AA1	BIRQ5	Interrupt lines	AA2	+5V	
AB1	BIRQ6		AB2		
AC1	BDAL16 L	Data/Control	AC2	GND	
AD1	BDAL17 L	lines	AD2		
AE1			AE2	BDOUT L	Bus Control lines
AF1			AF2	BRPLY L	
AH1			AH2	BDIN L	
AJ1	GND		AJ2	BSYNC L	
AK1			AK2	BWTBT L	
AL1			AL2	BIRQ4 L	Interrupt lines
AM1	GND		AM2	BIAKI L	
AN1	BDMR L	DMA Control	AN2	BIAKO L	
AP1	BHALT L		AP2	BBS7 L	Bus Control
AR1	BREF L	Q-Bus control	AR2	BDMGI L	DMA Control
AS1			AS2	BDMGO L	
AT1	GND	FUSE H sink	AT2	BINIT L	Bus Control
AU1			AU2	BDAL0 L	Data/Address lines
AV1			AV2	BDAL1 L	
BA1	BDCOK H	DC Power OK	BA2	+5V	
BB1	BPOK	Power OK	BB2		
BC1	BDAL18 L	Data/Address lines	BC2	GND	
BD1	BDAL19 L		BD2	+12V (XCVR)	Transceiver
BE1	BDAL20 L		BE2	BDAL2 L	Data/Address lines
BF1	BDAL21 L		BF2	BDAL3 L	
BH1			BH2	BDAL4 L	

Table 4–5 (Cont.) Q-Bus Connector Pins and Signals

Pin	Signal	Description	Pin	Signal	Description
BJ1	GND		BJ2	BDAL5 L	Data/Address lines
BK1			BK2	BDAL6 L	
BL1			BL2	BDAL7 L	
BM1	GND		BM2	BDAL8 L	
BN1	BSACK L	DMA Control	BN2	BDAL9 L	
BP1	BIRQ7 L	Interrupt	BP2	BDAL10 L	
BR1			BR2	BDAL11 L	
BS1			BS2	BDAL12 L	
BT1	GND	(XCVR) Transceiver	BT2	BDAL13 L	
BU1			BU2	BDAL14 L	
BV1	+5V		BV2	BDAL15 L	
CA1			CA2		
CB1			CB2		
CC1			CC2	GND(XCVR)	
CD1			CD2	+12V(XCVR)	
CE1			CE2		
CF1			CF2		
CH1			CH2		
CJ1	GND(XCVR)		CJ2		
CK1			CK2		
CL1			CL2		
CM1	GND(XCVR)		CM2		
CN1			CN2		
CP1			CP2		
CR1			CR2		
CS1			CS2		
CT1	GND(XCVR)		CT2		

Table 4–5 (Cont.) Q-Bus Connector Pins and Signals

Pin	Signal	Description	Pin	Signal	Description
CU1			CU2		
CV1			CV2		

Notes:

1. **(XCVR)** indicates isolated power connections through the DESQA to the Ethernet transceiver cable connector.
2. **BC2** is tied to the ground plane of the module and serves as a pull-down sink through approximately 14K Ohms for the Transceiver Connector signal FUSE H.

4.4.2 Initialisation

The QIC is reset from the Q-Bus when BDCOK is regated and BINITL is asserted. The assertion of BINITL also causes a reset sequence to the remainder of the DESQA module. On the falling edge of RINITH a 200ns pulse resets the QNA2 and a 150ms pulse resets the 68000 and the LANCE.

4.4.3 DMA Functions

The QIC can request, accept, and relinquish ownership of the Q-Bus, for both data DMA and control DMA accesses to the host database.

There are separate addressing and buffering mechanisms for data DMA and control DMA. This allows control DMA requests to take priority over data transfers in progress; when the control DMA is complete, the data transfer is restarted from where it left off.

4.4.3.1 Data DMA

During data DMA transfers, the 21-bit Q-Bus address counter is connected to DAL<21:01>, with DAL<00> held at zero. During a transfer, data transferred between the Q-Bus and the backport bus is double-buffered within the QIC. The 16-bit backport address counter is connected to BPDAL<00,15:01> to support transfers of up to 128 Kbytes.

Double buffering inside the QIC allows for the signal setup, hold and access times of Q-Bus and backport bus, and for high speed, pipelined transfers through the chip.

If control DMA is attempted during data DMA, the QIC:

1. Relinquishes control of the Q-Bus
2. Requests control for a control DMA transfer (non-block-mode).
3. Completes the control DMA transfer, and relinquishes control of the bus.
4. Requests control of the bus in order to resume data DMA transfers.

The QIC recognises 16-word boundaries because there is no RREF (Refresh) from the bus slave. To change to non-block mode, the QIC negates and re-asserts TSYNC (Synchronise).

4.4.3.2 Control DMA

During Control DMA transfers, the 21-bit Q-BUS address counter is connected to DAL<21:01>, with DAL<00> held at zero.Control information is passed along a latched path through the QIC using single byte or single word transfers.

This is the same path that is used when the host accesses the QIC as bus slave. If the host requests an I/O page access at the same time as the 68000 requests a control DMA transfer, the host request takes priority.

4.5 ETHERNET INTERFACE

4.5.1 Overview

The LANCE (Local Area Network Controller for Ethernet) and the SIA (Serial Interface Adapter) operate together to provide a complete interface from the backport bus to the Ethernet. To provide electrical isolation between the DESQA and the Ethernet transceiver, a transformer passes the signals between the SIA and the transceiver differential pairs. Between the SIA and the Ethernet is a selector for choosing Thinwire or Thickwire transmission. In the Thickwire mode the signals are routed to the D-Type connector and for the Thinwire mode, they are routed to the DP8392 transceiver chip and from there to the BNC Thinwire connector.

On the backport side of the LANCE, two bus transceivers isolate the LANCE from the backport data bus, and two latches isolate the LANCE from the backport address bus (arbitrated by the QNA2). The SIA takes a 20MHz clock input from the DESQA clock, and uses it to generate a 10MHz clock, which it also passes to the LANCE for synchronisation.

The SIA provides Manchester encoding of data for transmission, and decodes received data. The LANCE supplies and checks the 32-bit CRC that is appended to all Ethernet packets.

During transmission the LANCE polls the transmit buffer descriptor list for buffer descriptors that it owns. When it finds an entry that it owns it DMA's the data from shared RAM into internal FIFO. It prefixes the 64 bit preamble to the packet and sends the data to the SIA as a serial bit stream. When the packet has been transmitted, or if an error occurs, the LANCE will update the relevant buffer descriptor.

A DC-to-DC converter provides the -9V supply required for the operation of the DP8392 Thinwire transceiver chip. This -9V supply is DC isolated from the +12V input power supply.

The Thinwire Interface Circuit which includes the DP8392 transceiver chip is used to isolate DC voltages from the Ethernet to the DESQA. This function is carried out by the H4xxx when Thickwire Ethernet mode is selected.

During reception, the SIA detects and synchronises to the incoming bit stream from the Ethernet. The LANCE checks the destination address contained in the package against its list of addresses. If it finds a match, the LANCE will DMA the data from its internal FIFO in to buffer memory. When the reception is completed, or if an error occurs, the LANCE will update the relevant entry in the receiver buffer descriptor list.

The SIA communicates only with the LANCE and the Thinwire Transceiver or Thickwire interface. The LANCE acts as backport bus master for all data transfers, and as bus slave for the 68000 microprocessor to program its internal registers. It communicates with the QNA2 in order to access the backport bus and transfer data to and from shared RAM.

Figure 4–4 shows the Ethernet interface.

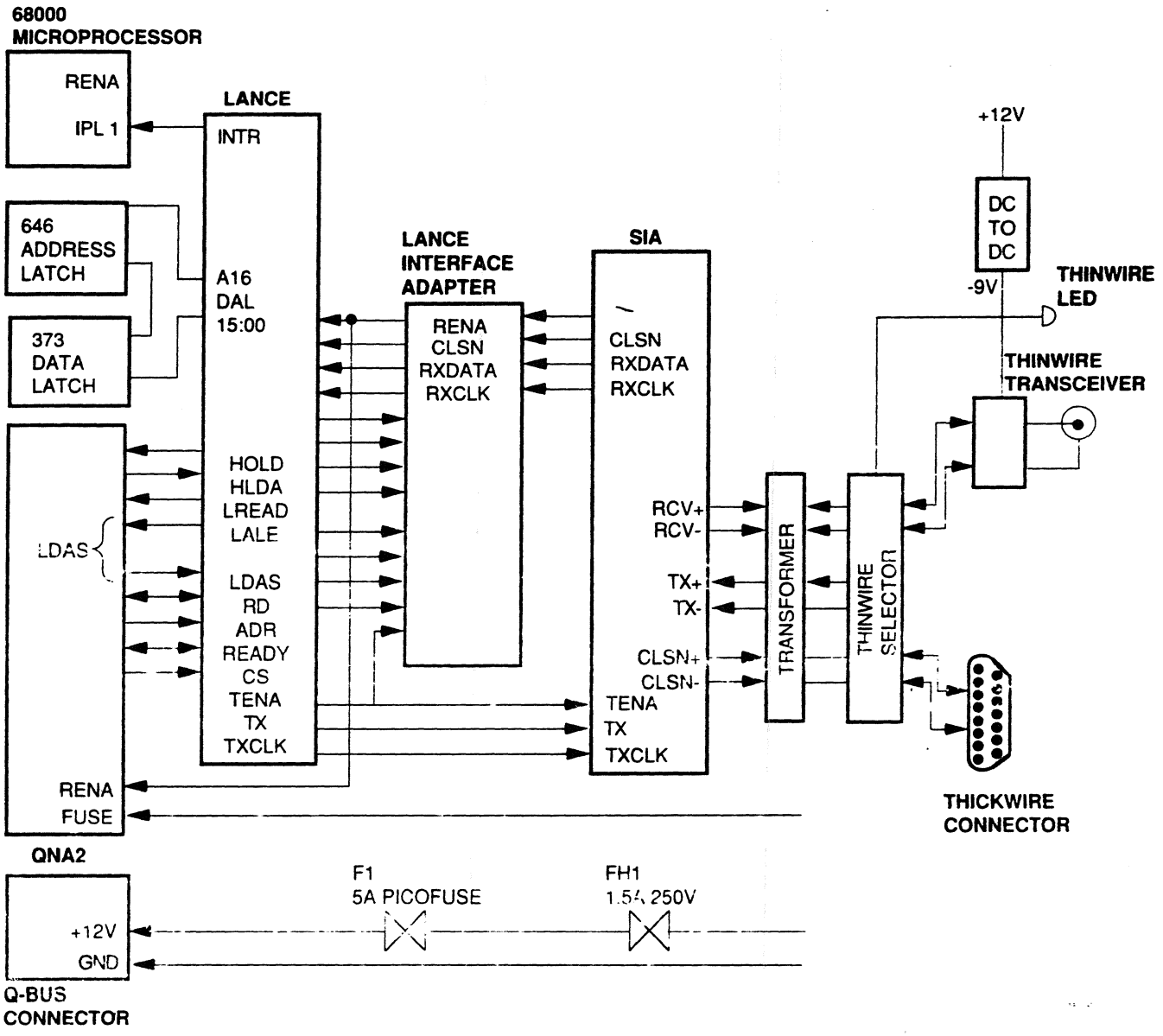


Figure 4–4 Ethernet Interface (hardware)

4.5.2 Initialisation

The LANCE initialises itself on request from the 68000. The 68000 writes an Initialisation block from 68000 ROM to shared RAM, and programs the LANCE CSR to locate the block. The LANCE reads the block to determine mode of operation, address recognition parameters, and buffer descriptor ring pointers.

4.5.2.1 Destination Addresses

The LANCE can be programmed to match three different types of address from an incoming packet, as follows:

1. **48-bit destination address**

   This is programmed as the unique address of the DESQA. Usually, the initialisation block is programmed to include the physical address from the SA (Station Address) PROM, but DECnet software reprograms the address in order to reflect the current DECnet node number. (SA ROM still stores a permanent record of the default physical address.)

2. **Multi-cast address filter**

   This is programmed using a logical address mask in the initialisation block.

3. **Promiscuous mode**

   In this mode the LANCE is programmed to accept all incoming packets. This mode is set using the mode register in the initialisation block.

4.5.2.2 LANCE Memory Structures

There are three memory structures accessed by the LANCE:

1. **Initialisation Block** (12 words) contains the operating parameters for the device: mode of operation; physical address; logical address mask; location of receive and transmit descriptor rings; number of entries in receive and transmit descriptor rings.

2. **Receive Message Descriptor Rings (rMDR) and Transmit Buffer Descriptor Rings (tMDR)**. Two ring structures, one each for incoming and outgoing packets. Each entry is four words long, comprising: data buffer address, length, and status.

3. **Data buffers** for packet buffering.

These memory structures are set up in shared RAM by the 68000 microprocessor. The 68000 then writes the Initialisation Block Start Address (IADR) to CSR1 and CSR2 in the LANCE, and the LANCE loads itself with the information contained in the Initialisation Block.

Refer to Appendix A for details.

4.5.3 Transmission

The LANCE polls the transmit Message Descriptor Ring (MDR) automatically every 1.6 milliseconds whenever it is not searching out receive buffers for incoming packets.

The polling consists of searching the status words of the transmit Message Descriptor Ring (MDR) for a buffer descriptor marked as OWNed by the LANCE. Then the LANCE executes two more read operations to collect the transmit buffer address and its buffer byte count.

Where the buffers are chained, the LANCE looks ahead once during the current transfer in order to find the next buffer. If it does not OWN the next MDR entry, it sets the UFLO and BUFF error bits in CSR0. When it empties a buffer, the LANCE clears the OWN status bit for that buffer.

The interpacket gap time on the Ethernet is a minimum of 9.6 microseconds, starting on the falling edge of the SIA signal RENA.

In transmit mode, the host supplies the destination address, source address, and length field. The LANCE appends preamble, sync, and CRC to the frame.

If an error occurs, the current buffer transmission is abandoned, causing an interrupt to the 68000 microprocessor.

4.5.4 Reception

When an incoming packet arrives,the LANCE checks to see if it own a receive buffer. If not it will poll the receive ring once for a buffer. If it does not own the buffer it will set the MIBB bit in CSR0,and will not poll the receive ring until the packet ends.

The polling consists of searching the status words of the receive Message Descriptor Ring (MDR) for a buffer descriptor marked as OWNed by the LANCE. Then the LANCE executes two more read operations to collect the receive buffer address and its buffer byte count.

If the LANCE OWNs a receive packet when an incoming packet arrives, it looks ahead once between transfers to find the next OWN buffer. This involves three separate one-word read operations. If the buffers for incoming packet need to be chained, and the LANCE has not found the next buffer, it sets the BUFF and/or the OFLO bits in CSR0.

The interpacket gap time on the Ethernet is a minimum of 9.6 microseconds, starting on the falling edge of the SIA signal RENA. If a new packet arrives within 4.1 microseconds of this edge, it can only be received correctly if at least eight bits of the preamble are left.

in receive mode, the LANCE strips the preamble and sync bits, and transfers data and CRC to shared RAM. The LANCE discards runt packets of less than 64 bytes (which are usually due to a collision).

The LANCE can handle up to seven dribbling bits when a received packet terminates, so long as the CRC accounts for them. If there are both dribbling bits and a CRC error, then a Framing Error is flagged in the Receive Message Descriptor.

If an error occurs, the current buffer reception is abandoned, causing an interrupt to the 68000 microprocessor.

4.5.5 Collision Detection

A collision on either the Thickwire or the Thinwire Ethernets is reported by assertion of the collision detect differential pair. The SIA reports a collision to the LANCE by asserting CLSN. It leaves TENA asserted for between 32 and 40 additional bit times in order to transmit the collision jam signal pattern. The jam pattern is any pattern other than the CRC bytes. The LANCE completes the preamble or the current byte transmission before starting to transmit the jam pattern.

The LANCE makes up to 16 retransmission attempts before reporting a collision.

If CLSN is asserted during reception, the reception is terminated at once, either because of an address mismatch with an internal pointer, or as a runt packet. A late collision (occurring after 64 byte times, or 51.2 milliseconds) is not recognised in receive mode.

4.5.6 Buffering

The LANCE operates two types of DMA transfer to shared RAM: single word DMA, and burst mode DMA.

Single word DMA is used to access the Transmit and Receive Message Descriptor Rings MD, and to read the initialisation block. Burst mode is used to transfer Transmit or Receive messages in eight consecutive reads or writes.

Bus cycles are a minimum of 600 nanoseconds. Burst mode transfers are separated by a gap of at least 700 nanoseconds.

Separate MD describe transmit and receive operations. Up to 128 tasks may be queued upon each MDR.

Each message descriptor entry is four words long. Each descriptor in a ring is marked as OWNed by either the LANCE or the host (the 68000 microprocessor). The status of a descriptor can only be changed while the descriptor is OWNed, and ownership can only be relinquished, never taken.

The location of the descriptor rings is programmed in the initialisation block.

4.5.7 Thinwire Ethernet Transceiver

The Thinwire Ethernet Transceiver is made up of a DP8392 transceiver chip, an isolation transformer and a DC-DC converter. The transformer isolates the Thinwire Ethernet Transceiver from the rest of the DESQA circuitry. The -9 Vdc needed for the DP8392 chip, is converted by the DC-DC converter from +12 Vdc which is supplied by the host. A functional block diagram of this transceiver component set is shown in Figure 4-5

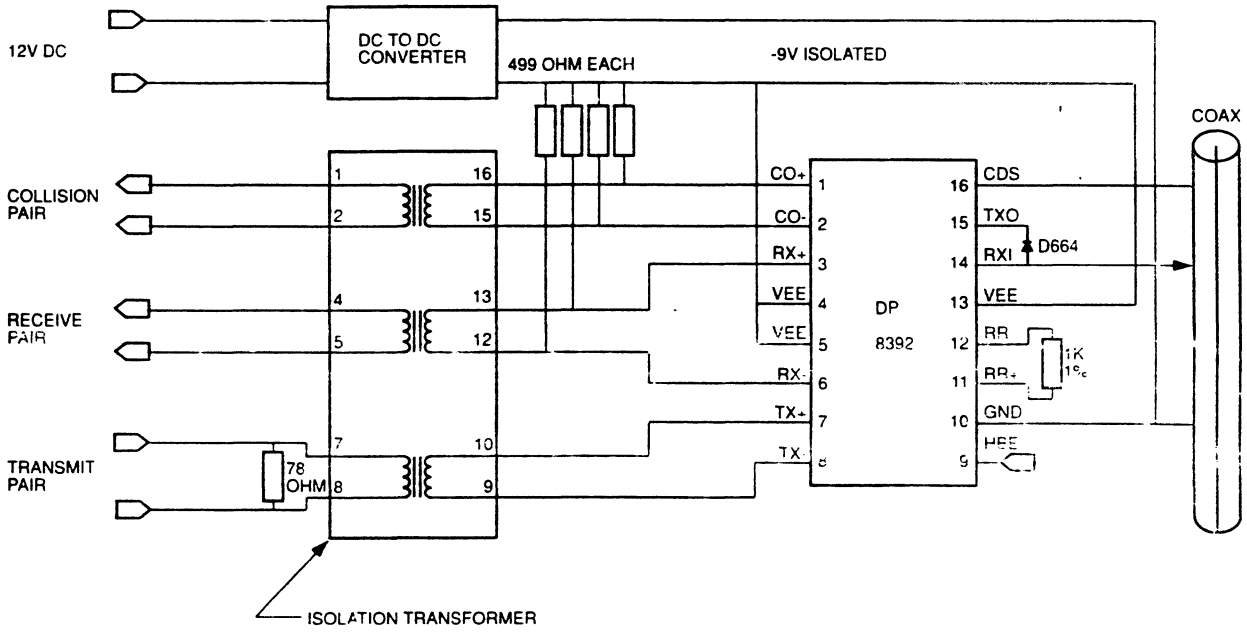


Figure 4–5 Functional Block Diagram of Thinwire Ethernet Transceiver

The DP8392 chip block diagram is shown in Figure 4-6 and is discussed in the following sections.

Section 4.5.7.1 Receive Functions
Section 4.5.7.2 Transmit Functions
Section 4.5.7.3 Collision Detect Functions
Section 4.5.7.4 Jabber Functions

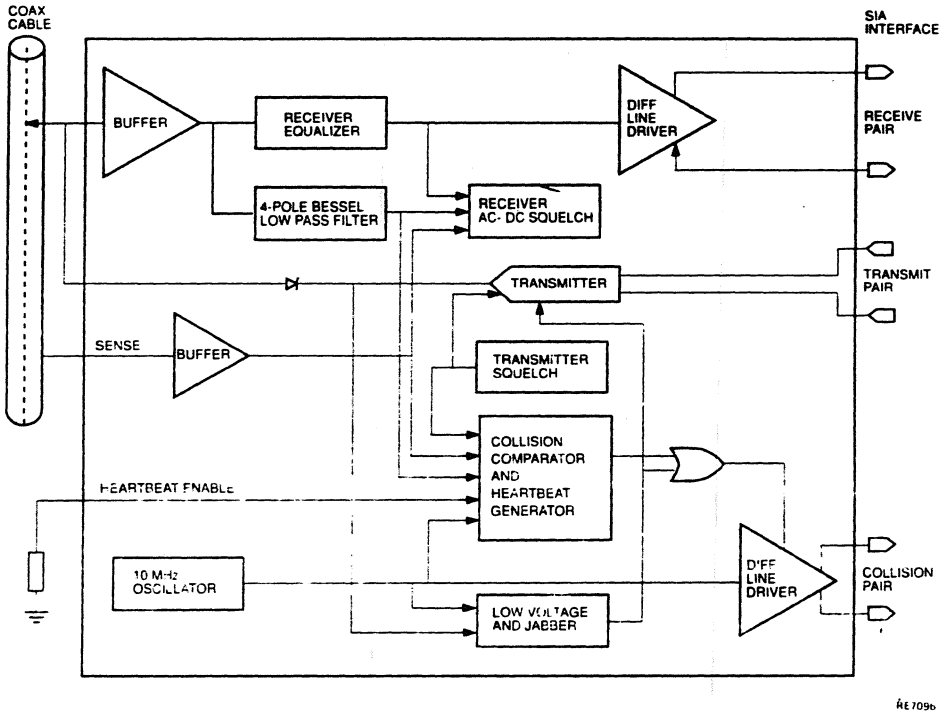


Figure 4–6 DP8392 Block Diagram

4.5.7.1 DP8392 Receive Functions

The receiver circuitry of the DP8392 transceiver chip is made up of the following components.

