QD32 DISK CONTROLLER

TECHNICAL MANUAL

(MSCP COMPATIBLE)

EMULEX
3545 Harbor Boulevard
Costa Mesa, California 92626
(714) 662-5600 TWX 910-595-2521

QD3251001 Rev C
July, 1986

WARNING

This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the technical manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of Federal Communications Commission (FCC) Rules, which are designed to provide reasonable protection against such interference when operating in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

The information in this manual is for information purposes and is subject to change without notice.

Emulex Corporation assumes no responsibility for any errors which may appear in the manual.

Printed in U.S.A.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

Section Title Page

FIVE TROUBLESHOOTING

SIX REGISTERS AND PROGRAMMING

SEVEN FUNCTIONAL DESCRIPTION

EIGHT INTERFACES

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

EMULEX PRODUCT WARRANTY

CONTROLLER WARRANTY: Emulex warrants for a period of twelve (12) months from the date of shipment that each Emulex controller product supplied shall be free from defects in material and workmanship.

CABLE WARRANTY: All Emulex provided cables are warranted for ninety (90) days from the time of shipment.

The above warranties shall not apply to expendable components such as fuses, bulbs, and the like, nor to connectors, adaptors, and other items not a part of the basic product. Emulex shall have no obligation to make repairs or to cause replacement required through normal wear and tear or necessitated in whole or in part by catastrophe, fault or negligence of the user, improper or unauthorized use of the product, or use of the product is such a manner for which it was not designed, or by causes external to the product, such as but not limited to, power failure or air conditioning. Emulex's sole obligation hereunder shall be to repair or replace any defective product, and, unless otherwise stated, pay return transportation cost for such replacement.

Purchaser shall provide labor for removal of the defective product, shipping charges for return to Emulex and installation of its replacement. THE EXPRESSED WARRANTIES SET FORTH IN THIS AGREEMENT ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, AND ALL OTHER WARRANTIES ARE HEREBY DISCLAIMED AND EXCLUDED BY EMULEX. THE STATED EXPRESS WARRANTIES ARE IN LIEU OF ALL OBLIGATIONS OR LIABILITIES ON THE PART OF EMULEX FOR DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF, OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THE PRODUCT.

RETURNED MATERIAL: Warranty claims must be received by Emulex within the applicable warranty period. A replaced product, or part thereof, shall become the property of Emulex and shall be returned to Emulex at

Purchaser's expense. All returned material must be accompanied by a RETURN MATERIALS AUTHORIZATION (RMA) number assigned by Emulex.

BLANK

1.1 INTRODUCTION

The QD32 Disk Controller, designed and manufactured by Emulex Corporation, is a MSCP-compatible controller that interfaces with SMD disk drives. This manual is designed to help you install and use your QD32 Disk Controller. The contents of the eight sections and four appendices are described as follows:

Section 1	**General Description:** This section contains an overview of the QD32 Disk Controller.
Section 2	**Controller Specification:** This section contains the specification for the QD32 Disk Controller.
Section 3	**Planning the Installation:** This section contains the information necessary to plan your installation, including MSCP subsystem and operating system considerations.
Section 4	**Installation:** This section contains the information needed to set up and physically install the controller, including switch settings and cabling.
Section 5	**Troubleshooting:** This section describes fault isolation procedures that can be used to pinpoint trouble spots.
Section 6	**Registers and Programming:** This section describes the QD32's LSI-11 bus registers and presents an overview of the Mass Storage Control Protocol (MSCP).
Section 7	**Functional Description:** This section describes the controller architecture.
Section 8	**Interfaces:** This section describes the controller LSI-11 bus and SMD interfaces.
Appendix A	**Autoconfigure:** This appendix describes the DEC algorithm for the assignment of CSR addresses and vector addresses.
Appendix B	**PROM Removal and Replacement:** This appendix contains instructions to remove and replace the firmware so that the user can upgrade the QD32 Disk Controller in the field.
Appendix C	**Utilities and Diagnostics:** This appendix lists the utilities and diagnostics for the QD32.
Appendix D	**Disk Drive Configuration Parameters:** This appendix contains configuration parameters for supported SMD disk drives.

1.2 SUBSYSTEM OVERVIEW

The QD32 Disk Controller connects high-capacity mass storage peripherals to the LSI-11 bus on computers manufactured by Digital Equipment Corporation (DEC). The QD32 implements DEC's Mass Storage Control Protocol (MSCP) to provide a software-transparent interface for the host DEC computer. To provide traditional Emulex flexibility in peripheral selection, the QD32 uses the industry standard SMD interface as its peripheral interface. The QD32 supports the extended cylinder addressing functions of the SMD-E (extended) interface. SMD-E and SMD-0 interfaces are electrically and logically compatible. For more information on the QD32's SMD interface, see subsection 8.3.

1.2.1 MASS STORAGE CONTROL PROTOCOL (MSCP)

MSCP is a software interface designed to lower the host computer's mass storage overhead by offloading much of the work associated with file management into an intelligent mass storage subsystem. In concert with SMD compatible peripherals, the QD32 provides just such a subsystem. The QD32 relieves the host CPU of many file maintenance tasks. The QD32 Disk Controller performs these MSCP functions: error checking and correction, bad block replacement, seek optimization, command prioritizing and ordering, and data mapping.

This last feature is, perhaps, the most important. This feature allows the host computer's operating system software to store data in logical blocks that are identified by simple logical block numbers (LBNs). Thus, the host does not need to have detailed knowledge of the peripheral's geometry (cylinders, tracks, sectors, etc.). This feature also makes autoconfiguration a simple matter. During system start-up, the host operating system queries the subsystem to find its capacity (the number of logical blocks that the subsystem can store).

Because the host operating system does not need to have detailed knowledge of its mass storage subsystem, the complexity of the operating system itself has been reduced. This reduction comes about because only one or two software modules are required to allow many different subsystems to be connected to a host.

1.3 PHYSICAL ORGANIZATION OVERVIEW

The QD32 Disk Controller is a modular, microprocessor-based disk controller that connects directly to the host computer's LSI-11 bus backplane. The microprocessor architecture ensures excellent reliability and compactness.

The QD32 is contained on a single dual-wide printed circuit board assembly (PCBA) that plugs directly into an LSI-11 bus backplane slot.

The QD32 supports up to two physical or four logical disk drives. Aggregate data storage capacities are limited only by the capacities of the peripherals. Figure 1-1 shows one possible SMD configuration.

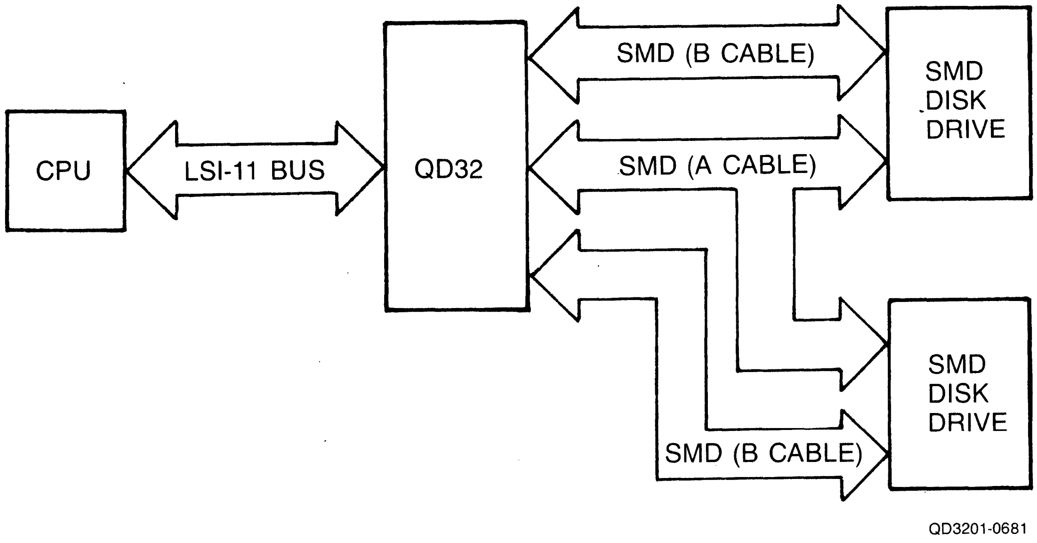


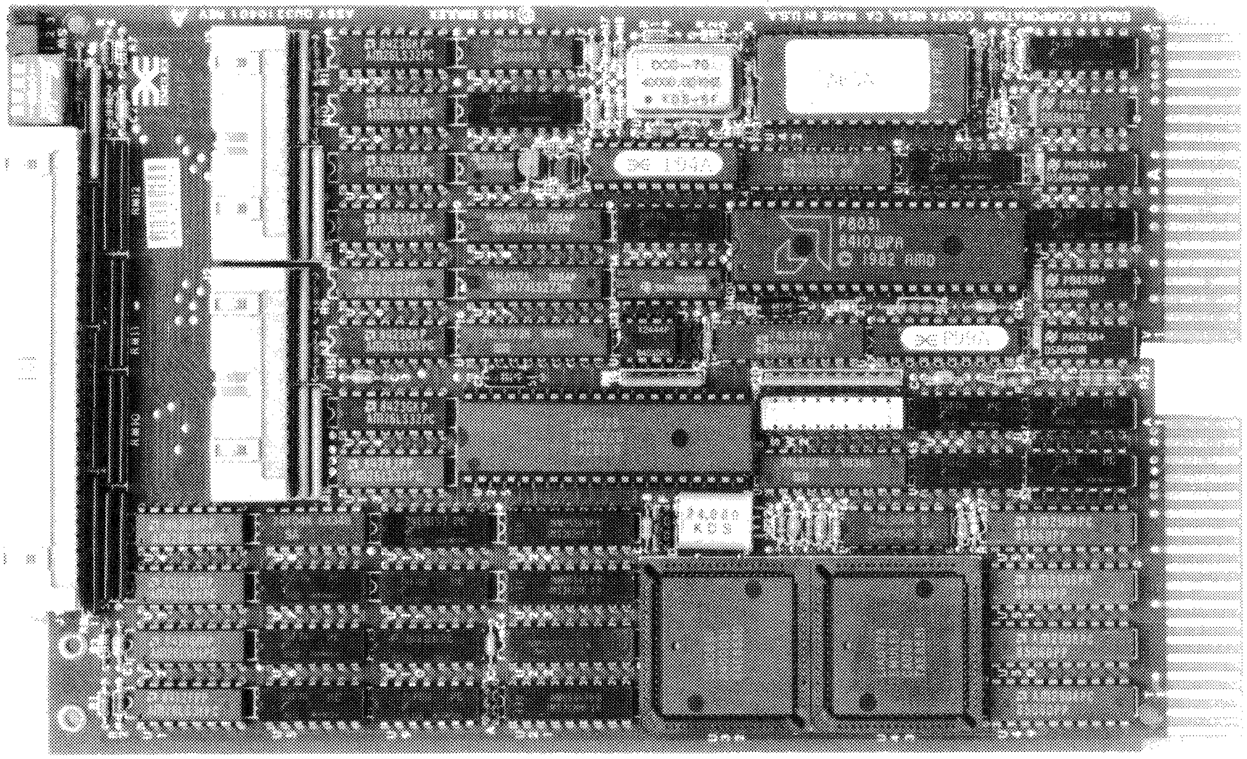
Figure 1-1. QD32 Subsystem Configuration

1.4 SUBSYSTEM MODELS and OPTIONS

The QD32 Disk Controller, with appropriate peripherals, provides a DEC MSCP-compatible mass storage subsystem. The QD32 is pictured in Figure 1-2. The QD32 is identified by a top level assembly tag that is glued to the 8031 microprocessor chip on the PWB. The QD32 top level assembly number is given in Table 1-1 along with the part numbers of the items that are delivered with the QD32.

Table 1-1. Basic Contents

Itm	Qty	Description	Part Number
1	1	QD32 Disk Controller	QD3210201
2	1	22-Bit Addressing Kit	QD0113002
3	1	QD32 Technical Manual	QD3251001



QD3201-0823

Figure 1-2. QD32 Disk Controller

1.4.1 SUBSYSTEM OPTIONS

Table 1-2 lists the options that can be ordered to tailor your QD32 to your particular application. Software programs are offered in distribution kits that include media and documentation.

Table 1-2. QD32 Options

Option	Description
PX995180n-0n[1]	Includes the Emulex LSI/PDP MSCP Formatter Program (QXMX8B). Distribution kit is per customer order.
VX9951804	Includes the Emulex MicroVAX MSCP Disk Formatter Program (FVD32M).
QD3113003	QD32 Cabling Kit for MICR0/PDP-11 and MicroVAX I and II Tower. Includes cables, panels, and instructions.
SU7811212-0n[2]	SMD A-Cable, shielded
SU7811219-0n[2]	SMD A-Cable, extension
SU7811213-0n[2]	SMD B-Cable. shielded
SU7811218-0n[2]	SMD-B Cable, extension
SU1110201	Cable I/O Adapter Panel
CU2220301	Rack-Mount Panel
SU7813104	Peripheral Cable Adapter Panel Kit

[1] See Appendix C for distribution kit part numbers.

[2] See Table 4-7 for part numbers for specific cable lengths.

Options are specified as separate line items on a sales order. An example of an actual sales order is shown in Figure 1-3.

Item	Qty	Model Number	Comment/Description
1.	1	QD32	Disk Controller implementing DEC MSCP with SMD drives.
2.	1	VX9951804	Includes the Emulex MicroVAX MSCP Disk Formatter Program (FVD32M).

Figure 1-3. Sales Order Example

1.5 FEATURES

The following features enhance the usefulness of the QD32 Disk Controller.

1.5.1 MICROPROCESSOR DESIGN

The QD32 design incorporates an eight-bit, high-performance CMOS microprocessor to perform all controller functions. The microprocessor approach provides a reduced component count, high reliability, easy maintainability, and the microprogramming flexibility that allows MSCP to be implemented without expensive, dedicated hardware.

1.5.2 CONFIGURATION FLEXIBILITY

The QD32 Disk Controller can support many different SMD drive configurations by using the QD32's Nonvolatile Random Access Memory (NOVRAM). The QD32's NOVRAM can be programmed for two physical drive configurations. In addition, the user can change these stored drive configurations by altering the NOVRAM using Emulex software or the QD32's extended command set. The flexibility of the QD32 configuration NOVRAM eliminates the need for special configuration PROMs and field upgrade kits.

1.5.3 SELF-TEST

The QD32 incorporates an internal self-test routine which exercises all parts of the microprocessor, the on-board memory, the buffer controller, and the Host Adapter Controller (HAC). Although this test does not completely test all circuitry, successful execution indicates a very high probability that the disk controller is operational. If the QD32 fails the self-test, it leaves three light-emitting diodes (LEDs) ON and sets an error bit in the Status and Address (SA) register (base address plus two).

1.5.4 ERROR CONTROL

The disk controller presents error-free media to the operating system by correcting soft errors and retrying operations without intervention by the host.

1.5.5 SEEK OPTIMIZATION

The QD32 is able to pool the various seeks that need to be performed and determine the most efficient order in which to do them. This is an especially important feature in heavily loaded systems. The disk controller's ability to arrange seeks in the optimum order saves a great deal of time and makes the entire system more efficient.

1.5.6 COMMAND BUFFER

The QD32 contains a buffer that is able to store 13 MSCP commands. This large buffer allows the subsystem to achieve a higher throughput and to operate at a very efficient level.

1.5.7 ADAPTIVE DMA

During each DMA data transfer burst, the QD32 monitors the LSI-11 bus for other pending DMA requests and suspends its own DMA activity to permit other DMA transfers to occur. The host processor programs the DMA burst length during the MSCP initialization sequence or the QD32 defaults to 16 words per burst. In addition, the QD32 firmware design includes a switch selectable DMA burst delay to avoid data late conditions. Because of these adaptive DMA techniques, the QD32 ensures that CPU functions, including interrupt servicing, are not locked out for excessive periods of time by high-speed disk transfers.

1.5.8 BLOCK-MODE DMA

The QD32 supports block-mode DMA for accessing memory. In this mode, the initial address of the data is transmitted, followed by a burst of up to 16 words of data. The memory address is automatically incremented to accommodate this burst. Block mode transfers considerably reduce the overhead associated with DMA operations.

1.5.9 TWENTY-TWO-BIT ADDRESSING

The QD32 supports the 22-bit addressing capability of the extended LSI-11 bus.

1.6 COMPATIBILITY

1.6.1 DIAGNOSTICS

Emulex offers two diagnostic programs to support the use and maintenance of the QD32:

- Emulex PDP/LSI MSCP Formatter Program (QXMX8B)
- Emulex MicroVAX MSCP Disk Formatter Program (FVD32M)

1.6.2 OPERATING SYSTEMS

The QD32 implements MSCP. Emulex supports its implementation of MSCP beginning with the indicated version of the following DEC operating systems:

Operating System	Version
Micro/VMS	4.0
RSTS/E	8.0
RSX-11M	4.1
RSX-11M-PLUS	2.1
RT-11	5.1
Ultrix-32m	1.1

1.6.3 HARDWARE COMPATIBILITY

The QD32 Disk Controller complies with DEC LSI-11 bus protocol, and it directly supports 22-bit addressing and block-mode DMA. The QD32 also supports scatter-write and gather-read operations on the MicroVAX I.

The disk drives supported by the QD32 are not media compatible with comparable DEC MSCP products.

The QD32 Disk Controller supports hard-sectored disk drives that use the SMD interface. The QD32 also supports the extended cylinder addressing functions of the SMD-E interface (see subsection 8.3). The QD32 supports disk transfer rates up to 2.5M byte per second. Emulex has certified the following disk drives for full support:

- Control Data Corporation (CDC) 9457 LMD
- CDC 9710-80
- CDC 9715-340
- CDC 9715-515
- CDC 9720
- CDC 9771 XMD
- Century Data Systems (CDS) 315
- Fujitsu 2298

- Fujitsu 2321/22
- Fujitsu M2333
- Fujitsu M2351A
- Fujitsu M2361A
- NEC 2352

BLANK

2.1 OVERVIEW

This section contains the general. environmental, physical, electrical, and port specifications for the QD32 Disk Controller.

Subsection	Title
2.2	General Specification
2.3	Environmental Specification
2.4	Physical Specification
2.5	Electrical Specification

2.2 GENERAL SPECIFICATION

Table 2-1 contains a general specification for the QD32 Disk Controller.

Table 2-1. QD32 General Specifications

Parameter	Description
FUNCTION	Providing mass data storage to Digital Equipment Corporation (DEC) computers that use the LSI-11 bus
Logical CPU Interface	Emulates DEC's Mass Storage Control Protocol (MSCP)
Diagnostic Software	Emulex PDP/LSI MSCP Disk Formatter Program (QXMX8B) and MicroVAX MSCP Disk Formatter Program (FVD32M)
Operating System Compatibility	Micro/VMS V4.0 and above RSTE/S V8.0 and above RSX-11M V4.1 and above RSX-11M PLUS V2.1 and above RT-11 V5.1 and above Ultrix-32m V1.1 and above

(continued on next page)

Table 2-1. QD32 General Specifications (continued)

Parameter	Description
CPU I/O Technique	Direct Memory Access (DMA), including adaptive techniques and block mode; supports scatter-write and gather-read operations on the MicroVAX I
INTERFACE	
CPU Interface	Extended LSI-11 bus interface
Device Base Address	
Standard	17772150_8
Alternates	17772154_8 17760334_8 17760340_8 17760344_8 17760350_8 17760354_8 17760360_8
Vector Address	Programmable
Priority Level	BR4 and BR5
Bus Loading	1 DC Load, 2.5 AC Loads
Peripheral Interface	SMD-E (Extended) up to 20 MHz SMD-0
Disk Transfer Rate	Up to 2.5M bytes per second
Number of Physical Drives Supported	2
Drive Sectoring	Hard Sectored
Maximum Cable Lengths	
A Cable (daisy-chain)	35 ft (10.5 m) cumulative
B Cable (radial)	25 ft (7.5 m)

2.3 ENVIRONMENTAL SPECIFICATION

Table 2-2 contains the environmental specifications for the QD32 Disk Controller.

Table 2-2. QD32 Environmental Specifications

Parameter	Description
OPERATING TEMPERATURE	10°C (50°F) to 40°C (104°F), where maximum temperature is reduced 1.8°C per 1000 meters (1°F per 1000 feet) altitude
RELATIVE HUMIDITY	10% to 90% with a maximum wet bulb of 28°C (82°F) and a minimum dewpoint of 2°C (3.6°F)
COOLING	6 cubic feet per minute
HEAT DISSIPATION	82 BTU per hour

2.4 PHYSICAL SPECIFICATION

Table 2-3 contains the physical specifications for the QD32 Disk Controller.

Table 2-3. QD32 Physical Specifications

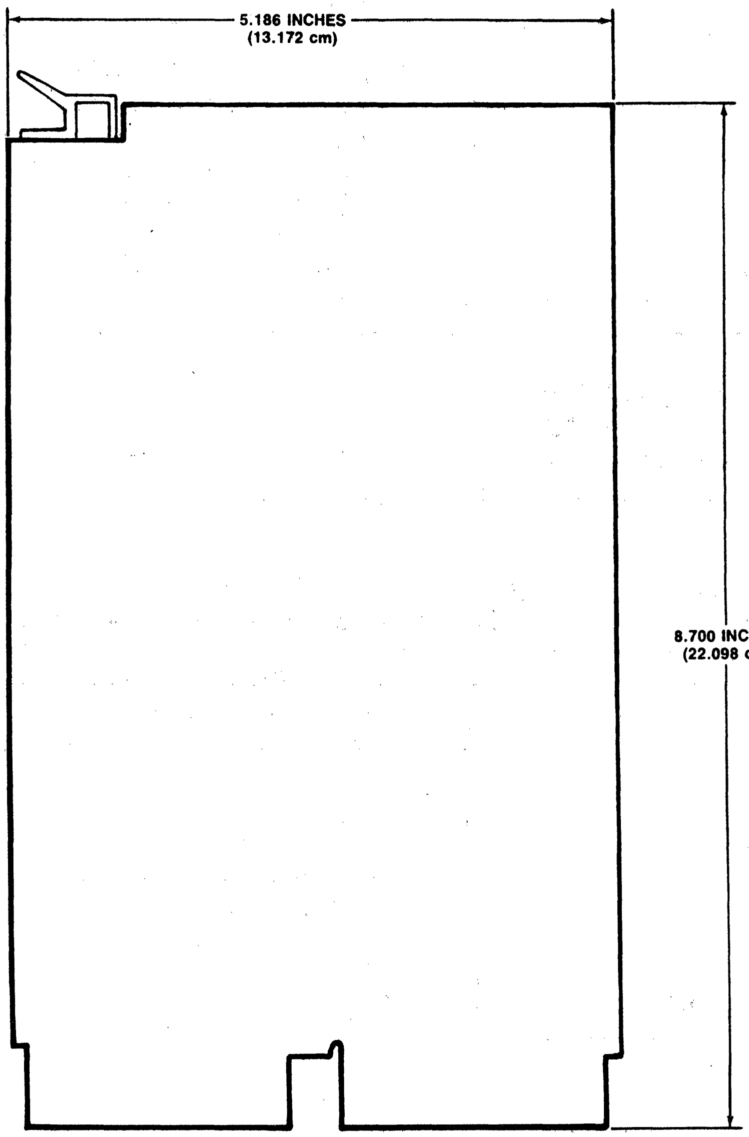
Parameter	Description
PACKAGING	Single, dual-wide, four-layer PCBA
Dimensions	5.186 by 8.70 inches 13.172 by 22.09 centimeters (see Figure 2-1)
Shipping Weight	3 pounds

2.5 ELECTRICAL SPECIFICATION

Table 2-4 lists and describes the electrical specification for the QD32 Disk Controller.

Table 2-4. QD32 Electrical Specifications

Parameter	Description
POWER	5 VDC ± 5%, 2.6 amperes (A)



QD3201-0634

Figure 2-1. QD32 Disk Controller Dimensions

Section 3
PLANNING THE INSTALLATION

3.1 OVERVIEW

This section is designed to help you plan the installation of your QD32 Disk Controller. Taking a few minutes and planning the configuration of your subsystem before beginning its installation should result in a smoother installation with less system down time. This section contains QD32 application examples and configuration procedures. The subsections are listed in the following table:

Subsection	Title
3.2	MSCP Subsystem Configuration
3.3	A DEC MSCP Subsystem
3.4	The QD32 MSCP Subsystem
3.5	Operating Systems, Device and Vector Addresses

3.2 MSCP SUBSYSTEM CONFIGURATION

The following paragraphs describe MSCP Subsystem concepts, including architecture, unit numbering, capacities, and related concepts.

3.2.1 ARCHITECTURE

MSCP is a protocol designed by DEC for mass storage subsystems using Digital Storage Architecture (DSA). In a MSCP mass storage subsystem, DSA comprises three functional and physical layers:

- **Host Layer.** An MSCP class-driver in the host system receives requests from the operating system and then relays data and commands to the controller in MSCP message packets.

- **Controller Layer.** The MSCP controller communicates with both the host layer and the mass storage layer. The controller transmits MSCP message packets to and from the host MSCP class-driver and performs data-handling functions for the mass storage devices. The QD32 functions as the controller layer.

- **Mass Storage Layer.** The mass storage peripheral devices communicate with the MSCP controller and send or receive data as specified by the MSCP controller.

MSCP defines the form of the message packets that are exchanged by the host and the MSCP controller. The QD32 implements MSCP in mass storage subsystems using SMD as the peripheral interface.

3.2.2 PERIPHERAL NUMBERING

Each MSCP peripheral on the system is identified to the operating system by an MSCP device name. An MSCP device name consists of a device class identifier and a unit number. The device class is indicated by a two-letter prefix; MSCP disk devices are indicated by the prefix DU.

With the exception of MicroVMS systems, DEC operating systems require that all MSCP peripherals on a system have different MSCP device numbers, even if they are managed by separate MSCP controllers at separate LSI-11 bus device addresses. For example, under RSX-11M-PLUS, if there are three peripherals on the first MSCP controller (at 772150_8), then the first peripheral on the second MSCP controller (in floating CSR address space) is numbered DU3.

3.2.3 PERIPHERAL CAPACITIES

The capacity of peripherals in an MSCP subsystem is measured in logical blocks. Each logical block contains 512 bytes of data. The MSCP controller can report the capacity of a peripheral to the operating system. For example, a Fujitsu M2361A (689M byte) disk drive supported by the QD32 is able to store 1,139,136 logical blocks.

3.3 A DEC MSCP SUBSYSTEM

Figure 3-1 shows the organization of a typical DEC MSCP subsystem for the LSI-11 bus. The MSCP host and controller functions (see section 3.2.1) are combined in a single piece of hardware, in this example the DEC KDA50. The KDA50 plugs directly into the LSI-11 bus and is attached to RA81 disk drives via a disk-drive-native interface.

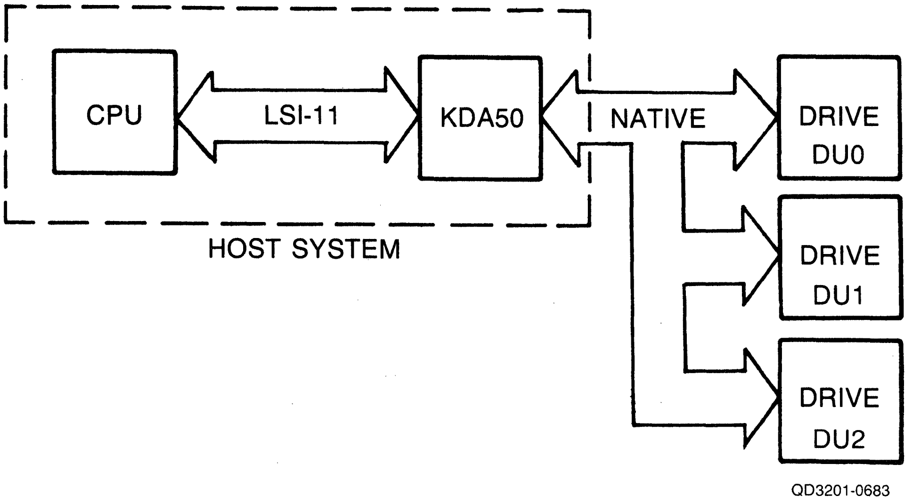


Figure 3-1. DEC MSCP Subsystem Logical and Physical Configuration

3.4 THE QD32 MSCP SUBSYSTEM

Figure 3-2 illustrates a typical QD32 MSCP subsystem. As with the DEC implementation, the QD32 is connected directly to the LSI-11 bus. However, the QD32 uses the SMD peripheral interface to communicate with one or two disk drives.

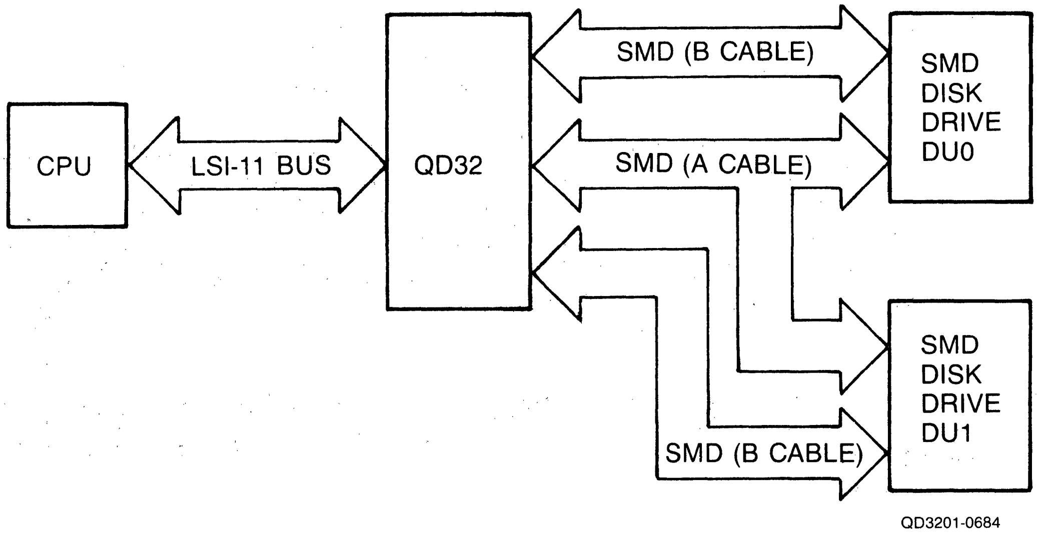


Figure 3-2. QD32 Subsystem Logical and Physical Configuration

The MSCP subsystem provided by the QD32 is essentially analogous to the DEC MSCP subsystem. As in the DEC subsystem, the QD32 MSCP controller connects directly to the LSI-11 bus and performs many of the same functions as the KDA50. As an MSCP controller, the QD32 receives requests from the host, optimizes the requests, generates SMD commands to perform the operations, transfers data to and from the host, transfers data to and from the device, and buffers data as necessary. When the command is complete, the controller sends a response to the host.

The QD32 also performs all of the functions of the peripheral controller, including serialization and deserialization of data. The QD32 connects to the peripherals it supports via the SMD interface.

3.4.1 LOGICAL UNIT NUMBERS

As noted in subsection 3.2.2, most DEC operating systems do not allow any MSCP disk devices on one system to have the same unit number, even though they may be controlled by separate MSCP controllers at different base addresses.

DEC MSCP-type drives can accept unit identification plugs that define addresses from 0 to 255. Disk drives controlled by the QD32 do not have this flexibility; the QD32 can detect only two unique drive addresses at its SMD interface - 0 and 1. To prevent a unit-number conflict between the QD32's drives and another MSCP controller's drives, the QD32 employs switches to change the drive logical unit number that is reported to the operating system.

Example 3-1   An MSCP controller at a standard base address supports four disk drives; a QD32 at an alternate base address supports two disk drives. An offset of 4 is specified for the QD32. This causes the QD32 disk with address 0 to be reported to the operating system as logical unit number (LUN) 4. The QD32 disk 1 is reported as LUN 5.

The offset for the logical unit number is specified by using switches SW1-2 through SW1-4 on the QD32. See subsection 4.3.3.2.2 for switch setting information.

3.4.2 QD32 MSCP SUBSYSTEM LOGICAL CONFIGURATION

This subsection explains the algorithm used by the QD32 to map logical MSCP peripherals onto the physical disk drives provided by the QD32 subsystem.

3.4.2.1 Logical Devices

The phrase "logical MSCP disk drive" refers to the disk drive as it appears to the operating system. That is, the operating system associates a disk drive of known type (in this case, an MSCP disk drive) with a unit number and a capacity. The QD32 MSCP controller presents that information to the operating system after initialization on command.

Because the MSCP controller is responsible for establishing the relationship between unit number and capacity, it is possible for the controller to divide its physical disk drives into more than one logical unit. For example, if a physical disk drive has a capacity of 1,402,832 blocks, the MSCP controller can divide that capacity into two parts of 701,416 blocks each. (Each part may have a different capacity.) Each part is then assigned a separate unit number, and the the unit number and capacity of each part is presented to the operating system.

The operating system then sees the two parts as separate disk drives, even though the data is actually stored on the same physical drive. The two parts are called logical disk drives, and the numbers that identify them are called MSCP unit numbers.

A drive configuration that supports multiple logical units is specified by the data that is stored in the configuration Nonvolatile Random Access Memory (NOVRAM). Information for programming the configuration NOVRAM is given in Section 6, Registers and Programming. The field that causes a drive to be divided into multiple logical units is called the Split Code (word 11). There are four types of split codes: no split, cylinder split, head split, and reverse head split:

- When no split is specified, the entire physical drive is one logical drive.

- Cylinder split codes divide a physical drive by cylinders. A Starting Cylinder Offset field in the NOVRAM specifies the first cylinder of the second logical drive. Alternate cylinders are divided evenly between drives. For example, a Fujitsu M2351A, which has 842 cylinders, might be divided so that Drive 0 is assigned cylinders 0 through 420 and Drive 1 assigned cylinders 421 through 841. In this example, the Starting Cylinder Offset has a value of 421.

- Head split codes divide the drive by data heads. A Starting Head Offset field in the NOVRAM specifies the first head of the second logical drive. When the drive is split by data heads, each logical drive has its own platter or group of platters; consequently, the lower blocks of one logical drive are in the same physical cylinder as the lower blocks of the other logical drive. For example, a CDC 9457 LMD might be divided so that Drive 0 is assigned heads 0 and 1 and Drive 1 is assigned heads 2 and 3. The Starting Head Offset has a value of 2. In this example, the CDC 9457 removable media is assigned logical unit number (LUN) 0 and the fixed media assigned LUN 1.

  Reverse head split codes also divide the drive by data heads, but assign the lower numbered heads to drive 1 and the higher numbered heads to drive 0. If you entered a reverse split code for the previous CDC 9457 example, Drive 0 is assigned heads 2 and 3 and Drive 1 is assigned heads 0 and 1. The Starting Head Offset has a value of 2. One advantage of the reverse head split codes is that you may use a fixed media unit as your system disk without modifications to the operating system.

The head splitting technique has a performance advantage over the cylinder method. Typically, most disk accesses are made in the lower cylinders of a disk because many system-oriented files are located there, including the drive's directory. Because the low (and high) cylinders are vertically aligned between the two logical drives when the head splitting technique is used, switching between head-split logical drives requires less head movement than switching between cylinder-split drives.

3.4.2.2 Device Numbers

The drives supported by the QD32 are assigned MSCP device names by the operating system. As described in subsection 3.2.2, an MSCP device name consists of a device class prefix and a device unit number. Drives are assigned MSCP device numbers beginning with zero (0). The conventions for numbering multiple MSCP drives vary by operating system.

Under RSX-11M, RSX-11M-PLUS and RT-11, DU0 is assigned to the first drive on the first MSCP controller, where "first" means the controller located at the standard base address. Unit number 1 would be the second drive on the first controller, etc. If there are two MSCP controllers on the system, the units installed on the second begin numbering at n+1, where n equals the highest unit number of the first MSCP controller.

RSTS/E requires that drives supported by a standard MSCP controller be numbered starting at 0 and drives supported by an alternate MSCP controller be numbered starting at 4.

Because MSCP device names under MicroVMS designate the supporting MSCP controller, the unit numbering is less restricted. For example, two drives which are supported by a standard MSCP controller might be DUA0 and DUA1 and a third drive supported by an alternate MSCP controller might be DUB0.

Table 3-1 is an MSCP unit numbering example under the RSX-11M operating system which shows the MSCP number versus the actual physical addresses assigned to all the components. The physical disk drive (unit number 0) of the second controller is split into two logical units. Note that two device names are associated with that drive.

Table 3-1. Subsystem Configuration Example

QD32 Address	Device Description	Drive Unit Number	Device Name
772150	Fujitsu M2361A	0	DU0
	Fujitsu M2361A	1	DU1
760334 (Floating)	CDC 9457 LMD (head split)	0	DU2 DU3
	Fujitsu M2351A	1	DU4

NOTE

All of the MSCP peripherals supported by the QD32 use the same device identifier - RA81. Unique device identifiers are not available to the host because of the nature of the NOVRAM.

3.5 OPERATING SYSTEMS, DEVICE AND VECTOR ADDRESSES

Before the installation of any peripheral device can be considered complete, the computer's operating system must be made aware of the new resource. The information provided in this section is intended to supplement your DEC operating system resources and to be used as an aid in planning the installation of your QD32.

An operating system can be made aware of a new resource in three ways:

- The operating system can poll the computer's I/O device address space.
- The device can be manually connected using CONNECT or CONFIGURE statements.
- The user can tell the operating system about a device during an interactive SYSGEN procedure.

The first technique is referred to as autoconfigure, and it is essentially automatic. The second technique requires that CONNECT statements be placed in a special command file that is executed each time the computer is bootstrapped. The third technique, interactive SYSGEN, creates a configuration file that the operating system references when the system is bootstrapped. All techniques accomplish the same result: they associate a specific device type with a bus address and interrupt vector.

Most recent versions of DEC operating systems use autoconfigure to some extent, and try to follow the same rules. The RT-11 operating system does not use autoconfigure, but can locate devices that reside at a standard address. There are some differences among operating systems, however, especially with regard to MSCP controllers at alternate LSI-11 bus addresses. The following paragraphs address these differences for each supported operating system. This discussion includes information on choosing appropriate LSI-11 bus device addresses and interrupt vectors for the subsystem.

The following operating system discussions give procedures for choosing LSI-11 bus addresses for the first MSCP controller and any subsequent controllers in the host configuration. No instructions are provided for programming the chosen address into the QD32. See subsection 4.5.1 for detailed switch setting information.

MSCP-type controllers contain two registers that are visible to the LSI-11 bus I/O page. They are the Initialization and Polling (IP) register (base address) and the Status and Address (SA) register (base address plus 2). The IP register, the CSR address, LSI-11 bus address and the base address all refer to the same register. All of the operating systems described in the following subsections use the standard LSI-11 bus address of 772150_8 for the first controller on the host system.

Vector addresses for MSCP controllers are not selected by using switches on the controller, but are programmed into the controller during the Initialization process. Many operating systems select the vector address automatically. If an operating system requires manual input of the vector, the procedure notes that fact.

Again, although DEC has attempted to standardize treatment of peripherals by operating systems, some differences do exist. Table 3-2 lists and describes the device names assigned to MSCP devices under five operating systems. Two controller names and two drive names are given to indicate the numbering scheme.

Table 3-2. Device Names

Operating System	Controller First, Second	Drives Supported by First Controller
RSTS/E	RU0, RU1	DU0, DU1
RSX-11M	---- ----	DU0, DU1
RSX-11M-PLUS	DUA, DUB	DU0, DU1
RT-11	Port0, Port1	DU0, DU1
MicroVMS	PUA, PUB	DUA0, DUA1
Ultrix-32m	uda0, uda1	ra0, ra1

The information regarding operating systems in these subsections is subject to change. The following discussions are based on three assumptions:

- This is the first pass that is being made through SYSGEN; therefore, no saved answer file exists. Answer **N** (no) to questions such as "Use as input saved answer file?"
- Your host system configuration conforms to the standard LSI-11 device configuration algorithm (otherwise autoconfigure results are not reliable).
- You are generating a mapped version of the operating system on the appropriate hardware (unless you are using RT-11).

3.5.1 RSTS/E OPERATING SYSTEMS (V8.0 and above)

RSTS/E scans the hardware to determine configuration each time the system is bootstrapped. The scanning program is called INIT.SYS and it relies on the same hardware configuration conventions as do the other DEC operating systems.

The RSTS/E operating system can support two MSCP controllers. The first MSCP controller must be located at the standard LSI-11 bus address, 772150_8. According to DEC documentation, the second unit should be located in floating address space. For an alternate QD32, Emulex suggests specifying a LSI-11 bus address of 760334_8 using the HARDWARE option of the INIT.SYS program.

The INIT.SYS program uses a user-specified table, located in the currently installed monitor, to make exceptions to the autoconfigure algorithm. This table is modified by the HARDWARE option of the INIT.SYS program. Use of this table allows an MSCP controller to be placed at virtually any address on the I/O page. Note that this table must be reset any time a new monitor is installed. Emulex suggests using an LSI-11 bus address of 760334_8 for an alternate QD32. An MSCP controller must be located at the standard address to be a bootstrap device.

Interrupt vector addresses are assigned to the MSCP controllers by INIT.SYS and programmed into the devices during initialization.

3.5.1.1 Adding MSCP Support

Support for an MSCP controller must be included in a monitor at SYSGEN time. To include support for an MSCP controller in a RSTS/E monitor, respond to the SYSGEN question "number of MSCP controllers" with the number of MSCP controllers on the system.

Units connected to MSCP controllers will be accessible to an on-line RSTS/E system only after the monitor is successfully SYSGENed and installed with the INSTALL sub-option of the INIT.SYS program, and the units have been successfully initialized with the DSKINT sub-option of INIT.SYS.

3.5.2 RT-11 OPERATING SYSTEMS (V5.1 and above)

The RT-11 Operating System supports up to four MSCP controllers with up to 256 devices (total) on the four controllers. The following paragraphs discuss the LSI-11 bus and vector addresses for MSCP controllers under RT-11 in host systems with only one MSCP controller and in those with more than one controller. Disk partitioning, a unique feature of RT-11 that is applicable regardless of the number of controllers, is also discussed.

3.5.2.1 Installing a Single MSCP Controller

If your host system includes only one MSCP controller, install it with a LSI-11 bus address of 772150_8. RT-11 will find and install the handler (driver) for that controller. In single MSCP controller configurations, it is not necessary to run SYSGEN. You may use one of the pregenerated monitors that are provided with the RT-11 Distribution. Emulex recommends that you modify the system start-up command file, STARTx.COM, to properly partition the disk drives. See subsection 3.5.2.3.

3.5.2.2 Installing Multiple MSCP Controllers

If your host system includes more than one MSCP controller, you may either modify the MSCP handler as described in the RT-11 Software Support Manual or perform a SYSGEN. The following procedure describes the SYSGEN technique (user input is in **boldface** print):

1. Initiate SYSGEN:

   ```
   IND SYSGEN<return>
   ```

   Answer the next group of questions appropriately.

2. Indicate that you want the system to use the start-up command file when booting:

   ```
   Do you want the start-up indirect
   file (Y)? Y<return>
   ```

   The start-up command file is required to allow additional MSCP controller LSI-11 bus addresses to be specified and to partition the disks consistently when the system is bootstrapped. Answer the next set of questions appropriately.

3. Indicate that you want MSCP support when the Disk Options question appears:

   ```
   Enter the device name you want support for
   [dd]: DU<return>
   ```

4. Indicate the number of MSCP controllers on your system in response to this question:

   ```
   How many ports are to be
   supported (1)? 2<return>
   ```

   RT-11 refers to individual MSCP controllers or controllers as ports. Each port has its own LSI-11 bus and vector addresses.

5. Specify support for all other devices in your host system configuration as well. Indicate that there are no more devices by entering a period:

   ```
   Enter the device name you want support for
   [dd]: .<return>
   ```

6. You must specify the addresses of all MSCP controllers (ports) using the SET CSR keyboard command. To ensure that this is done consistently and automatically on power-up, you must add the commands to the system start-up command file, STARTx.COM. The x stands for the monitor that is being used, where x is S, F, or X for single-job, foreground/background, or extended memory, respectively. Edit the command file to include the following statements:

   ```
   SET DU CSR=772150          (Default)
   SET DU CSR2=760334₈        (Floating)
   SET DU VECTOR=154          (Default)
   SET DU VEC2=300
   ```

   The LSI-11 bus for the second device can be any unused address in the I/O page which is supported by QD32 address switch settings; the vector address can be any unused address in the vector page. Default statements are not required.

7. Complete SYSGEN according to the DEC documentation.

3.5.2.3 Disk Partitioning

RT-11 is unable to handle drives with a capacity of more than 65,535 blocks (33.5M bytes). To allow drives with larger capacities to be used, RT-11 allows individual physical drives to be partitioned into multiple logical drives. This is done by assigning as many logical drive names (DU0, DU1, etc.) to a physical drive as that drive can support. The statements that make that assignment should be placed in the system start-up command file. This ensures that the drives are automatically partitioned every time the system is bootstrapped and that the partitions are always the same. Use the following procedure to determine the total number of logical drives to be assigned to each physical drive.

1. Determine the drive configuration(s) that you intend to use. You need to know the LUN of each logical drive and the data storage capacity (in logical blocks) of each logical unit. Refer to Appendix D for the logical block capacity of supported SMD drives. If the QD32 is at an alternate LSI-11 bus address (not 772150_8), then you must specify an MSCP device number by using switches SW1-2 through SW1-4 (see subsection 4.3.3.2.2).

2. Divide the capacity for each MSCP Unit by 65,535. If the result is a number greater than 1, then that MSCP Unit should be partitioned into multiple logical units. (The last partition on a disk may be smaller that 65,535 blocks.) Round the result up to the nearest whole number. That whole number equals the number of logical disks into which that MSCP unit should be partitioned.

3. You must then include a series of statements in the system start-up command file, STARTx.COM, that assigns logical names to each partition. Each statement has the following format:

```
SET DUn UNIT=y PART=x PORT=z
```

where **n** is the logical device name, **y** is the physical MSCP unit number, **x** is the partition number, and **z** is the controller number. You must do this for each partition on each drive, including drives that can hold only one partition.

Example: You have selected a Fujitsu M2351A drive that has a capacity of 787,156 blocks.

```
787,156
-------    = 12.01 (13 logical units)
65,535
```

(RT-11 only allows a maximum of 8 addressable units.)

Dividing the unit capacities by 65,535 and rounding the result up to the nearest whole number gives the number of logical units into which each should be partitioned.

You begin assigning logical names to the partitions beginning with DU0. For the previous example, the assignments are made as follows:

```
SET DU0 UNIT=0 PART=0 PORT=0
SET DU1 UNIT=0 PART=1 PORT=0
SET DU2 UNIT=0 PART=2 PORT=0
SET DU3 UNIT=0 PART=3 PORT=0
SET DU4 UNIT=0 PART=4 PORT=0
SET DU5 UNIT=0 PART=5 PORT=0
SET DU6 UNIT=0 PART=6 PORT=0
SET DU7 UNIT=0 PART=7 PORT=0
```

NOTE

The port command must be removed is only one controller is present. If two controllers are present, then the command is valid.

Modify the system start-up command file to include the disk partitioning statements.

3.5.3 RSX-11M OPERATING SYSTEMS (V4.0 and above)

RSX-11M SYSGEN is an interrogative program that allows a complete, running RSX-11M system to be configured for a particular hardware environment. SYSGEN is well documented in the RSX-11M System Generation and Installation Guide, and you are expected to rely primarily on that manual. This explanation is provided only to remove some ambiguities that the installation of the QD32 may present.

SYSGEN supports autoconfigure, and MSCP controllers are detected by autoconfigure. However, autoconfigure detects only the MSCP controller that is located at the standard LSI-11 bus address. Additional MSCP controllers at alternate addresses must be attached to the operating system manually.

NOTE

If the QD32 controls the system disk, you must select 22-bit addressing (SW2-6 ON) even if your system has only 256K bytes of memory.

3.5.3.1 Installing a Single MSCP Controller

If you have only one QD32, install it at the standard address (772150_8) and use autoconfigure to connect your peripherals. The procedure given in the RSX-11M System Generation and Configuration Guide is adequate for this purpose.

3.5.3.2 Installing Multiple MSCP Controllers

If you have two MSCP controllers, say a DEC MSCP controller and a QD32, you must perform a complete manual initialization. We recommend that the DEC MSCP controller be installed at the standard LSI-11 bus address. Locating the QD32 at the alternate LSI-11 bus address does not prevent its being used as the system device. Both MSCP controllers are connected to the operating system by using the following procedure.

1. Invoke SYSGEN.

```
> SET /UIC=[200,200]<return>
> @SYSGEN<return>
```

2. To indicate that you want to use autoconfigure, answer **Y** (yes) to the following question:

```
*      Autoconfigure the host system hardware?
       [Y/N]: Y<return>
```

3. To indicate that you do not want to override autoconfigure results, answer **N** (no) to this question:

```
*      Do you want to override Autoconfigure
       results? [Y/N]: N<return>
```

   Answer the rest of the questions in the SETUP section appropriately, and continue to the next section, TARGET CONFIGURATION. In TARGET CONFIGURATION, the defaults presented for the first group of questions should be accurate for your system because autoconfigure was requested.

4. In response to the question regarding devices, indicate that you have two MSCP-type controllers:

```
*      Devices: DU=2<return>
       Devices: .<return>
```

   This entry supersedes the value of 1 that autoconfigure has determined. Typing a period (.) terminates device input.

   Continue through the next four sections, HOST CONFIGURATION, EXECUTIVE OPTIONS, TERMINAL DRIVER OPTIONS, and SYSTEM OPTIONS, answering questions appropriately.

5. When you reach the PERIPHERAL OPTIONS section, SYSGEN asks you questions that pertain only to the MSCP devices on your system. (Unless you indicated that you wished to override other autoconfigure results when you responded to the Devices question, SYSGEN asks questions on those devices.)