NOTE
Refer to the *National Semiconductor Corporation DP8392 Data Sheet* for detailed information.

1. The Input Buffer

   This buffer provides high input impedance and low input capacitance to minimise loading and reflections on the coaxial.

2. The Cable Equaliser

   This equaliser is a high pass filter that compensates for the low pass effect of the cable. The composite result of the maximum length cable and equaliser is a flatband response at the signal frequencies to minimise jitter.

3. The 4-Pole Bessel Low Pass Filter

   This extracts the average dc level on the coaxial cable, which is used by both the receiver squelch and the collision detection circuits.

4. The Squelch Circuit

   The receiver squelch circuit prevents any noise on the coaxial cable from falsely triggering the receiver in the absence of a signal. The squelch turns off at the beginning of the packet when the dc level from the low pass filter is lower than the dc threshold. At the end of the packet, a quick squelch turn on rejects dribble bits.

5. The Differential Line Driver

   This driver provides ECL-like signals to the rest of the DESQA circuitry that have rise and fall times of 5 ns. In the idle state, differential line driver outputs go to differential zero to prevent dc standing currents in the isolation transformer.

4.5.7.2 DP8392 Transmit Functions

The transmitter of the DP8392 transceiver chip receives differential signals from the rest of the DESQA circuitry through the isolation transformer of the Thinwire interface. It uses a squelch circuit that provides noise immunity, and an open collector current output driver to drive the coaxial cable when transmitting.

The open collector current output driver meets Ethernet specifications for signal levels. It controls rise and fall times to minimise higher harmonic components.

4.5.7.3 DP8392 Collision Detect Functions

The collision detect circuitry of the DP8392 transceiver chip, consists of the following components.

1. The Low Pass Filter

   This filter extracts average dc level on the coaxial.

2. The Comparator

   The comparator monitors the dc level from the low-pass filter. If the level is more negative than the collision threshold, the collision output is enabled.

3. The Heartbeat Generator

   The heartbeat generator enable is tied to ground.

4. The 10-MHz Oscillator

   This oscillator generates the signal for the collision/heartbeat functions, and is used as the timebase for all jabber functions.

5. The Collision Differential Line Driver

   This driver transfers the 10-MHz signal to the collision detector (CD) +/- input if collision, jabber, or heartbeat conditions are present. It also features zero differential idle state.

4.5.7.4 DP8392 Jabber Functions

The jabber timer monitors the transmitter for faults and inhibits transmission when faults exist. A fault exists if the transmitter is active for longer than 32 ms. The jabber timer enables collision outputs for the duration of faults. After faults are removed, the jabber timer waits 500 ms before re-enabling the transmitter. The transmit input must stay inactive during unjab time which is 500 ms.

4.5.7.5 Thinwire DC-to-DC Converter

The onboard DC-to-DC converter takes the +12V from the backplane and converts it into the -9V required for their operation of the DP8392 transceiver.

The type of converter used for this purpose is a pulse width modulated flyback type switching power supply.

Regulation is achieved via a simple single stage zener referenced amplifier, opto-coupled to the pulse width modulator circuit.

The result is a steady -9V output that is isolated from the +12V input.

Figure 4-7 shows the block diagram of the DC-to-DC Converter.

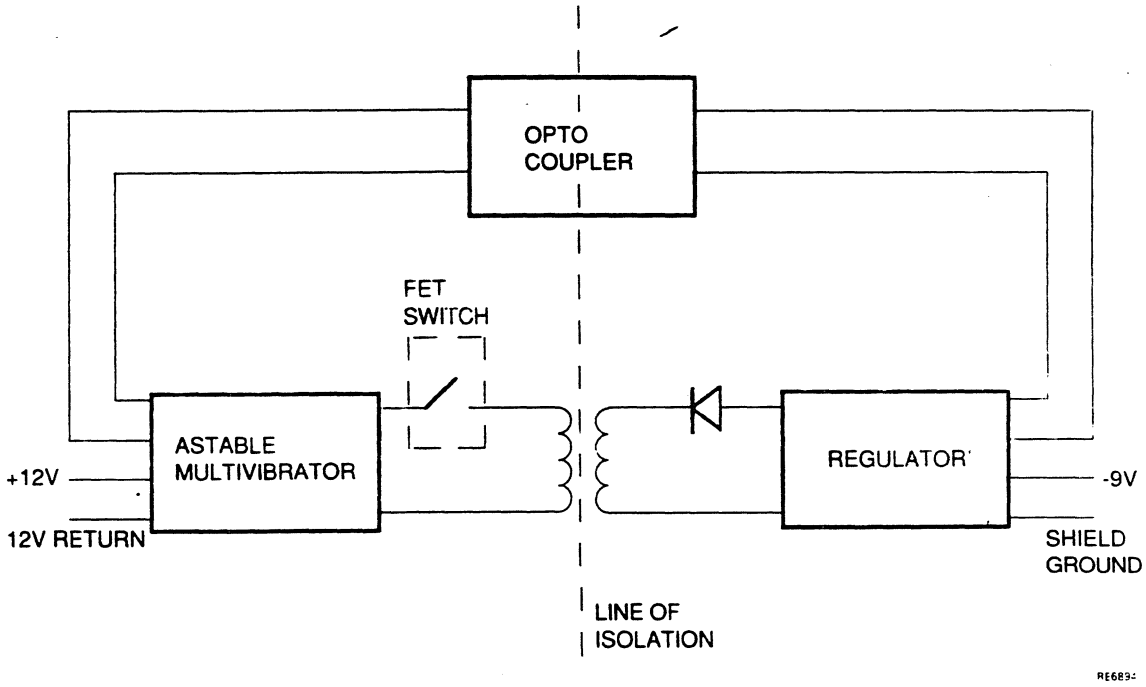


Figure 4–7 DC-to-DC Converter Block Diagram

4.6 BACKPORT BUS & QNA2

4.6.1 Overview

The DESQA module is built around a backport bus which passes data between the QIC, shared RAM, and the Lance.

The QNA2 (Q-Bus Network Arbitrator) arbitrates requests for access to the backport bus. It then implements the control function appropriate to the access request that it has granted. Priority of access is in the order QIC, 68000, LANCE.

The QNA2 input, FUSE, monitors the power indicator from the H4*xxx* Ethernet transceiver. The QNA2 also drives the on-board LEDs that indicate: powerup, loopback testing, citizenship (CQ) tests, and normal operation.

Table 4–6 shows how the memory space on the backport bus is used.

Table 4–6 Backport Address Map

IC	Word address Octal	Binary
QIC Base	40 377 440	100 000 011 111 111 100 100 000
QNA2 CSR	20 177 416	010 000 001 111 111 100 001 110
QNA2 IR	20 177 436	010 000 001 111 111 100 011 110

4.6.2 Backport Interface Timing

The 20 MHz CLOCK input to the QNA2 is shared with the QIC. This ensures the synchronous operation between the QNA2 and the QIC. The QNA2 further divides this 20MHz clock to produce a 10MHz clock for the 68000 microprocessor.

The antiphase version of the QNA2 clock is passed to the SIA which uses it to generate a 10MHz clock. This clock is passed to he LANCE as TXCLK.

4.7 MODULE CONTROL

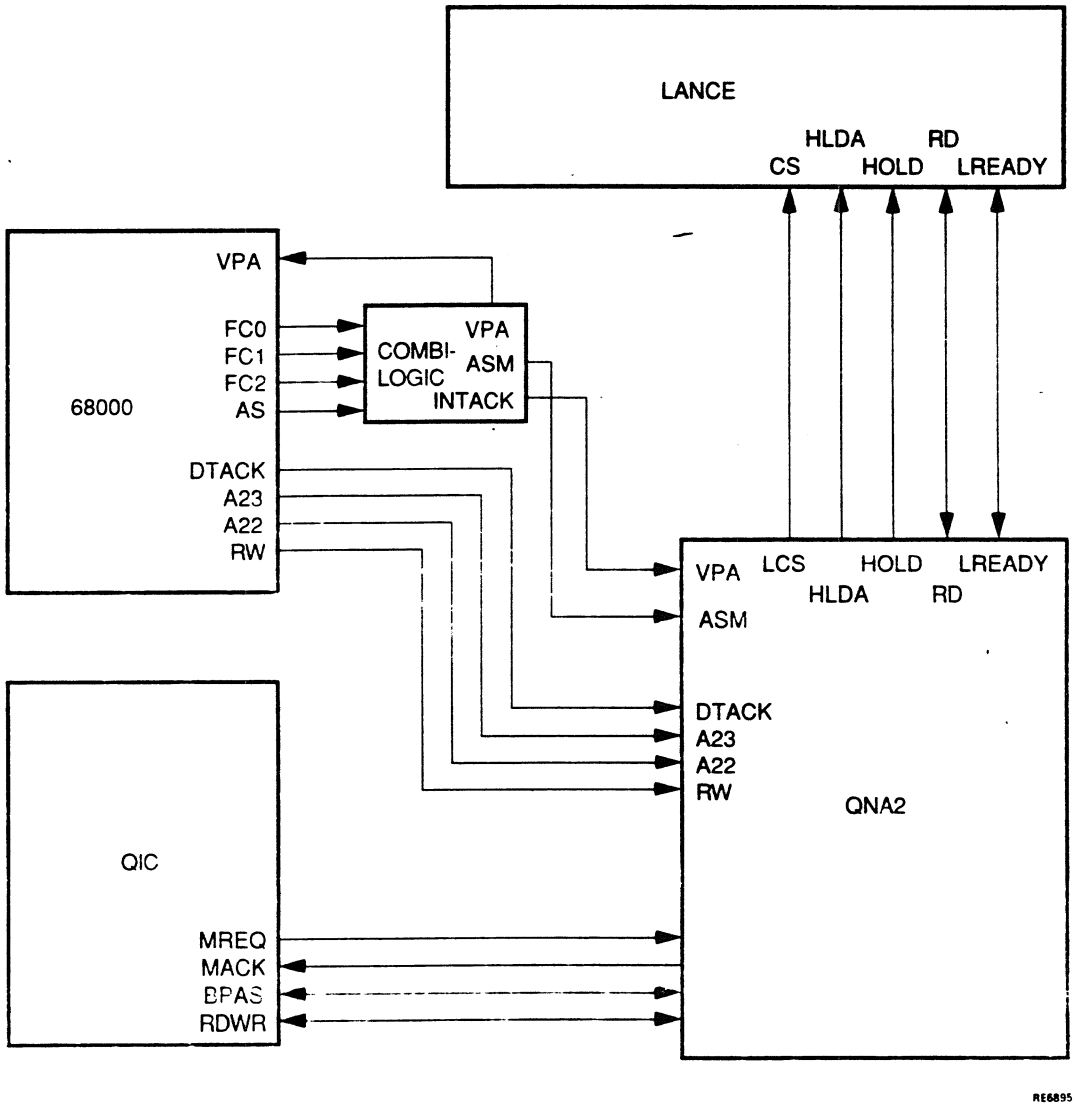


Figure 4–8 DMA Control Signals

4.7.1 Interrupts

In the DESQA module there are three interrupt lines to the 68000 microprocessor, as follows:

1. From the QIC
2. From the QNA2
3. From the LANCE

The ATTN (Attention) signal from the QIC to the 68000 is the result of a logical OR operation on several status condition bits within the QIC. It alerts the microprocessor, which can then interrogate the QIC backport attention register. Bits in this register are cleared when written to one, or when the QIC is reset by the QBUS.

Figure 4-9 shows the main interrupt control logic.

4.7.2 Error Recovery

Errors in accesses to host memory, and in Ethernet transmit and receive operations, are flagged in the Control and Status Register.

The error conditions are recorded originally in the QIC Attention Register or the LANCE CSR0. The 68000 responds to a backport interrupt from these devices by examining the relevant register, transferring the information to the DESQA CSR, and generating a Q-Bus interrupt to the host system.

The LANCE flags network errors in CSR0, to be examined by the 68000 following a backport interrupt. System errors include:

1. Babbling transmitter attempts to transmit more than 1518 bytes
2. Collision detection circuitry not functioning
3. Packet missed due to insufficient buffers
4. Memory acknowledge timeout.

Packet errors are recorded in the relevant buffer descriptor entries. Packet errors include:

1. Invalid CRC
2. Framing error
3. Overflow or underflow
4. Insufficient buffers.

The LANCE contains a 10MHz time domain reflectometry counter to aid the location of faults in an Ethernet cable.

Figure 4-6 shows the main DMA control signals.

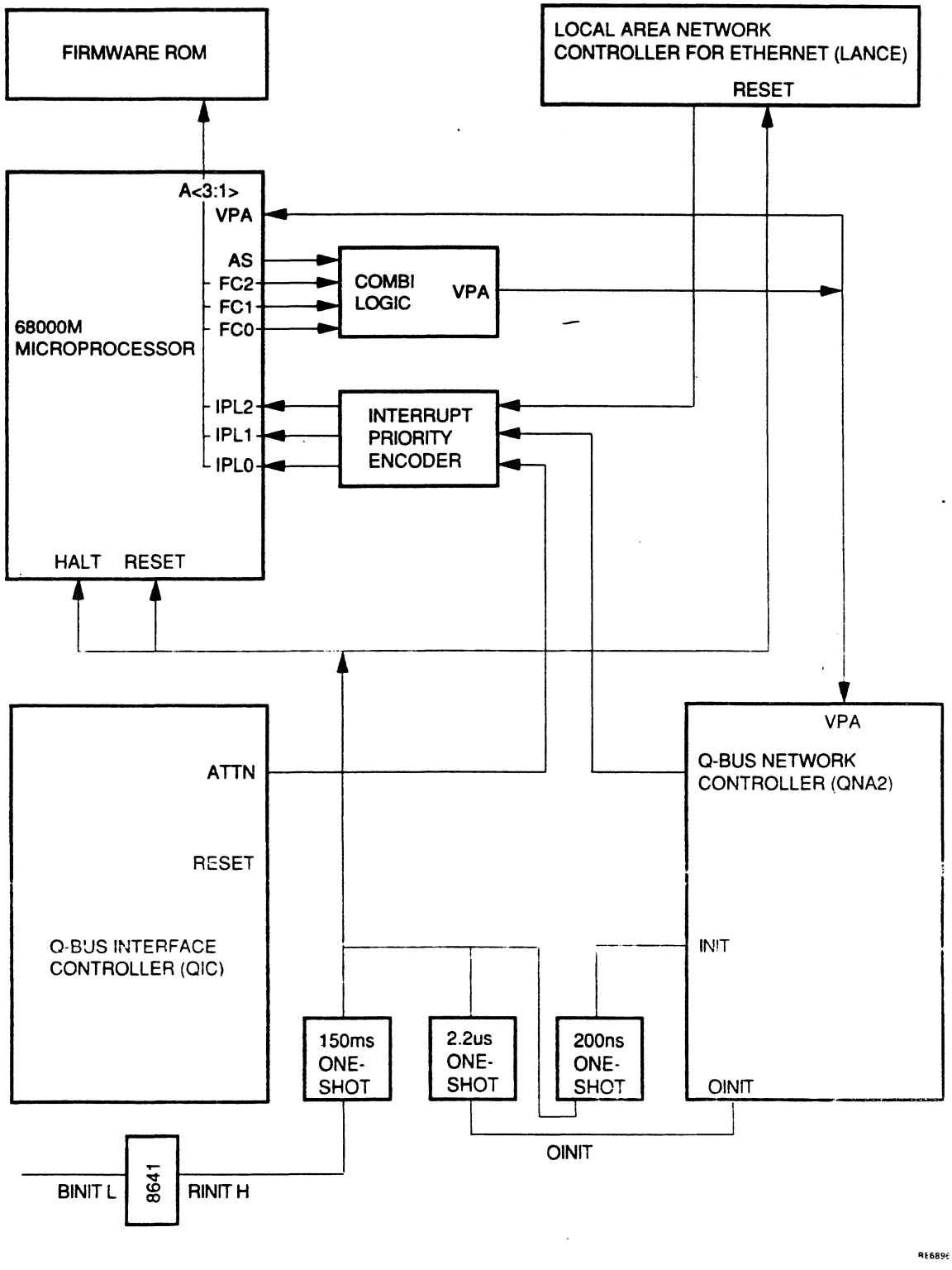


Figure 4–9 DESQA Interrupt and Reset Signals

4.7.3 Sanity Timer

The sanity timer is enabled and reset by the host software. After the timer is enabled, the software must reset it periodically; otherwise it counts to its preset limit and times out.

When the sanity timer times out, the Q-Bus line DCOK is negated for approximately 3.6 microseconds. This resets the host system, and invokes the primary bootstrap (depending how the jumpers are set up in the host processor).

On reset the timeout defaults to four minutes, but it can be changed during setup mode to any factor of four within the range 1/4 second to 64 minutes. The timer can also be disabled by host software command.

4.7.4 Initialisation

There are two initialisation mechanisms:

1. Q-Bus powerup/reset
2. Software reset

Powerup initialises the LANCE and 68000 for 150 milliseconds. The Q-Bus signal RINIT is used to trigger this reset.

NOTE
The QIC is initialised by the combination of the RINIT asserted and RDCOK negated.

Software Reset initialises the LANCE and 68000 for 2.2 microseconds. The signal OINIT is used to trigger this reset. OINIT is the inversion of CSR01.

The assertion of both these resets causes the QNA2 to be reset for 200ns.

CHAPTER 5
MAINTENANCE

5.1 SCOPE

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

Section 5.2	**Maintenance philosophy**
Section 5.3	**Built-in diagnostics**
Section 5.5	**IEEE 802.3 Network Support: Null link-layer Service Access Points**
Section 5.6	**Network diagnostics**
Section 5.7	**Module diagnostics**

WARNING

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

ATTENTION

Les procédures décrites dans ce chapitre nécessitent l'enlévement des capots du système. Elles ne pourront être effectuées que par du personnel qualifié.

VORSICHT!

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

¡ ATENCION!

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

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.

RE5146

Figure 5–1 Warnings

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。

אזהרה

הפעולות המתוארות בפרק זה, כרוכות בהסרת המכסים של המערכת וייבוצעו 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.

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.

VARNING

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

RES145

Figure 5–2 Warnings

5.2 MAINTENANCE PHILOSOPHY

5.2.1 Preventive Maintenance

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

5.2.2 Corrective Maintenance

The DESQA 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)

5.2.2.1 DESQA Thinwire Troubleshooting (Thinwire Mode)

The following figure shows a flowchart for troubleshooting the DESQA when configured in the Thinwire mode.