   The first question requests information about the controller's interrupt vector address, LSI-11 bus address, the number of DU-type disk drives (there is no default value for this parameter), the number of command rings, and the number of response rings. The question is asked twice, once for controller 0 and once for controller 1, because we have specified two DU-type controllers. The dialog uses the abbreviation contr to indicate controller.

```
* DU contr 0 [D:154,172150,,4,4]
154,172150,3,4,4<return>
```

The standard vector address for MSCP controllers is 154_8. The vector for a second unit should be allocated from floating vector address space. Any unused vector between 300_8 and 774_8 can be allocated. See Appendix A for a description of DEC's algorithm for assigning floating vectors.

The standard LSI-11 bus address for MSCP controllers is 772150_8. Emulex recommends that second unit be located at in floating LSI-11 bus address space. See Appendix A for a description of the DEC algorithm for assigning floating addresses.

The number of DU-type disk drives depends on the configuration that you have selected for the QD32, or on the number of drives that are attached to a DEC MSCP controller.

When you select a configuration for the QD32, you are taking into account the number of physical disk drives that you are attaching to the QD32's SMD interface. When you select a configuration, you are also specifying a logical arrangement for the QD32 MSCP subsystem. Some configurations split one physical drive into two logical drives to make file management easier. You determine the configuration of each SMD disk drive when you program the QD32's NOVRAM; see Section 6.

The following types of disk drives can be attached to DEC MSCP controllers:

- RX50
- RD51
- RD52
- RD53
- RC25
- RA60
- RA80
- RA81

The RX50 drive contains two 5.25-inch floppy diskettes; count an RX50 as two drives. The RC25 has both fixed and removable hard media; count an RC25 as two drives.

RSX-11M supports up to eight command and eight response rings; the number of command and response rings that you specify depends on your application. Four command and four response rings are reasonable and adequate for most applications. For further information about command and response rings, refer to the MSCP documentation listed in subsection 6.3 of this manual. In most instances, further information is not required to install the QD32.

6. SYSGEN then asks you to specify the type of disk drive(s) on each controller:

```
* DU contr 0 unit 0. is an RA60/80/81/RC25/RD51/RX50
[D:RA81]<return>
```

For the DEC MSCP controller, indicate the appropriate peripherals.

For the QD32, indicate that you have one RA81 for each logical disk drive.

RSX-11M does not tolerate gaps in sequence; the unit numbers must be contiguous. In addition, the unit numbers specified for each controller must be the same as those reported by the controller during initialization.

7. Complete the SYSGEN procedure according to DEC documentation.

3.5.4 RSX-11M-PLUS OPERATING SYSTEMS (V2.1 and above)

RSX-11M-PLUS SYSGEN is an interrogative program that allows a complete, running RSX-11M-PLUS system to be configured for a particular hardware environment. SYSGEN is well documented in the RSX-11M-PLUS System Generation and Installation Guide, and you are expected to rely primarily on that manual. This explanation is provided only to remove some ambiguities that the installation of the QD32 may involve.

SYSGEN supports autoconfigure, and MSCP controllers are detected by autoconfigure. However, autoconfigure detects only the MSCP controller that is located at the standard LSI-11 bus address. Additional MSCP controllers at alternate addresses must be attached to the operating system manually.

3.5.4.1 Installing a Single MSCP Controller

If you have only one QD32, install it at the standard address (772150_8) and use autoconfigure to connect your peripherals. The procedure given in the RSX-11M-PLUS System Generation and Configuration Guide is adequate for this purpose.

3.5.4.2 Installing Multiple MSCP Controllers

If your initial system configuration includes two MSCP controllers, connect the alternate MSCP controller to the operating system during the initial SYSGEN. We recommend that you use autoconfigure to connect the first controller at the standard address (772150_8). We recommend that the DEC MSCP controller be installed at the standard LSI-11 bus address; locating the QD32 at the alternate LSI-11 bus address does not prevent its being used as the system device.

If you are adding the second MSCP controller to the system configuration, use the Add a Device option of SYSGEN or a complete SYSGEN. The following procedure describes the Add a Device process (user input is in **boldface** print:

1. Invoke SYSGEN.

```
> SET /UIC=[200,200]<return>
> @SYSGEN<return>
```

2. To indicate that you want to do a subset of the SYSGEN procedure, answer **N** (no) to the following questions:

```
*    Do you want to do a complete SYSGEN?
     [Y/N D:Y]: N<return>

*    Do you want to continue a previous SYSGEN
     from some point? [Y/N D:Y]: N<return>
```

3. To indicate that you want to execute a specific module of the SYSGEN procedure, answer **Y** (yes) to this question:

```
*    Do you want to do any individual sections
     of SYSGEN? [Y/N D:Y]: Y<return>
```

4. Select the Add a Device section of SYSGEN:

```
*    Which sections would you like to do?
     [S R:0.-15.]: H<return>
```

   Type the letter H to select the Add a Device section. SYSGEN now asks you all of the questions in the Choosing Peripheral Configuration section.

The questions that SYSGEN asks pertain to the type and number of controllers that are installed on your system. There is one question for each type of controller that RSX-11M-PLUS can support. Answer 0 (zero) for all types of controllers until you are prompted for the number of UDA-type devices.

5. When you are asked to specify the number of MSCP-type devices, answer appropriately:

   * How many MSCP disk controllers do you have? [D R:0.-63. D:0.] **2<return>**

6. Give the total number of MSCP disk drive (on all controllers) installed on the system.

   * How many MSCP disk drives do you have? [D R:0.-n. D:1.] **5<return>**

   The answer to this question depends on the configuration that you have selected for the QD32 and on the number of drives that are attached to any DEC MSCP controllers.

   When you select a configuration for the QD32, you are taking into account the number of physical disk drives that you are attaching to the QD32's SMD interface. When you select a configuration, you are also specifying a logical arrangement for the QD32 MSCP subsystem. Some configurations split one physical drive into two logical drives to make file management easier. You determine the configuration of each SMD disk drive when you program the QD32's NOVRAM (see Section 6).

   The following types of disk drives can be attached to DEC MSCP controllers:

   - RX50
   - RD51
   - RD52
   - RD53
   - RC25
   - RA60
   - RA80
   - RA81

   The RX50 drive contains two 5.25-inch floppy diskettes; count an RX50 as two drives. The RC25 drive has both fixed and removable hard media; count an RC25 as two drives.

7. SYSGEN then asks you to specify controllers per disk drives.

   ```
   *    To which DU controller is DU0: connected?
        [S R:1-1]: A<return>
   ```

   This question is asked as many times as the number of MSCP drives that you have indicated are on the system. RSX-11M-PLUS does not tolerate gaps in sequence; the MSCP unit numbers must be contiguous. In addition, the unit numbers specified for each controller must be the same as those reported by the controller during initialization. Use A for the primary controller and B for the alternate controller.

8. Enter the vector address for each MSCP controller:

   ```
   *    Enter the vector address of DUA
        [O R:60-774 D:154]
   ```

   The standard vector address for MSCP controllers is 154_8. The vector for a second unit should be allocated from floating vector address space. Any unused vector between 300_8 and 774_8 can be allocated. See Appendix A for a description of DEC's algorithm for assigning floating vectors.

9. Enter the CSR address for each MSCP controller:

   ```
   *    What is its CSR address?
        [O R:160000-177700 D:172150]
   ```

   The standard CSR address for MSCP controllers is 772150_8. Emulex recommends that the second unit be located in floating CSR address space. See Appendix A for a description of the DEC algorithm for assigning floating addresses.

10. Specify the number of command rings for each MSCP controller:

    ```
    *    Enter the number of command rings for DUA
         [D R:1.-8. D:4.] 4<return>
    ```

    RSX-11M-PLUS supports up to eight command rings. The value you specify depends on your application. Four command rings are reasonable and adequate for most applications. For further information about command and response rings, refer to the MSCP documentation listed in subsection 6.3 of this manual. In most instances, further information is not required to install the QD32.

10. Specify the number of response rings for each MSCP controller:

```
*    Enter the number of response rings for DUA
     [D R:1.-8. D:4.] 4<return>
```

RSX-11M-PLUS supports up to eight response rings. The value you specify depends on your application. Four response rings are reasonable and adequate for most applications.

3.5.5 MicroVMS OPERATING SYSTEMS

MicroVAX/MicroVMS supports MSCP controllers at the standard address, 772150_8, and in floating address space. MicroVMS has a software utility called SYSGEN which can be used to determine the LSI-11 bus address and interrupt vector address for any I/O devices to be installed on the computer's LSI-11 bus. A running MicroVAX/MicroVMS computer system is required in order to use this utility.

If you do not have access to a running system, you must determine the LSI-11 bus addresses and vector addresses manually (although autoconfigure can still be used to connect the devices to the computer automatically on power-up). See Appendix A for a description of the algorithm used by SYSGEN to determine LSI-11 bus addresses.

The following procedure tells how to use MicroVMS SYSGEN to determine LSI-11 bus addresses and interrupt vectors.

1. Login to the system manager's account. Run the SYSGEN utility:

```
$ RUN SYS$SYSTEM:SYSGEN<return>
SYSGEN>
```

The SYSGEN> prompt indicates that the utility is ready to accept commands.

2. Obtain a list of devices already installed on the MicroVAX LSI-11 bus by typing:

```
SYSGEN> SHOW/CONFIGURATION<return>
```

```
Name: PUA  Units: 1  Nexus: 0  CSR: 772150   Vector1: 154  Vector2: 000
Name: TTA  Units: 1  Nexus: 0  CSR: 760100*  Vector1: 300* Vector2: 304*
Name: TXA  Units: 1  Nexus: 0  CSR: 760500*  Vector1: 310* Vector2: 000

*Floating address or vector
Note: All addresses and vectors are expressed in octal notation.
```

Figure 3-3. Sample SHOW CONFIGURATION

SYSGEN lists by logical name the devices already installed on the LSI-11 bus. Make a note of these other devices with floating addresses (greater than 760000_8) or floating vectors (greater than 300_8) that you plan to re-install with your QD32.

3. To determine the LSI-11 bus addresses and vectors that autoconfigure expects for a particular device type, execute the CONFIGURE command:

```
SYSGEN> CONFIGURE<return>
DEVICE>
```

Specify the LSI-11 bus devices to be installed by typing their LSI-11 bus names at the DEVICE prompt (the device name for MSCP controllers under MicroVMS is UDA).

```
DEVICE> UDA,2<return>
DEVICE> DHV11<return>
DEVICE> DZ11<return>
```

A comma separates the device name from the number of devices of that type to be installed. The number of devices is specified in decimal.

In addition to the QD32, you need only specify devices that have floating addresses or vectors. Devices with fixed addresses or vectors do not affect the address or vector assignments of devices with floating addresses and vectors.

4. Indicate that all devices have been entered by pressing the <ctrl> and Z keys simultaneously:

```
DEVICE> ^Z
```

SYSGEN lists the addresses and vectors of the devices entered in the format shown in Figure 3-4.

```
SYSGEN>  CONFIGURE
DEVICE> DZ11
DEVICE> DHV11
DEVICE> UDA,2
DEVICE> ^Z
Device: UDA     Name: PUA    CSR: 772150     Vector: 154     Support: yes
Device: DZ11    Name: TTA    CSR: 760100*    Vector: 300*    Support: yes
Device: UDA     Name: PUB    CSR: 760354*    Vector: 310     Support: yes
Device: DHV11   Name: TXA    CSR: 760500*    Vector: 320     Support: yes

*Floating address or vector
Note: All addresses and vectors are expressed in octal notation.
```

Figure 3-4. CONFIGURE Command Listing

5. Note the CSR addresses listed for the LSI-11 bus devices in floating address space. Program the listed addresses into non-Emulex devices as instructed by the manufacturer's documentation. For the QD32, program the address given for the QD32 (lowest numerical address) into the board as described in subsection 4.3.1.

6. Complete SYSGEN according to the DEC documentation.

If you want to select a non-standard address for the QD32, that is one that differs from the address selected by the CONFIGURE command, you must enter CONNECT statements in the SYCONFIG.COM file that is in the system manager's account, SYS$MANAGER. Use the syntax of the CONNECT statements as described in the DEC documentation on MicroVMS SYSGEN.

NOTE

> Do not alter the STARTUP.COM or UVSTARTUP.COM command files in the main system account, SYS$SYSTEM.

3.5.6 ULTRIX-32m OPERATING SYSTEMS

The Emulex MSCP class disk disk subsystems emulate the DEC DSA UDA-50/KDA-50/RA81 (MSCP) disk subsystem. They report that they are of controller type "DU" and of device type "RA81". However, when asked for the number of logical blocks, they do not return a size value that matches that of a "real" DEC RA81.

3.5.6.1 The Kernel

Support for MSCP controllers must be included in a monitor when rebuilding the kernel. The configuration file is edited to reflect the number of controllers and the number of drives connected to each controller. The ULTRIX-32m system supports two MSCP disk controllers. The following example shows two controllers, the first with two drives, the second with one:

```
controller    uda0   at uba0 csr 0172150    vector udintr
disk          ra0    at uda0 drive 0
disk          ra1    at uda0 drive 1
controller    uda1   at uba0 csr 0160334    vector udintr
disk          ra2    at uda1 drive 2
```

Ultrix-32m does not require that the units be in sequential order. The units numbers of the drives **MUST** match the specifications in the configuration file. In this example, the first unit on the second controller must be unit number two; regardless of the units on the first controller.

3.5.6.2 Special files

The Ultrix operating system communicates with devices on the system by the use of special files. These files contain pointers into a system table that lists the entry points for a corrasponding driver for that device. There must be a special file for each device (and each partition for disks) on the system in order for Ultrix to communicate with that device. Some devices will have two special files associated with a device: one for use with character mode, and the other for block mode. These special files exist in the account '/dev'.

There is a shell script, called "MAKEDEV" (uppercase important), on the Ultrix-32m system to help build these special files. The format of this command is:

```
% /dev/MAKEDEV device ...
```

This script passes your input to the program "mknod" to create the special files. You should use this command file to create the special files for each disk you wish to connect to the system. An example for two disks is:

```
% /dev/MAKEDEV ra4 ra5
```

This example assumes that you have already added the device into the configuration file, and you chose the logical names ra4 and ra5 for your disks.

3.5.6.3 Autoconfigure

At boot time, Ultrix-32m attempts to auto-configure the devices included in the booted monitors configuration file. If the device was not included in the configuration files, it will not be configured into the running system. If the device is not present, Ultrix will skip it.

When Ultrix-32m finds a device at autoconfigure time it prints a message as follows:

```
rqd0 at csr 172150 vec 774, ipl 17
ra0 at rqd0 slave 0
ra1 at rqd0 slave 0
rqd1 at csr 160334 vec 770, ipl 17
ra2 at rqd1 slave 0
```

The CSR address were set in the configuration file. The vectors are assigned sequentially in reverse order by the operating system. If the CSR or unit numbers don't match the configuration file, the device will be skipped (and no message will be printed).

3.5.6.4 Disk Partitions

Unix/Ultrix allows a user to logically subdivide a disk into sections called "partitions". Disk partitions were created because the first Unix operating systems could access only a limited amount of space on large disks. Disk partitioning lets several Unix file systems reside on the same disk, one file system per partition. This allows the operating system to utilize the entire disk.

Each disk has a partition table that defines the starting location and size (both in blocks) of each partition on that disk. When a disk is first opened by the operating system (for the very first time), it writes the partition size table into the super block of partition "a" (the first partition) on the disk.

3.5.6.5 Disk Partition Modifications

Modifications to a disks partition table is done with the "chpt" command each time a disk is initialized or re-initialized. The "chpt" command allows a system manager to alter a particular partition's location and size characteristic.

The operating system initializes the disks partition table with that of a real DEC RA81's size table (found in the disk driver) on its very first open. The system manager should then edit these sizes (with "chpt" command) to match his/her system needs.

3.5.6.6 Default Partition Modifications

It is also possible to modify the default RA81 partition size table, which is stored in the device driver; this would eliminate the need for editing the partition table each time the disk is initialized.

When DEC reorganized the Berkeley 4.2 Unix system to create Ultrix-32(m) they set it up to allow the distribution of the operating system in a binary format. This allowed them to distribute a minimum amount of source code to binary licence holders. They separated each of the drivers and system kernel modules into two sections: a code portion and a data portion. The code portion does not require recompilation depending on the selected options at sysgen time; this is supplied in object format (xx.**o**). The data portion requires selection parameters based on sysgen answers; this is supplied in source code format (xx**_data.c**). Making changes to this table will alter the default partition size characteristics for new disks. An example of the changes to the 'uda' driver is included here.

/usr/sys/data/uda_data.c:

```
}, ra81_sizes[8] ={
        15884,  0,              /* A=blk 0 thru 15883 */
        66880,  15884,          /* B=blk 15884 thru 82763 */
        -1,     0,              /* C=blk 0 thru end */
        0,      0,              /* D= not used */
        0,      0,              /* E= not used */
        0,      0,              /* F= not used */
        -1,     82764,          /* G=blk 82764 thru end */
        0,      0,              /* H= not used */
};
```

The **-1** above indicates the end of the disk.

3.5.6.7 Newfs

The newfs program speeds up the creating of a file system on a partition. It looks up information, in the file /etc/disktab, on the disk specified by the system manager and creates the file system according to those default values. An example of the changes to the /etc/disktab file have been included here.

/etc/disktab:

```
qd32|QD32|Emulex QD32 Fujitsu Eagle M2351A Winchester:\
        :ty=winchester:ns#47:nt#20:nc#840:\
        :pa#15884:ba#4096:fa#512:\
        :pb#66880:bb#4096:fb#512:\
        :pc#789600:bc#4096:fc#1024:\
        :pg#706836:bg#4096:fg#1024:
```

3.5.6.8 Suggestions/Warnings

There is a maximum of eight partitions per disk. The partitions form logical boundaries on the disk, separating each file system from all others. These logical divisions are useful for disk management because you can put similar types of users, files, directories or projects all on the same file system. Because a file system can never exceed its partition in size, you can use partitions to regulate disk use.

There are certain areas of the disk which, by default, is reserved for the operating system. By mounting the swap space, for example, on its own partition, important data can not be overwritten when data from memory is swapped to the disk. The Ultrix-32m systems use partition "b" for the swap file. If you plan to use your own partition values, be sure to allocate an area on your system disk for a swap file.

For more information on disk partitioning and modifications to the partition sizes, see Appendix C "Disk Partitioning" in the Ultrix-32m System Manager's Guide.

The program "diskpart" is used to create entries for the disk driver or for the "disktab" file. It creates a template based on the default rules used at Berkeley. The following is a table defining the Berkeley defaults:

Partition	**20-60 MB**	**61-205MB**	**206-355 MB**	**356+ MB**
a	15884	15884	15884	15884
b	10032	33440	33440	66880
c *	all	all	all	all
d	15884	15884	15884	15884
e	unused	55936	55936	307200
f *	unused	end	end	end
g *	end	end	end	end

* The 'c' partition is, by convention, used to access the entire disk. In normal operation, either the 'g' partition is used, or the 'd', 'e', and 'f' partitions are used. The 'f' and 'g' partitions are variable sized, occupying whatever space remains after allocation of the fixed sized partitions.

NOTE

Ultrix-32m V1.0 did not support self sizing disks and is therefore unusable with the Emulex MSCP controllers. The "diskpart" program was not included on the Ultrix-32m V1.1 distribution kit.

BLANK

4.1 OVERVIEW

The procedure for installing the QD32 Disk Controller is described in this section. The subsection titles are listed below to serve as an outline of the procedure.

Subsection	Title
4.2	Inspection
4.3	Disk Controller Setup
4.4	Physical Installation
4.5	SMD Drive Preparation
4.6	Cabling
4.7	Integration and Operation

If you are unfamiliar with the subsystem installation procedure, Emulex recommends reading this Installation Section before beginning.

4.1.1 SUBSYSTEM CONFIGURATIONS

This section is limited to switch setting data and physical installation instructions. No attempt is made to describe the many subsystem configurations that are possible. **If you are not familiar with the possible configurations, we strongly recommend reading Section 3, PREPARING THE INSTALLATION, before attempting to install this subsystem.**

When you are installing the subsystem, you should make a record of the subsystem configuration and environment. Figure 4-1 is a Configuration Record Sheet that lists the information required and shows where the data can be found. This information will be of help to an Emulex service representative should your subsystem require service.

QD32 CONFIGURATION REFERENCE SHEET

GENERAL INFORMATION

- Host computer type
- Host computer operating system
  Version
- Other MSCP Controllers; Type , LSI-11 Bus Address

DRIVE CONFIGURATION PARAMETERS

- Drive Manufacturer (0) Model
- Drive Manufacturer (1) Model
- NOVRAM Parameters:

DRIVE 0

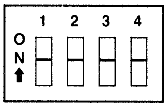
1	Number Units	
2	Type Code	
3	Head Offset	
4	Sectors/Track	
5	Heads	
6	Cylinders	
7	Spare Sectors	
8	Alternate Cylinders	
9	Configuration Bits	
10	Split Code	
11	Removable Media	
12	Gap 0	
13	Gap 1	
14	Gap 2	
15	Cylinder Offset	
16	Spiral Offset	

DRIVE 1

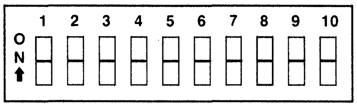
1	Number Units	
2	Type Code	
3	Head Offset	
4	Sectors/Track	
5	Heads	
6	Cylinders	
7	Spare Sectors	
8	Alternate Cylinders	
9	Configuration Bits	
10	Split Code	
11	Removable Media	
12	Gap 0	
13	Gap 1	
14	Gap 2	
15	Cylinder Offset	
16	Spiral Offset	

QD32/D CONFIGURATION

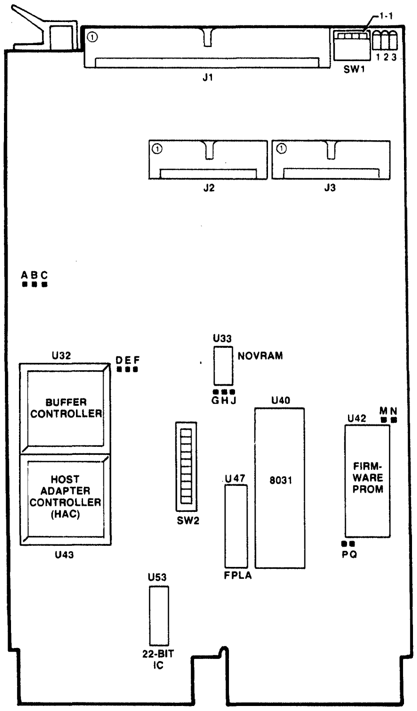
- Firmware revision number
- Warranty expiration date
- Top assembly number
- Serial number
- LSI-11 bus address
- Interrupt vector address
- 22-bit addressing IC (7438) installed (Y or N)
- Switch settings (□ = OFF ■ = ON)



SW1



SW2



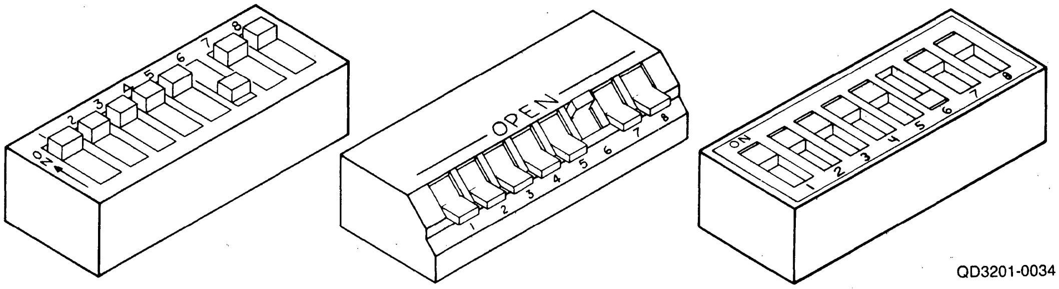
QD3201-0685

Figure 4-1. QD32 Configuration Reference Sheet

4.1.2 DIP SWITCH TYPES

Switch-setting tables in this manual use the numeral one (1) to indicate the ON (closed) position and the numeral zero (0) to indicate the OFF (open) position.

The three DIP switch types used in this product are shown in Figure 4-2. Each is set to the code shown in the switch setting example.



SW1							
1	2	3	4	5	6	7	8
1	1	1	1	1	0	1	1

Figure 4-2. Switch Setting Example

4.1.3 MAINTAINING FCC CLASS A COMPLIANCE

Emulex has tested the QD32 Intelligent Disk Controller with DEC computers that comply with FCC Class A limits for radiated and conducted interference. When properly installed, the QD32 does not cause compliant computers to exceed Class A limits.

There are two possible configurations in which the QD32 and its associated SMD peripherals can be installed:

- With both the QD32 Disk Controller and the SMD disk drives both mounted in the same cabinet, and

- With the QD32 mounted in the CPU cabinet and the SMD drives mounted in a separate cabinet.

To limit radiated interference, DEC completely encloses the components of its computers that generate or could conduct radio-frequency interference (RFI) with a grounded metal shield (earth ground). During installation of the QD32, nothing must be done that would reduce this shield's effectiveness. That is, when the QD32 installation is complete, no gap in the shield that would allow RFI to escape can be allowed.

Conducted interference is generally prevented by installing a filter in the AC line between the computer and the AC outlet. Most power distribution panels that are of current manufacture contain suitable filters.

The steps that must be taken to maintain the integrity of the shield and to limit conducted interference are explained fully in subsection 4.1.2.

4.2 INSPECTION

Emulex products are shipped in special containers designed to provide full protection under normal transit conditions. Immediately upon receipt, the shipping container should be inspected for evidence of possible damage incurred in transit. Any obvious damage to the container, or indications of actual or probable equipment damage, should be reported to the carrier company in accordance with instructions on the form included in the container.

Unpack the QD32 subsystem and, using the shipping invoice, verify that all equipment is present. Verify also that model or part numbers (P/N), revision levels, and serial numbers agree with those on the shipping invoice. Paragraph 1.4 explains model numbers and details kit contents. These verifications are important to confirm warranty. If evidence of physical damage or identity mismatch is found, notify an Emulex representative immediately. If the equipment must be returned to Emulex, it should be shipped in the original container.

4.2.1 QD32 DISK CONTROLLER INSPECTION

Visually inspect the QD32 Disk Controller after unpacking. Check for such items as bent or broken connector pins, damaged components, or any other evidence of physical damage.

Examine all socketed components carefully to ensure that they are properly seated.

4.3 DISK CONTROLLER SETUP

Several configuration setups must be made on the QD32 Disk Controller before inserting it into the chassis. These setups are made by option switches SW1 and SW2.

Figure 4-3 shows the locations of the configuration switches referenced in the following paragraphs.

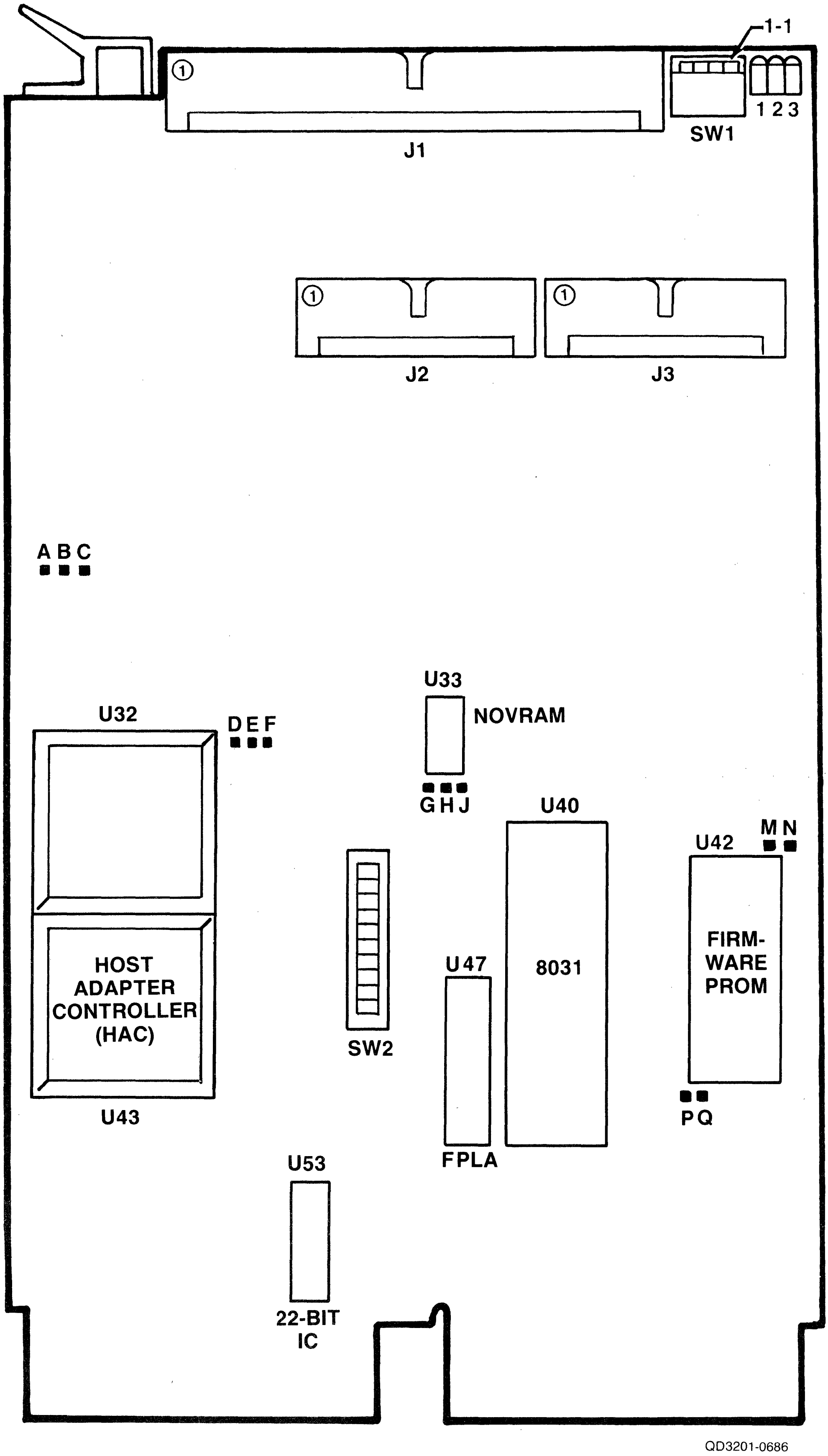


Figure 4-3. QD32 Disk Controller Assembly

NOTE

> If you change a switch position on the QD32, you must also reset the QD32 so that the host operating system's initialization sequence reads the codes established by the switch settings. To reset the QD32, either toggle switch SW1-1 (ON then OFF), or power-down and power-up the system.

Table 4-1 defines the function and factory configuration of all switches on the QD32 controller. The factory configuration switch settings are representative of most QD32 Disk Controller applications.

Table 4-1. QD32 Switch Definitions and Factory Configuration

SW	OFF(0)	ON(1)	Fact	Function	Section
SW1-1	Run	Reset/Halt	OFF(0)	Run vs. Reset/Halt	
SW1-2	-	-	OFF(0)	MSCP Device Number (LSB)	4.3.3.2
SW1-3	-	-	OFF(0)	MSCP Device Number	4.3.3.2
SW1-4	-	-	OFF(0)	MSCP Device Number (MSB)	4.3.3.2
SW2-1	Disable	Enable	OFF(0)	Loop on Self-Test Error	
SW2-2	Disable	Enable	OFF(0)	Automatic Bootstrap	4.3.3.1
SW2-3	-	-	OFF(0)	LSI-11 Bus Address	4.3.1
SW2-4	-	-	OFF(0)	LSI-11 Bus Address	4.3.1
SW2-5	-	-	OFF(0)	LSI-11 Bus Address	4.3.1
SW2-6	18-bit	22-bit	OFF(0)	Twenty-Two-Bit Addressing	4.3.3.3
SW2-7	4 µsec	8 µsec	OFF(0)	DMA Burst Delay	4.3.3.4
SW2-8	-	-	OFF(0)	Reserved	
SW2-9	B Cable	A Cable	OFF(0)	Index and Sector	4.3.3.5
SW2-10	-	-	OFF(0)	Reserved	

ON(1) = Closed
OFF(0) = Open

NS = No standard
Fact = Factory switch setting

Table 4-2 lists the function and factory configuration of all jumpers on the controller.

Table 4-2. QD32 Jumper Definitions and Factory Configuration

Jumper	OUT	IN	FACT	Comment
A-B-C	Not Used	Not Used	OUT	Must be OUT
D-E-F	Not Used	Not Used	OUT	Must be OUT
G	Normal Operation	Ground (Test)	OUT	Must be OUT
H-J	Disable Clock	Enable Clock	IN	Must be IN
M-N	16K PROM Select	32K PROM Select	OUT	Must be OUT
P-Q	Normal Operation	Factory Test	OUT	Must be OUT

Cntrl = Controller
FACT = Factory Setting

4.3.1 DISK CONTROLLER BUS ADDRESS

Every LSI-11 bus I/O device has a block of several registers through which the system can command and monitor that device. The registers are addressed sequentially from a starting address assigned to that controller, in this case an MSCP-class Disk Controller.

The address for the first of the QD32's two LSI-11 bus registers is selected by DIP switches SW2-3 through SW2-5. See Table 4-3 for register address switch settings. For more information on determining the LSI-11 bus address, see Section 3 and Appendix A.

Table 4-3. Controller Address Switch Settings

Bus Address (in octal)	-- SW2 -- 3	4	5	Factory
772150	0	0	0	✓
772154	1	0	0	
760334	0	1	0	
760340	1	1	0	
760344	0	0	1	
760350	1	0	1	
760354	0	1	1	
760360	1	1	1	

4.3.2 INTERRUPT VECTOR ADDRESS

The interrupt vector address for the QD32 is programmed into the device by the operating system during power-up. See subsection 3.5 for a discussion of vector addresses.

4.3.3 OPTIONS

There are other QD32 options that can be implemented by the user. These features are selected by physically installing the option on the PCBA or by enabling the option using a switch.

4.3.3.1 Automatic Bootstrapping

The automatic bootstrapping option causes the system to boot automatically from logical unit 0 through 3 on power-up when the QD32 is at the standard base address. To enable this option, set SW2-2 ON and set switches SW1-2 through SW1-4 as described in Table 4-4. This option cannot be enabled with a MicroVAX or in a system that uses an 11/73B CPU module.

Switch	OFF	ON	Factory
SW2-2	Disable	Enable	OFF

The automatic bootstrapping process requires that the LSI-11 CPU module be configured for power-up mode 0. The following table lists the configuration settings for several popular LSI-11 CPUs.

CPU	Configuration Setting
11/73A	Install W3 and W7
11/23+	Remove J18-J19 and J18-J17
11/23	Remove W5 and W6
11/02	Remove W5 and W6

If the bootstrap device is not powered-up or safe (e.g., it failed its self-test), the autoboot routine in the QD32 halts the CPU after 1 minute. This causes the CPU to enter Console ODT. You can then examine the Status and Address (SA) register (base address plus 2) for an MSCP error code (Table 5-3) and bootstrap the system from another device.

4.3.3.2 MSCP Device Number

QD32 switches SW1-2 through SW1-4 specify MSCP device numbers. The functions of these switches are dependent on the options you select for your QD32:

- If the QD32 is installed at the standard LSI-11 bus address, these switches identify the MSCP device number of the drive from which to bootstrap. The QD32 automatic bootstrap option supports only MSCP units 0 through 3 at the standard address. See subsection 4.3.3.2.1.

- If the QD32 is installed at an alternate LSI-11 bus address, these switches identify the MSCP device number of the first drive supported by that alternate QD32. The first drive supported by the QD32 at alternate address may be drive 1 through 8. See subsection 4.3.3.2.2.

4.3.3.2.1 Logical Unit to Autoboot From

If the QD32 automatic bootstrapping option is enabled (SW2-2 ON) and the QD32 is at the standard LSI-11 bus address (772150_8), switches SW1-2 through SW1-4 define the MSCP device number of the drive from which the QD32 bootstraps. By using these switches, you may select one of four logical units to bootstrap from. Table 4-4 defines the MSCP device numbers selected by switches SW1-2 through SW1-4 if the QD32 is at a standard address.

Table 4-4. Bootstrap MSCP Device Number

Bootstrap MSCP Device Number	----- SW1 ----- 2 (LSB)	 3	 4 (MSB)	Factory
0	0	0	0	√
1	1	0	0	
2	0	1	0	
3	1	1	0	

4.3.3.2.2 First Logical Unit Number for an Alternate QD32

If your QD32 is installed at an alternate address, switches SW1-2 through SW1-4 select the MSCP device number of the first drive supported by the QD32. MSCP device numbering schemes may vary by DEC operating system (see subsection 3.4.2.2). Table 4-5 defines the MSCP device numbers selected by switches SW1-2 through SW1-4 if the QD32 is at an alternate address.

Example 4-1: Your system operates under RSX-11M-PLUS and has two QD32 Disk Controllers. The first QD32 is at the standard base for MSCP controllers, 772150_8, and it supports three logical drives. Unit 0. Unit 1, and Unit 2. The second QD32 is at an alternate base address, and it supports two logical drives. RSX-11M-PLUS requires that the first drive on the alternate QD32 have an MSCP device number of 3 and that the second drive have an MSCP device number of 4. On the alternate QD32. set switches SW1-2 in the ON position, SW1-3 in the ON position, and SW1-4 in the OFF position to specify a MSCP device number of 3 for the first drive.

This example would also apply if the first MSCP controller were a DEC MSCP with three logical drives.

Table 4-5. MSCP Device Number for the First Drive Supported by a QD32 at an Alternate Address

Starting MSCP Device Number	-----	SW1	-----
	2 (LSB)	3	4 (MSB)
1	1	0	0
2	0	1	0
3	1	1	0
4	0	0	1
5	1	0	1
6	0	1	1
7	1	1	1
8	0	0	0

4.3.3.3 22-Bit Memory Addressing

Twenty-two-bit addressing capability is a standard option for the QD32 The QD32 22-Bit Addressing Kit, part number QD0113002, is supplied with the QD32. To enable 22-bit addressing, install the single 7438 IC in socket U53 on the QD32 PCBA and set SW2-6 ON (1).

CAUTION

> Some manufacturers of LSI-11 bus backplanes use the backplane lines now devoted to extended addressing for power distribution. Installing a QD32 with the extended addressing option in such a system **will** damage the option IC. Before installing the option, confirm that there is neither positive nor negative potential between lines BC1, BD1, BE1, BF1, and logic ground. A QD32 without the extended addressing option will not be damaged if power is present on those lines.

4.3.3.4 DMA Burst Delay

The QD32 firmware design includes a switch-selectable DMA burst delay to avoid data late conditions. Switch SW2-7 selects either a 4-microsecond or 8-microsecond delay between DMA bursts. Even with the QD32 adaptive DMA, some applications may require a longer burst delay to allow other devices adequate time on the bus.

Switch	OFF	ON	Factory
SW2-7	4 usec	8 usec	OFF

4.3.3.5 Index and Sector

The QD32 uses switch SW2-9 to determine whether the controller will look for the index and sector pulses on the A cable or B cable.

Switch	OFF	ON	Factory
SW2-9	B cable	A cable	OFF

4.4 PHYSICAL INSTALLATION

4.4.1 SYSTEM PREPARATION

To prepare your CPU to accept the QD32, use the following procedures:

MICRO/PDP-11 and MicroVAX I and II Preparation:

1. Power down the system by switching OFF the main AC breaker.
2. Remove the rear cover from the chassis so that the patch panel is exposed. The rear cover is held on by snap pads. Grasp the cover at the top and bottom, and pull straight back.
3. Loosen the captive screws from the patch panel using a standard screwdriver.
4. Remove the patch panel.
5. Find the flat-ribbon cable that connects the CPU module to the patch panel. For easier board installation, you may disconnect the CPU flat-ribbon cable from the patch panel.

LSI-11 Series Preparation:

1. Power down the system by switching OFF the main AC breaker.
2. Remove the cover from the chassis so that the backplane is exposed.

Do not replace the covers or patch panels until the installation is verified (subsection 4.7).

4.4.2 SLOT SELECTION

The QD32 may be assigned to any desired slot because it uses the LSI four-level interrupt scheme to perform distributed interrupt arbitration.

Be sure to find out whether your backplane is straight or serpentine and choose a slot accordingly. On straight backplanes, the QD32 must be plugged into connectors A and B, since connectors C and D carry no signals. On a serpentine backplane, the QD32 can be plugged into either connectors A and B or connectors C and D.

There must be no unused slots, however, between the CPU and the QD32. If you have a DEC RQDX1 in your backplane, be sure to install the QD32 in front of the RQDX1; not all RQDX1 controllers pass grant signals.

4.4.3 MOUNTING

The QD32 Disk Controller PWB should be plugged into the LSI-11 backplane with components oriented in the same direction as the CPU and other modules. Always insert and remove the boards with the computer power OFF to avoid possible damage to the circuitry. Be sure that the board is properly positioned in the throat of the board guides before attempting to seat the board by means of the extractor handle.

4.5 SMD DISK DRIVE PREPARATION

The disk drive(s) must be configured for the proper number of sectors and have an ID plug or address selection switches properly configured.

4.5.1 DRIVE PLACEMENT

Uncrate and install the disk drives according to the manufacturer's instructions. Position and level the disk drives in their final places before beginning the installation of the QD32. This positioning allows the I/O cable routing and length to be accurately judged.

4.5.2 LOCAL/REMOTE

The local/remote switch controls whether the disk drive can be powered up from the drive (local) or from the controller (remote). Place the switch in the REMOTE position. With the CPU powered down, press the Start switch on the front panel of each of the drives (the Start LED will light, but the drive will not spin up and become ready). When the CPU is powered up, the drives spin up sequentially. This sequential power-up prevents the heavy current draw that would be caused if all the drives were powered up at once. When in the remote mode, the drives power down when the CPU is powered down. While the CPU is powered on, the drives can be powered up and down individually (to change disk media, for example) by using the drive Start switch.

4.5.3 SECTORING

See Appendix D, Drive Parameters, for the correct sector count settings for the disk drives in use. Consult the appropriate drive manual for additional information.

4.5.4 DRIVE NUMBERING

An address from 0 to 1 must be selected for each drive. Be careful that no two drives are assigned the same number. The logical unit number is determined by the address given to the drive. See subsection 3.2.2.

CDC drive addresses are selected by means of an ID plug. Drives from other manufacturers have their addresses selected by switches on one of the logic cards. Consult the appropriate drive manual for the exact procedure.

4.5.5 SECTOR AND INDEX SIGNALS

The QD32 supports SMD drives that provide the index and sector pulses on either the A or B cable without drive modification. Switch SW2-9 indicates the signal location (see subsection 4.3.3.5).

4.6 CABLING

The QD32 Disk Controller interfaces with each disk drive that it controls via one control cable (A cable) and a data cable (B cable). The A cable originates from connector J1 on the QD32 and is daisy-chained to all of the supported drives, terminating on the last drive. Maximum cumulative cable length for the A cable is 35 feet (10.5 meters). The B cables originate from connectors J2 and J3. A radial (B) cable is connected directly from the QD32 to each supported disk drive. Maximum cable length for the B cable is 25 feet (7.5 meters).

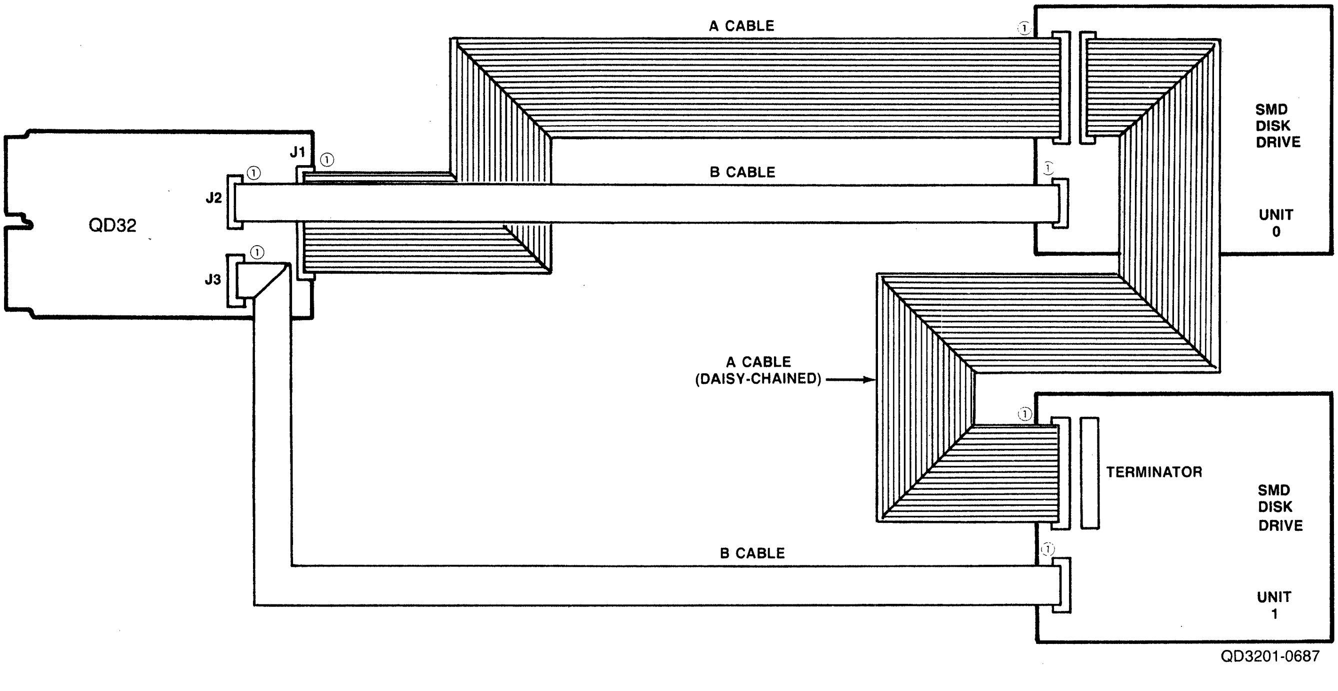


Figure 4-4. Drive Cabling

Table 4-6 and 4-7 list cables offered by Emulex for the QD32. Figure 4-4 shows basic cable installation. In addition, Emulex offers the QD32 Cabling Kit (P/N QD3113003) for installation in a MICRO/PDP or MicroVAX I or II Tower. The QD32 Cabling Kit, which includes cables and panels for separate cabinet installation, is documented in the QD32 Cabling Kit Instruction Sheet (QD3252401).

To prevent excessive RFI, DEC surrounds its computers with a grounded metal shield. These shields are built into the computer cabinet.

As noted in subsection 4.1.3, cabling has a direct effect on the amount of electromagnetic interference radiated by a computer system. When installing the QD32 and its drives, you must take steps to preserve the integrity of the shield built into FCC-compliant DEC cabinets.

If both the controller and the peripheral are installed in the same cabinet, then you need only replace the shields that you have removed to keep the computer compliant with FCC regulations.

If the controller and the disk drives are located in separate cabinets, then you must shield the cables that run between the cabinets. Also, you must install the cables so that their points of cabinet exit and entry do not cause the computer installation to exceed FCC limits for RFI. To allow you to do this easily, Emulex makes shielded A cables, B cables, and cable I/O adapter panels that are designed to keep RFI within FCC limitations.

As previously noted, the QD32 and its SMD disk drives can be installed in either of two configurations:

- With the QD32 Disk Controller and the drives mounted in the same cabinet

- With the QD32 mounted in the CPU cabinet and the drives mounted in a separate cabinet

The following paragraphs describe the cabling between the QD32 and drives for both these configurations. The separate-cabinet installations rely on Emulex cabling kits to limit EFI, and thus the procedures for installing the kits are described in subsection 4.6.2.

4.6.1 SAME CABINET INSTALLATIONS

When the QD32 and its SMD drives are installed in the same cabinet, it is possible that the cabinet itself provides sufficient shielding. In such cases, it is not necessary to shield the A and B cables that connect the subsystem.

NOTE

> If the cabinet in which the QD32 and CPU are installed was manufactured before 1 October 1983, it may not provide sufficient shielding or filtering to prevent excessive RFI radiation or conduction. In case of complaint, it is the operator's responsibility to take what ever steps are necessary to correct the interference.

Emulex makes unshielded A and B cables in several lengths. Table 4-6 lists the available cables.

Table 4-6. Unshielded Cables

Item	Part Number	Description	Length	Interface
1	SU1111201	Cable, Unshielded	8 ft	SMD A-Cable
	SU1111203	Cable, Unshielded	15 ft	SMD A-Cable
	SU1111205	Cable, Unshielded	25 ft	SMD A-Cable
	SU1111207	Cable, Unshielded	35 ft	SMD A-Cable
2	SU1111202	Cable, Unshielded	8 ft	SMD B-Cable
	SU1111204	Cable, Unshielded	15 ft	SMD B-Cable
	SU1111206	Cable, Unshielded	25 ft	SMD B-Cable

The items listed in Table 4-6 can be ordered from your Emulex sales representative or directly from the factory. The factory address is:

Emulex Customer Service
3545 Harbor Boulevard
Costa Mesa, CA 92626
(714) 662-5600   TWX 910-595-2521

To cable the subsystem, see Figure 4-4 and use the following procedures:

4.6.1.1 A Cable

1. Look at the header at either end of the A cable. Find the molded-in arrow that identifies pin 1 of the connector.

2. Find the arrow that that is molded into connector J1 on the QD32. Align the arrow on the cable header with the connector arrow and press the header into the connector.

3. Find the molded-in arrow on the cable header at the other end of the A cable.

4. Find the arrow on the disk drive's A cable connector. Most SMD drives have two A cable interfaces to allow daisy-chaining; use either. Align the arrow on the cable header with the connector arrow and press the header into the connector. Make sure that the locking tabs on the connector are fully flush with the sides of the cable header.

5. Connect the first drive to the second drive supported by the QD32 using another A cable. Run the cable from the second A cable connector on the first drive to the second drive using steps 1 through 4.