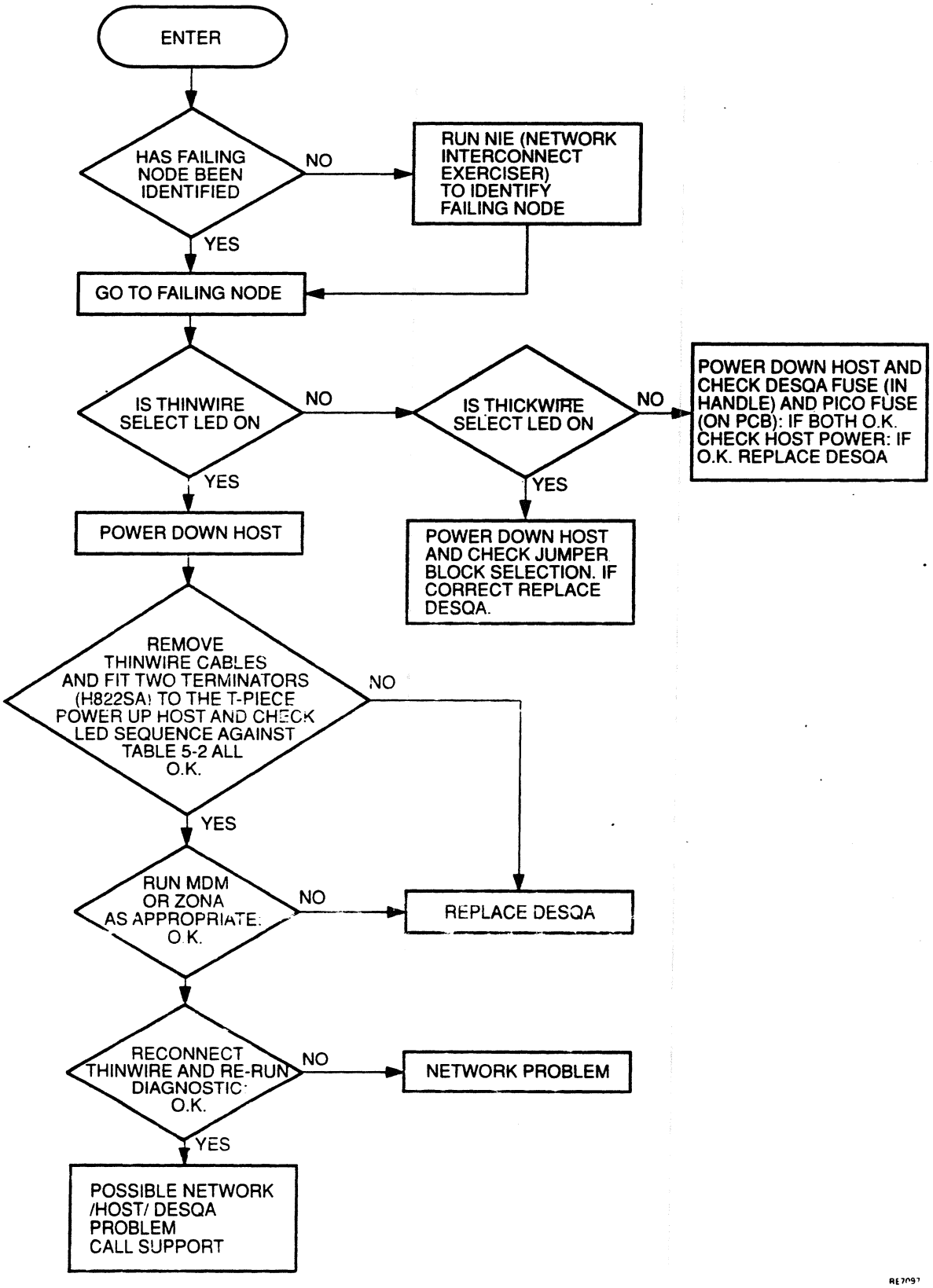
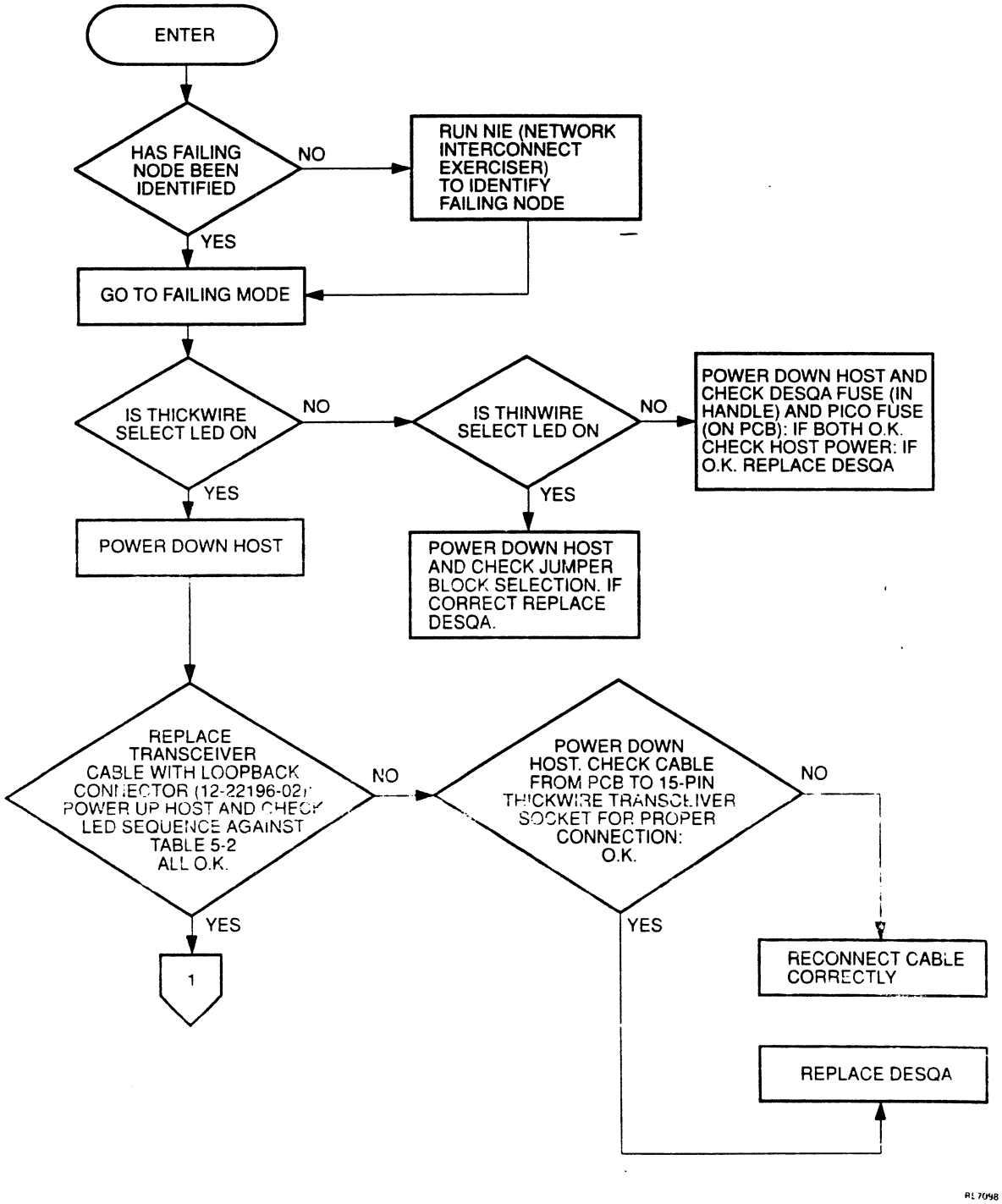


Figure 5–3 DESQA Troubleshooting (Thinwire Mode)

5.2.2.2 DESQA Thickwire Troubleshooting (Thickwire mode)

The following figure shows a flowchart for troubleshooting the DESQA when configured in the Thickwire mode



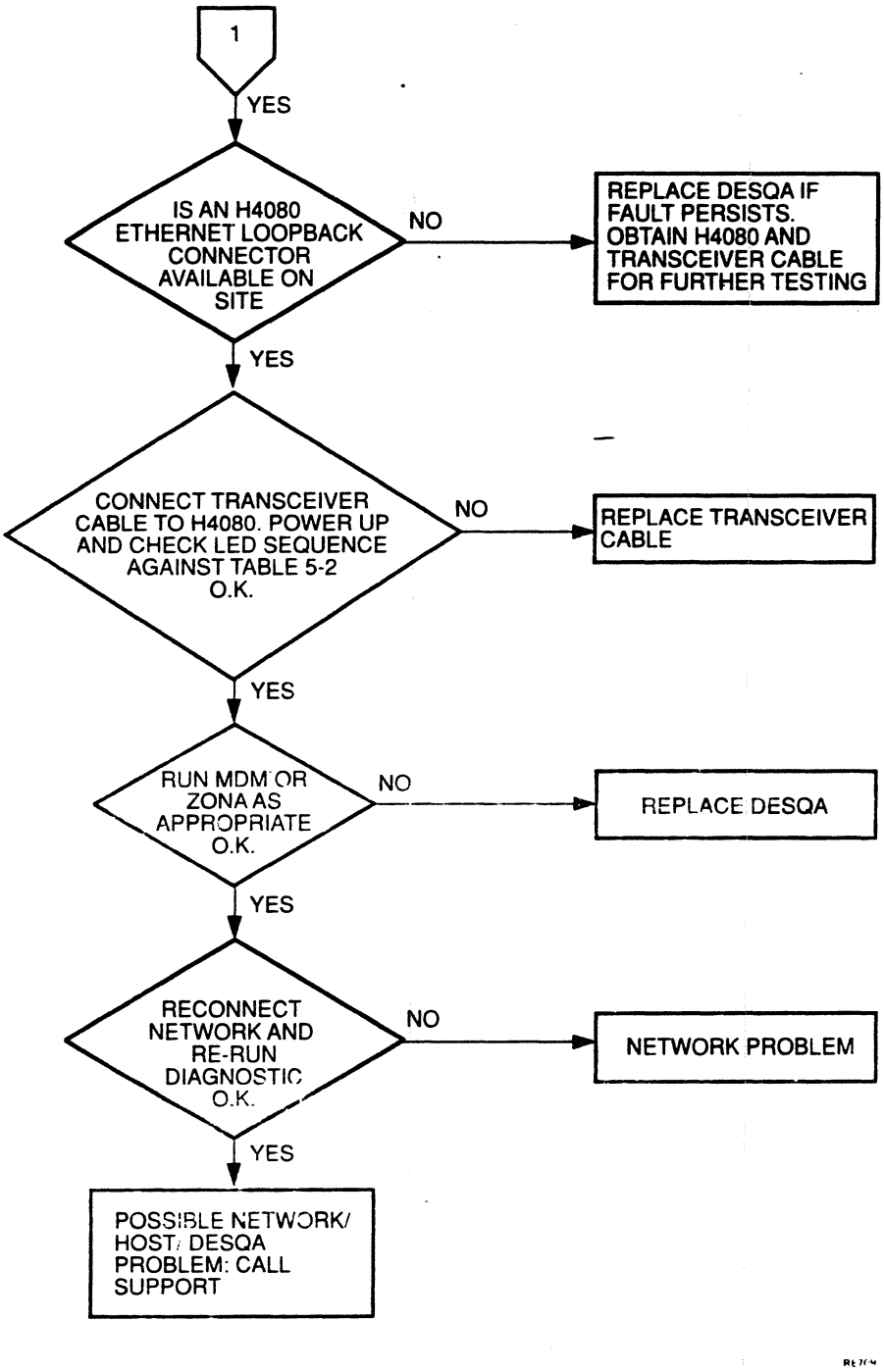


Figure 5–4 DESQA Troubleshooting (Thickwire Mode)

NOTE

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

5.2.3 Field Replaceable Units (FRUs)

The Field Replaceable Units (FRUs) are:

- The DESQA module
- Bulkhead assembly fuse
- Transceiver and transceiver cable (or bulkhead loopback connector)
- Module picofuse.

NOTE

This picofuse should only be replaced by trained personnel.

Figure 5-5 shows the field replaceable units in the DESQA installation.

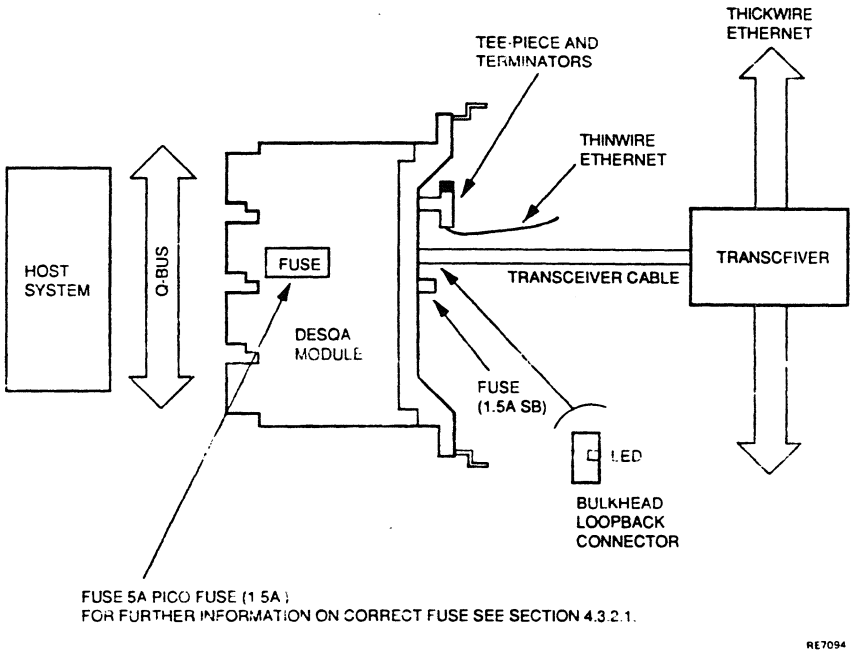


Figure 5–5 Field Replaceable Units (FRUs)

5.2.4 Diagnostic Procedure

The general strategy for identifying a fault is as follows:

1. Check the DESQA 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.

5.3 SELF-TEST

The DESQA 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 DESQA 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 DESQA 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, multi-cast, and physical address filtering
   - The internal loopback
   - The external loopback.

NOTE

A successful pass of the self-test requires that the DESQA be connected to a valid Ethernet network, or an external loopback connector. For Thickwire use the Bulkhead loopback connector (12-22196-02) or alternatively an H4080 Ethernet tester. For Thinwire, the "T" piece with two terminations (12-26318-01) attached should be connected to the BNC connector.

Table 5–1 Module LED Sequences

LED1	LED2	LED3	Definition
			Normal Mode — Power-Up Sequence
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 after this time, and irrecoverable failure has occurred. See Table 5–2.

Table 5–2 Irrecoverable Failure Indications

Failure Indication	LED1	LED2	LED3
Q-bus failure	F	OFF	OFF
LANCE failure	OFF	F	OFF
68000 exception	OFF	OFF	F
Firmware Fault	F	F	F

5.3.1 Extended Primary Bootstrap

A PDP-11 host can boot from a DESQA using a method similar to that for a mass-storage device. Part of the BD ROM on board the DESQA 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 DESQA 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 DESQA to access the Ethernet.

5.3.2 Citizenship Test

The DESQA citizenship test (CQ) is a series of diagnostic test routines that determine whether the DESQA 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 DESQA, 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 DESQA 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 initialised state to either a functional or a nonfunctional state.

5.3.2.1 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 DESQA 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 buffers. 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 DESQA. 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	External loopback operation status check

5.3.2.2 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 DESQA is set up as follows.

- All three DESQA 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 multi-cast address filtering are off.
- The DESQA 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) DESQA 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 DESQA passed all the CQ tests except those which require a connected transceiver.

Table 5–3 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 DESQA device interrupt

Table 5–3 (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 DESQA address

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

In NORMAL mode the DESQA 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.

5.4.1 TEST Message

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

5.4.2 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 DESQA 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.

5.5 NETWORK DIAGNOSTICS

5.5.1 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 DESQA 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.

5.5.2 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 DESQA, but the communications link to which it is connected; therefore, the NIE assumes that the DESQA has successfully completed the citizenship test.

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

Refer to Appendix B for further information.

5.6 MODULE DIAGNOSTICS

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

- The DESQA, DELQA or DEQNA modules
- The fuse.

5.6.1 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 DESQA simultaneously with other system devices

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

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

The field functional diagnostic program (ZQNA??) tests the DESQA 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 DESQA 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 DESQA 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.

5.6.3 PDP DEC/X11 Exerciser

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

NOTE

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

5.7 OPERATING ENVIRONMENTS

5.7.1 PDP-11 Diagnostic Runtime Services (DRS)

The Diagnostic Runtime Services (DRS) run under the XXDP+ monitor. To start the DRS, use the following procedure.

1. Boot XXDP+
2. To the "." prompt, type :R <NAME>

   where <NAME>is either:

   VNIA?? for the PDP NIE or

   ZQNA?? for the functional diagnostic
3. To the "DR>" prompt type START[/SWITCHES][/FLAGS].

   For details of switches and flags, sections B.2.1.2 and B.2.1.3

   respectively.

   DRS prompts CHANGE HW (L)?

   Respond with y(es) (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 DESQA/DEQNA?

   Respond with the address of the I/O page register assigned for one of the DESQA devices; refer to Chapter 3 for details.

   INTERRUPT VECTOR ADDRESS?

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

   WHAT IS THE PRIORITY LEVEL?

   Respond with the DESQA interrupt priority level.
4. DRS prompts: **CHANGE SW (L) ?**

   Respond with N(o).

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

5.7.1.1 DRS Commands

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

Table 5-4 lists the DRS commands.

Table 5–4 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 initialising.
START	Start the diagnostic from an initial state.
ZFLAGS	Clear all flags.

5.7.1.2 DRS Switches

To modify supervisor operation, several switches can be appended to each DRS command. The system will recognise 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 5-1 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 5-5 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 5–5 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 5–6 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)				

5.7.1.3 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 5-7.

Table 5–7 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).

5.7.2 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 DESQA 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 5.11 describes the DESQA MDM tests in more detail.

Refer to the *MicroVAX Diagnostic Monitor User's Guide* for further information about MDM.

5.8 NETWORK INTERCONNECT EXERCISER (NIE)

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 (DESQA), but the communications link to which it is connected; therefore, the NIE assumes that the DESQA 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 5.11.1.2), the NIE resumes testing where it left off after reporting the error. However, note that the NIE does not test the DESQA 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.

5.9 OPERATING MODES

The NIE is command-driven; that is, it executes commands given by the user. Commands are described in Paragraph Section 5.11 In addition to entering commands, the user can select one of two operating modes: unattended or operator directed.

5.9.1 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

5.9.1.1 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.

5.9.1.2 Direct Loop Message Test

This test checks the ability of a node to respond to a loopback request. (See Paragraph Section 5.11.2, RUN TEST command, DIRECT test.) A node has a maximum of 8 seconds to respond; three attempts are made to contact each node.

5.9.1.3 Pattern Test

This test sends six different loop direct messages to each node in the node table. (See Paragraph Section 5.11.2, RUN TEST command, PATTERN test.)

5.9.1.4 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.

5.9.2 Operator Directed Mode

The commands available in this mode are listed below and described in Paragraph Section 5.11.2.

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

5.10 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
- DESQA 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.

5.11 COMMAND DESCRIPTION

5.11.1 DRS Commands

The 11 DRS commands are listed in Table 5-8, with a brief description of each. The system will recognise a command by its first three characters; for example, you can type STA instead of START.

Table 5–8 . DRS Commands

Command	Description
START	Start the diagnostic from an initial state.
RESTART	Start the diagnostic without initialising.
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 5.11.1.2)
ZFLAGS	Clear all flags (see Section 5.11.1.2)

5.11.1.1 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 recognise a switch by its first three characters. For example, you can type /TES:1–5 instead of /TESTS:1–5. Table 5-9 lists the switches that can be used with each command.

Table 5–9 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 5.11.1.2)
/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 5–10 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)				

5.11.1.2 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 5-11.

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 5–11 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 5.12.1.

5.11.2 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 5–12 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 5–12 (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 default parameters 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 in Table 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 5–12 (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 5–12 (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. 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 5-9, 5-10, and 5-11.) 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 5-11) 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.

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 DESQA as a loop assist or target node on the Ethernet. The other tests ignore forwarding requests, and will not transmit console IDs.

Table 5–12 (Cont.) NIE Commands

Command	Description
	Examples:
	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 5–12 (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. 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 5–12 (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**

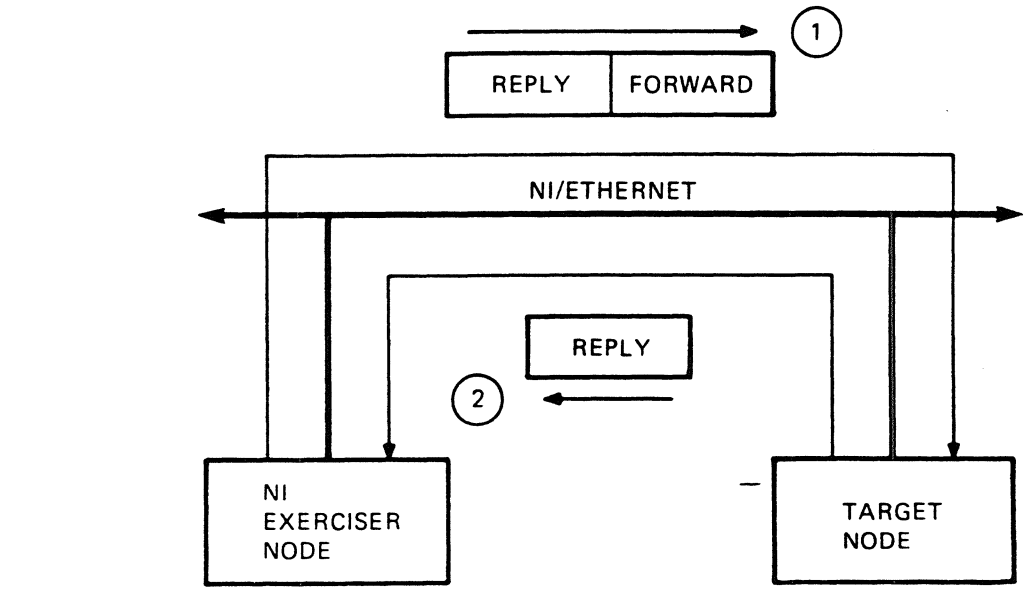


Figure 5–6 Loop Direct Messages Test Path

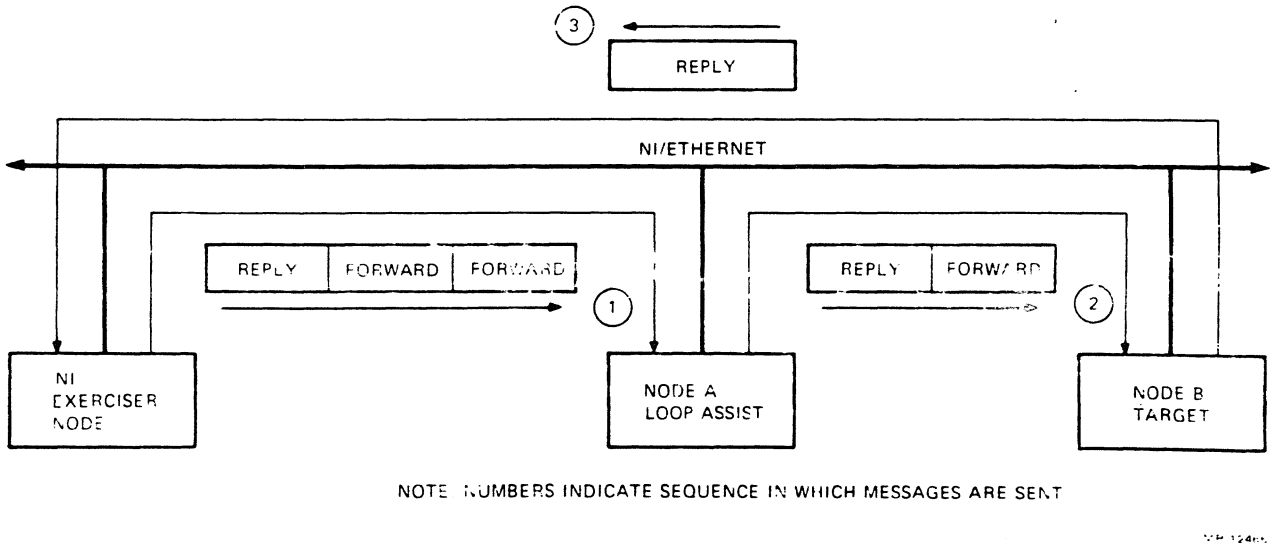
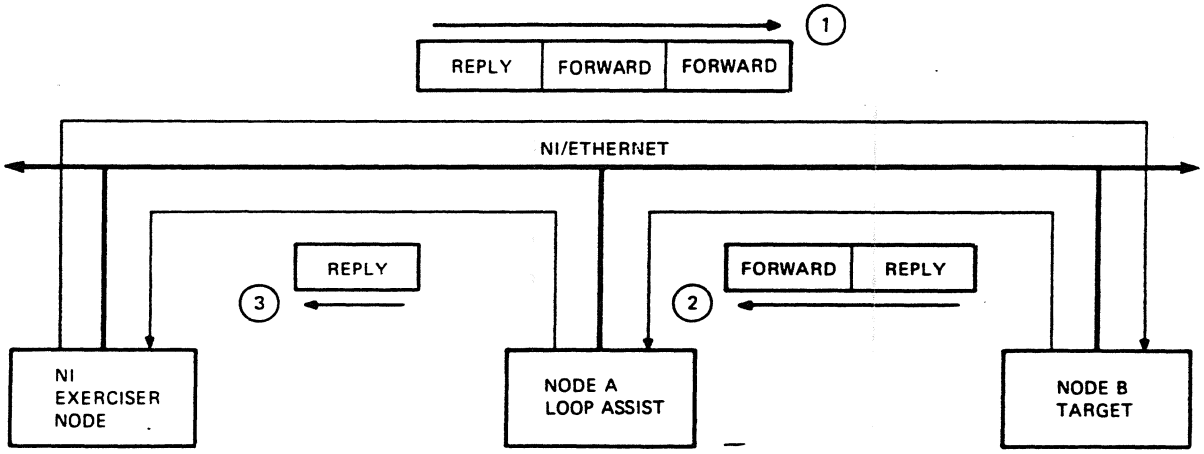