6. Install the A cable terminator (supplied with the drive) on the second A-cable connector of the last drive.

End of Procedure

4.6.1.2 B Cable

1. Look at the header at either end of the B cable. Find the molded-in arrow that identifies pin 1 of the connector.

2. Find the arrow that that is molded into connector J2 on the QD32. Align the arrow on the cable header with the connector arrow and press the header into the connector.

3. Find the molded-in arrow on the cable header at the other end of the B cable.

4. Find the arrow on the first disk drive's B cable connector. If the drive is a dual port model, make sure you connect the B cable to the same port to which you connected the A cable. Align the arrow on the cable header with the connector arrow and press the header into the connector. Make sure that the locking tabs on the connector are fully flush with the sides of the cable header.

5. Repeat steps 1 through 4 for the other drive, using J3 on the QD32.

End of Procedure

4.6.1.3 Grounding

For proper operation of the disk subsystem, there must be a good connection between the disk drive logic ground and the CPU logic ground. The ground connection should be braid (preferably insulated) that is 0.25 inches wide or wider, or AWG No. 10 wire or heavier. The grounding wire may be daisy-chained between disk drives. If the disk drive has a switch or jumper that connects the logical signal ground to the cabinet ground (DC ground to AC ground), this connection should be removed once the disk drive is placed online. It can be connected for performing local offline maintenance on the disk drive.

NOTE

> Failure to observe proper signal grounding methods generally results in marginal operation with random error conditions.

4.6.2 SEPARATE CABINETS

If the disk drives are mounted in a separate cabinet from the QD32 Disk Controller, then the A and B cables that connect the drives to the QD32 must be shielded because they run outside the shielded cabinet environment.

Emulex makes extension cables, cable I/O adapter panels, and shielded cables that are designed to be used with the QD32 in separate-cabinet installations. Extension cables connect the controller (or drive) with the cable I/O adapter panels. Cable I/O adapter panels ground the shields on the shielded cables and maintain the integrity of the cabinet shield. Shielded cables are run between cabinets.

The cables are available in various lengths, and there are several different types of cable I/O adapter panels. Table 4-7 gives the part numbers of these accessories and describes their application. The items listed in Table 4-7 can be ordered from your Emulex sales representative or directly from the factory. See the address given in subsection 4.6.1.

Table 4-7. Shielded Cables and Cable I/O Adapter Panels

Item	Part Number	Description	Length	Qty Rqd	Application
1	SU7811212-01	Cable, Shielded	4	1	SMD A-Cable
	SU7811212-02	Cable, Shielded	8	1	SMD A-Cable
	SU7811212-03	Cable, Shielded	15	1	SMD A-Cable
	SU7811212-04	Cable, Shielded	25	1	SMD A-Cable
	SU7811212-05	Cable, Shielded	35	1	SMD A-Cable
2	SU7811219-01	Cable, Extension	2	2	SMD A-Cable
	SU7811219-02	Cable, Extension	4	2	SMD A-Cable
	SU7811219-03	Cable, Extension	6	2	SMD A-Cable
	SU7811219-04	Cable, Extension	8	2	SMD A-Cable
	SU7811219-05	Cable, Extension	10	2	SMD A-Cable
3	SU7811213-01	Cable, Shielded	4	1-2	SMD B-Cable
	SU7811213-02	Cable, Shielded	8	1-2	SMD B-Cable
	SU7811213-03	Cable, Shielded	15	1-2	SMD B-Cable
	SU7811213-04	Cable, Shielded	25	1-2	SMD B-Cable
4	SU7811218-01	Cable, Extension	2	2-4	SMD B-Cable
	SU7811218-02	Cable, Extension	4	2-4	SMD B-Cable
	SU7811218-03	Cable, Extension	6	2-4	SMD B-Cable
	SU7811218-04	Cable, Extension	8	2-4	SMD B-Cable
	SU7811218-05	Cable, Extension	10	2-4	SMD B-Cable
	SU7811218-06	Cable, Extension	12	2-4	SMD B-Cable
5	SU1110201	Cable I/O Adapter Panel	NA	2-3	
6	CU2220301	Rack-Mount Panel	NA	2-3	(optional)
7	SU7813104	Peripheral Cable Adapter Panel Kit	NA	1-2	Freestanding disk drive cable adapter

The cable I/O adapter panels are designed to fit directly into the I/O bulkhead that is built into most FCC-compatible DEC CPU expansion cabinets. See Figure 4-5. If there is no I/O bulkhead in the cabinet, item 6, the rack-mount panel, holds two cable I/O adapter panels and mounts in any standard 19-inch RETMA rack. The rack-mount panel is shown in Figure 4-6.

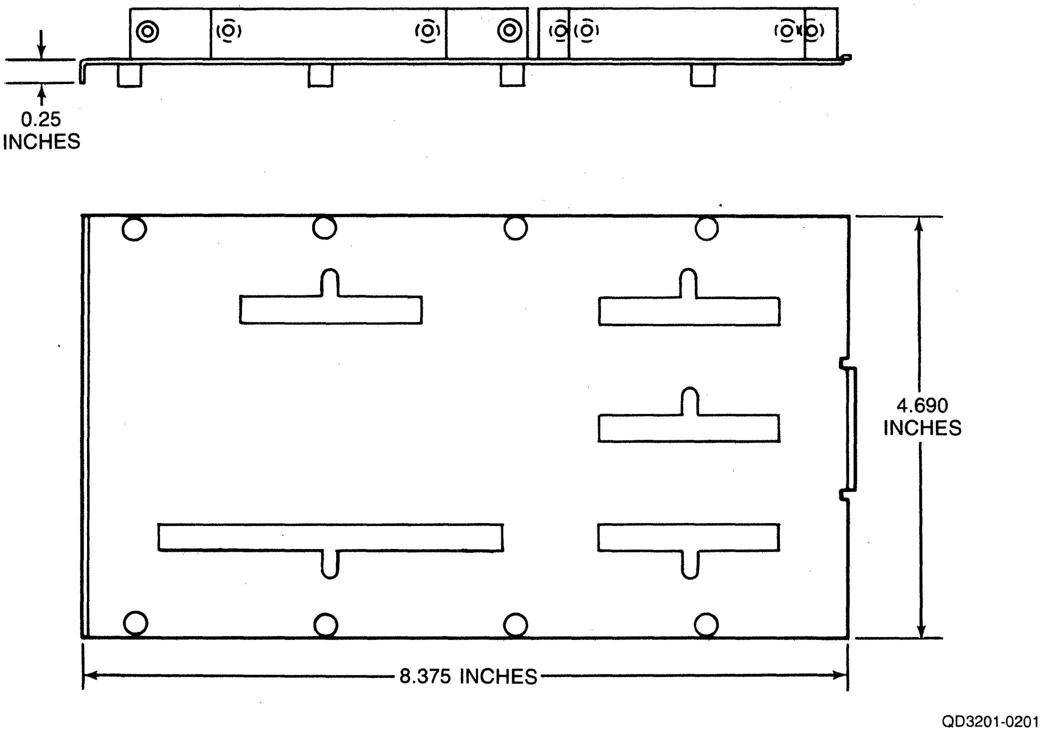


Figure 4-5. Cable I/O Adapter Panel

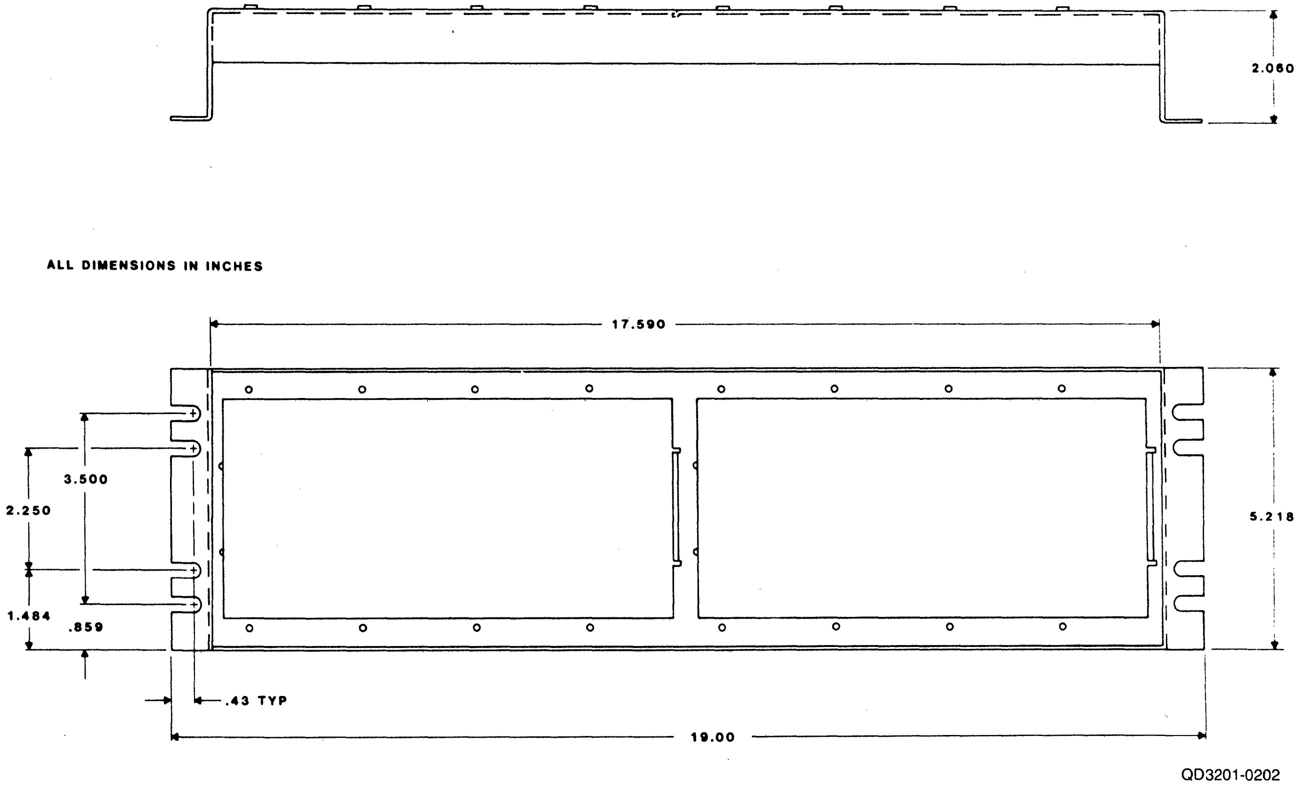


Figure 4-6. Rack-Mount Panel

There are two possible separate cabinet configurations. In one, the drives are mounted in a freestanding RETMA rack (a RETMA rack attached to a PDP CPU cabinet would not count as a separate cabinet installation at all). In the other configuration, the drives have their own cabinets and are themselves freestanding. These two configurations are shown in Figures 4-7 and 4-8. The cabling procedure that follows is a generic one that is for use with either configuration. Consult the illustrations for details.

1. Open the rear bulkhead door or panel of the equipment cabinet.

2. Install the QD32 disk controller in an appropriate CPU bus slot.

3. Install a cable I/O adapter panel in a convenient aperture in the rear bulkhead of the CPU or expansion cabinet and secure it with the eight captive screws. Tighten the screws finger-tight. Make sure that no gaps are present above or below the cable I/O adapter panel.

4. Install a rack-mount panel in the RETMA rack (Figure 4-7) or peripheral cable adapter kit in each of the drive cabinets (Figure 4-8).

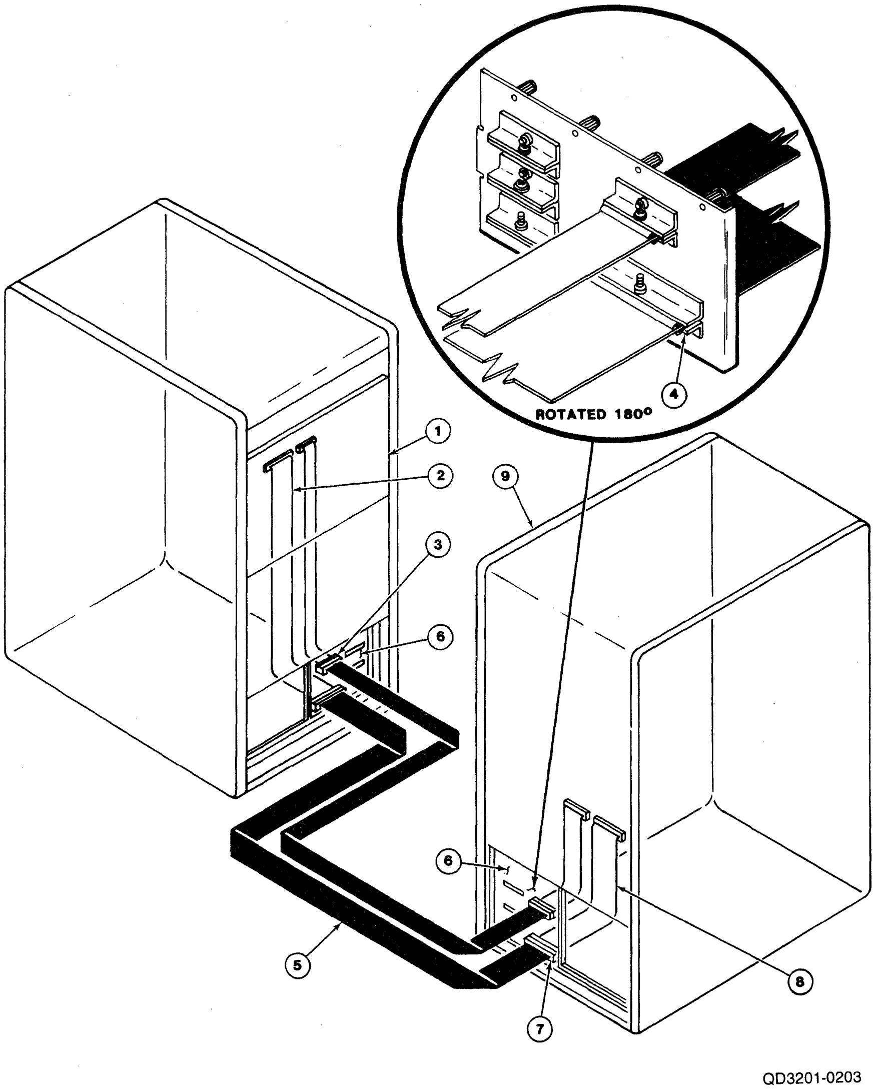
5. Select an extension A cable that is long enough to reach from J1 of the QD32 to the cable I/O adapter panel in the CPU or expansion cabinet.

6. Find the arrow that is molded into the female cable header of the extension cable. Align the cable arrow with the corresponding arrow in connector J1 and press the header into the connector.

7. Select an extension B cable that is long enough to reach from J2 of the QD32 to the cable I/O adapter panel in the CPU or expansion cabinet.

8. Find the arrow that is molded into the female cable header of the extension cable. Align the cable arrow with the corresponding arrow in connector J2 and press the header into the connector. Repeat this step for J3 if that port is to have a drive attached to it.

9. Select a shielded A cable that is long enough to reach from the CPU or expansion cabinet to the RETMA rack or the first freestanding drive.



1. Disk Controller PCBA Enclosure
2. Nonshielded Extension Cable
3. Cable Connectors, Extension Cable to Shielded Cable
4. Clamp - Shield of Shielded Cable Clamped Within
5. Shielded/Jacketed Cable, External to Equipment Cabinets
6. Cable I/O Adapter Panel
7. Cable Connectors, Shielded Cable to Extension Cable
8. Nonshielded Extension Cable
9. Peripheral Device

Figure 4-7. Rack-Mount Drive Cabling Configuration

10. Remove (or loosen) the clamping bars on the widest slot of the cable I/O adapter panel in the CPU or expansion cabinet.

11. Route one end of the prepared cable through the widest slot in the cable I/O adapter panel (see detail in Figure 4-5). Clamp the bare shielding against the cable I/O adapter panel with the clamping bar.

12. Find the arrow that is molded into the male cable header of the extension cable. Align this arrow with the corresponding arrow in the female header of the shielded cable and press the headers together.

13. Repeat steps 10 through 13 for each B cable, substituting B cable for A cable as appropriate and using the narrow slots in the cable I/O adapter panels.

Use the Following Steps for Rack Mount Configurations

14. Repeat steps 10 and 11 at the disk end of the cable.

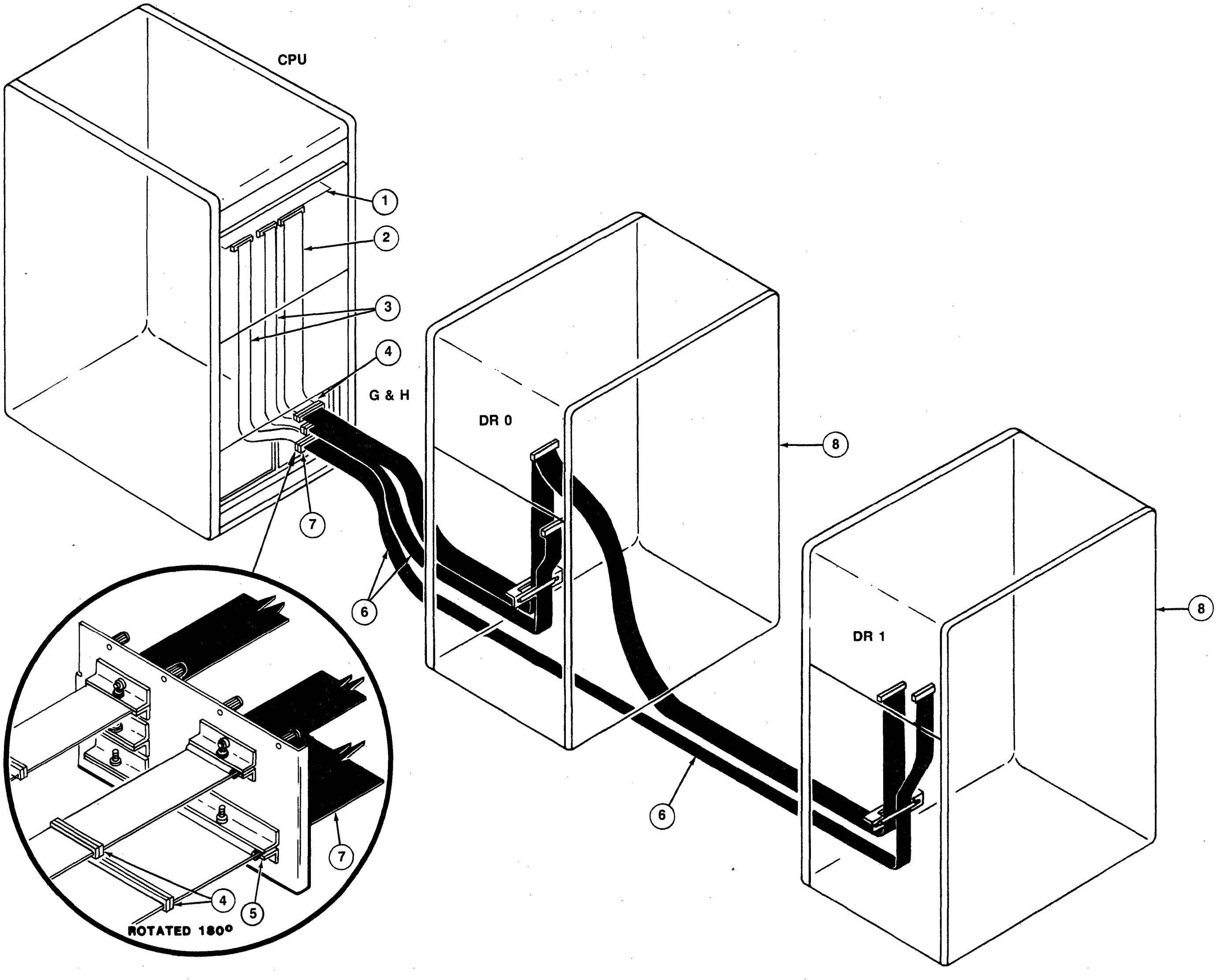
15. Select an extension A cable that will reach from the cable I/O adapter panel in the RETMA rack to the first (lowest) disk drive (the position in the A cable daisy chain has no effect on unit number).

16. Find the arrow that is molded into the female cable header of the extension cable. Align the cable arrow with the corresponding arrow of one of the disk drive's A cable connectors and press the header into the connector. Make sure that the latches on the connector fully engage the cable header.

17. Select an extension B cable that will reach from the cable I/O adapter panel in the RETMA rack to the first (lowest) disk drive.

18. Find the arrow that is molded into the female cable header of the extension cable. Align the cable arrow with the corresponding arrow on the disk drive's B cable connector and press the header into the connector. Make sure that the latches on the connector fully engage the cable header.

19. Daisy-chain the A cable from the first disk drive to the remaining drives as described in step 5 of subsection 4.6.1.1. (You can use the unshielded cables listed Table 4-6 to interconnect the drives.)



1. Disk Controller PCBA
2. Nonshielded Extension A Cable
3. Nonshielded Extension B Cable
4. Cable Connectors, Extension Cables to Shielded Cables
5. Clamp - Shield of Shielded Cable Clamped Within
6. Shielded/Jacketed Cable, External to Equipment Cabinets
7. Cable I/O Adapter Panel
8. Peripheral Device

Figure 4-8. Freestanding Drive Cabling Configuration

20. Repeat step 17 of this procedure for the other disk drive in the rack.

21. Terminate the A cable at the last drive in the daisy chain by installing a terminator (included with the drive) on the drive's unused A cable connector.

22. Connect a ground cable from the logic ground of the CPU to the logic ground of the disk drives.

Use the Following Steps for Freestanding Drive Configurations

23. Select two extension A cables long enough to reach from the drive's A cable connector to the peripheral cable adapter. (Alternatively, you can strip enough insulation off the shielded A cable to allow the shield to be clamped at the adapter and the cable head to connect at the drive's A cable connectors. If you use this approach, select a shielded A cable that is long enough to reach to the next drive in the daisy chain at this point, and prepare it by stripping the shield insulation at both ends.)

24. Connect both A cables to the drive's A cable connectors by matching pin 1 (molded-in arrow on cable header) and pressing the header onto the connector.

25. Select a shielded A cable that is long enough to reach the next drive in the daisy-chain.

26. Clamp both shielded A cables in the wide section of the peripheral cable adapter.

27. Connect the extension cables to the shielded cables as described in step 12.

28. Select an extension B cable that will reach from the peripheral cable adapter rack to the drive's B cable connector.

29. Clamp the shielded B cable from the controller in the peripheral adapter. Connect the extension cable and the shielded cabled as described in step 17.

30. Repeat steps 23 through 29 for the other drive in the subsystem.

31. Terminate the A cable at the last drive in the daisy chain by installing a terminator (included with the drive) on the drive's unused A cable connector.

End of Procedure

4.7 INTEGRATION AND OPERATION

Before you can use the QD32, you must load the NOVRAM with the configuration parameters of the drives that are controlled by the QD32. This can be done in two ways: by using a console emulator or by using a software program. Subsection 6.3.1.1 describes the console emulator technique. The Emulex PDP/LSI MSCP Formatter Program (QXMX8B) and the MicroVAX MSCP Disk Formatter Program (FVD32M) are supplied on several Emulex diagnostic program distributions as listed and described in Appendix C.

NOTE

> Drive configuration parameters must be loaded into the NOVRAM before the controller can be used. The NOVRAM is not loaded at the factory, and its contents are indeterminate as delivered by the factory.

4.7.1 DRIVE FORMATTING

Before data can be stored on the QD32's disk drives, the drives must be formatted. The same program that Emulex provides to load the QD32's drive configuration NOVRAM can also format the disk drives, verify the disk media, and reassign the blocks that it finds to be bad. The QD32 implements a format option that allows it to format its disk drives without help from system software; however, the QD32 does not verify the disk media or reassign bad blocks. See subsection 6.3.1.2. We strongly recommend using the Emulex software programs to format and verify the disk media. See subsection 1.4 for ordering information.

4.7.2 TESTING

Successfully loading the NOVRAM and formatting the disk drives gives good indication that the QD32 and its disk drives are in good operating condition. Figure 5-1 illustrates the loading and formatting procedure in flow chart format, and the figure gives fault isolation information that can be used with the procedure. The Emulex PDP/LSI MSCP Formatter Program (QXMX8B) program follows a similar path, and the fault isolation information can be used with the program as well.

4.7.3 OPERATION

There are no operational instructions. The QD32 is ready for MSCP initialization as soon as it is powered up.

4.7.3.1 Indicators

There are three light emitting diodes (LED) on the QD32 PWB. These LEDs are used for both diagnostics and for normal operations.

If switch SW2-1 is OFF, the QD32 executes a self-test at the following times:

- on power-up
- after a reset condition
- after a bus initialization
- after a write operation to the Initialization and Polling (IP) register (base address)

The self-test routine consists of two test sequences: preliminary and self-test. The preliminary test sequence exercises the 8031 microprocessor chip and the Disk Formatter chip. When the QD32 successfully completes the preliminary test, LED3 illuminates indicating that the QD32 is waiting for the MSCP initialization sequence.