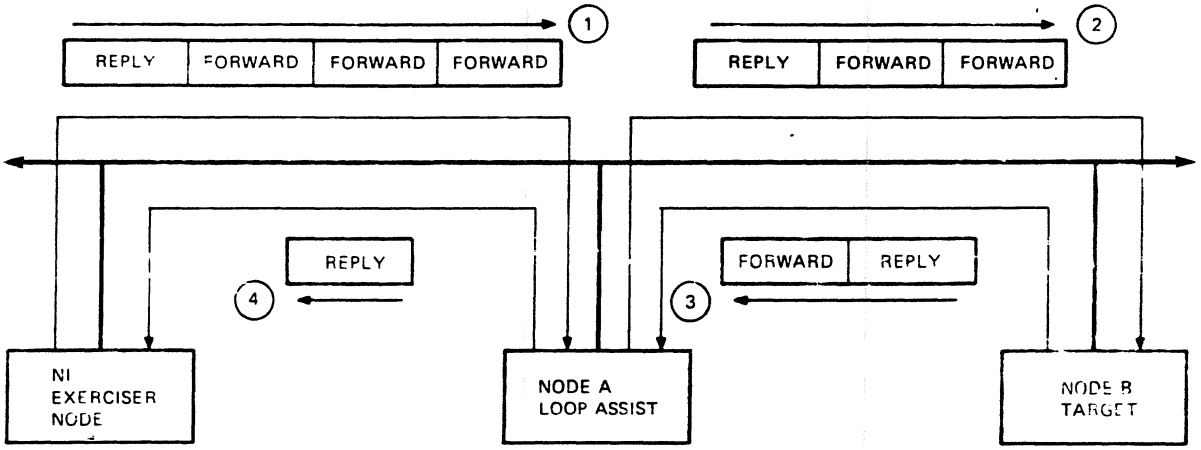
Figure 5–7 Transmit Assist Loopback Message Test Path



NOTE: NUMBERS INDICATE SEQUENCE IN WHICH MESSAGES ARE SENT

MR 12467

Figure 5–8 Full Assist Loopback Message Test Path



NOTE: NUMBERS INDICATE SEQUENCE IN WHICH MESSAGES ARE SENT

MR 12468

Figure 5–9 Receive Assist Loopback Message Test Path

5.12 ERRORS

5.12.1 Error Messages

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

5.12.1.1 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

5.12.1.2 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.

5.12.1.3 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

5.12.2 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.

5.13 PDP-11 FUNCTIONAL DIAGNOSTIC (ZQNA)

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

- DESQA, DEQNA or DELQA
- Fuse

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

NOTE

Early versions of ZQNA?? only work with the DEQNA. ZQNA1 is the first version to be compatible with both DEQNA and DESQA.

NOTE

A successful pass of the self-test requires that the DESQA be connected to a valid Ethernet network, or an external loopback connector. For Thickwire use the Bulkhead loopback connector (12-22196-02) or alternatively an H4080 Ethernet tester. For Thinwire, the "T" piece with two terminators (12-26318-01) attached should be connected to the BNC connector.

5.13.1 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).

5.13.2 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 loopback path verification test
15. Carrier bit test
16. Sanity timer test

5.13.3 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 5-2.

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 DESQA

Return status messages:

TRANSMIT STATUS ERROR

RECEIVE STATUS ERROR

CSR STATUS ERROR

5.14 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 DESQA 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 DESQA details.
2. Run the Verify tests (FUNCTIONAL and EXERCISER) to check DESQA 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.

5.14.1 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:
  - — In Thickwire Mode– Bulkhead loopback connector (order number 12-22196-02)
  - — Cable and Ethernet transceiver to provide external loopback.
  - — In Thinwire Mode– "T" piece and two terminators (order number 12-26318-01)
- 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.

5.14.2 MDM Service Test Descriptions

5.14.2.1 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 Initialise 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 multi-cast filtering
Test normal filtering

TEST7 (*Normal mode only*) Verify the operation of Extended setup packet by looping several back through the DESQA.
Verify correct processing of the MOP element blocks

5.14.2.2 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 DESQA-cable-transceiver loop. (Internal loopback is tested in the MODE routines.)

TEST1 Send packets using external loopback mode

5.14.2.3 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

5.14.3 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 cables attached to the DESQA.

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**

5.15 DEC/X11 EXERCISER

The DEC/X11 Exerciser exercises one DESQA at maximum activity rates. It transmits and receives random-length packets (using either 18- or 22-bit physical address space). The DESQA 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 DESQA is dropped from further testing if any of the following occurs.

- The DESQA 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.

5.15.1 Environment

It is assumed that, prior to running this exerciser, both the DESQA 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 in Thickwire Mode, the DESQA under test must be connected to the transceiver, or the external loopback connector must be connected.

In the Thinwire Mode,the terminators must be connected.

The options for Software Register 1 (SR1) bits 0 and 1 are described in Table 5-13.

Table 5–13 DESQA DEC/X11 Exerciser Software Register Bits

BIT 1	BIT 0	Description
	Clear	Exerciser runs in internal extended loopback mode (default). The network is not needed.
	Set	Exerciser runs in external loopback mode. A valid network or external loopback connector is required.
Clear		Print error messages.
Set		Do not print error messages.

5.15.2 Command Descriptions

To set external loopback mode, type the following commands.

MOD QNAA0 16 <RETURN>

1 <RETURN>

To test a DESQA 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 A
IC DESCRIPTIONS

A.1 SCOPE

This appendix contains some basic data on the main Integrated Circuits in the DESQA. The sections are as follows:

Section A.2	**68000 Microprocessor**
Section A.3	**Q-Bus Interface Controller(QIC)**
Section A.4	**QNA2 Q-Bus Network Arbitrator**
Section A.5	**AM7990 Local Area Controller for Ethernet (Lance)**
Section A.6	**AM7991 Serial Interface Adapter(SIA)**
Section A.7	**DP8392 Thinwire Ethernet Transceiver**

Each section includes the following (where relevant):

1. Overview.
2. Signals and pinout.
3. Addressing, memory structures, and registers.

For more detailed information, refer to the manufacturer's data sheets. Other components on the DESQA board are well described in the standard references, and are not included here.

NOTE

Not all features described in these sections are used by the DESQA.

A.2 68000 MICROPROCESSOR

A.2.1 Overview

The 68000 is a 16-bit microprocessor which has a 32-bit internal architecture. The 68000 is located on the top side of the DESQA board. Its main features are:

- 16-bit asynchronous data bus
- 23-bit asynchronous address bus, capable of addressing 16 Mbytes in conjunction with the data strobe signals UDS and LDS
- Eight 32-bit data registers
- Seven 32-bit address registers
- Memory-mapped I/O
- Compatibility with 6800-series peripheral ICs
- Single +5V power supply
- Mounted in a PLCC package.

The internal registers of the 68000 are shown in simplified form in Figure A–1.

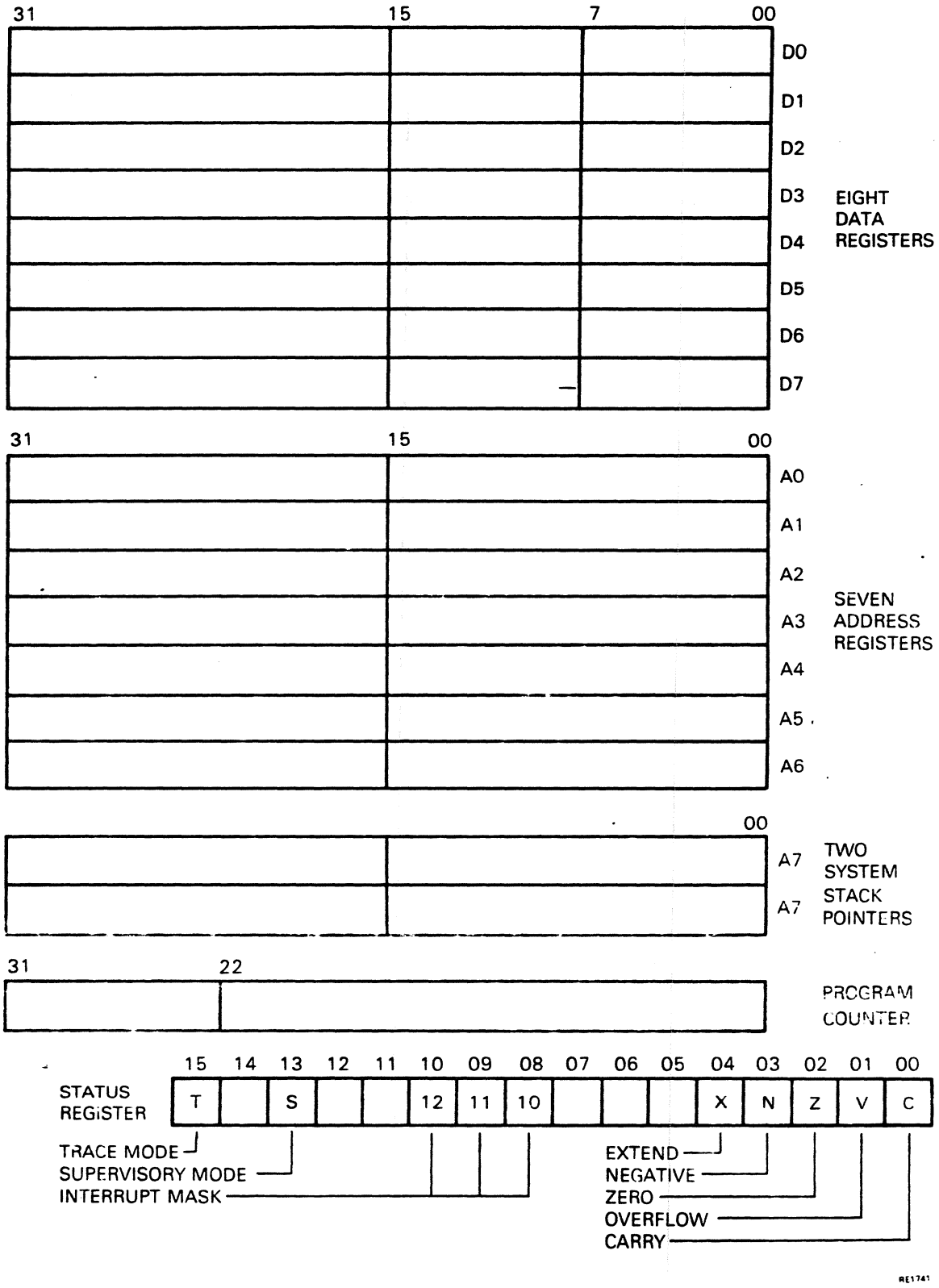


Figure A–1 68000 Internal Registers

Table A–1 68000 Signal Descriptions

Address and Data Bus

Pin	Name	Sense†	Description
01:05 54:64	D<04:00> D<15:05>	I/O Hi	Data Bus Lines D0:D15. 16-bit bus to transfer words or bytes. Lines D0:D7 are also used to receive a vector number during an interrupt-acknowledge cycle.
29:32 33:48 50:52	A<01:04> A<05:20> A<21:23>	O Hi	Address Bus Lines A1:A23. 23-bit bus to address 16 MBytes in conjunction with Data Strobes UDS and LDS. Lines A3:A1> are also used to signal the interrupt level while an interrupt is being serviced. In the DESQA, A20 selects the firmware ROM.

Asynchronous Bus Control

Pin	Name	sense†	Description
06	AS	O Lo	Address Strobe to indicate that a valid memory address is on the address bus.
07 08	UDS LDS	O Lo	Data Strobes to indicate whether data for transfer is on the upper, the lower, or both bytes of the data bus. (No Connection)
09	R/W	O Hi	Read to indicate direction of data.
		Lo	Write transfer to the data bus latches.
10	DTACK	I Lo	Data Transfer Acknowledge to complete a data transfer. The data bus cycle is extended until DTACK so that slow devices or memories can synchronise with the cycle.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

A.2.2 Signals and Pinout

The signals to and from the 68000 microprocessor fall into logical groups. The functions, signals and pinouts of these groups are described in Table A-1, and shown in Figure A–2.

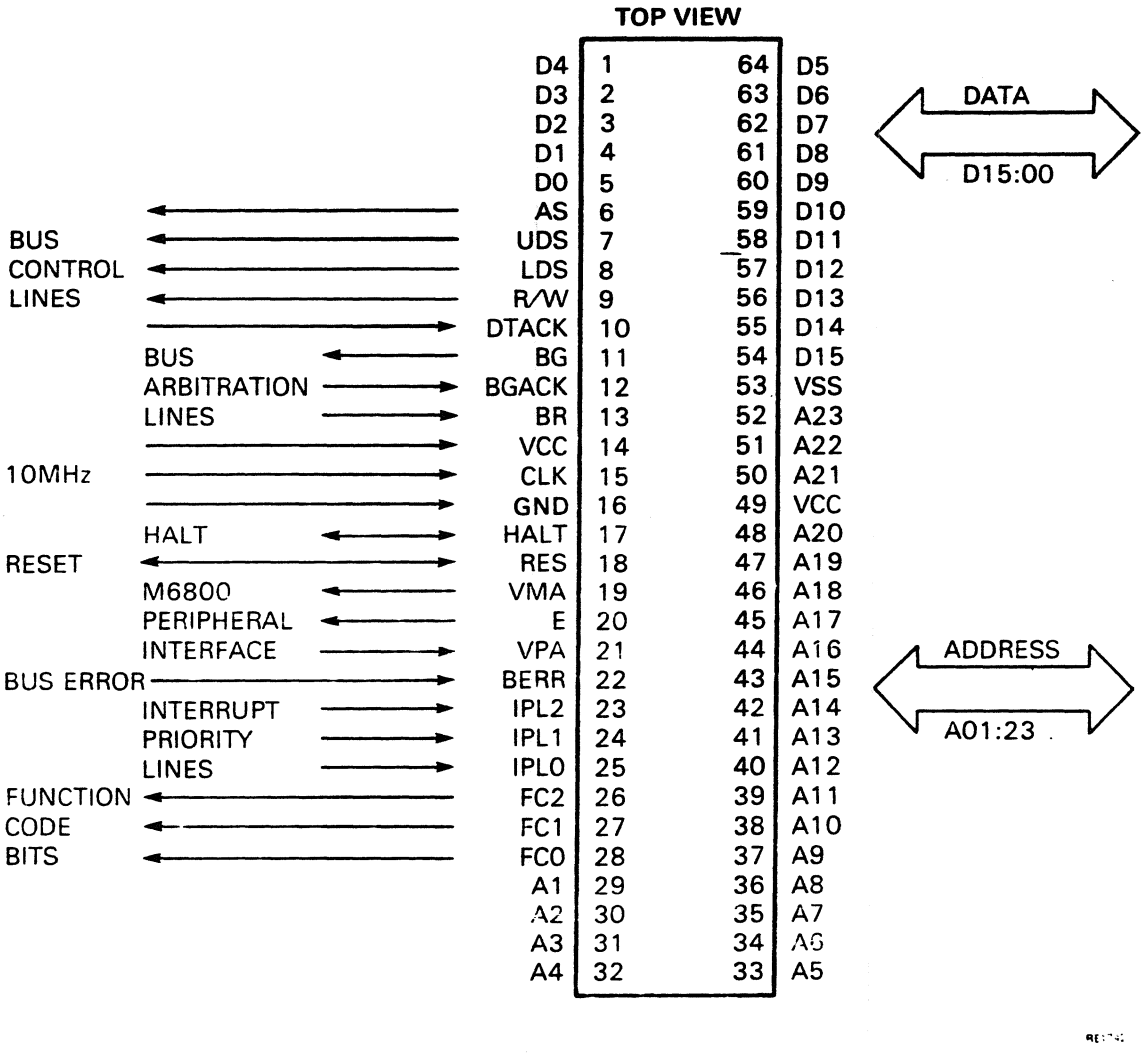


Figure A–2 68000 Pinouts

Bus Arbitration (Not used)

Pin	Name	sense†	Description
13	BR	I Lo	Bus Request from a device for bus control.
11	BG	O Lo	Bus Grant gives control of the bus.
12	BGACK	I Lo	Bus Grant Acknowledge from a device confirms that it has taken control of the bus.
Interrupt Priority			
23 24 25	IPL2 IPL1 IPL0	I Lo	Interrupt Priority Lines IPL<2:0> encode the priority of an interrupting device or process in the range 0 (no interrupt) to 7 (highest priority). IPL2 is the MSB.
Function Code			
28 27 26	FC0 FC1 FC2	O Hi	Function Code Lines FC<2:0> indicate the state (user or supervisor) and type of cycle (data, program, interrupt) executed:

FC2	FC1	FC0	State	Cycle
0	0	0	User	Reserved
0	0	1	User	Data
0	1	0	User	Program
0	1	1	User	Reserved
1	0	0	Supervisor	Reserved
1	0	1	Supervisor	Data
1	1	0	Supervisor	Program
1	1	1	Supervisor	Interrupt

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

M68000 Peripheral Interface

Pin	Name	sense†	Description
21	VPA	I Lo	Valid Peripheral Address indicates that the device or memory addressed is type M68000 and that data transfer should be synchronised to Enable (E).
19	VMA	O Lo	Valid Memory Address indicates, in response to VPA, that a valid address is on the address bus and that the 68000 is synchronised to the Enable Signal. (Not used.)
20	E	O Hi	Enable is the standard enable clock signal for M68000 systems. (Not used.)

System Control and Timing

17	HLT	I/O Lo	Halt is a bidirectional line that either receives an external halt from, or sends a halt signal to, external devices. When generated internally the instruction halts the processor. When HLT is asserted by an external device, the 68000 halts at the end of the current bus cycle. A halted 68000 can be restarted only by an external Reset. HLT also interacts with RES and BERR.
18	RESET	I/O Lo	Reset is a bidirectional line that either receives an external reset from, or sends a reset signal to, external devices. When generated internally the instruction resets all external devices without affecting the 68000. When RES and HLT are asserted by an external device, all devices are reset and the 68000 executes its reset vector at interrupt level 7 (non-maskable: highest priority).
22	BERR	I Lo	Bus Error terminates the current bus cycle due to an error. If HLT is asserted, the 68000 prepares to rerun the cycle, and does so when HLT is negated. If HLT is negated, the 68000 executes its bus error vector. (Pullup resistor.)
15	CLK	I Hi	Clock input from the QNA2, frequency 10MHz.

Power Supply

Pin	Name	sense†	Description
14,49	Vcc	I	+5V Power rail.
16,53	Vss		Gnd Earth rail.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

A.2.3 Addressing

The 68000 uses an asynchronous bus structure with separate address and data buses. The processor executes read and write cycles to transfer bytes and words from and to shared buffer RAM, using the function code and bus control signals to control the transfer.

The Test and Set (TAS) instruction operates on bytes only to provide interprocessor communication. The instruction is executed by an indivisible read-modify-write cycle; during this cycle AS (Address Strobe) is asserted throughout.

Bus arbitration is controlled using the Bus Arbitration signals to determine whether the 68000 or an external device is acting as bus master.

A.3 Q-BUS INTERFACE CONTROLLER (QIC)

A.3.1 Overview

This section describes the DIGITAL Q-Bus Interface Controller (QIC). It is intended only to describe the functions of the QIC that are used in the DESQA module. It is not a complete QIC specification; nor does it include details of Q-Bus operations.

The QIC is a general-purpose Q-Bus Interface Controller, which provides complete Q-Bus slave control logic. The QIC is packaged in an 84-pin plastic leaded chip-carrier (plcc). Together with 2 8641-2 and 4 DC021 transceivers, the QIC forms a complete Q-Bus interface design.

The DESQA supports both block-mode and non-block mode DMA on the Q-Bus.

On the Qbus side, the QIC supports control DMA (based on Buffer Descriptor Lists) and data DMA, using block-mode to achieve the highest possible speeds.

On the backport side, the QIC uses DMA to transfer data to and from shared buffer RAM and on-board registers (if supported by the module). The backport has 16 data/address lines and up to four control lines. The main controller can program the QIC registers and initiate DMA transfers to and from the Q-Bus.

Internally the QIC provides:

- Complete Q-Bus slave control logic
- I/O page addressing, with: slave address matching and programmable base-address register.
- DMA arbitration and control, both block-mode and non-block-mode
- 22-bit Q-Bus DMA address register/counter
- 15-bit DMA word-count register
- 16-bit backport DMA address register/counter and control
- Interrupt control
- Non-existent-memory (NXM) timeout
- Holdoff timer to separate DMA requests so that other units can access the Q-Bus
- Ability to initiate host reboot.

The architecture of the QIC is shown in the block diagram Figure A-3.

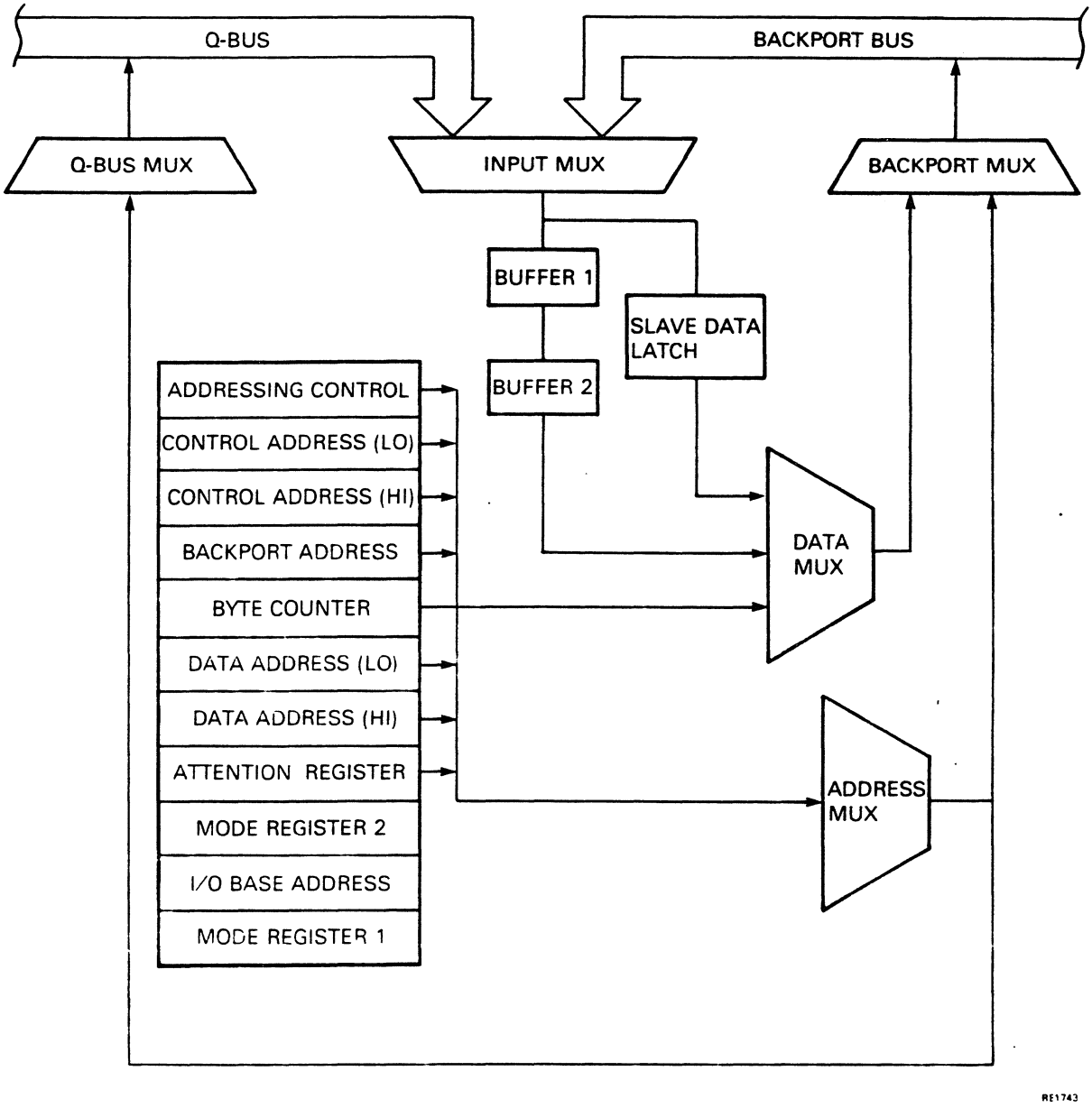


Figure A–3 QIC Block Diagram

A.3.2 Signals and Pinouts

The signals for each pin of the QIC are shown in Figure A–4, and the signals are described in Table A–2. Some of the QIC signals share pins on the IC, and the circuit designer will have chosen one function or the other.

DESQA does not use all the functions provided by the QIC. Unused outputs are not connected; unused inputs are tied high or low as appropriate.

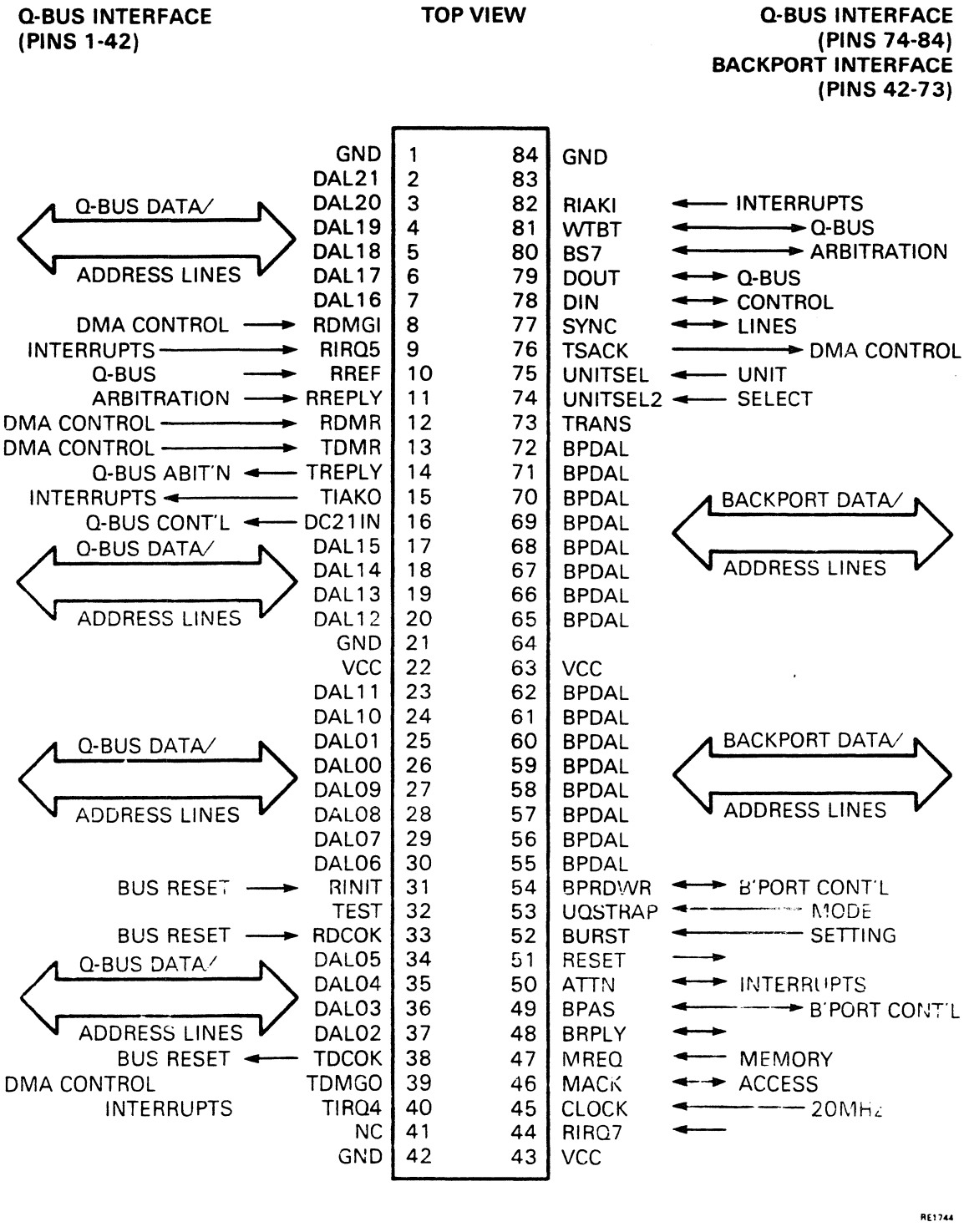


Figure A–4 QIC Pinouts

Table A–2 QIC Signal Descriptions

QIC to Q-Bus Interface: Address and Data Bus Control

02:05	DAL<21:18>	I/O Hi	Data/Address Lines, connected as follows to the DC021 transceivers:
06:07	DAL<17:16>		DAL<15:08> to DC021 number 3;
17:20	DAL<15:12>		DAL<21:20,07:02> to DC021 number 2;
23:24	DAL<11:10>		DAL<19:16,01:00> to DC021 number 1;
25:26	DAL<01:00>		DAL<00> is held low for data DMA, high for control DMA transfers.
27:30	DAL<09:06>		
34:37	DAL<05:02>		
			DAL<21:18> are Address lines.
			DAL<17:16> are Memory parity/Address lines.
			DAL<15:00> are Data/Address lines.

QIC to Q-Bus Interface: DMA Control

Pin	Name	sense†	Description
08	RDMGI	I Hi	Receive DMA Grant In (with pulldown resistor), from the Q-Bus signal BDGMI (receive), using DC021 number 1.
12	RDMR	I Hi	Receive DMA Request, which connects to the Q-Bus signal BDMR (receive), using 8641 transceiver number 2.
13	TDMR	O Hi	Transmit DMA Request, which is sent to the Q-Bus signal BDMR (transmit), using 8641 transceiver number 2.
39	TDMGO	O Hi	Transmit DMA Grant Out, which is sent to the Q-Bus signal BDMGO (transmit), using 8641 transceiver number 1 (with a pulldown resistor).
76	TSACK	O Lo	Transmit DMA Selection Acknowledge, which controls the direction of transfer of DC021 number 4.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

QIC to Q-Bus Interface: Interrupts control

Pin	Name	sense†	Description
09	RIRQ5	I Hi	Receive Interrupt Request Level 5 (with pulldown resistor), from the Q-Bus signal BIRQ5 (receive), using DC021 transceiver number 1.
15	TIAKO	O Hi	Transmit Interrupt Acknowledge Out, which is sent to the Q-Bus signal BIAKO, using DC021 number 4 (with a pulldown resistor).
40	TIRQ4	O Hi	Transmit Interrupt Request Level 4, which is sent to the Q-Bus signal BIRQ4 (transmit), using 8641 transceiver number 2.
41	TIRQ6	O Hi	No connection.
44	RIRQ7	I Hi	Tied inactive low.
82	RIAKI	I Hi	Receive Interrupt Acknowledge In (with pulldown resistor), from the Q-Bus signal BIAKI, using DC021 number 4.
83	RIRQ6	I Hi	Receive Interrupt Request Level 6 (with pulldown resistor), from the Q-Bus signal BIRQ6 (receive), using DC021 number 1.

QIC to Q-Bus Interface: Address and Data Bus Control

Pin	Name	sense†	Description
11	RREPLY	I Hi	Receive Reply, from the Q-Bus signal BRPLY (receive), using 8641 transceiver number 2.
14	TREPLY	O Hi	Transmit Reply, which is sent to the Q-Bus signal BRPLY (transmit), using 8641 transceiver number 2.
80	BS7	I/O Hi	Bank Select 7 (tristate signal), which connects to the Q-Bus signal BBS7, using DC021 number 4.
81	WTBT	I/O Hi	Write Byte (tristate signal), which connects to the Q-Bus signal BWTBT, using DC021 number 4.
16	DC021IN	O Hi	DC021 Input controls the direction of transfer of DC021 numbers 1 to 3.

QIC to Q-Bus Interface: Address and Data Bus Control

Pin	Name	sense†	Description
77	SYNC	I/O Hi	Synchronise (tristate signal), which connects to the Q-Bus signal BSYNC, using DC021 number 4.
78	DIN	I/O Hi	Data Input (tristate signal), which connects to the Q-Bus signal BDIN, using DC021 number 4.
79	DOUT	I/O Hi	Data Output (tristate signal), which connects to the Q-Bus signal BDOUT, using DC021 number 4.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

QIC to Q-Bus Interface: Bus Control

Pin	Name	sense†	Description
10	RREF	I Hi	Refresh, from the Q-Bus signal BREF, using 8641 transceiver number 1.
31	RINIT	I Hi	Receive Initialise, from the Q-Bus signal BINIT, using 8641 transceiver number 1. This signal is asserted each time the Q-Bus host executes a Reset instruction.
33	RDCOK	I Lo	Receive DCOK, from the Q-Bus signal BDCOK (receive), using 8641 transceiver number 1. When RINIT is active at the same time, the QIC asserts ATTN to alert the 86000 microprocessor.
38	TDCOK	O Lo	Transmit DCOK, which is sent to the Q-Bus signal BDCOK (transmit), using 8641 transceiver number 1.

QIC to Q-Bus Interface: Unit Select

74 75	UNITSEL<2> UNITSEL<1>	I Hi	Unit Select pins at powerup is reflected in mode register 2, bits <15:14> UNITSEL<1> is connected to a test point to provide an external select facility.
75	EXTSEL	I Hi	External Select. This pin is used to request QIC access to the backport memory

QIC to Q-Bus Interface: Power

Pin	Name	sense†	Description
1, 21, 42, 64, 84	Gnd		Signal ground.
22,43, 63	Vcc		+5V Power rail.

QIC to Backport Interface: Address and Data Bus

Pin	Name	sense†	Description
72:65	BPDAL<15:08>	I/O Hi	Backport Data/Address Lines (tristate)
62:55	BPDAL<07:00>		

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

QIC to Backport Interface: System Control and Timing

Pin	Name	sense†	Description
32	TEST	I Lo	Test mode. Tied inactive high.
45	CLOCK	I	20MHz TTL clock, shared with QNA2.
53	UQSTRAP	I Hi	Tied inactive low.

QIC to Backport Interface: Memory Access

47	MREQ	O Hi	Memory Request, sent to QNA2 during backport DMA to request access to shared buffer RAM, and negated one clock pulse after MACK is received in response.
46	MACK	I Hi	Memory Acknowledge received in response to MREQ during backport DMA. This enables both the DMA address onto the backport bus, BPDAL<15:00>, and also BPAS, BPRDWR, and BPRPLY. This signal must be synchronous to the 20MHz clock.

QIC to Backport Interface: Backport Bus Control

Pin	Name	sense†	Description
49	BPAS	I/O Lo	Backport Address Strobe (tristate, with pullup resistor) indicates that a valid address on BPDAL<15:00> can be latched on the asserting edge.
51	BPRDWR	I/O Hi LO	Backport Read/Write (tristate signal) Lo indicates the direction of transfer. BPRDWR is asserted through most of the transfer cycle. In Read operations, BPRDWR is negated after the QIC latches the data, so that the backport master can start reading.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

QIC to Backport Interface: DMA Control

Pin	Name	sense†	Description
48	BPRPLY	I/O Hi	Tied high.
52	DMARDY	I Lo	This pin is used to indicate to the QIC that backport transfers for DATA DMA should not be attempted, when negated (high). Host I/O page accesses, vector fetches and control DMA are unaffected by the state of this pin.
73	TRANS	I/O Hi	Not used. Tied low.

QIC to Backport Interface: Interrupt Control

Pin	Name	sense†	Description
50	ATTN	O Lo	Attention asserts an interrupt to the 68000, in order to indicate an error or a completion.
51	RESET	O Lo	Not used.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

A.4 QNA2 Q-BUS NETWORK ARBITRATOR

A.4.1 OVERVIEW

The QNA2 is packaged in a 68-pin, surface-mounted plastic leaded chip-carrier (plcc). The device is created with an LSI logic 2220-cell gate array.

The QNA2 controls access to the multiport memory system on the a backport bus. It arbitrates between requests from the QIC, the LANCE, and the 68000 microprocessor, granting access rights to these three devices on a shared time basis.

Access is granted according to the following order of priority:

1. QIC
2. 68000 microprocessor
3. LANCE.

The QNA2 implements the following control functions:

- Arbitration of access rights for QIC, 68000 microprocessor and LANCE
- Read/Write control logic for RAM and ROM accesses
- Read/Write control logic for QIC registers and LANCE registers
- Read/Write control logic for QNA2 registers
- LANCE and QIC DMA control
- 68000 control.

The architecture of the QNA2 is shown in the block diagram.

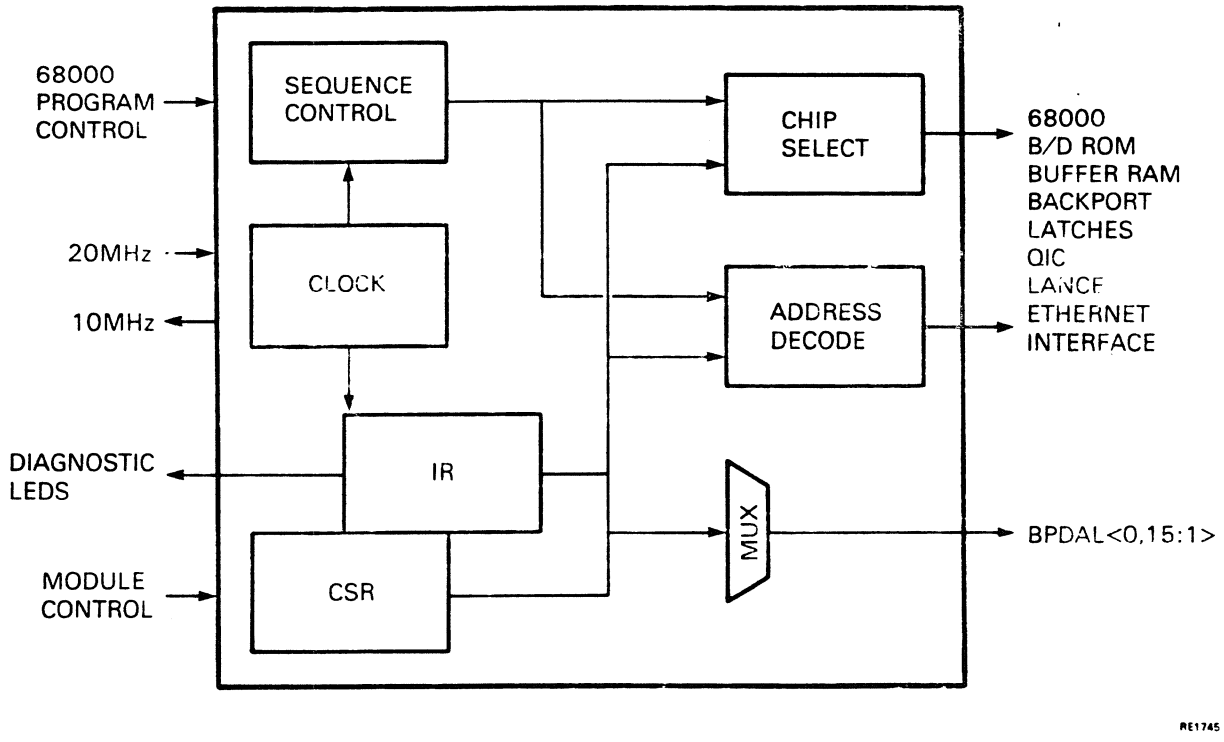


Figure A–5 QNA2 Block Diagram

A.4.2 SIGNALS AND PINOUT

The QNA2 controls all the gating on the DESQA board. It employs combinational logic to examine the access requests asserted on the backport bus, and then executes the appropriate control function for the request with the highest priority.

The three control sequences implemented by the QNA2 are:

1. QIC DMA transfers:
   a. Read/Write access to the QNA2 CSR.
   b. Read/Write access to shared RAM.
   c. Read only access to Station Address ROM.
2. 68000 microprocessor:
   a. Read/Write access to shared RAM.
   b. Read/Write access to QIC and QNA2 registers.
   c. Read/Write access to the LANCE.
   d. Read access to firmware ROM.
   e. Read access to Station address ROM.
3. LANCE DMA transfers:
   a. Read/Write access to shared RAM.

A.4.2.1 QNA2 Signal Descriptions

Table A–3 QNA2 Signal Descriptions

Backport Interface: Address and Data Bus

Pin	Name	Sense†	Description
02:04	BPDAL<13:15>	I/O Hi	Backport Data/Address lines, used as follows:
47:50	BPDAL<00:03>		BPDAL<15:00> data word transfers;
53:56	BPDAL<08:11>		BPDAL<07:00> control byte transfers;
63:67	BPDAL<04:07,12>		BPDAL<00,15:01> Backport DMA.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

Backport Interface: QIC

Pin	Name		Sense†	Description
05	MREQ	I	Hi	QIC request for access to RAM or to QNA2 registers. No synchronisation needed since QIC and QNA2 share a 20MHz clock.
06	MACK	O	Hi	Acknowledge MREQ.
07	BPRDWR	I/O	Hi/Lo	Backport Read/Write indicates direction of data transfer. Hi-to-Lo transition terminates a slave read access by enabling the tristate gates.
08	BPAS	O	Lo	Backport Strobe latches a QIC register address for a slave read or write.