During the MSCP initialization sequence, initiated by host software control, the QD32 executes a second self-test that exercises the buffer controller chip, the Host Adapter Controller (HAC) chip and its associated circuitry, the on-board RAM, and the control memory PROM. If the QD32 passes this sequence of its self-test successfully, all the LED indicators on the edge of the QD32 are OFF.

If a fatal error is detected either during self-test or while the system is running, all three of the edge-mounted LED indicators are ON (illuminated). If the QD32 fails to pass its power-up self-tests, you can select a special diagnostic mode (switch SW2-1 ON) which causes the LED indicators to display an error code. See Self-Test Error Reporting, in Section 5, TROUBLESHOOTING.

During normal operation, LED1 and LED2 flicker occasionally. These LEDs are used to indicate LSI-11 bus activity and SMD disk drive activity respectively.

BLANK

5.1 OVERVIEW

This section describes the several diagnostic features with which the QD32 Disk Controller is equipped, and outlines fault isolation procedures that use these diagnostic features.

5.2 SERVICE

Your Emulex QD32 Disk Controller was designed to give years of trouble-free service, and it was thoroughly tested before leaving the factory.

Should one of the fault isolation procedures indicate that the QD32 is not working properly, the product must be returned to the factory or to one of Emulex's authorized repair centers for service. Emulex products are not designed to be repaired in the field.

Before returning the product to Emulex, whether the product is under warranty or not, you must contact the factory or the factory's representative for instructions and a Return Materials Authorization (RMA) number.

Do not return a component to EMULEX without authorization. A component returned for service without an authorization will be returned to the owner at the owner's expense.

In the continental United States, Alaska, and Hawaii contact:

Emulex Technical Support
3545 Harbor Boulevard
Costa Mesa, CA 92626
(714)662-5600 TWX 910-595-2521

Outside the United States, contact the distributor from whom the subsystem was initially purchased.

To help you efficiently, Emulex or its representative requires certain information about the product and the environment in which it is installed. During installation, a record of the switch setting should have been made on the Configuration Reference Sheet. This sheet is contained in the Installation Section, Figure 4-1.

After you have contacted Emulex and received an RMA, package the component (preferably using the original packing material) and send the component **postage paid** to the address given you by the Emulex representative. The sender must also insure the package.

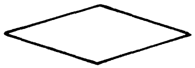
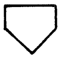
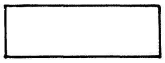
5.3 FAULT ISOLATION PROCEDURE

This fault isolation procedure is provided in flowchart format. The procedure is based on the self-diagnostics incorporated into the QD32. The procedure is designed to be used if the product's self-diagnostic fails or if many errors are flagged by the subsystem during normal operation. If neither of these events happens, it is not necessary to follow these procedures.

The Fault Isolation Chart is shown in Figure 5-1. The chart symbols are defined in Table 5-1.

If the fault isolation procedure indicates that a component needs to be returned to Emulex, see subsection 5.2 for instructions on how to do so.

Table 5-1. Flow Chart Symbol Definitions

Symbol	Description
	Start point, ending point.
	Decision, go ahead according with YES or NO.
	Connector, go to same-numbered symbol on another sheet.
	Process.

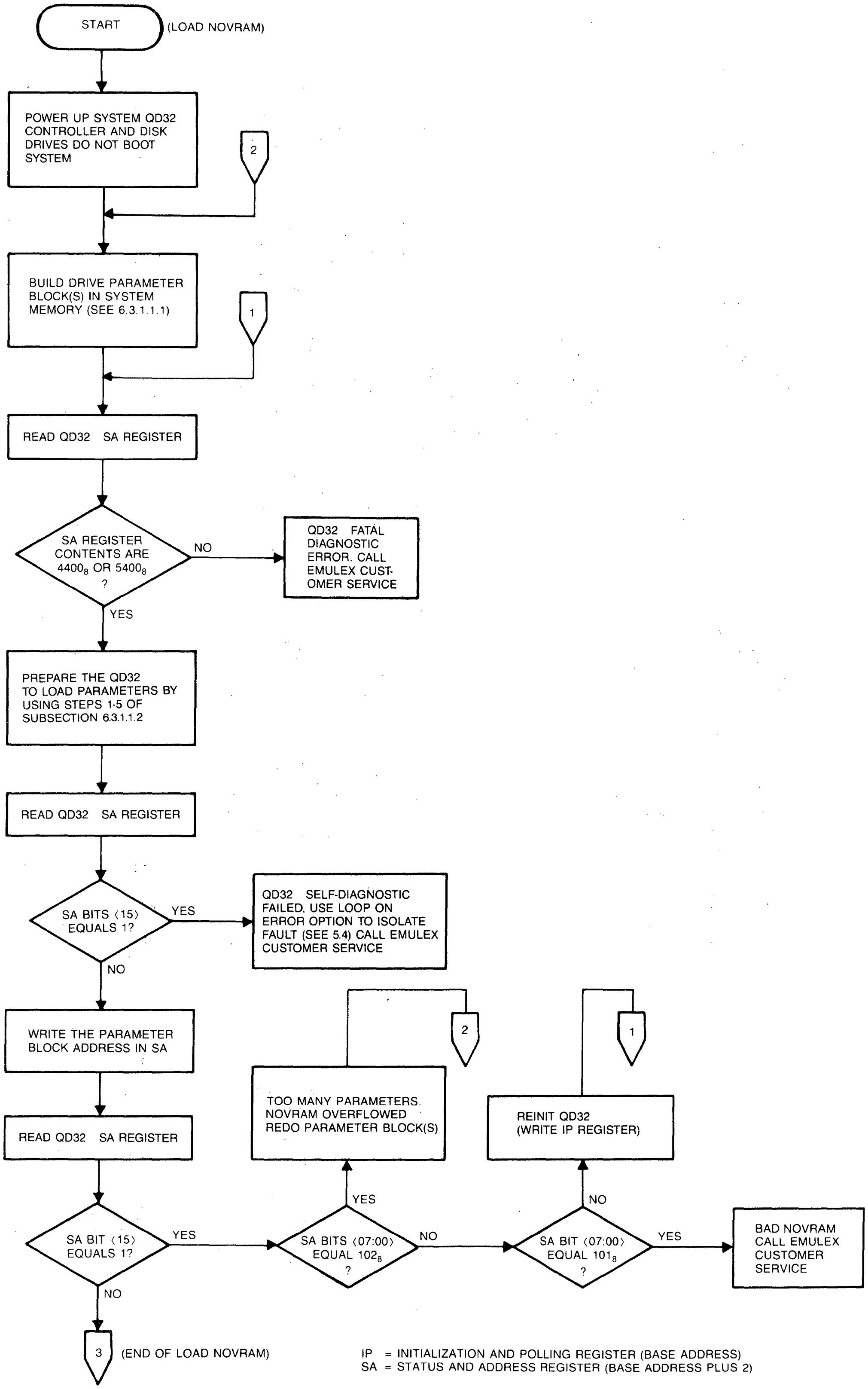
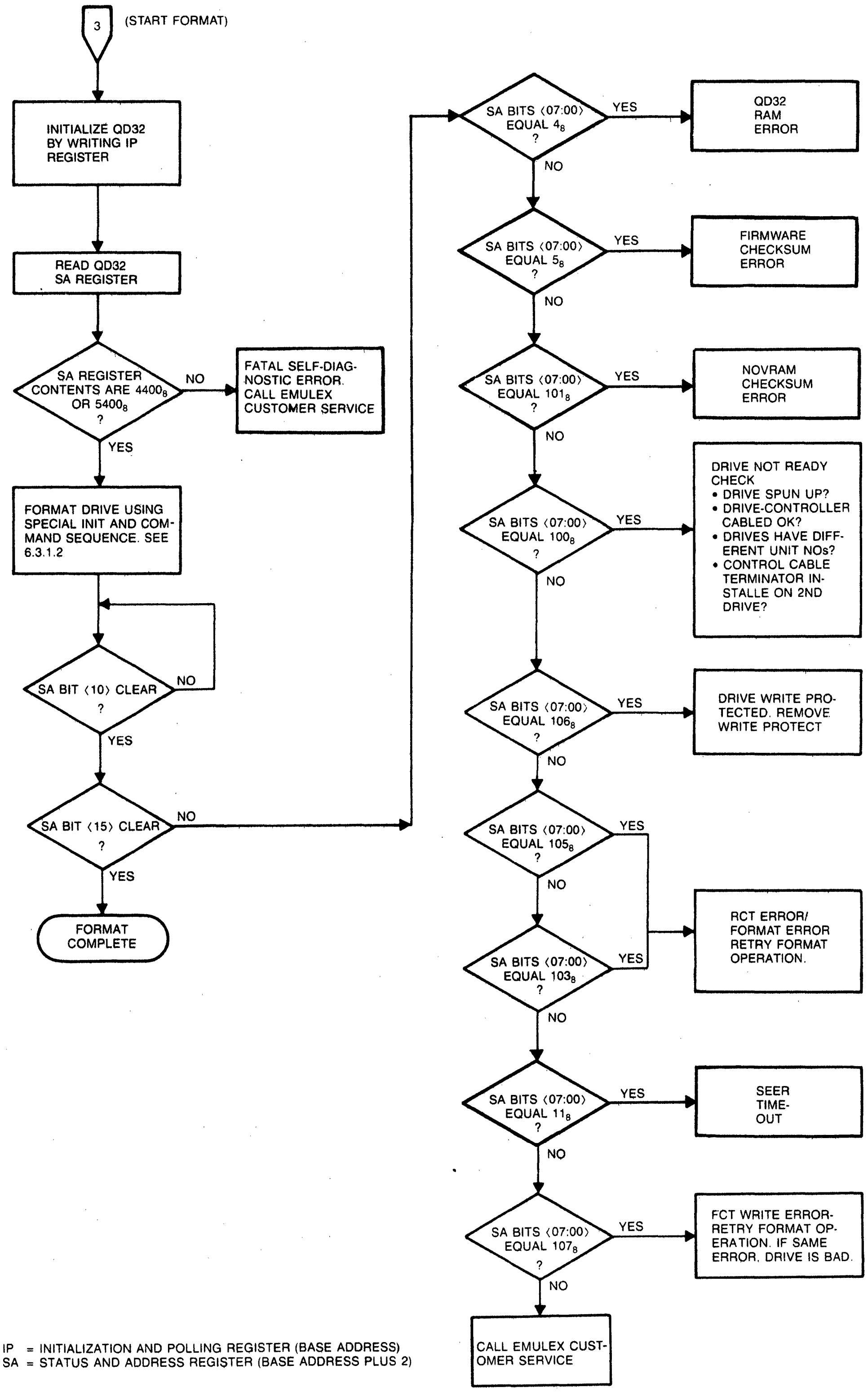

START
(LOAD NOVRAM)
POWER UP SYSTEM QD32 CONTROLLER AND DISK DRIVES DO NOT BOOT SYSTEM
2
BUILD DRIVE PARAMETER BLOCK(S) IN SYSTEM MEMORY (SEE 6.3.1.1.1)
1
READ QD32 SA REGISTER
SA REGISTER CONTENTS ARE 4400_8 OR 5400_8 ?
NO
QD32 FATAL DIAGNOSTIC ERROR. CALL EMULEX CUSTOMER SERVICE
YES
PREPARE THE QD32 TO LOAD PARAMETERS BY USING STEPS 1-5 OF SUBSECTION 6.3.1.1.2
READ QD32 SA REGISTER
SA BITS ‹15› EQUALS 1?
YES
QD32 SELF-DIAGNOSTIC FAILED, USE LOOP ON ERROR OPTION TO ISOLATE FAULT (SEE 5.4) CALL EMULEX CUSTOMER SERVICE
NO
WRITE THE PARAMETER BLOCK ADDRESS IN SA
READ QD32 SA REGISTER
SA BIT ‹15› EQUALS 1?
YES
SA BITS ‹07:00› EQUAL 102_8 ?
YES
TOO MANY PARAMETERS. NOVRAM OVERFLOWED REDO PARAMETER BLOCK(S)
2
NO
SA BIT ‹07:00› EQUAL 101_8 ?
NO
REINIT QD32 (WRITE IP REGISTER)
1
YES
BAD NOVRAM CALL EMULEX CUSTOMER SERVICE
NO
3
(END OF LOAD NOVRAM)
IP = INITIALIZATION AND POLLING REGISTER (BASE ADDRESS)
SA = STATUS AND ADDRESS REGISTER (BASE ADDRESS PLUS 2)

Figure 5-1. Fault Isolation Chart (sheet 1 of 2)


3
(START FORMAT)
INITIALIZE QD32 BY WRITING IP REGISTER
READ QD32 SA REGISTER
SA REGISTER CONTENTS ARE 4400_8 OR 5400_8 ?
NO
FATAL SELF-DIAGNOSTIC ERROR. CALL EMULEX CUSTOMER SERVICE
YES
FORMAT DRIVE USING SPECIAL INIT AND COMMAND SEQUENCE. SEE 6.3.1.2
SA BIT ⟨10⟩ CLEAR ?
NO
YES
SA BIT ⟨15⟩ CLEAR ?
NO
YES
FORMAT COMPLETE
SA BITS ⟨07:00⟩ EQUAL 4_8 ?
YES
QD32 RAM ERROR
NO
SA BITS ⟨07:00⟩ EQUAL 5_8 ?
YES
FIRMWARE CHECKSUM ERROR
NO
SA BITS ⟨07:00⟩ EQUAL 101_8 ?
YES
NOVRAM CHECKSUM ERROR
NO
SA BITS ⟨07:00⟩ EQUAL 100_8 ?
YES
DRIVE NOT READY CHECK
• DRIVE SPUN UP?
• DRIVE-CONTROLLER CABLED OK?
• DRIVES HAVE DIFFERENT UNIT NOs?
• CONTROL CABLE TERMINATOR INSTALLE ON 2ND DRIVE?
NO
SA BITS ⟨07:00⟩ EQUAL 106_8 ?
YES
DRIVE WRITE PROTECTED. REMOVE WRITE PROTECT
NO
SA BITS ⟨07:00⟩ EQUAL 105_8 ?
YES
NO
SA BITS ⟨07:00⟩ EQUAL 103_8 ?
YES
RCT ERROR/ FORMAT ERROR RETRY FORMAT OPERATION.
NO
SA BITS ⟨07:00⟩ EQUAL 11_8 ?
YES
SEER TIME-OUT
NO
SA BITS ⟨07:00⟩ EQUAL 107_8 ?
YES
FCT WRITE ERROR-RETRY FORMAT OPERATION. IF SAME ERROR, DRIVE IS BAD.
NO
CALL EMULEX CUSTOMER SERVICE

IP = INITIALIZATION AND POLLING REGISTER (BASE ADDRESS)
SA = STATUS AND ADDRESS REGISTER (BASE ADDRESS PLUS 2)

Figure 5-1. Fault Isolation Chart (Sheet 2 of 2)

5.4 POWER-UP SELF-DIAGNOSTIC

The QD32 executes an extensive self-diagnostic to ensure that the disk controller is in good working order. The self-diagnostic is divided into several parts. Table 5-2 lists the tests in the order in which they are performed.

The first two tests are executed immediately after power-up, a reset, a bus INIT, or a write to the IP register (base address). The other tests are executed as the controller interacts with the MSCP initialization routine. If the QD32 fails any of the tests, it posts an MSCP fatal error code in the low-byte of the SA register (base address plus 2) and turns on three LEDs which are located on the outside edge of the PWB. The MSCP fatal error codes used by the QD32 are listed in Table 5-3.

To help determine the location of the problem, the operator can select a special diagnostic mode that causes the LEDs to display an error code. To enable this diagnostic mode, place the CPU halt switch in the ON position and set QD32 switch SW2-1 ON (1). After setting SW2-1 ON, the host computer must be powered down or QD32 switch SW1-1 must be toggled (turned ON and then OFF) to cause the QD32 to again perform its self-test.

Upon encountering an error, the host microprocessor halts and the LEDs display an error code. The error codes are listed and described in Table 5-2.

If the QD32 completes the diagnostic mode without errors, all three LEDs are OFF. Set switch SW2-1 in the OFF position and reset the QD32 controller before using.

Table 5-2. LED Error Codes

LED 3	 2	 1	Error Description
0	0	1	CPU Chip Test failed
0	1	0	Formatter Chip Test failed
1	0	0	Controller idle, waiting for initialization
0	1	1	Buffer Controller or External Memory Test failed
1	0	1	HAC Test failed
1	1	0	Emulation PROM Checksum Test failed
0	0	0	Self-Diagnostic complete without errors

5.5 FATAL ERROR CODES

If the QD32 encounters a fatal error anytime during operation, all three LEDs are illuminated and an error code is posted in the low-byte of the SA register (base address plus 2). Table 5-3 lists the MSCP fatal error codes used by the QD32.

Table 5-3. MSCP Fatal Error Codes used by the QD32

Octal Code	Hex Code	Description
0	0	No information in message packet.
1	1	Possible parity or timeout error when the QD32 attempted to read data from a message packet.
2	2	Possible parity or timeout error when the QD32 attempted to write data to a message packet.
4	4	QD32 diagnostic self-test indicated a controller RAM error.
5	5	QD32 diagnostic self-test indicated a firmware checksum error.
6	6	Possible parity or timeout error when the QD32 attempted to read an envelope address from a command ring.
7	7	Possible parity or timeout error when the QD32 attempted to write an envelope address to a command ring.
11	9	Host did not communicate with QD32 within the time frame established while bringing the controller online. (If this code appears during firmware formatting, it indicates a seek timeout. See Table 5-4.)
12	A	Operating system sent more commands to the QD32 than the controller can accept.

Table 5-3. MSCP Fatal Error Codes used by the QD32 (continued)

Octal Code	Hex Code	Description
13	B	Controller unable to perform DMA transfer operation correctly.
14	C	QD32 diagnostic self-test indicated controller fatal error.
16	E	The MSCP connection identifier is invalid.
23	13	An error occurred during the MSCP initialization sequence.

In addition, fatal error messages may appear during the firmware formatting procedure (subsection 6.7). These error codes are listed in Table 5-4.

Table 5-4. Error Codes Unique to Firmware Formatting

Octal Code	Hex Code	Description
004	04	RAM error
005	05	Firmware checksum error
011	09	Seek timeout
100	40	Drive not ready
101	41	NOVRAM checksum error
103	43	RCT Write error
105	45	Format error
106	46	Drive write protected
107	47	FCT Write error

BLANK

6.1 OVERVIEW

This section contains an overview of the QD32 device registers that are accessible to the LSI-11 bus and that are used to monitor and control the QD32 Disk Controller. The registers are functionally compatible with DEC implementations of MSCP controllers.

The following table outlines the contents of this section.

Subsection	Title
6.2	Overview of MSCP Subsystem
6.3	Programming
6.4	Registers
6.5	NOVRAM Commands
6.6	Bootstrap Command
6.7	Firmware Formatting Command

6.2 OVERVIEW OF MSCP SUBSYSTEM

Mass Storage Control Protocol (MSCP) is the protocol used by a family of mass storage controllers and devices designed and built by Digital Equipment Corporation. MSCP allows a host system to be connected to subsystems with a variety of capacities and geometries. This flexibility is possible because MSCP defines data locations in terms of sequential, logical blocks, not in terms of a physical description of the data's location (i.e., cylinder, track, and sector). This scheme gives the MSCP subsystem the responsibility for converting MSCP logical block numbers into physical addresses that the peripheral device can understand.

This technique has several implications. First, the MSCP subsystem must have detailed knowledge of the peripheral's capacity, geometry, and status. Second, the ability to make the translation between logical and physical addresses implies considerable intelligence on the part of the subsystem. Finally, the host is relieved of responsibility for error detection and correction because its knowledge of the media is insufficient to allow error control to be done efficiently.

There are several advantages to this type of architecture. First, it provides the host with an "error free" media. Second, it provides for exceptional operating system software portability because, with the exception of capacity, the characteristics of all MSCP subsystems are the same from the operating system's point of view.

In terms of implementation, this protocol requires a high degree of intelligence on the part of the subsystem. Essentially, this intelligence is a process that runs on a microprocessor and is referred to as an MSCP controller. The MSCP controller has all of the responsibilities outlined above.

The host computer runs corresponding software processes which take calls from the operating system, convert them into MSCP commands, and cause the resulting command to be transferred to the MSCP controller.

In summary, an MSCP subsystem is characterized by an intelligent controller that provides the host with the view of a perfect media. It is further characterized by host independence from a specific bus, controller, or device type.

For more information about MSCP subsystems, see subsections 3.2, 3.3, and 3.4.

6.3 PROGRAMMING

A complete description of MSCP commands and the corresponding status responses which the QD32 Disk Controller posts is beyond the scope of this manual.

6.3.1 UNSUPPORTED COMMANDS

No currently available MSCP Controller supports the entire range of MSCP commands. The following subsections describe the extent of MSCP command support by the QD32.

6.3.1.1 Minimal Disk Subset

The QD32 Disk Controller supports the entire minimal disk subset of MSCP commands.

6.3.1.2 Diagnostic and Utility Protocol (DUP)

The QD32 Disk Controller does not support any of the DUP commands or maintenance read/write commands. Therefore, the QD32 is not compatible with DEC diagnostics that use the MSCP DUP commands.

6.4 REGISTERS

During normal operation, the QD32 Disk Controller is controlled and monitored using the command and status packets that are exchanged by the Class Driver (host) and the MSCP Controller. The QD32 has two 16-bit registers in the LSI-11 Bus I/O page that are used primarily to initialize the subsystem. During normal operation, the registers are used only to initiate polling or to reset the subsystem. These registers are always read as words. The register pair begins on a longword boundary. Table 6-1 lists the octal and hexadecimal values for the Initialization and Polling (IP) register (base address) and the Status and Address (SA) register (base address plus 2) supported by the QD32.

The IP register (base address) has two functions as detailed below:

- When written with any value, it causes a hard initialization of the MSCP Controller.

- When read while the port is operating, it causes the controller to initiate polling.

The SA register (base address plus 2) has four functions as listed below:

- When read by the host during initialization, it communicates data and error information relating to the initialization process.

- When written by the host during initialization, it communicates certain host-specific parameters to the port.

- When read by the host during normal operation, it communicates status information including port and controller-detected fatal errors.

- When zeroed by the host during either initialization or normal operation, it signals the port that the host has successfully completed a bus adapter purge in response to a port-initiated purge request.

Registers

The detailed operation of these registers is discussed in the Storage System UNIBUS Port Description (AA-L621A-TK) available from DEC as referenced in subsection 6.3. Note that only word transfers to and from IP and SA are permissible; the behavior of byte transfers is undefined.

Table 6-1. QD32 IP and SA Registers

Register	Octal	Hexadecimal
IP	772150	20001468
SA	772152	2000146A
IP	772154	2000146C
SA	772156	2000146E
IP	760334	200000DC
SA	760336	200000DE
IP	760340	200000E0
SA	760342	200000E2
IP	760344	200000E4
SA	760346	200000E6
IP	760350	200000E8
SA	760352	200000EA
IP	760354	200000EC
SA	760356	200000EE
IP	760360	200000F0
SA	760362	200000F2

6.5 NOVRAM COMMANDS

The QD32 allows the user to specify media geometry for the SMD disk drives that it supports. The geometry data is stored in a Nonvolatile Random Access Memory (NOVRAM) on the QD32. When used with the QD32, the NOVRAM can store configurations for two physical (four logical) SMD disk drives.

The NOVRAM is programmed while the QD32 is installed by loading the drive geometry parameters into the host computer's memory and by executing a command sequence that causes the QD32 to read those parameters. Both operations can be performed manually by using the CPU console (or a console emulator such as ODT), or automatically by using a special utility. Emulex provides utilities for VAX-11, and PDP-11 and LSI-11 processors: the MicroVAX MSCP Disk Formatter Program (FVD32M) and the Emulex LSI/PDP MSCP Formatter Program (QXMX8B). See subsection 1.4.1, Subsystem Options, for ordering information.

The following subsections describe the geometry parameters that are required, their format, and the command sequence that causes the parameters to be stored in the NOVRAM.

6.5.1 DRIVE GEOMETRY PARAMETERS

The QD32 NOVRAM defines a maximum of two physical drive configurations and uses a separate parameter information block to define each drive configuration type. Each block contains the 16 words that define the drive parameters. Figure 6-1 shows the format of the parameters and lists them. Emulex recommends that you document the parameters you select for your drive types using the Drive Configuration Parameter Block Worksheet in Figure 6-1.

The blocks are placed anywhere in the first 64K of memory and must be contiguous. The first word of the parameter block can be stored at any even address. The addresses of subsequent words and parameter blocks are contiguous and numerically higher. Note that for MicroVAX II systems, the first word must be located at 0 (see subsection 6.5.2, Loading the NOVRAM).

The parameter block at the lowest numerical address defines the first drive connected to the QD32. If a second drive is configured exactly as the first drive, that first parameter block can define both drives. If a second drive is configured differently, a second parameter block must be defined. If you are using the console emulator, be sure that the memory location following the last word in the last parameter block contains 0.