Module Control and Timing

Pin	Name		Sense†	Description
09	CLOCK	I	Hi	20 MHz clock (shared with QIC).
13	INIT	I	Lo	Initialise pulse clears all register bits and output functions, and sets the control sequencer to idle.
16	JMPR3	I	Lo	Select access time for 68000 ROM: Hi = 150ns.; Lo = 250ns. Preset to Lo. If Hi, DTACK is generated 100 ns. earlier.
21	JMPR2	I	Hi	Select Option, connected to option switch S4 of the 5-way switchpack, and reflected in IR bit 10. For DEQNA-Lock mode (mode switch S3=0): select sanity timer. For Normal mode (mode switch S3=1): select MOP Remote Boot.
22	JMPR1	I	Hi	Select Mode, connected to switch S3 of the 5-way switchpack, and reflected in IR bit 5. S3=1=Normal mode; S3=0=DEQNA-Lock mode.
32	OINIT	O	Lo	The state of this pin is the inversion of CSRbit 1 (SR)

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

Diagnostic Signals

Pin	Name		Sense†	Description
10	LED3	O	Lo	Diagnostic output to LED2 driver.
11	LED2	O	Lo	Diagnostic output to LED1 driver.
12	LED1	O	Lo	Diagnostic output to LED0 driver.
20	ELOOP	O	Hi	Ethernet Loopback, reflects CSR bit 9 and selects the MSB of Station Address ROM.

Ethernet Interface

Pin	Name		Sense†	Description
14	RENA	I	Hi	Receive Enable signal from the SIA.
15	FUSE	I	Hi	Power indicator from the BULKHEAD FUSE. This signal is fed directly to CSR bit 12.

Backport Interface: LANCE

Pin	Name		Sense†	Description
19	LADR	O	Hi	LANCE Address Select, (re)set each time ADS (Address Strobe) is asserted with ADR21 (un)set.
23	LREAD	I/O	Hi/Lo	LANCE Read indicates the direction of transfer (Hi=read: Lo=write) for both slave and DMA accesses.
24	LDAS	I/O	Lo/Hi	LANCE Data Strobe to/from the LANCE,asserted by:. a. 68000-to-LANCE read/write control sequence. b. LANCE DMA read/write cycle.(Pullup resistor).
25	LHLDA	O	Lo	LANCE Hold Acknowledge, asserted in response to LHOLD only when there is no valid slave access by the 68000.
26	LHOLD	I	Lo	LANCE DMA Request. (Pullup resistor).

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

Pin	Name		Sense†	Description
36	LREADY	I/O	Lo/Hi	LANCE DMA Ready, used as follows: Lo - the QNA2 control sequence responds to a LANCE DMA cycle; Hi - the LANCE indicates completion of 68000 slave accesses.
37	LCS	O	Lo	LANCE Chip Select, asserted for each complete 68000 read/write cycle.
43	LEN	O	Lo	LANCE Data Latch Enable (backport bus).
44	LCAB	O	Hi	LANCE Data Latch from backport bus, allowing the LANCE control sequencer to terminate in its own time when: a. 68000 writes to the LANCE; b. LANCE DMA reads RAM.
45	LDIR	O	Hi	LANCE Data Latch Direction, presented before LENL to avoid tristate overlaps.
57	LAOE	O	Lo	LANCE Address Output enables the address latches on DMA cycles. (The address is latched by LALE from the LANCE).

Backport Interface: 68000

Pin	Name		Sense†	Description
27	IPL2	O	Lo	Interrupt Level 2, asserted to 68000 as the result of a host write to the CSR, VAR, TXBDL, RXBDL or after the QNA2 timer has expired.
28	R/W	I	Hi/Lo	Read/Write from 68000.
29	ADS	I	Lo	Address Strobe from
30	CPCLK	O	Hi	10MHz clock for the 68000 microprocessor.
31	VPA	I	Lo	Valid Peripheral Access, the result of a NAND operation on FC0+FC1+FC2 and the inverse of ASM, indicating an active interrupt acknowledge cycle. This inhibits 68000 access cycles.
33	DTACK	O	Lo	Data Transfer Acknowledge to 68000.
38	A21TES	I/O	Hi/Lo	Address Bit 21/Test, used with ADS to clock in BPDAL21 and assert LADR.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

Pin	Name		Sense†	Description
39:40	ADR<22:23>	I	Hi	Address Bits <23:22> indicate to the control sequencer the device addressed: 00 68000 to ROM access 01 68000 to RAM access 10 68000 to QIC access 11 68000 to LANCE access
46	CPDIR	O	Hi	Control Port Direction for the 68000 data transceivers.
58	CPDOE	O	Lo	Control Port Data Output Enable for the 68000 data transceivers.
59	CPAOE	O	Lo	Control Port Address Output Enable for the 68000 address latch.

Backport Interface: Memory

Pin	Name		Sense†	Description
41	SELROM	O	Lo	Select 68000 Firmware ROM.
42	SELSAR	O	Lo	Select Station Address ROM.
61	MWE	O	Lo	Memory Write Enable.
62	MCSL	O	Lo	Memory Chip Select.

Backport Interface: Backport Bus Control

Pin	Name		Sense†	Description
60	MALE	O	Lo	Memory Address Latch selects the two F373 latches on the backport bus. The associated DMA cycle may be initiated by 68000 or QIC.

Power lines

Pin	Name	Description
01,18,35,52	Vss	Signal ground.
17,34,51,68	Vcc	+5V Power rail.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

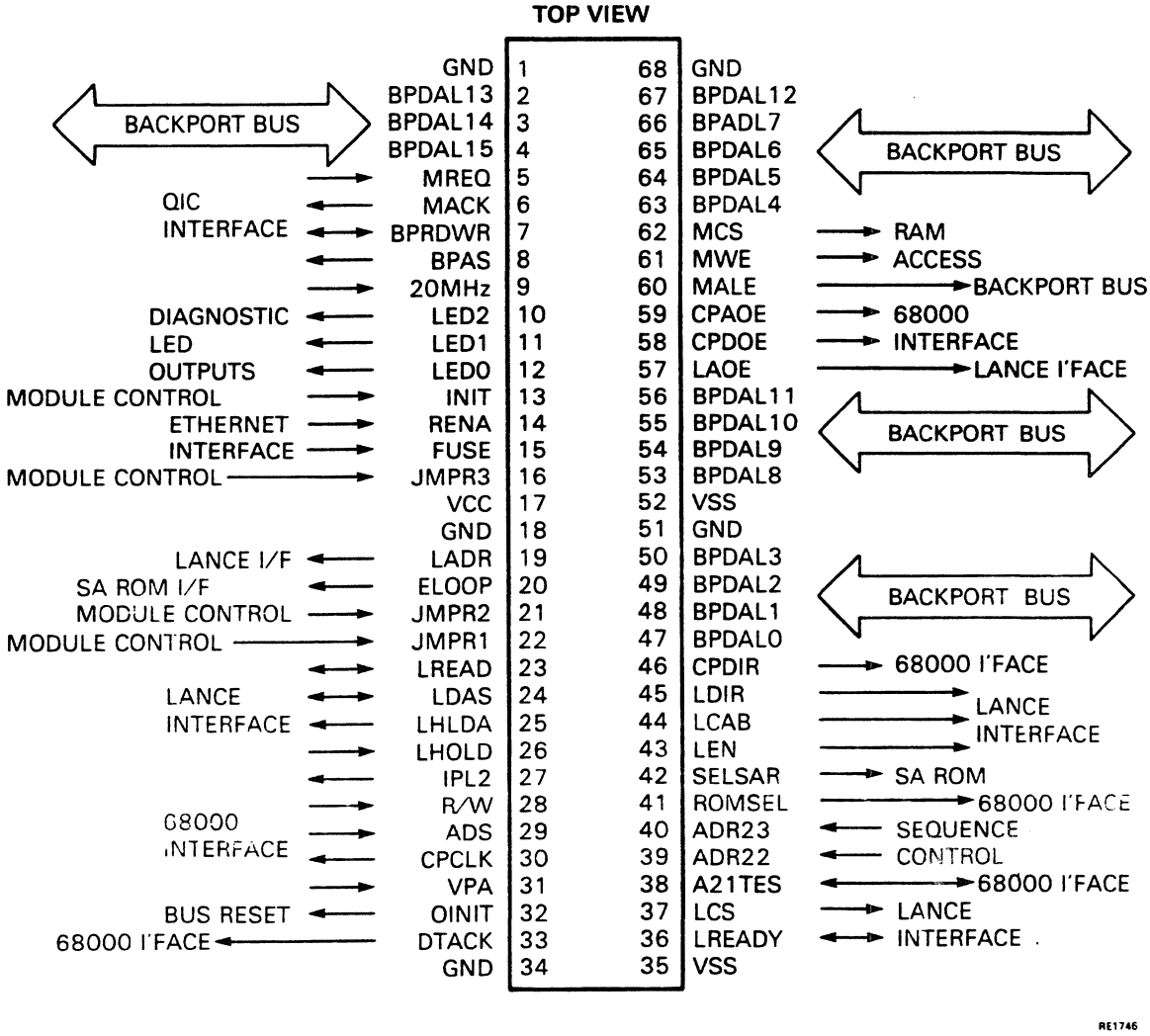


Figure A–6 QNA2 Pinout

A.4.2.3 Addressing

Table A-4 lists the QNA2 registers together with their backport memory map addresses.

Table A–4 QNA2 Registers

Backport Address	Register
40FF0E	Control and status Register (CSR)
40FF1E	Interrrupt Register (IR)

A.5 AM7990 LOCAL AREA NETWORK CONTROLLER FOR ETHERNET (LANCE)

A.5.1 Overview

The LANCE operates with the SIA (from the same family of LSI devices) to provide a complete interface between the backport bus and the Ethernet cable. The LANCE is a 10Mbit/sec MOS device in a 48-pin package. It implements the following facilities for the link level of the Ethernet protocol:

- CSMA/CD network access
- Memory management using on-board DMA
- Error reporting
- Packet handling
- Microprocessor interface.

In transmit mode, the LANCE initiates a DMA cycle to transfer data from shared RAM. It prefaces the data with the standard preamble and synchronisation pattern, and it calculates and appends a 32-bit CRC.

In receive mode, the SIA asserts Carrier Present to the LANCE, in order to indicate that two signals are available: Receive Data and Receive Clock. The LANCE uses Receive Clock to clock in the Receive Data signals. Then it calculates and compares the CRC to the CRC checksum at the end of the packet. If the two values do not match, the LANCE sets an error bit and flags an interrupt to the 68000 microprocessor.

There are three addressing modes:

- Physical addressing, in which a 48-bit destination address at the front of an incoming packet is compared with the node address written to the LANCE at initialisation
- Multi-cast addressing, in which the LANCE accepts all incoming packets for a certain class of node
- Promiscuous operation, in which the LANCE accepts all incoming packets.

In its buffers, the LANCE sets up circular task queues known as descriptor rings. The tasks are used for transmit and receive operations; up to 128 tasks may be queued for execution.

The architecture of the QIC is shown in Figure A–7.

A.5.2 Signals and Pinout

The signals to and from each QIC port fall into logical groups. The functions, signals and pinouts of these groups are shown in Figure A–8 and described in Table A–5.

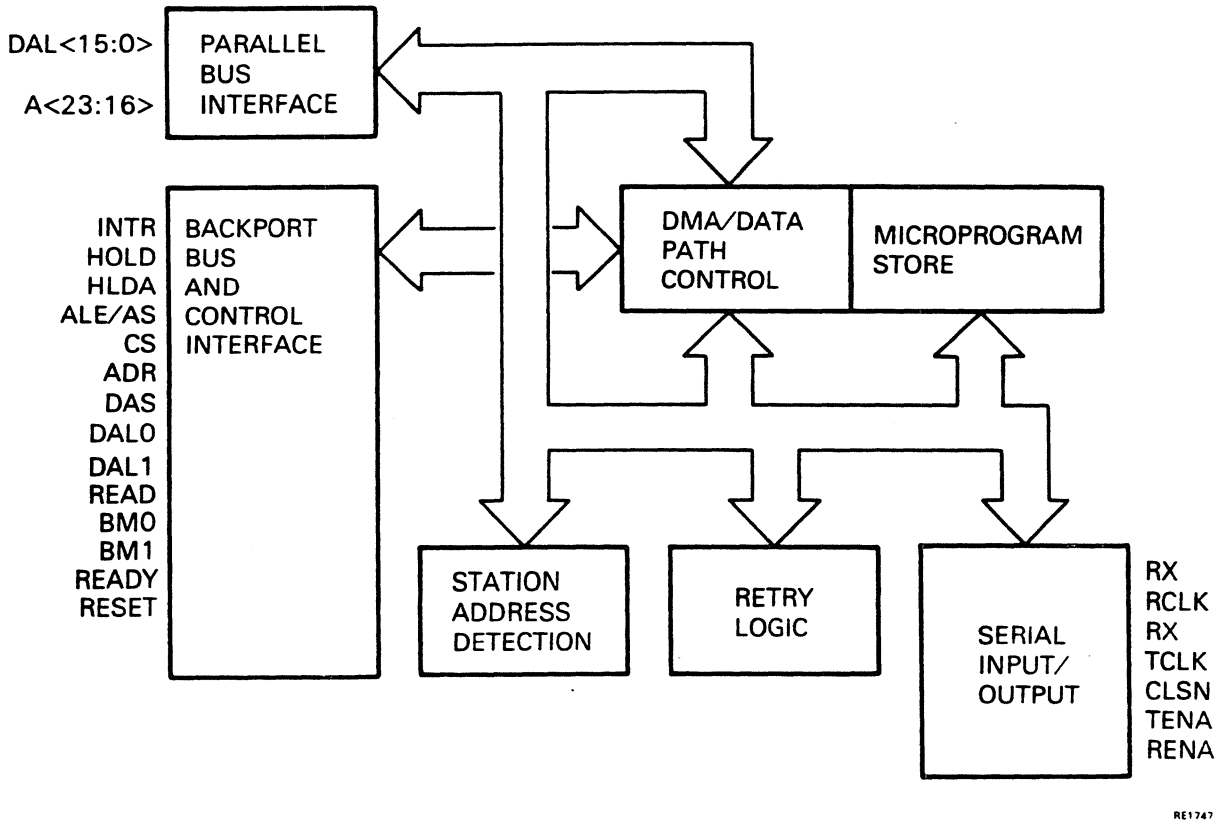


Figure A–7 AM7990 LANCE Block Diagram

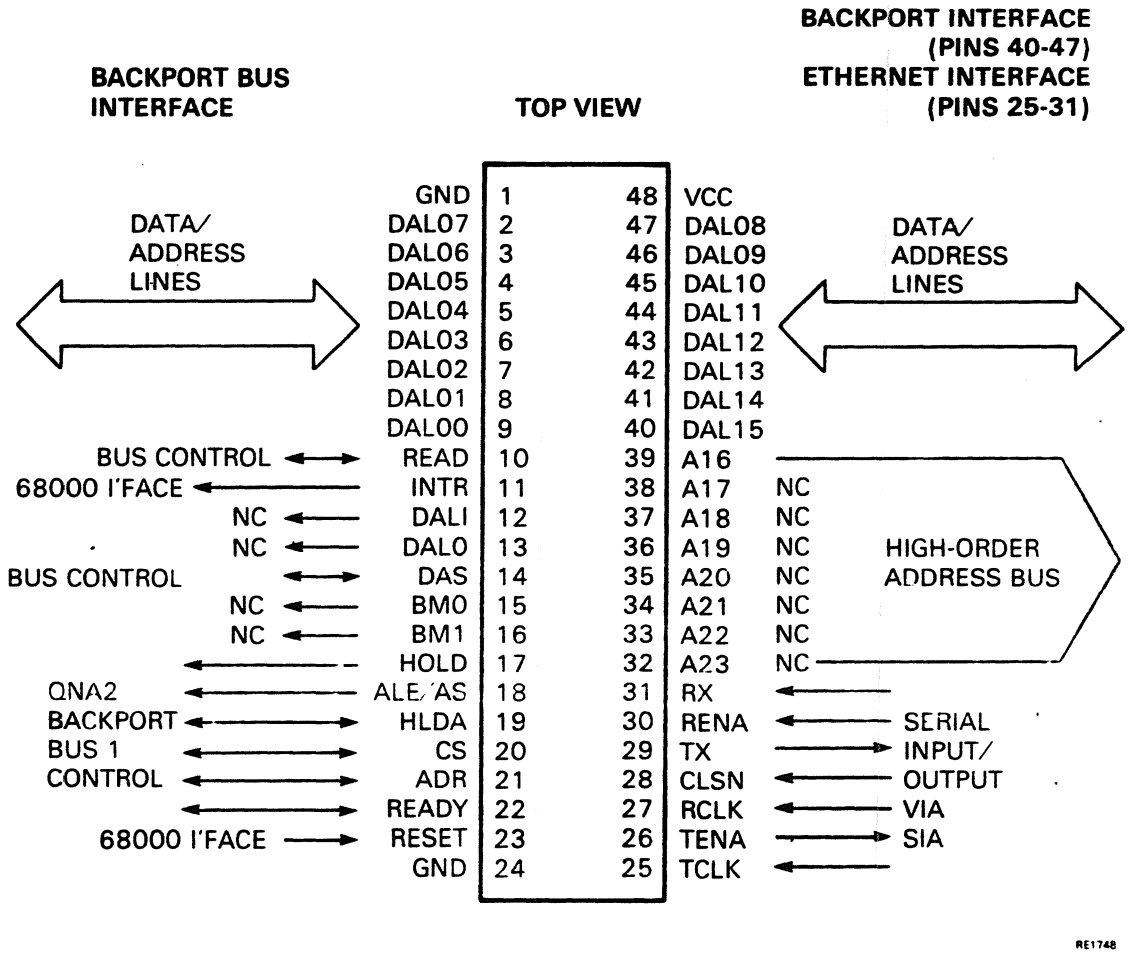


Figure A–8 AM7990 LANCE Pinouts

Table A–5 AM7990 LANCE Signal Definitions

Backport Interface: Address and Data Bus

Pin	Name	Sense†	Description
02:09	DAL<07:00>	I/O Hi	Data/Address lines (tristate). During
40:47	DAL<15:08>		address select, A<23:16> contains the
32:39	A<23:16>	O Hi	upper 16 address bits.

Backport Interface: Backport Bus Control

Pin	Name	Sense†	Description
10	READ	I/O Hi	Read indicates the direction of data transfer. Reading when the LANCE is bus master; writing when LANCE is bus slave.
12	DALI	O Lo	Data/Address Line In (tristate).
13	DALO	O Hi	Data/Address Line Out (tristate). (Not used).
14	DAS	I/O Lo	Data Strobe (tristate).
15	BM0/BYTE	O Lo/Hi	Byte Mask/Byte.
16	BM1/BUSAKO	O Lo/Lo	Byte Mask/Bus Request Daisy Chain Output.
			These tristate signals are programmable through bit 0 of CSR3. (No Connection.)
17	HOLD/BUSRQ	O Lo/Lo	Bus Hold Request (open drain). This signal is programmable through bit 0 of CSR3 (Pullup resistor).
18	ALE/AS	O Lo/Hi	Address Latch Enable. This tristate signal is programmable through bit 1 of CSR3 (Pullup resistor).
19	HLDA	I Lo	Bus Hold Acknowledge, asserted in response to HOLD to make the LANCE bus master.
20	CS	I Lo	Chip Select puts the LANCE into bus slave mode for a data transfer to or from the LANCE registers. This signal must not be asserted with HLDA.
21	ADR	I Hi	Register Address Port Select.
22	READY	I/O Lo	Data Transfer Ready (open drain).

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

SIA Interface: Ethernet I/O

Pin	Name	sense†	Description
10	READ	I/O Hi	Read indicates the direction of data.
25	TCLK	I Hi	Transmit Clock (10MHz).
26	TENA	O Hi	Transmit Enable.
27	RCLK	I Hi	Receive Clock (10MHz square wave).
28	CLSN	I Hi	Collision (logical input).
29	Tx	O Hi	Transmit output bit stream.
30	RENA	I Hi	Receive Enable (logical input) indicates that a carrier is present on the Ethernet channel.
31	Rx	I Hi	Receive input bit stream (input).

SIA Interface: 68000 Interrupts Control

Pin	Name	sense†	Description
11	INTR	O Lo	Interrupt (open drain; pullup resistor).
23	RESET	I Lo	Bus Reset Request causes LANCE to stop, clear its internal logic, and idle.

Power lines

Pin	Name	Description
01	Vss	Signal ground.
48	Vcc	+5V Power rail.

† The Sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

A.5.3 LANCE Addressing

The LANCE has four Control and Status Registers, which are programmed from the 68000 microprocessor through two ports on the backport bus: the Register Data Port (RDP), and the Register Address Port (RAP).

To access the registers, the QNA2 first asserts CS to put the LANCE into bus slave mode. The 68000 then writes a CSR select code to the LANCE, which stores the code in its Register Address Port. Subsequent read/write accesses affect only the CSR selected.

A.5.4 LANCE Registers

Table A–6 lists the LANCE registers.

Table A–6 LANCE Registers

Register		Description
RAP	Register Address Port	Read/Write.
RDP	Register Data Port	Read/Write.
CSR0	Control and Status Register 0	Transmit/Receive error flags.
CSR1	Control and Status Register 1	Initialisation Block address.
CSR2	Control and Status Register 2	Initialisation Block address.
CSR3	Control and Status Register 3	Defines bus master interface.

A.5.5 LANCE Buffers

Table A–7 lists the contents of the Initialisation Block. Table A–8 lists the contents of the Descriptor Rings.