Parameter	Drive Type 1	Drive Type 2
Word 1 Number of Units Of this Type		
Word 2 Type Code		
Word 3 Starting Head Offset		
Word 4 Number of Sectors Per Track		
Word 5 Number of Heads		
Word 6 Number of Cylinders		
Word 7 Number of Spare Sectors per Track		
Word 8 Number of Alternate Cylinders		
Word 9 Configuration Bits		
Word 10 Split Code		
Word 11 Removable Media Flag		
Word 12 Gap 0 Parameter		
Word 13 Gap 1 Parameter		
Word 14 Gap 2 Parameter		
Word 15 Cylinder Offset		
Word 16 Spiral Offset		

Figure 6-1. Drive Configuration Parameter Block Worksheet

Number of Units of this Type (1)

This word specifies the number of physical disk drives that this parameter block defines. This number cannot be larger than 2. If this word is 0, then words 2 through 16 are ignored.

Type Code (2)

This word indicates the type of disk drive. The only valid value is 1.

Starting Head Offset (3)

This word specifies the physical drive head that is to be used as the first head of the second logical drive. This field has meaning only if a Split Code 2 or 3 (word 10) is specified. The valid range is from 0 through 31. If a Split Code 0 or 1 is selected, this word must be 0.

Number of Sectors per Track (4)

This word specifies the number of logical sectors per physical track. Spare sectors are not included in this number (but are specified in word 7). The valid range is from 1 through 255. See Appendix D for recommended parameters for certified drives.

Number of Heads (5)

This word specifies the number of data heads per physical drive. The valid range is from 1 through 63.

Number of Cylinders (6)

This word specifies the number of logical cylinders per physical drive. Spare cylinders are not included in this number (but are specified in word 8). The valid range is from 1 through 4095.

Number of Spare Sectors per Track (7)

This word specifies the number of spare sectors reserved per track. This number plus the number of logical sectors per track (word 4) equals the total number of physical sectors per track. The valid range is 0 or 1. If 0 is specified, no spare sectors are reserved. Emulex recommends a value of 1 for MSCP implementation of the replace block command.

Number of Alternate Cylinders (8)

This word specifies the number of spare cylinders per physical drive. This number plus the number of logical cylinders (word 6) equals the total number of physical cylinders. The valid range is from 0 through 15. At least one cylinder must be specified as an alternate for MSCP implementation of the replace block command. (If spare sectors are specified, the sector replacement algorithm needs one track for working space.)

If Split Code 1 is used, you must specify twice the normal number of alternate cylinders because they are divided evenly between the two logical drives. A minimum of 2 alternate cylinders must be specified if block replacement is to function with a cylinder split.

Configuration Bits (9)

This word defines some additional configuration parameters of the drive. The valid range is from 0 through 7. This word has a 3-bit field that is defined as follows:

Bit 0: This bit specifies whether or not the drive negates the On Cylinder signal during a head select operation. The valid range for this bit is 0 or 1. If this bit is 0, the On Cylinder signal remains on during a head select. If this bit is 1, the On Cylinder signal is negated during a head select.

Bit 1: This bit specifies whether or not the drive can perform early or late data strobe operations. The valid range for this bit is 0 or 1. If this bit is 0, the drive cannot perform early or late data strobe operations. If this bit is 1, the drive is capable of performing early or late data strobe operations.

Bit 2: This bit specifies whether or not the drive is capable of head offset operations. The valid range for this bit is 0 or 1. If this bit is 0, the drive cannot perform head offset operations. If this bit is 1, the drive is capable of performing head offset operations.

Split Code (10)

This word allows the drive(s) defined by this parameter block to be split into two logical disk units (two each, if more than one drive is defined by this block). The split codes are:

Code 0: No split.

Code 1: The cylinders are divided between the two logical drives. A starting cylinder offset value (word 15) specifies the first cylinder of the second logical drive.

Code 2: The drives data heads are divided between the two logical drives. A starting head offset value (word 3) specifies the first head of the second logical drive. If you select a head split code on a drive with both fixed and removable media, the removable media may be configured as logical unit number (LUN) 0 and the fixed media as LUN 1.

Code 3: Identical to Code 2 except the logical assignments for the physical drives are reversed. Reverse head split codes also divide the drive by data heads, but assign the lower numbered heads to drive 1 and the higher numbered heads to drive 0.

By using Codes 2 and 3, you can partition drives that use both fixed and removable media so that either the fixed media or removable media is the lower logical unit number.

The QD32's seek-ordering and overlapped seek processing capabilities are not effective when both logicals of a split physical drive are active. Therefore, Emulex does not recommend using this option unless necessary to allow for convenient file backup and restoration.

For more information on split codes, see subsection 3.4.2.1.

Removable Media (11)

This word indicates whether the disk media is fixed or removable. If you are defining 1 physical/logical drive, this word uses a 1-bit field and valid values are 0 and 1 where 0 indicates fixed media and 1 indicates removable media.

If you are defining a drive with a logical split, this word uses a 2-bit field and the valid range is 0 though 3:

Parameter Word Value	Bit 1 0	Definition
0	0 0	LUN 0 and LUN 1 are both fixed.
1	0 1	LUN 0 is removable, LUN 1 is fixed.
2	1 0	LUN 0 is fixed, LUN 1 is removable.
3	1 1	LUN 0 and LUN 1 are both removable.

Gap 0 Parameter (12)

This word, in conjunction with words 13 and 14, specifies the recording format for each sector on the drive. The only valid values are contained in Appendix D, Disk Drive Configuration Parameters. These values are factory parameters and are the only values to be used with Emulex certified drives. If any of these factory parameters are altered, the QD32 may not support the disk drive.

Gap 1 Parameter (13)

This word, in conjunction with words 12 and 14, specifies the recording format for each sector on the drive. The only valid values are contained in Appendix D, Disk Drive Configuration Parameters. These values are factory parameters and are the only values to be used with Emulex certified drives. If any of these factory parameters are altered, the QD32 may not support the disk drive.

Gap 2 Parameter (14)

This word, in conjunction with words 12 and 13, specifies the recording format for each sector on the drive. The only valid values are contained in Appendix D, Disk Drive Configuration Parameters. These values are factory parameters and are the only values to be used with Emulex certified drives. If any of these factory parameters are altered, the QD32 may not support the disk drive.

Cylinder Offset (15)

This word specifies the physical cylinder that is to be used as the first cylinder of the second logical drive. This field has meaning only if a Split Code 1 (word 10) is specified. If a Split Code 0, 2, or 3 is selected, this word must be 0.

Spiral Offset (16)

This word specifies the number of sectors by which sector 0 is offset from sector 0 of the previous track. Offsetting sector 0 from one track to the next is a technique that is used to reduce latency when performing write or read operations that cross a track boundary. When the drive is formatted, sector 0 of a track is offset a certain number of sectors from the position of sector 0 on the previous track. When this is done, spiral write and read operations are more efficient because the drive has time to seek from track to track before encountering sector 0.

The valid range is from 0 through 15. See Appendix D for recommended parameters for certified drives.

6.5.2 LOADING THE NOVRAM

A special sequence of commands causes the QD32 to load the parameter blocks from memory into the NOVRAM. The process uses the Initialization and Polling (IP) register (QD32 base address) and the Status and Address (SA) register (base address plus 2). See section 6.4 for register octal and hexadecimal notation.

To load the NOVRAM on a MicroVAX II, you must first prepare the system for DMA transfer operations:

1. Apply power to the system and initialize the LSI-11 bus by depositing 0 in internal memory register 37. If you have a running system, you must bring it down before executing this step.

   ```
   >>>D/I 37 0<return>
   ```

2. Enable external access to local memory by loading 20_{16} into the Interprocessor Communications Register (ICR) at address $20001F40_{16}$:

   ```
   >>>D/P/W 20001F40 20<return>
   ```

3. Set the map register for the first page in memory to 0 by depositing 80000000_{16} in map register 20088000_{16}:

   ```
   >>>D/P/L 20088000 80000000<return>
   ```

4. Proceed to step 1 of the NOVRAM command sequence. Be sure to store the NOVRAM parameters at memory location 0.

To load the NOVRAM:

1. Initialize the QD32 by writing any value into the IP register (base address). The QD32 performs its self-test and begins the initialization dialog.

Register	Octal	Hexadecimal
IP: Write	000001	0001

2. The QD32 indicates that initialization step 1 has begun by setting bit 11 in the SA register (base address plus 2). The host must poll the register for this value (no interrupt is generated). Bit 8 should also be set. If 22-bit addressing is enabled, bit 9 will be set.

Register	Octal	Hexadecimal	Addressing
SA: Read	004400	0900	18-Bit
	005400	0B00	22-Bit

3. When the controller indicates that step 1 of the initialization dialog is begun, load the SA register (base address plus 2) with the "special initialization code:"

Register	Octal	Hexadecimal
SA: Write	030003	3003

4. The controller acknowledges the initialization code with: 00400.

Register	Octal	Hexadecimal
SA: Read	000400	0100

5. Write the Define Unit Geometry command into the SA register (base address plus 2):

Register	Octal	Hexadecimal
SA: Write	041000	4200

6. The QD32 finishes its self-test (about two seconds) before acknowledging with:

Register	Octal	Hexadecimal
SA: Read	001001	0201

You must wait until the QD32 acknowledges before proceeding.

7. Write the 16-bit memory address of the first word of the first parameter block into the SA register (base address plus 2).

   For example, if you loaded the first word of the first parameter block at memory address 001000:

Register	Octal	Hexadecimal
SA: Write	001000	0200

8. The QD32 begins loading the parameter blocks. Word 17 of the last parameter block must be 0 to indicate that there is no more data (Word 17 would be the first word of the next parameter block, if there was another).

   After the QD32 has stored the parameter data in the NOVRAM, it reads the data from the NOVRAM and computes a one-byte checksum. It places the checksum in the SA register (base address plus 2). The host knows that the checksum is available when bit 09 of SA register (base address plus 2) is clear (0).

   If the QD32 sets SA register (base address plus 2) bit 15 after it clears bit 09, an error has occurred. The low byte contains the error code.

Error Code		Description
101_8	41_{16}	Checksum error (NOVRAM may be bad)
102_8	42_{16}	NOVRAM capacity exceeded (too many parameters)

6.6 BOOTSTRAP COMMAND

To allow the system to be easily bootstrapped from peripherals attached to the QD32 Disk Controller, Emulex has incorporated a Bootstrap Command into the controller. This feature is not part of the standard MSCP command set nor is it supported on the MicroVAX or on systems using an 11/73B CPU module.

The Bootstrap Command can be issued from the console after the system is powered up, or it may be incorporated into a firmware routine that is located in a Bootstrap ROM. (The ROM would not be located on the QD32 PWB, but on some other module in the system.) The Bootstrap Command causes the QD32 to load the first logical block from the selected peripheral into host memory starting at location 00000.

Registers

To issue the Bootstrap Command to the QD32:

1. Initialize the QD32 by writing any value into the IP register (base address). The QD32 performs self-test and begins the initialization dialog.

Register	Octal
IP: Write	000001

2. The QD32 indicates that initialization step 1 has begun by setting bit 11 in the SA register (base address plus 2). The host must poll the register for this value (no interrupt is generated). Bit 8 should also be set. If 22-bit addressing is enabled, bit 9 will be set.

Register	Octal	Addressing
SA: Read	0044000	18-Bit
	005400	22-Bit

3. When the controller indicates that step 1 of the initialization dialog is begun, load the SA register (base address plus 2) with the "special initialization code:"

Register	Octal
SA: Write	030003

4. The controller acknowledges the initialization code with 00400.

Register	Octal
SA: Read	000400

5. Load the SA register (base address plus 2) with $04000n_8$ or $400n_{16}$, where n is the MSCP logical unit number of the unit to bootstrap from. In this example, the unit is 0.

Register	Octal
SA: Write	040000

6. Load CPU register R0 with the MSCP unit number of the unit to bootstrap from. In this example, the unit is 0.

Register	Octal
R0: Write	00000

7. Load CPU register Rl with the base address of the QD32. In this example, the QD32 is at the standard base address.

Register	Octal
Rl: Write	772150

8. Load the Processor Status Word (PSW) register with 340. The PSW register is 777776 for 18-bit systems and 1777776 for 22-bit systems.

Register	Octal
PSW Write	304

9. Load CPU register R7 with 0.

Register	Octal
R7: Write	0

10. At the console emulator prompt, enter P to begin:

```
@P
```

6.7 FORMAT DRIVE COMMAND

The QD32 also has the ability to format the disk drives attached to it. This format operation is performed autonomously by the QD32 in response to a special initialization command. The process uses the IP register (base address) and SA register (base address plus 2). Refer to subsection 6.4 for octal and hexadecimal notation. To initiate the format operation, use the following procedure.

1. Initialize the QD32 by writing any value into the IP register (base address). The QD32 performs self-test and begins the initialization dialog.

Register	Octal	Hexadecimal
IP: Write	000001	0001

2. The QD32 indicates that initialization step 1 has begun by setting bit 11 in the SA register (base address plus 2). The host must poll the register for this value (no interrupt is generated). Bit 8 should also be set. If 22-bit addressing is enabled, bit 9 will be set.

Register	Octal	Hexadecimal	Addressing
SA: Read	004400	0900	18-Bit
	005400	0B00	22-Bit

3. When the controller indicates that step 1 of the initialization dialog is begun, load the SA register (base address plus 2) with the "special initialization code:"

Register	Octal	Hexadecimal
SA: Write	030003	3003

4. The controller acknowledges the initialization code with 00400.

Register	Octal	Hexadecimal
SA: Read	000400	0100

5. Write the Format Unit command into the SA register (base address plus 2):

Register	Octal	Hexadecimal
SA: Write	0420nn	440n

where n is the number of the logical unit to be formatted. Valid values for a maximum of 11 logical drives are 0-13_8 and 0-C_{16}. If the logical drive is supported by an alternate QD32 controller, add the unit offset specified by switches SW1-2 through SW1-4.

6. The QD32 acknowledges the command with:

Register	Octal	Hexadecimal
SA: Read	001000	0200

7. Write the 16-bit volume serial number into the SA register (base address plus 2). This number may be any value from 1 to 177777_8 ($FFFF_{16}$).

Register	Octal	Hexadecimal
SA: Write	000001	0001

8. The QD32 acknowledges the serial number with:

Register	Octal	Hexadecimal
SA: Read	002000	0400

9. Write the format parameter word into the SA register (base address plus 2). (The format parameter word is not defined and is reserved for future use. Write all zeros into the register.) The QD32 begins formatting the selected drive.

Register	Octal	Hexadecimal
SA: Write	000000	0000

10. Poll the SA register (base address plus 2) until the QD32 clears SA bit 10 to indicate that the format operation concluded. If the operation was not successful, the QD32 sets bit 15 in the SA register (base address plus 2). The low-byte of the register contains the error code:

Error Octal	Code Hex	Description
004	04	RAM error
005	05	Firmware checksum error
011	09	Seek timeout
100	40	Drive not ready
101	41	NOVRAM checksum error
103	43	RCT Write error
105	45	Format error
106	46	Drive write protected
107	47	FCT Write error

BLANK

7.1 OVERVIEW

This section contains a description of the QD32 Disk Controller's architecture.

7.2 QD32 DISK CONTROLLER ARCHITECTURE

The QD32 is a microprocessor-based emulating disk controller that is contained on a single dual-wide PCBA. The QD32's major functional blocks are shown in Figure 7-1. The disk controller is organized around the eight-bit 8031 microprocessor. The board has an eight-bit internal data bus with 16-bit addressing capability. The Host Adapter Controller, the Formatter Controller, and the Buffer Controller are addressed as memory (memory-mapped I/O).

The 8031's primary task is to decode and implement commands from the host. At command completion, the microprocessor is also responsible for generating status and transmitting it to the host. A large part of the microprocessor's job while performing those duties involves setting up the Host Adapter Controller and the Buffer Controller for the large data transfers that are their specialties.

The QD32 uses a 27128 erasable programmable read-only memory (EPROM), which contains the control program, and 32K bytes of random access memory (RAM), which is used for data buffering and working storage.

The LSI-11 bus interface contains 22 lines. Sixteen of the lines are multiplexed for both address and data; six are used for only address. The Host Adapter Controller is used for programmed I/O, CPU interrupts, and DMA data transfers. The microprocessor responds to all programmed I/O and carries out the I/O functions required for the addressed disk controller register. The Host Adapter Controller has automatic LSI-11 bus address generation capability that, in conjunction with a byte counter, allows the interface to conduct LSI-11 bus DMA transfers without direct microprocessor intervention after the interface is set up for a transfer. This automatic DMA capability is used with the QD32 Buffer Controller to transfer large blocks of data directly between host memory and the QD32's RAM.

The Buffer Controller is implemented on a single chip. This multichannel DMA is responsible for moving large blocks of data between the 32K RAM buffer and the SMD interface, and between the LSI-11 bus interface and the 32K RAM buffer. After being set up for an operation by the microprocessor, either interface requests DMA service from the Buffer Controller by driving an individual request signal active. The transfer then proceeds without direct intervention by the microprocessor. This allows high-speed data transfers to occur while the microprocessor is focused on other processes.

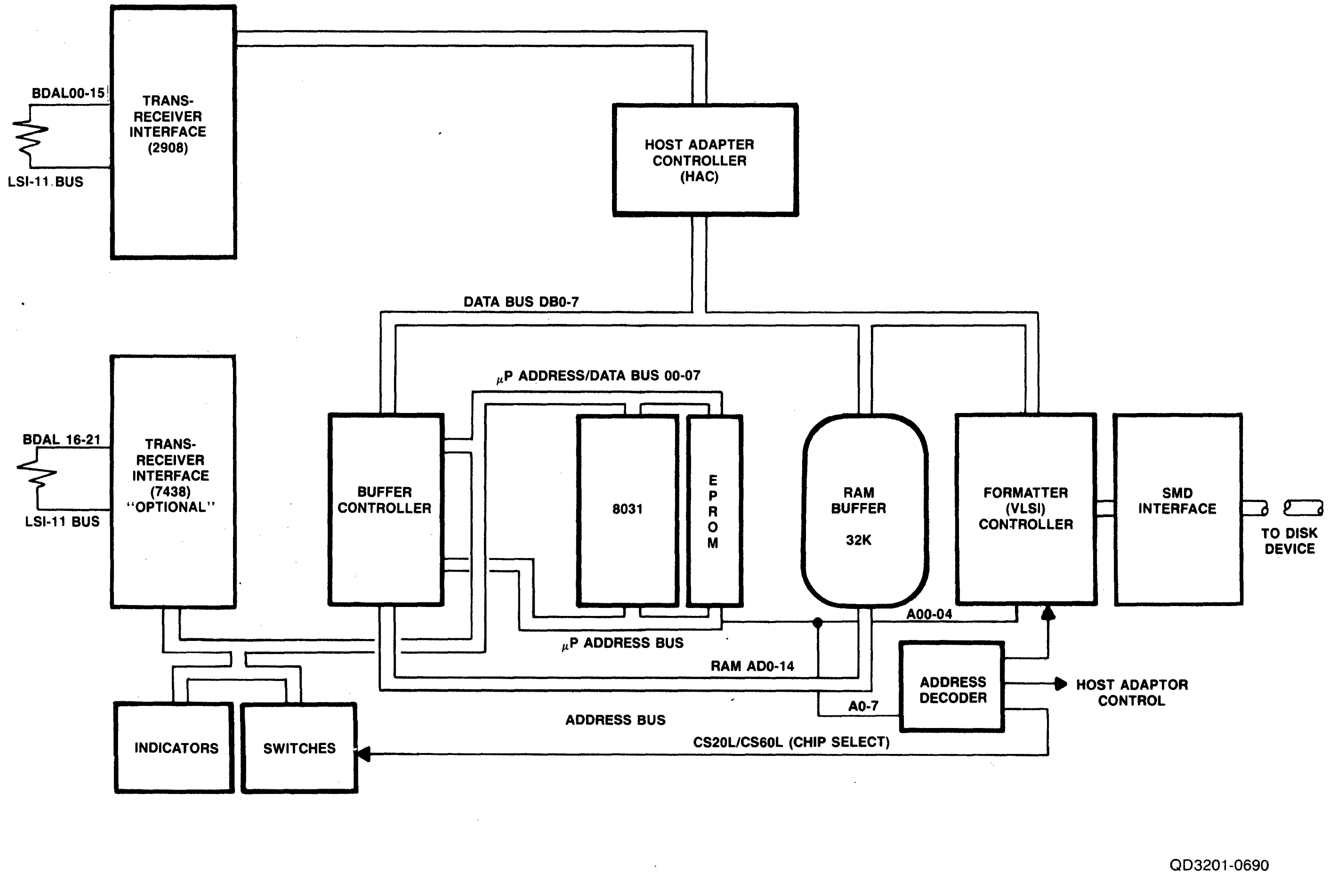


Figure 7-1. QD32 Block Diagram

8.1 OVERVIEW

This section describes the interfaces that the QD32 Disk Controller incorporates. It includes information on the QD32 implementation of SMD interface electrical and mechanical requirements. Excluding this overview, the section is divided into the following subsections.

Subsection	Title
8.2	QD32 LSI-11 Bus Interface
8.3	QD32 SMD Drive Interface

8.2 LSI-11 BUS INTERFACE

The LSI-11 bus between the CPU and the QD32 Disk Controller contains 42 bidirectional signal lines and two unidirectional signal lines on connectors A and B, and two unidirectional signal lines on connector C. LSI-11 bus interface pin assignments are listed and described in Table 8-1. These signal lines provide the means by which the CPU and the QD32 Disk Controller communicate with each other.

The LSI-11 bus interface is used for programmed I/O, CPU interrupts, and DMA data transfer operations. Addresses, data, and control information are sent along these signal lines, some of which contain time-multiplexed information. The LSI-11 bus interface lines are grouped in the following categories:

- **Twenty-two Data/Address Lines** <BDAL00:BDAL21> The four Data/Address lines which carry the most significant bits (MSB) are lines BDAL21:BDAL18. They are used for addressing only and do not carry data. Lines BDAL17 and BDAL16 reflect the parity status of the 16-bit data word during a Write or Read Data Transfer operation via the LSI-11 bus cycle.

- **Six Data Transfer Control Lines** BBS7, BDIN, BDOUT, BRPLY, BSYNC, and BWTBT.

- **Six Direct Memory Access (DMA) Control Lines** BDMR, BSACK, BDMGI, and BDMGO (the last two are on both connectors A and C).

- **Seven Interrupt Control Lines** BEVNT, BIAKI, BIAKO, BIRQ4, BIRQ5, BIRQ6, and BIRQ7.

- **Five System Control Lines** BDCOK, BHALT, BINIT, BPOK, and BREF.

Table 8-1. LSI-11 Bus Interface Pin Assignments

Connector A Signal			Connector B Signal		
Component Side	Pin	Solder Side	Component Side	Pin	Solder Side
BIRQ5	A	+5V	BDCOK	A	+5V
BIRQ6	B		BPOK	B	
BDAL16	C	0V (GND)	BDAL18	C	0V (GND)
BDAL17	D		BDAL19	D	
	E	BDOUT	BDAL20	E	BDAL02
	F	BRPLY	BDAL21	F	BDAL03
	H	BDIN		H	BDAL04
0V (GND)	J	BSYNC	0V (GND)	J	BDAL05
	K	BWTBT		K	BDAL06
	L	BIRQ4		L	BDAL07
0V (GND)	M	BIAKI	0V (GND)	M	BDAL08
BDMR	N	BIAKO	BSACK	N	BDAL09
BHALT	P	BBS7	BIRQ7	P	BDAL10
BREF	R	BDMGI	BEVNT	R	BDAL11
	S	BDMGO		S	BDAL12
0V (GND)	T	BINIT	0V (GND)	T	BDAL13
	U	BDAL00		U	BDAL14
	V	BDAL01		V	BDAL15
Connector C Signal			Connector D Signal		
Component Side	Pin	Solder Side	Component Side	Pin	Solder Side
	A	+5V		A	+5V
	B			B	
	C	0V (GND)		C	0V (GND)
	D			D	
	E			E	
	F			F	
	H			H	
0V (GND)	J		0V (GND)	J	
	K			K	
	L			L	
0V (GND)	M	BIAKI	0V (GND)	M	
	N	BIAKO		N	
	P			P	
	R	BDMGI		R	
	S	BDMGO		S	
0V (GND)	T		0V (GND)	T	
	U			U	
	V			V	

All signals, except BDCOK and BPOK, are low active.

8.2.1 INTERRUPT PRIORITY LEVEL

The QD32 is hardwired to issue both level 4 and level 5 interrupt requests. The level 4 request is necessary to allow compatibility with either an LSI-11 or LSI-11/2 CPU.

8.2.2 REGISTER ADDRESS

The QD32 Disk Controller has two registers visible to the LSI-11 bus. Their addresses are determined by DIP switches SW2-3 through SW2-5. See Section 4 for detailed address and switch setting information.

8.2.3 DMA OPERATIONS

All DMA data transfer operations are performed under microprocessor control. When doing a Read or Write From Memory operation, a check is made for memory parity or non-existent memory (NXM) errors. If an error is detected, an MSCP status error is returned.

8.2.4 SCATTER/GATHER

The QD32 Disk Controller supports the MicroVAX I I/O technique of scatter-write operations and gather-read operations.

8.3 QD32 SMD-E DISK DRIVE INTERFACE

The QD32 Controller interfaces with each SMD disk drive via a 60-pin control cable and one of two 26-pin data cables. A 60-pin male connector at reference designator J1 on the QD32 Controller connects directly into the SMD disk drive control cable. The QD32 Controller contains two 26-pin male connectors, one at reference designator J2 and one at reference designator J3.