Table A–7 LANCE Initialisation Block

Bit/Name	Description
IADR + 0	Mode Register.
15 PROM	Promiscuous addressing mode.
14:07	Reserved.
06 INTL	Internal Loopback.
05 DRTY	Disable Retry on transmission.
04 COLL	Force Collision (diagnostic).
03 DTCR	Disable Transmit CRC (enables Multi-cast addressing in external loopback).
02 LOOP	Enable Loopback.
01 DTX	Disable Transmitter.
00 DRX	Disable Receiver.
IADR + 02	Physical Address (PA) .
47:00 PADR	Physical Address of LANCE .
IADR + 08	Logical Address Filter (LAF) .
63:00 LADRF	64-bit logical address mask.

Table A–8 LANCE Descriptor Ring Pointers

Bit/Name	Description
IADR + 18	Receive Descriptor Ring Pointer.
15:13 RLEN	Receive Ring Length, expressed as power of two.
12:08	Reserved.
07:00 RDRA	Receive Descriptor Ring (base) Address.
IADR + 20	Transmit Descriptor Ring Pointer.
15:13 TLEN	Transmit Ring Length, expressed as power of two.
12:08	Reserved.
07:00 TDRA	Transmit Descriptor Ring (base) Address.

Table A–9 LANCE Four-Word Descriptor Ring Formats

Receive Message Descriptor 0 (RMD0)

Bit/Name	Description
15:00 LADR	Low order 16 address bits of buffer described. Written by the host and unchanged.

Receive Message Descriptor 1 (RMD1)

Bit/Name	Description
15 OWN	Descriptor owned by LANCE (not host).
14 ERR	Error = OR operation on FRAM, OFLO, CRC, BUFF.
13 FRAM	Framing Error (not just CRC error).
12 OFLO	Overflow (internal silo).
11 CRC	CRC Error.
10 BUFF	Buffer Error (OWN bit zero, or silo overflow).
09 STP	Start of Packet.
08 ENP	End of Packet.
07:00 HADR	High order 8 address bits of buffer described.

Receive Message Descriptor 2 (RMD2)

15:12	Must be ones
11:00 BCNT	Buffer Byte Count (2's complement), written by host and unchanged.

Receive Message Descriptor 3 (RMD3)

Bit/Name	Description
15:12	Reserved.
11:00 MCNT	Message Byte Count (in BCD), when ERR is clear and ENP is set. Written by host and unchanged.

Transmit Message Descriptor 0 (TMD0)

Bit/Name	Description
15:00 LADR	Low order 16 address bits of buffer described. Written by the host and unchanged.

Table A–9 (Cont.) LANCE Four-Word Descriptor Ring Formats

Bit/Name	Description
Transmit Message Descriptor 1 (TMD1)	
Bit/Name	**Description**
15 OWN	Descriptor owned by LANCE (not host).
14 ERR	Error = OR operation on LCOL, LCAR, UFLO, RTRY.
13	Reserved .
12 MORE	More than one retry needed.
11 ONE	Exactly one retry needed.
10 DEF	Transmit Deferred, due to busy channel.
09 STP	Start of Packet.
08 ENP	End of Packet.
07:00 HADR	High order 8 address bits of buffer described.
Transmit Message Descriptor 2 (TMD2)	
Bit/Name	**Description**
15:12	Must be ones.
11:00 BCNT	Buffer Byte Count (2's complement), written by host and unchanged.
Transmit Message Descriptor 3 (TMD3)	
15 BUFF	Buffer Error (OWN bit zero, or silo underflow).
14 UFLO	Underflow error. Message truncated due to late data from memory.
13	Reserved.
12 LCOL	Late Collision (channel slot time elapsed).
11 LCAR	Loss of Carrier (RENA negated during transmission).
10 RTRY	Retry Error. after either one or 16 attempts.
09:00 TDR	Time Domain Reflectometry (valid for RTRY).

A.5.6 Logical Address Filter

The LANCE maps each incoming 48-bit logical address from the Ethernet to the 64-bit logical address mask (stored at IADR+08 in the Initialisation Block).

If the first bit of the incoming physical address is set (PADR<0>=1), the address is passed to the logical address filter, which contains four 16-bit registers. All 48-bits of the incoming address are passed through the CRC circuit. The six high-order bits of the resultant 32-bit CRC are strobed into a register, where they are decoded to select a single bit position in the LAF. If that bit in the LAF is set, the incoming address is accepted.

This provisional assessment is then checked by comparing the full incoming logical address with the list of valid logical addresses that is stored in shared RAM.

The broadcast address does not go through this decoding process.

Table A–10 shows how LAF bit positions correspond to logical addresses.

Table A–10 LANCE Logical Address Filter Mask

Logical Filter Register	LAF bit	Destination Address (hex)	Logical Filter Register	LAF bit	Destination Address (hex)
LAF0	0	85	LAF2	32	21
LAF0	1	A5	LAF2	33	01
LAF0	2	E5	LAF2	34	41
LAF0	3	C5	LAF2	35	71
LAF0	4	45	LAF2	36	E1
LAF0	5	65	LAF2	37	C1
LAF0	6	25	LAF2	38	81
LAF0	7	05	LAF2	39	A1
LAF0	8	2B	LAF2	40	8F
LAF0	9	0B	LAF2	41	BF
LAF0	10	4B	LAF2	42	EF
LAF0	11	6B	LAF2	43	CF
LAF0	12	EB	LAF2	44	4F
LAF0	13	CB	LAF2	45	6F
LAF0	14	8B	LAF2	46	2F
LAF0	15	BB	LAF2	47	0F
LAF0	16	C7	LAF2	48	63
LAF0	17	E7	LAF2	49	43

Table A–10 (Cont.) LANCE Logical Address Filter Mask

Logical Filter Register	LAF bit	Destination Address (hex)	Logical Filter Register	LAF bit	Destination Address (hex)
LAF0	18	A7	LAF2	50	03
LAF0	19	87	LAF2	51	23
LAF0	20	07	LAF2	52	A3
LAF0	21	27	LAF2	53	83
LAF0	22	67	LAF2	54	C3
LAF0	23	47	LAF2	55	E3
LAF0	24	69	LAF2	56	CD
LAF0	25	49	LAF2	57	ED
LAF0	26	09	LAF2	58	AD
LAF0	27	29	LAF2	59	8D
LAF0	28	A9	LAF2	60	0D
LAF0	29	89	LAF2	61	2D
LAF0	30	C9	LAF2	62	6D
LAF0	31	E9	LAF2	63	4D

A.6 AM7991 SERIAL INTERFACE ADAPTER (SIA)

A.6.1 Overview

The SIA is a Manchester encoder/decoder which interfaces between the LANCE TTL logic and the Ethernet transceiver. It is a 24-pin package, and is located close to the Ethernet connector.

The SIA acquires clock and data within six bit-times, and decodes Manchester data with less than +/-20ns phase jitter at 10MHz. In order to minimise false start conditions, the SIA provides both guaranteed signal threshold limits and transient noise suppression circuitry in both data and collision paths.

The architecture of the SIA is shown in the block diagram Figure A-9.

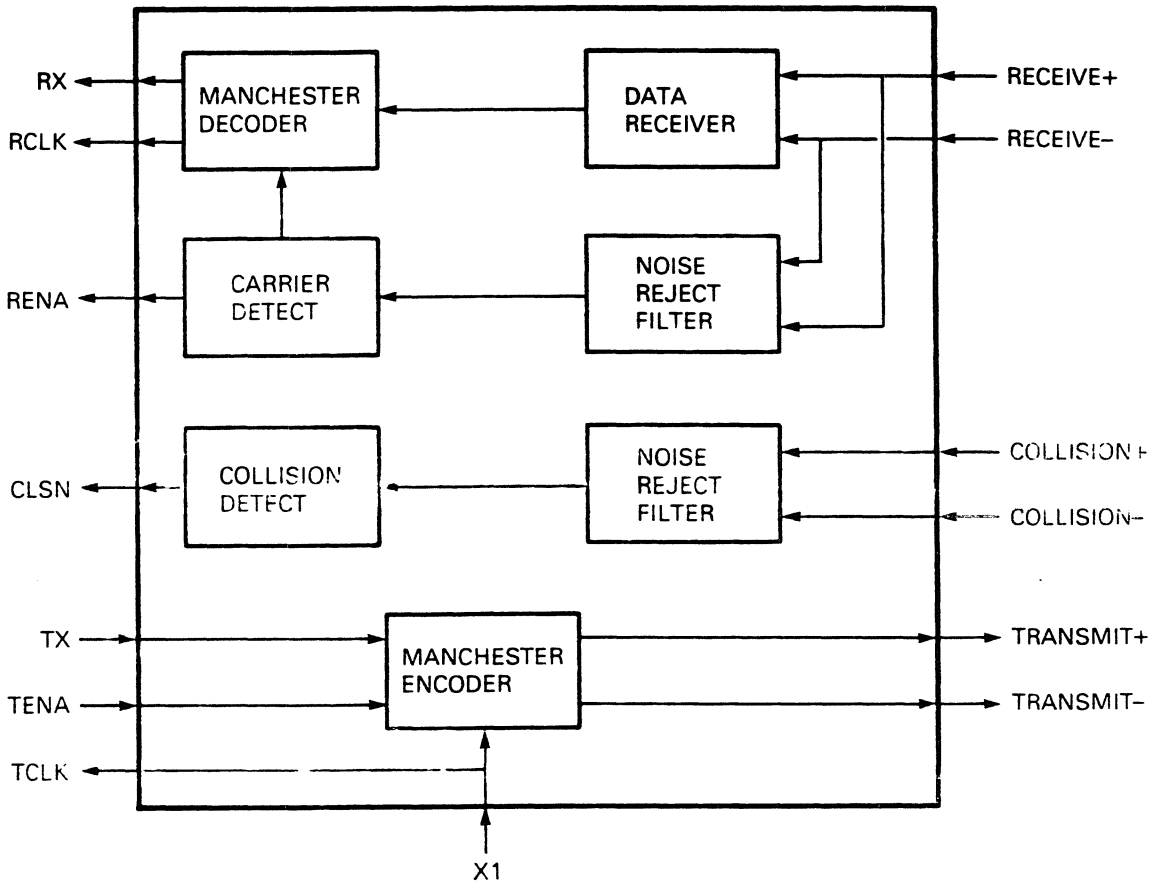


Figure A–9 AM7991 SIA Block Diagram

A.6.2 Signal Description

The signals to and from the SIA fall into two logical groups. The functions, signals and pinouts of these groups are shown in Figure A–10 and described in Table A–11.

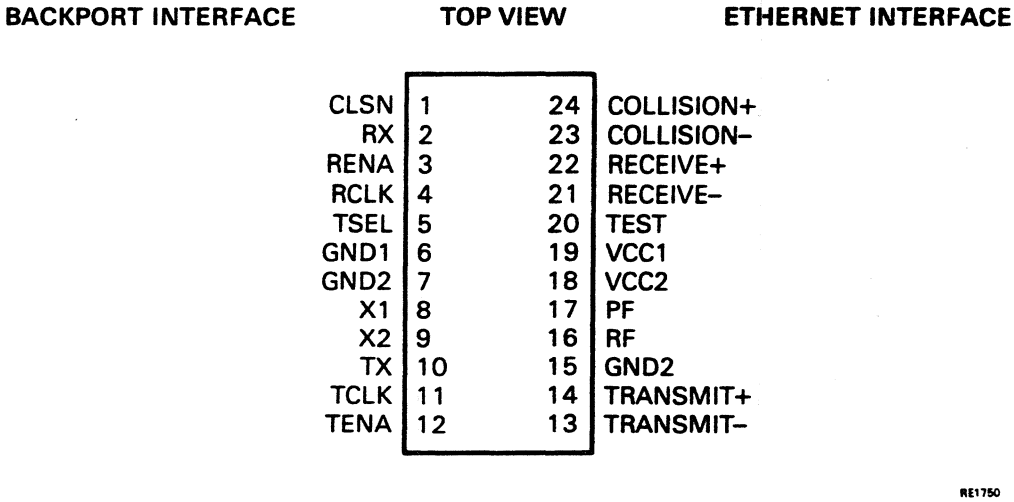


Figure A–10 AM7991 SIA Pinouts

Table A–11 AM7991 SIA Signal Definitions

Backport Interface

Pin	Name	sense†	Description
01	CLSN	O Hi	Collision
02	Rx	O Hi	Receive Data (TTL output)
03	RENA	O Hi	Receive Enable (TTL output)
04	RCLK	O Hi	Receive Clock (TTL output)
05	TSEL	I/O Hi	Transmit Mode Select (open collector output/sense amplifier input). When the transmitter is idle: TSEL low selects Transmit + positive with respect to Transmit - , TSEL high selects Transmit + equal to Transmit -
06	Gnd1		High current ground
07	Gnd2		Logic ground
08	X1	I Hi	Biased crystal oscillator, used to generate TCLK
09	X2		Biased crystal oscillator, used to generate TCLK
10	Tx	I Hi	Transmit (TTL compatible)
11	TCLK	O Hi	Transmit Clock (TTL; symmetrical). This is also input to the LANCE
12	TENA	I Hi	Transmit Enable (TTL compatible)
15	Gnd3		Ground for voltage controlled oscillator
16	RF	O	Receive Frequency. A reference voltage for the receive path phase detector and and timing noise immunity circuits
17	PF	I	Phase Filter control voltage for phase lock loop damping (receive path)
18	Vcc2		+5V Power rail for voltage controlled oscillator in phase lock loop
19	Vcc1		High current and logic supply
20	TEST	I Lo	Test control

† The sense column is used to show if a signal in an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

Table A–11 (Cont.) AM7991 SIA Signal Definitions

Pin	Name	sense†	Description
Ethernet Interface			
13	Transmit -	O	Transmit. Differential line outputs into terminated transmission lines
14	Transmit +	O	Transmit. Differential line outputs into terminated transmission lines
21	Receive -	I	Differential line inputs
22	Receive +	I	Differential line inputs
23	Collision +	I	Differential collision inputs, which do not affect data path functions.
24	Collision -	I	Differential collision inputs, which do not affect data path functions.

† The sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

A.6.3 Noise Margins

On input the static noise margin for received carrier detection is -175mV to -275mV. Transient noise of less than 10nsec. in the collision path, and 16nsec. in the data path, are also rejected. The SIA will decode jittered data of up to +/-20nsec.

A.7 THINWIRE ETHERNET TRANSCEIVER

A.7.1 Functionality

The Thinwire Ethernet Transceiver is made up of a DP8392 transceiver chip, an isolation transformer and a DC-DC converter. The transformer isolates the Thinwire Ethernet Transceiver from the rest of the DESQA circuitry. The -9 Vdc needed for the DP8392 chip, is converted by the DC-DC converter from +12 Vdc which is supplied by the host. A functional block diagram of this transceiver component set is shown in Figure A–11.

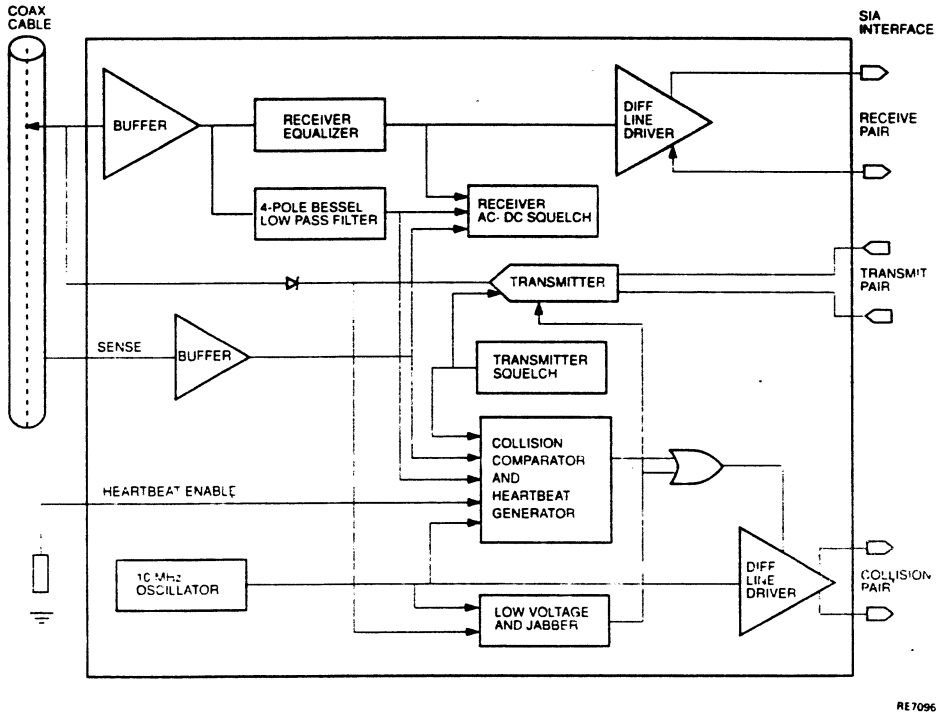


Figure A–11 Functional Block Diagram of Thinwire Ethernet Transceiver

The DP8392 chip block diagram is shown in Figure A–12

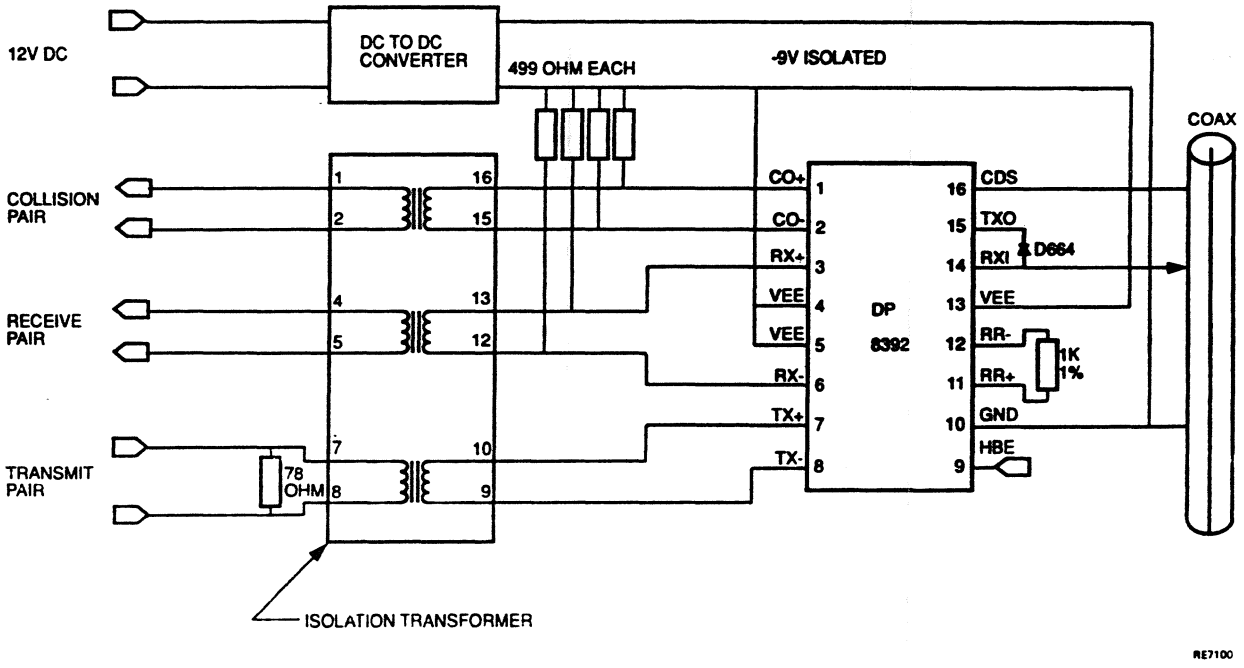


Figure A–12 DP8392 Block Diagram

Pin	Name	sense†	Description
01	CD+	0	Balanced differential line driver outputs from Collision Detect Circuitry.
02	CD-	0	The 10MHz signal from the internal oscillator is gated to these outputs in the event of collision, excessive transmission (jabber), or during SQE test. These outputs are open emitter; pulldown resistors to VEE are required. When drawing a 78 ohm transmission line, these resistors should be 500 ohm.
03	RX+	0	Balanced differencial line driver outputs from the receiver.
06	RX-	0	These outputs also require 500 ohm pulldown resistors.
07	TX+	I	Balanced differencial line receiver inputs to the transmitter.
08	TX-	I	
04	VEE		Supply connection. VEE is negative with respect to ground.
05	VEE		
13	VEE		
10	GND		Ground connection.
09	HBE	I	This input enables CD Heartbeat when at VIH, disables it when at VIL.
14	RXI	I	Receiver input from the coaxial cable. Signals meeting receiver squelch requirements are equalized for intersymbol distortion, amplified, and outputted at RX+.
15	TXO	O	Transmit output; connects to coaxial cable.
16	CDS	I	Ground sense connection for collision detect circuitry. This pin should be connected seperately to the shield to avoid voltage drops through the coax shield from altering the receive mode collision threshold.

†The sense column is used to show if a signal is an input I or output O from the chip. Logic High (Hi) or Low (Lo) states are shown for signal assertion.

APPENDIX B
NETWORK MANAGEMENT

B.1 SCOPE

This appendix outlines the Ethernet Network Management protocol.

B.2 NETWORK CONTROL

To control access, Ethernet uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD). This is a distributed channel allocation procedure under which every station receives all the transmissions from every other station. Each station can detect overlapping transmissions by other stations, and must wait for the bus to become idle before transmitting its message. There is no central or hierarchical control for this distributed process.

B.2.1 Ethernet Transmission

The data link layer checks that the network bus is clear of any baseband signal before it starts to transmit. When the channel is clear, the data link layer passes the data packet to the physical layer as a stream of bits. The physical layer precedes the data with an encoded preamble that allows the other nodes on the channel to synchronise their clocks. Then the physical layer starts to translate the binary encoded data into Manchester-phase-encoded signals.

Throughout the transmission, the physical layer monitors the correct energy level for transmission without contention in the channel. If the energy level exceeds this, then the physical layer sends a collision detect signal to the data link layer.

The collision detect signal can only be detected by a node that is transmitting.

B.2.2 Ethernet Reception

In the absence of other traffic on the channel, the physical layers of all other nodes in the network sense the carrier signal from a transmitting node. They alert their respective data link layers with a carrier sense signal, in order to delay any conflicting attempts to transmit.

The physical layer of each receiving node synchronises to the incoming preamble, receives the encoded bits from the cable, and translates the phase-encoded signal to binary encoded data, discarding the preamble.

At each of the receiving nodes, the binary bits are passed to the data link layer, which has been alerted to its arrival by the carrier sense signal. As the bit stream arrives, the data link layer checks the destination address. If the data is intended for that node, the packet is inspected for damage and alignment, before being passed on to higher levels of protocol with an indication of its condition. If the packet is not intended for that node, the bit stream is halted.

The physical layer continues to receive encoded bits from the channel until the carrier goes off.

B.2.3 Ethernet Contention

When two nodes simultaneously determine that the channel is free, and start to transmit, a collision occurs. Ethernet is designed to cater for such collisions by re-transmission following on a collision detect signal.

The Ethernet packet is specified to have a minimum length of 64 bytes because the time it takes to transmit such a packet over the maximum Ethernet configuration is slightly less than the time it takes for a collision detect signal to travel from one end of the network to the other and back again. (The elapse time for a round-trip signal is known as the network **slot time**).

This means that the collision detect signal from a node at the other end of the network has time to reach any originating node before that node has completed transmission of the packet that caused the collision. The originating node can then re-transmit the correct packet.

On receipt of a collision detect signal the data link layer at the transmitting node:

1. Continues to transmit the packet, in order to cause a **jam** that will be noticed by all other nodes.

2. **Backs off** and schedules a re-transmission attempt. The delay before re-transmission is an integral multiple of the slot time. The multiple is selected at random by each node for each collision.

The bits from a collision are known as a **runt packet** because the physical layers at all the receiving nodes accept the bits as though they were from a valid packet. However, the data link layers reject runt packets because they are always shorter than the shortest Ethernet packet.

B.2.4 Ethernet Fault Detection

Data integrity is maintained by the 32-bit CRC for each packet. This is handled by the data link layer protocol. Several network maintenance features are also built into the system:

1. **Time Domain Reflectometry**. Whenever a packet is transmitted, the transmitting node starts a timer and notes the times of all collisions. If the collision times are not random, but predictable and due to some physical fault, then the location of the fault be determined by time/speed/distance calculations.

2. **Time-limited operations**. Both controller and transceiver are designed to time out any transmission that is significantly longer than the Ethernet specification.

3. **Power supply monitor**. The controller monitors the power line direct from the transceiver; and the transceiver provides a positive test signal to the controller after every transmission to verify its operation.

4. **Loopback diagnostics**, both local and remote, are provided.

APPENDIX C
VECTOR ASSIGNMENTS

This appendix lists the rank of vector assignments for Q-Bus systems.

The DESQA 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 DESQA and a floating vector assignment for the second DESQA. 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.

C.1 THE FLOATING VECTOR ASSIGNMENTS

If a host node has a KXV11 and an RXV21 and a DESQA, the DESQA 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 DESQA module is configured as a DMA device, in the same way as a DEQNA module. The first DEQNA/DESQA vector is fixed by host system software at 120(octal). Subsequent DEQNA/DESQA 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.

C.2 FLOATING VECTORS

The DESQA 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).

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

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

Rank	Device	Size (Decimal)	Modulus (Octal)
45	ISB11	4	10
46	DMV11	4	10
47	DEUNA DEQNA/DESQA/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/DHQ11	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.

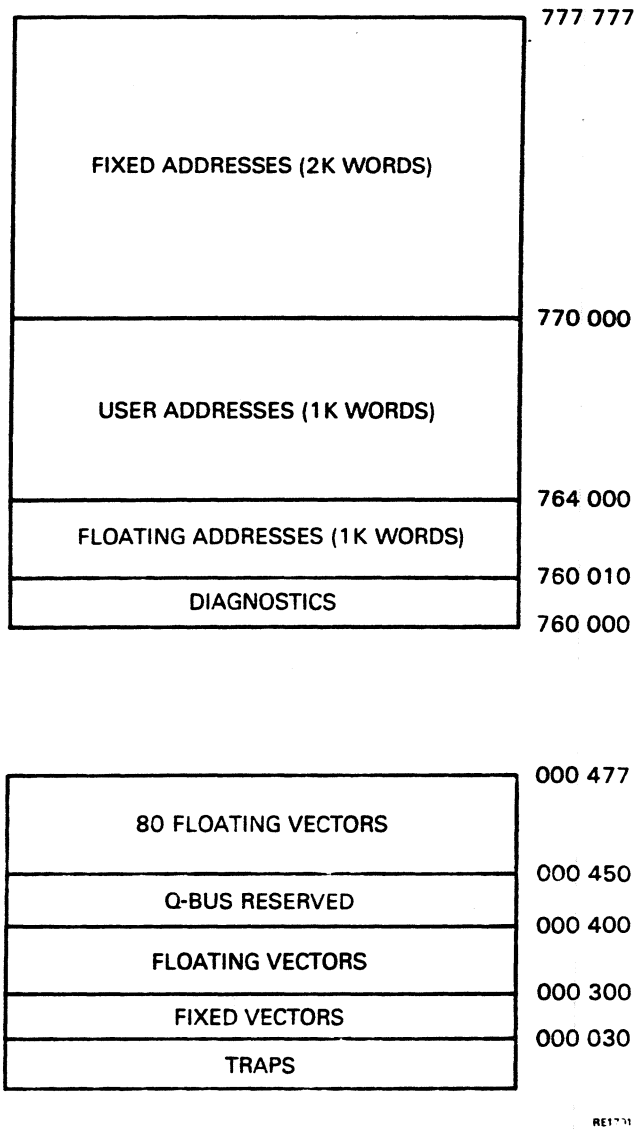


Figure C–1 Q-bus Address Map

APPENDIX D
PROGRAMMING EXAMPLES FOR PDP-11 SYSTEMS

This appendix contains programming examples written in MACRO-11 for the DESQA. They are presented only as a general guide for the prospective user, and not as the best or only method of driving the DESQA. These programs are not guaranteed or supported by Digital Equipment Corporation.

Programming examples are provided for the following.

1. Data definitions
2. Resetting the DESQA
3. Configuring the DESQA
4. A simple interrupt handler
5. Data transmission
6. Data reception
7. Executing on-board diagnostics

D.1 DATA DEFINITIONS

The following data definitions are used throughout the sample programs. Note that all numbers are octal unless otherwise specified.

Table D–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 DESQA installed at I/O page base address 17774440, and the following definitions of I/O port registers apply throughout.

```
 qarll: .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
```

D.2 RESETTING THE DESQA

The DESQA undergoes a software reset when CSR bit 1 is cleared from 1 to 0.

```
;
; *-------------------------------------------------------------*
; *                                                             *
; *         A routine to Software Reset the DESQA.              *
; *                                                             *
; *-------------------------------------------------------------*
;
lqares: bis     #sr,@lqacsr     ; set SR in CSR
        bic     #sr,@lqacsr     ; clear SR to generate sw reset
        rts     pc              ; return to caller
```

D.3 CONFIGURING THE DESQA

The DESQA 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 DESQA 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 DESQA 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 DESQA 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 DESQA 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     #xleis,sttxdl+bdes ; assume an even length packet
        bit     #1,r0              ; IF odd length setup pkt THEN DO
        beq     95$                ;   BEGIN
        inc     r4                 ;   increment 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     #xe,@lqacsr        ; disable interrupts via CSR IE
;
; Validate the Receive and Transmit buffer descriptor lists.
;
        mov     #strxdl,@lqarll ; write Rx BDL addr to the DESQA
        clr     @lqarlh         ; and validate it
        mov     #sttxdl,@lqatll ; write Tx BDL addr to the DESQA
        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
```

D.4 A SIMPLE INTERRUPT HANDLER

```
;
; *-------------------------------------------------------------*
; *                                                             *
; *     Subprogram LQAINT                                       *
; *                                                             *
; *     This subprogram handles DESQA 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
```

D.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 DESQA interrupts disabled.
- Enabling the DESQA to interrupt the host when transmission is complete.

```
;
; *-------------------------------------------------------------*
; *                                                             *
; *      Routine to transmit a packet and poll the CSR to       *
; *      detect transmission completion.                        *
; *                                                             *
; *-------------------------------------------------------------*
;
txpkt1: bic     #ie!el,@lqacsr      ; disable DESQA interrupts...
                                    ; ... and loopback modes
        bis     #il,@lqacsr         ; disable Internal Loopback
        mov     #tx.bdl,@lqatll     ; write addr of Tx BDL to DESQA
        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 DESQA to      *
; * interrupt the host to signify that the transmission is      *
; * complete.                                                   *
; *                                                             *
; *-------------------------------------------------------------*
;
txpkt2:
lqavec  =       400             ; DESQA host interrupt vector
pri07   =       340             ; Processor Status of ...
                                ; ...interrupt handler - IPL = 7
        bic     #ie!el,@lqacsr  ; disable DESQA interrupts...
                                ; ...and External Loopback
        bis     #il,@lqacsr     ; disable Internal and ...
                                ; ...Internal Extended Loopback
;
; We must write the address of the DESQA 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 DESQA interrupts
        mov     #tx.bdl,@lqatll ; write addr of Tx BDL to DESQA
        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
```

D.6 DATA RECEPTION

The steps required to initiate a reception are:

- Set RE to 1 in CSR. This is not necessary if the DESQA 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 DESQA interrupts disabled.
- Enabling the DESQA 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.                                                   *
; *                                                             *
; *-------------------------------------------------------------*
;
rxpkt1: bic     #ie!el,@lqacsr      ; disable DESQA interrupts...
                                    ; ...and loopback modes
        bis     #il!re,@lqacsr      ; disable Internal Loopback...
                                    ; ...and enable receiver
        mov     #rx.bdl,@lqarll     ; write addr of Rx BDL to DESQA
        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 DESQA to *
; * to determine that a packet has been received.               *
; *                                                             *
; *-------------------------------------------------------------*
;
lqavec  =       400             ; DESQA host interrupt vector
pri07   =       340             ; Processor Status of ...
                                ; ...interrupt handler - IPL = 7
rxpkt2: bic     #ie!el,@lqacsr  ; disable DESQA interrupts...
                                ; ...and loopback modes
        bis     #il!re,@lqacsr  ; disable Internal Loopback...
                                ; ...and enable receiver
;
; We must write the address of the DESQA 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 DESQA interrupts
        mov     #rx.bdl,@lqarll ; write addr of Rx BDL to DESQA
        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+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
```

D.7 EXECUTING ON-BOARD DIAGNOSTICS

The DESQA 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 DESQA.
; Error handling/reporting code should be placed here.
;
20$:    rts     pc                  ; return to caller
```

D.8 BUFFER DESCRIPTOR MANAGEMENT ALGORITHM

(BDLs). 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" it's 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 setup.
- 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 DESQA to transmit, it is recommended that the host use the following algorithm with the BD just issued to the DESQA:

```
if  (DESQA 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 DESQA
              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 DESQA CSR BIT XI

start_bd_unload:
  BD = NEXT BD TO BE RETURNED BY DESQA 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  (DESQA 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 DESQA 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 DESQA
                on the transmit BDL.

              end

     end (* END OF UNLOAD BD ALGORITHM *)
```

APPENDIX E
GLOSSARY

This appendix defines some of the terminology of DESQA and Ethernet.

baseband system

In a baseband system, information is encoded and transmitted in a single frequency band. This means that only one information signal at a time can use the transmission channel without disruption (see **collision**).

binary exponential backoff

The algorithm used to schedule retransmissions after a collision. So named because the interval between retransmissions is increased exponentially with repeated collisions.

bit cell

The length of time occupied by one encoded data bit (see **Manchester encoding**). Equivalent to bit time.

bit time

Equivalent to bit cell. At 10 MHz the bit time is 100 ns.

broadcast

In general, the mode of communications where all nodes are capable of receiving a signal transmitted by any other node. Also, a specific Ethernet addressing mode where the destination is all nodes.

bulkhead

That part of the host system cabinet that provides for external connections to system modules; also known as the system I/O panel, the distribution panel,or the patch and filter assembly.

bulkhead assembly

An assembly comprising the transceiver cable, socket, transceiver fuse, and bulkhead panel insert.

carrier sense

The means by which the physical layer determines that one or more nodes are currently transmitting on the Ethernet.

citizenship (CQ) test

This is a **go/no go** diagnostic test for the DESQA. The PDP-11 code that runs the test is held in the DESQA boot/diagnostic ROM. It can be loaded into host memory and executed by the host.

coaxial cable interface

The electrical, mechanical, and logical connection to the coaxial cable. In other words, the transceiver.

coaxial cable section

An unbroken piece of coaxial cable, fitted with coaxial connections at its ends, which is used to make up coaxial cable segments.

collision

The result of simultaneous transmissions by two or more nodes. Simultaneous transmissions distort each other's signals, and the data must be retransmitted separately. **Collision detect** be asserted only during transmission

collision enforcement

Each node continues to transmit encoded frame bits after a collision has been detected. This makes the collision last long enough for all transmitting stations to detect it. See also **jam bits**.

contention

Interference between colliding transmissions (see **collision**.) Ethernet link management resolves contentions (see **CSMA/CD**.)

controller

A unit, such as a DESQA, which connects a node to the Ethernet.

CRC (cyclic redundancy check)

An error detection scheme, in which a check character is generated by taking the remainder after dividing all the serialized bits in a block of data by a predetermined binary number. The check character is compared with the transmitter-generated check character as a safeguard against errors in transmission.

CSMA/CD

Carrier Sense Multiple Access with Collision Detection. This is the generic term for the class of link management procedure used by the Ethernet. So called because it:

- Allows multiple nodes to access the broadcast channel at will
- Uses carrier sense and deference to avoid contention
- Resolves contention through collision detection and retransmission.

data-link layer

The layer in which the modules create a communications path between two adjacent nodes. Data-link modules ensure the integrity of data transferred across the path. Data-link modules execute simultaneously and independently in this layer. This is the higher of the two layers in the Ethernet local area network. See also **physical layer**

DECnet boot

This is the PDP-11 code that the host calls to start DECnet. The code is stored in the DESQA Boot/Diagnostic ROM.

deference

A process by which the DESQA delays transmission when the channel is busy, in order to avoid contention with on-going transmissions.

DESQA primary boot

That part of the primary bootstrap code in the host that is specific to DESQA. See also **extended primary bootstrap**.

destination address

In the message packet, this is the 48-bit field that contains the address of the receiving node.

drop cable

See **transceiver cable**.

extended primary bootstrap

This is the PDP-11 code that the host calls from the DESQA Boot/Diagnostic ROM. The sequence is as follows:

1. The DESQA primary boot loads the extended primary bootstrap from the DESQA Boot/Diagnostic ROM into host memory.
2. The DESQA primary boot transfers control to the extended primary bootstrap.
3. The extended primary bootstrap uploads the entire contents of the Boot/Diagnostic ROM into host memory, calls the citizenship (CQ) test, and does one of the following:
   - The bootstrap calls the DECnet boot.
   - The bootstrap transfers control to a user-supplied address. If an error occurs, the bootstrap halts.

See also **DESQA primary boot**.

frame

See **packet**.

frame check sequence

A 16-bit error check polynomial that checks that the bit content of a frame is the same before and after transmission. The encoded check value is stored at the end of each frame, and enables transmission errors in the physical channel to be detected.

frame fragment

Any frame containing less than 64 bytes (512 bits) is defined to be a frame fragment resulting from a collision. See also **runt**.

heartbeat

A positive, functional verification provided by the H4000 transceiver within two milliseconds after every attempted transmission. It indicates that the H4000 is operating correctly with respect to collision detection. Also known as the SQE (Signal Quality Error) test.

interframe spacing

See **interpacket gap**.

interpacket gap

An enforced idle time between successive transmissions to allow receiving controllers and the physical channel to recover.

jam.

An encoded bit sequence (for example, part of the frame) that is transmitted to enforce a collision. A jam comprises at least 32 but not more than 48 bits.

late collision

A collision that occurs at least one slot time after the start of the next transmission.

Manchester encoding

A self-synchronizing method of encoding a serial data stream. Under this scheme, a phase reversal occurs in the center of every bit-cell.

message

See **packet**.

MOP (Maintenance Operation Protocol)

A formal set of messages and rules used to load and dump computer memory, as well as to test a communications link between two adjacent nodes.

multicast

An addressing mode where the destination is a group of nodes.

node

A single addressable entity (for example, a computer and its peripherals) that is connected to the Ethernet using a controller (for example, DESQA) and a transceiver.

OSI (Open Systems Interconnection)

A supplier-independent set of protocols for interconnecting communications devices, peripheral devices, and processors in a network.

packet

All data carried on the Ethernet is encapsulated in a packet; this may also be called a frame. A packet contains a preamble, destination address field, source address field, type field, data field, and frame check sequence. Packets are separated by the inter-packet gap.

packet serialization

The process of turning a series of parallel bytes into a serial data stream for transmission. The complete serial packet for Ethernet consists of: a 64-bit fixed format preamble; between 60 and 1514 data bytes; and a 32-bit CRC.

physical address

The unique address value of a given node on the network. By definition, an Ethernet physical address is distinct from all other physical addresses on all Ethernets.

physical channel

The implementation of the physical layer. For example, the DESQA, transceiver cable, transceiver, and coaxial cable.

physical layer

The communications layer in which the implementation depends on the nature of the medium that connects two adjacent nodes. The physical layer enables the data-link layer to disregard the particular characteristics of the connecting medium. For Ethernet, the physical layer is implemented using coaxial cable. This is the lower of the two layers in the Ethernet local area network. See also **data-link layer**.

preamble

A 64-bit sequence that is transmitted at the start of a frame in order to establish synchronization at the receiving node.

primary bootstrap

This is the PDP-11 code that the host executes when the system is powered on. The code is stored in the host memory, usually in ROM.

promiscuous mode

A receive mode, in which the controller accepts all packets and any address filtering is done by software in the host.

repeater

A device for connecting cable segments together, in order to extend the physical channel to its maximum end-to-end length.

round-trip propagation time

The worst-case time (measured in bit times) required for a transmitting node to assert collision detect due to normal contention for the channel. This time is the primary component of slot time and is defined to be 464 bit times (45.4 microseconds).

runt

This status is allocated to a partial packet (less than 64 bytes) that exists in the receive buffer because:

Either the destination address of the packet did not match the address of this node

Or a collision occurred while the packet was being retransmitted, and the buffer stack could not be flushed.

segment

A length of coaxial cable made up from one or more coaxial cable sections and terminated at each end with its characteristic impedance. The maximum segment length is 500 meters (1640.5 feet).

slot time

A parameter that is determined by the three important aspects of collision handling, namely:

- An upper bound on the acquisition time of the network
- An upper bound on the length of a collision-generated frame fragment
- The base value used to calculate the retransmission delay.

Slot time is defined to be 512 bit times (51.2 microseconds).

source address

In the message packet, this is the 48-bit field that contains the address of the transmitting node.

station

Equivalent to node.

station address

See **physical address**.

transceiver

The device that connects directly to the coaxial cable in order to send and receive the encoded signals on the cable, and to provide the required electrical isolation (for example, the H4005).

transceiver cable

The cable that runs between an Ethernet controller (for example, the DESQA) and a transceiver.

type

In the message packet, this is the 16-bit field that indicates how higher layers in the architecture are to interpret the data field.

Index

E

F

H

I

L

M

V

W

Z

N

O

P

Q

R

S

T

U

READER'S COMMENTS

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

1. **Which of the following most closely describes your job?** 1 a b c d e
   (a) Administrative support (d) Scientist/Engineer (g) Educator/Trainer
   (b) Programmer/Analyst (e) Systems Manager (h) Computer Operator
   (c) Software support (f) Sales (i) Other _____
2. **How many years of experience do you have with computers?** 2 a b c d e
   (a) Less than 1 (b) 1 to 3 (c) 4 to 6 (d) 7 to 9 (e) 10 or more
3. **What did you like *most* about this manual?**

4. **What did you like *least* about this manual?**

5. **How do you rate this manual?**
   Indicate your opinion of the quality of the manual. For each aspect of quality, darken your response on the five-point scale, where (1) = POOR and (5) = EXCELLENT

(a) Accuracy	1	2	3	4	5
(b) Completeness	1	2	3	4	5
(c) Usefulness of Examples/Figures	1	2	3	4	5
(d) Clearness of Language	1	2	3	4	5
(e) Helpfulness of Index/Table of Contents	1	2	3	4	5
(f) Consistency in Presenting Information	1	2	3	4	5
(g) Logical Organization	1	2	3	4	5
(h) Visual Appeal	1	2	3	4	5
(i) Relevance of Information	1	2	3	4	5
(j) Ease of Learning	1	2	3	4	5
(k) Ease of Use	1	2	3	4	5
(l) YOUR OVERALL IMPRESSION	1	2	3	4	5
(m) Quality Relative to Other Digital Manuals	1	2	3	4	5
(n) Quality Relative to Other Companies' Manuals	1	2	3	4	5

6. **List any errors you found in the manual. (Reference page, table, or figure numbers.)**

7. **Do you have any additional comments?**

Name _____ Company _____

Title _____ Department _____

Street _____ City _____ State/Country _____ Zip _____

Telephone No. _____ Date _____

Site Code: RE

FOLD HERE AND TAPE

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Educational Services/Quality Assurance
12 Crosby Drive BUO/E08
Bedford, MA 01730

FOLD HERE