The QD32 Controller can control a maximum of two disk drives. Either 26-pin connector (reference designator J2 or J3) can connect directly into the data cable for the first disk drive. If a second disk drive is configured, the unused 26-pin connector is connected into the data cable for that disk drive.

This subsection provides information on the QD32 implementation of the SMD interface electrical and mechanical requirements.

The QD32 controller's disk interface conforms to the SMD-E Interface Specification for 15 MHz and 24 MHz Devices (CDC Document No. 64712402) The controller has been tested with most drives using the SMD-E interface and is compatible with the electrical and timing characteristics disk drives up to 20 MHz.

All communications between the QD32 Controller and its drives pass through the interface. This communication includes all commands, status, control signals, and read/write data transmitted and received by the controller.

The following subsection describes both the I/O cables and I/O signal processing.

8.3.1 I/O CABLES

All the signal lines between the controller and drive are contained in two I/O cables. They are referred to as the A and B cables. Table 8-2 lists all lines (except those not used) in both cables.

8.3.1.1 A Cable

The 60-conductor A Cable is daisy-chained to all disk drives and terminated at the last drive. The purpose of the signals in this cable, along with their function when the control tag (Tag 3) is asserted, are listed in Table 8-3.

The A Cable should be a 30-twisted pair flat cable with an impedance of 100 ohms and a cumulative length not greater than 35 feet.

8.3.1.2 B Cable

The 26-conductor B Cable is radial to all drives and contains the data and clock signals. The function of the signals in this cable are listed in Table 8-4.

The B Cable should be a 26-conductor flat cable with ground plane and drain wire. The impedance should be 130 ohms and the length not greater than 25 feet.

8.3.1.3 Drivers and Receivers

The drivers for the A and B Cables are 26LS31, which are equivalent to 75110A drivers. The receivers are 26LS32 quad differential receivers, which are equivalent to 75108 receivers. The lines of the A Cable are terminated with 82 ohms to ground. The lines of the B Cable are terminated with 56 ohms to ground.

Table 8-2. SMD-E Interface Connections

Pins Lo,Hi	Signal	(Tag 3 Function)	From/To
A Cable:			
22,52	Unit Select Tag		To
23,53	Unit Select bit 0		To
24,54	Unit Select bit 1 /Tag 5[1]		To
26,56	Unit Select bit 2		To
27,57	Unit Select bit 3		To
1,31	Tag 1		To
2,32	Tag 2		To
3,33	Tag 3		To
4,34	Bit 0	(Write Gate)	To
5,35	Bit 1	(Read Gate)	To
6,36	Bit 2	(Servo Offset Plus)	To
7,37	Bit 3	(Servo Offset Minus)	To
8,38	Bit 4	(Fault Clear)	To
9,39	Bit 5	(AM Enable)	To
10,40	Bit 6	(Return to Zero)	To
11,41	Bit 7	(Data Strobe Early)	To
12,42	Bit 8	(Data Strobe Late)	To
13,43	Bit 9	(Release)	To
30,60	Bit 10		To
14,44	Open Cable Detect		To
15,45	Fault		From
16,46	Seek Error		From
17,47	On Cylinder		From
18,48	Index		From
19,49	Unit Ready		From
20,50	Address Mark Found		From
21,51	Busy (dual port only)		From
25,55	Sector		From
28,58	Write Protected		From
29	Power Sequence Hold		To
30,60	Tag 4[1]		To
59	Power Sequence Pick		To
B Cable:			
8,20	Write Data		To
6,19	Write Clock		To
2,14	Servo Clock		From
3,16	Read Data		From
5,17	Read Clock		From
10,23	Seek End		From
22,9	Unit Selected		From
12,24	Index		From
13,26	Sector		From

[1] Tag 4, Tag 5, and Tag 6 are part of the SMD-E extended functions. Tag 6 is the logical AND of Tag 4 and Tag 5. These lines are terminated but their functions are not used by the QD32.

8.3.2 I/O SIGNAL PROCESSING

I/O signals from the controller initiate and control all drive operations except power on. The I/O signals are sent to receivers in the drive and are routed from the receivers to the appropriate drive logic. The drive in turn sends information, concerning the operation back to the controller via the transmitters.

There are two basic types of I/O signals: (1) tag/bus and (2) discrete. The two types differ in that the tag and bus signals work in conjunction to perform a variety of functions while generally the discrete signals work independently each performing a specific function. Both types are described in the following subsections.

8.3.2.1 Tag/Bus Signals

All commands (except unit select) are sent to the drive via the tag and bus signal lines. the tag lines define the basic operation to be performed and the bus lines supply the parameters for the operation.

Table 8-3 explains all the tag/bus commands recognized by the drive.

8.3.2.2 Discrete Signals

In addition to the tag/bus signals, there are various discrete signal lines going between drive and controller. these lines carry clock, status, control and read/write data signals. The function of each of the discrete lines is also explained in Tables 8-3 and 8-4.

Table 8-3. A Cable Signal Line Functions

Signal	Function
Pick In	Used for power sequencing. When the controller is ON, this line is pulled low to power up the disk drive(s). The drive's LOCAL/REMOTE switch must be set to REMOTE and the START switch must be ON. This signal is daisy chained from drive to drive. It is not passed from one drive to the next until the first drive is up to speed. The ground is passed on to the next drive as Pick Out (Pick Out is terminated at last drive in daisy chain).
Hold	Used for power sequencing. This line is pulled low to power up the disk drive. The drive's LOCAL/ REMOTE switch must be set to REMOTE and the START switch must be ON. This signal is daisy chained from drive to drive. This line must be grounded at controller for drive to complete and hold remote power up sequence.
Open Cable	The controller holds this signal TRUE when it is powered on. When false, the drive cannot be selected. This prevents any unwanted command such as Write Gate when the A cable is disconnected or controller power is lost.
Unit Select Tag	Initiates unit select sequence, and in dual channel units it also reserves drive to that controller, provided unit selection is successful (refer to discussion on Unit Selection).
Unit Select lines 2^0 thru 2^3	Used to select the drive. The binary code on these lines must match the code of the drive logical address for the drive to be selected. These lines are used in conjunction with the Unit Select Tag pin A cable I/O) (refer to discussion on Unit Selection).
Tag 1 (Cylinder Select)	Initiates seek functions and used in conjunction with Bus Bit lines. This tag strobes the cylinder address, contained on Bus Bits lines, into drive logic. Drive must be on cylinder before this tag is sent. Bus Bits are interpreted as follows:

(continued on next page)

Table 8-3. A Cable Signal Line Functions (continued)

Signal	Function
	Bus Bit — Function — Bus Bit — Function 0 Cyl Adrs 2^0 5 Cyl Adrs 2^5 1 Cyl Adrs 2^1 6 Cyl Adrs 2^6 2 Cyl Adrs 2^2 7 Cyl Adrs 2^7 3 Cyl Adrs 2^3 8 Cyl Adrs 2^8 4 Cyl Adrs 2^4 9 Cyl Adrs 2^9
Tag 2 (Head Select)	Initiates head select functions and used in conjunction with Bus Bit lines. This tag strobes the head address, contained on bus bit lines, into drive logic. Bus Bits are interpreted as follows: Bus Bit — Functions — Bus Bit — Functions 0 Head Adrs 2^0 5 Not Used 1 Head Adrs 2^1 6 Not Used 2 Head Adrs 2^2 7* Cyl Adrs 2^{10} 3 Head Adrs 2^3 8* Cyl Adrs 2^{11} 4 Head Adrs 2^4 9 Not Used * The extended cylinder address bits are strobed with the head select bits for the SMD-E compatible drives.
Tag 3 (Control Select)	Initiates various operations to be performed by the drive. Used in conjunction with Bus Bit lines and specific operation initiated depends on content of these lines which is defined as follows: Bus Bit — Function 0 Write Gate - Enable write drivers. 1 Read Gate - Enables the digital read data lines. With PLO option, leading edge triggers read chain to sync on all-zeros pattern. 2 Servo Offset Plus - Offsets the actuator from the nominal on cylinder position toward the spindle.

(continued on next page)

Table 8-3. A Cable Signal Line Functions (continued)

Signal	Function	
Tag 3 (continued)	Bus Bit	Function
	3	Servo Offset Minus - Offsets the actuator from the nominal on cylinder position away from the spindle.
	4	Fault Clear - Pulse sent to drive to clear the fault summary flip-flop.
	5	Address Mark Enable - Not used.
	6	RTZ - Pulse sent to drive to cause actuator to seek to track zero.
	7	Data Strobe Early - Enables the PLO data separator (optional) to strobe the data at a time earlier than optimum
	8	Data Strobe Late - Enables the PLO data separator to strobe the data at a time later than optimum.
	9	Release - Releases dual channel drives from reserved and/or priority selected condition (refer to discussion on Unit Selection). Not used for single channel drives.
Tag 4 (Current Sector)		Line is terminated but function not used by QD32.
Tag 5 (Extended Status)		Line is terminated but function not used by QD32.

Table 8-3. A Cable Signal Line Functions (continued)

Signal	Function
Bus Bits (0 - 9)	Used in conjunction with Tags 1, 2, and 3 (also used with Unit Select Tag on units with 50 pin A Cable I/O.
Unit Ready	Indicates that drive is selected, up to speed, heads are loaded and no fault exists.
Busy (applicable only to dual channel units).	TRUE when a drive selection is attempted but the drive is already reserved by the other controller. This signal is returned to the controller attempting selection along with the unit selected signal (refer to discussion on Unit Selection).
On Cylinder	Indicates drive has positioned the heads over a track.
Seek Error	Indicates that the unit was unable to complete a move within 500 msec, or that carriage has moved to a position outside recording field.
Address Mark Found	Not used.
Write Protect	Indicates that drive's write circuits are disabled.
Fault	Indicates that one or more of these faults exists: DC power fault, head select fault, write fault, write or read while off cylinder, and Write Gate during a Read operation
Unit Selected	Indicates that the drive is selected. This line must be active before drive will respond to any commands from controller. However, on dual channel units, if Busy is returned in conjunction with Unit Selected, it indicates the drive is reserved to the other controller and selection was unsuccessful (refer to discussion on Unit Selection.

Table 8-4. B Cable Signal Functions

Signal	Function
Write Data	Carries NRZ data to be recorded on disk pack.
Write Clock	Synchronized to NRZ Write Data, it is a return of the Servo Clock. The drive transmits this signal continuously.
Index	Occurs once per revolution of disk pack and its leading edge is considered leading edge of sector zero.
Sector	Derived from servo surface of disk pack, this signal can occur any number of times per revolution of disk pack. Number of sector pulses occurring depends the configuration of the disk drive.
Servo Clock	Clock signals derived from the drive's servo track.
Read Data	Carries NRZ data recovered from disk pack.
Read Clock	Clock signals derived from NRZ Read Data.
Seek End	Seek End is a combination of ON CYL or SEEK ERROR indicating that a seek operation has terminated.

BLANK

Appendix A
AUTOCONFIGURE, CSR and VECTOR ADDRESSES

A.1 OVERVIEW

The following discussion presents the algorithm for assignment of floating addresses and vectors for all DEC operating systems. Bus addresses are discussed in subsection 3.3.2.

A.2 DETERMINING THE CSR ADDRESS FOR USE WITH AUTOCONFIGURE

The term Autoconfigure refers to a software utility that is run when the computer is bootstrapped. This utility finds and identifies I/O devices in the I/O page of system memory.

Some devices (like the DM11) have fixed addresses reserved for them. Autoconfigure detects their presence by simply testing their standard address for a response. Specifically, the control/status register (CSR) address, which is usually the first register of the block, is tested.

Addresses for those devices not assigned fixed numbers are selected from the floating CSR address space (760010 - 763776) of the Unibus input/output (I/O) page. This means that the presence or absence of floating devices will affect the assignment of addresses to other floating-address devices. Similarly, many devices have floating interrupt vector addresses. According to the DEC standard, vectors must be assigned in a specific sequence and the presence of one type of device will affect the correct assignment of vectors for other devices.

The CSR address for a floating-address device is selected according to the algorithm used during autoconfigure. The algorithm is used in conjunction with a Device Table, Table A-1.

Essentially, Autoconfigure checks each valid CSR address in the floating CSR address space for the presence of a device. Autoconfigure expects any devices installed in that space to be in the order specified by the Device Table. Also, the utility expects an eight-byte block to be reserved for each device that is not installed in the system. Each empty block tells Autoconfigure to look at the next valid address for the next device on the list.

When a device is detected, a block of addresses is reserved for the device according to the number of registers it employs. The utility then looks at the next CSR for that device type. If there is a device there, it is assumed to be of the same type as the one before it and a block is reserved for that device. If there is no response at the next address, that space is reserved to indicate that there are no more devices of that type. Then the utility checks the CSR address (at the appropriate boundary) for the next device in the table.

Table A-1. SYSGEN Device Table

Rank	Device	Number of Registers	Octal Modulus	Rank	Device	Number of Registers	Octal Modulus
1	DJ11	4	10	17	Reserved	4	10
2	DH11	8	20	18	RX11[2]	4	10
3	DQ11	4	10	18	RX211[2]	4	10
4	DU11,DUV11	4	10	18	RXV11[2]	4	10
5	DUP11	4	10	18	RXV21[2]	4	10
6	LK11A	4	10	19	DR11-W	4	10
7	DMC11	4	10	20	DR11-B[3]	4	10
7	DMR11	4	10	21	DMP11	4	10
8	DZ11[1]	4	10	22	DPV11	4	10
8	DZV11	4	10	23	ISB11	4	10
8	DZS11	4	10	24	DMV11	8	20
8	DZ32	4	10	25	DEUNA[2]	4	10
9	KMC11	4	10	**26**	**UDA50**[2]	**2**	**4**
10	LPP11	4	10	27	DMF32	16	40
11	VMV21	4	10	28	KMS11	6	20
12	VMV31	8	20	29	VS100	8	20
13	DWR70	4	10	30	TU81	2	4
14	RL11[2]	4	10	31	KMV11	8	20
14	RLV11[2]	4	10	32	DHV11	8	20
15	LPA11-K[2]	8	20	33	DMZ32	16	40
16	KW11-C	4	10	34	CP132	16	40

[1] DZ11-E and DZ11-F are treated as two DZ11s.

[2] The first device of this type has a fixed address. Any extra devices have a floating address.

[3] The first two devices of this type have a fixed address. Any extra devices have a floating address.

In summary, there are four rules that pertain to the assignment of device addresses in floating address space:

1. Devices with floating addresses must be attached in the order in which they are listed in the Device Table, Table A-1.

2. The CSR address for a given device type is assigned on word boundaries according to the number of UNIBUS- accessible registers that the device has. The following table relates the number of device registers to possible word boundaries.

Device Registers	Possible Boundaries
1	Any Word
2	XXXXX0, XXXXX4
3,4	XXXXX0
5,6,7,8	XXXX00,XXXX20,XXXX40,XXXX60
9 thru 16	XXXX00,XXXX40

   The Autoconfigure utility inspects for a given device type only at one of the possible boundaries for that device. That is, the utility does not look for a DMF32 (16 registers) at an address that ends in 20.

3. An 8-byte gap must follow the register block of any installed device to indicate that there are no more of that type of device. This gap must start on the proper CSR address boundary for that type of device.

4. An 8-byte gap must be reserved in floating address space for each device type that is not installed in the current system. The gap must start on the proper word boundary for the type of device the gap represents. That is, a single DJ11 installed at 760010 would be followed by a gap starting at 760020 to show a change of device types. A gap to show that there are none of the next device on the list, a DH11, would begin at 760040, the next legal boundary for a DH11-type device.

A.3 DETERMINING THE VECTOR ADDRESS FOR USE WITH AUTOCONFIGURE

There is a floating vector address convention that is used for communications and other devices which interface with the Unibus. These vector addresses are assigned in order starting at 300 and proceeding upwards to 777. Table A-2 shows the assignment sequence. For a given system configuration, the device with the highest floating vector rank would be assigned to vector address 300. Additional devices of the same type would be assigned subsequent vector addresses according to the number of vectors required per device, and according to the starting boundary assigned to that device type.

Table A-2. Priority Ranking for Floating Vector Addresses (starting at 300 and proceeding upwards)

Rank	Device	Number of Vectors	Octal Modulus
1	DC11	2	10
1	TU58	2	10
2	KL11[1]	2	10
2	DL11-A[1]	2	10
2	DL11-B[1]	2	10
2	DLV11-J[1]	8	40
2	DLV11,DLV11-F[1]	2	10
3	DP11	2	10
4	DM11-A	2	10
5	DN11	1	4
6	DM11-BB/BA	1	4
7	DH11 modem control	1	4
8	DR11-A, DRV11-B	2	10
9	DR11-C, DRV11	2	10
10	PA611 (reader+punch)	4	20
11	LPD11	2	10
12	DT07	2	10
13	DX11	2	10
14	DL11-C to DLV11-F	2	10
15	DJ11	2	10
16	DH11	2	10
17	VT40	4	20
17	VSV11	4	10
18	LPS11	6	40
19	DQ11	2	10
20	KW11-W, KWV11	2	10
21	DU11, DUV11	2	10
22	DUP11	2	10
23	DV11 + modem control	3	20
24	LK11-A	2	10
25	DWUN	2	10
26	DMC11	2	10
26	DMR11	2	10
27	DZ11/DZS11/DZV11	2	10
27	DZ32	2	10
28	KMC11	2	10
29	LPP11	2	10

(continued on next page)

Table A-2. Priority Ranking for Floating Vectors Addresses (starting at 300_8 and proceeding upwards) (continued)

Rank	Device	Number of Vectors	Octal Modulus
30	VMV21	2	10
31	VMV31	2	10
32	VTV01	2	10
33	DWR70	2	10
34	RL11/RLV11[2]	1	4
35	TS11[2], TU80[2]	1	4
36	LPA11-K	2	10
37	IP11/IP300[2]	1	4
38	KW11-C	2	10
39	RX11[2]	1	4
39	RX211[2]	1	4
39	RXV11[2]	1	4
39	RXV21[2]	1	4
40	DR11-W	1	4
41	DR11-B[2]	1	4
42	DMP11	2	10
43	DPV11	2	10
44	ML11[3]	1	4
45	ISB11	2	10
46	DMV11	2	10
47	DEUNA[2]	1	4
48	**UDA50[2]**	**1**	**4**
49	DMF32	8	40
50	KMS11	3	20
51	PCL11-B	2	10
52	VS100	1	4
53	Reserved	1	4
54	KMV11	2	10
55	Reserved	2	10
56	IEX	2	10
57	DHV11	2	10
58	DMZ32	6	20
59	CP132	6	20

[1] A KL11 or DL11 used as a console, has a fixed vector.

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

[3] ML11 is a Massbus device which can connect to a UNIBUS via a bus adapter.

Vector addresses are assigned on the boundaries indicated in the modulus column of Table A-2. That is, if the modulus is 10, then the first vector address for that device must end with zero (XX0). If the modulus is 4, then the first vector address can end with zero or 4 (XX0, XX4).

Vector addresses always fall on modulo 4 boundaries (XX0, XX4). That is, a vector address never ends in any number but four or zero. Consequently, if a device has two vectors and the first must start on a modulo 10 boundary, then, using 350 as a starting point, the vectors will be 350 and 354.

A.4 A SYSTEM CONFIGURATION EXAMPLE

Table A-3 contains an example of a system configuration that includes devices with fixed addresses and vectors, and floating addresses and/or vectors.

Table A-4 shows how the device addresses for the floating address devices in Table A-3 were computed, including gaps.

Table A-3. CSR and Vector Address Example

Controller	Vector	CSR
1 UDA50	154	772150
1 DZ11	300	760100
1 UDA50	310	760354
2 DHV11	320	760500
	330	760520

Table A-4. Floating CSR Address Assignment Example

Installed	Device		Octal Address
	DJ11	Gap	760010
	DH11	Gap	760020
	DQ11	Gap	760030
	DU11	Gap	760040
	DUP11	Gap	760050
	LK11A	Gap	760060
	DMC11	Gap	760070
---->	DZ11		760100
		Gap	760110
	KMC11	Gap	760120
	LPP11	Gap	760130
	VMV21	Gap	760140
	VMV31	Gap	760150
	DWR70	Gap	760170
	RL11	Gap	760200
	LPA11-K	Gap	760220
	KW11-C	Gap	760230
	Reserved	Gap	760240
	RX11	Gap	760250
	DR11-W	Gap	760260
	DR11-B	Gap	760270
	DMP11	Gap	760300
	DPV11	Gap	760310
	ISB11	Gap	760320
	DMV11	Gap	760340
	DEUNA	Gap	760350
---->	**UDA50 (QD32)**		**772150**[1]
---->	**UDA50 (QD32)**		**760354**
		Gap	760360
	DMF32	Gap	760400
	KMS11	Gap	760420
	VS100	Gap	761440
	TU81	Gap	761450
	KMV11	Gap	761460
---->	DHV11		761500
---->	DHV11		761520
		Gap	761530
	DMZ32	Gap	761540
	CP132	Gap	761600

[1] Fixed address

BLANK

Appendix B
PROM REMOVAL AND REPLACEMENT

B.1 OVERVIEW

This appendix provides instructions for replacing the QD32's firmware PROM.

B.2 EXCHANGING PROMS

The QD32 firmware PROM is located in the socket at U42. Remove the existing PROM from its socket using an IC puller or an equivalent tool.

The QD32 PROM is identified by the part numbers on top of the PROMs. Place the QD32 PROM in U42. Make certain that the PROM is firmly seated and that no pins are bent or misaligned. (If the two rows of PROM pins are too far apart to fit in the socket, grasp the PROM at its ends using your thumb and forefinger and bend one of the pin rows inward by pressing it against a table top or other flat surface.)

PROM Number	PCBA Location
A63	U42

BLANK

Appendix C
UTILITIES AND DIAGNOSITCS

C.1 OVERVIEW

This appendix contains a list of the diagnostics and utilities software that are available for use with the QD32. The list includes a description of the function of the software and a description of the media on which the software is distributed. The media and supporting documentation are supplied in diagnostic distribution kits as described in Table C-1.

All of the diagnostic and utility media listed contain all of the software provided for the QD32 by Emulex.

Table C-1. Utility and Diagnostic Software

Part Number	Media Type	Boot Type	Description
PX9951801-01	0.5-inch tape, 800 bpi	MT	All tape, disk, communications, and subsystem software
PX9951801-02	0.5-inch tape, 1600 bpi	MT	All tape, disk, communications, and subsystem software
PX9951801-03	0.5-inch tape, 1600 bpi	MS	All tape, disk, communications, and subsystem software
PX9951801-04	0.25-inch cartridge tape	MS	All tape, disk, communications, and subsystem software
PX9951802-01	IOMEGA disk cartridge	DL	Emulex Subsystem software (subset of above)
PX9951802-02	0.25-inch cartridge tape	MS	Emulex Subsystem software (subset of above)

(continued on next page)

Table C-1. Utility and Diagnostic Software (continued)

Part Number	Media Type	Boot Type	Description
PX9951802-03	IOMEGA Disk Cartridge	DU	Emulex Subsystem software (subset of above)
VX9951804-01	RX50 Floppy	DU	MicroVAX Diagnostics
VX9951804-02	TK50 Tape Cartridge	MU	MicroVAX Diagnostics

D.1 OVERVIEW

This appendix contains the configuration parameters and sector settings for the following drives which have been certified by Emulex for QD32 support:

Disk Drive	Tables
CDC LMD 9457 Lark	D-1
CDC RSD 9710-80	D-2, D-3
CDC 9715-515	D-4, D-5
CDC 9771 XMD	D-6, D-7, D-8, D-9
CDS-315	D-10, D-11
Fujitsu 2351A	D-12, D-13, D-14, D-15
Fujitsu 2361A	D-16, D-17
Fujitsu 2333	D-18, D-19
Fujitsu 2298	D-20
Fujitsu 2321/22	D-21
NEC 2352	D-22
CDC 9720	D-23
CDC 9715-340	D-24

The drive configuration parameters listed in this appendix relate to the physical geometry of the disk drives; options such as logical splits are left to the user (see subsection 6.3.1.1). The NOVRAM checksums, which indicate a correctly defined configuration, are listed in octal for PDP/LSI systems and hexadecimal for MicroVAX systems.

The correct sector count setting for each drive is specified after the configuration parameter table. If you require further instructions, consult the appropriate manufacturer's drive manual.

The user capacity for each drive is stated in logical blocks before each parameter table. Large capacity drives, such as the CDC 9771 XMD and Fujitsu 2361A, may require special considerations. Refer to subsection D.3 before entering drive parameters.

D.2 CONFIGURATION TABLES

The drive configuration parameters listed in this appendix are based on one spare sector per track. The drive configuration tables list the parameters to be entered under a console emulator in octal for PDP/LSI systems and hexadecimal for MicroVAX I and II systems. A decimal reference is included for Emulex software. If you are entering the parameters under a console emulator, the memory location following the last parameter block must be 0 to indicate no more configurations.

If configured as shown in Table D-1, the CDC LMD 9457 has a user capacity of 38,316 logical blocks.

Table D-1. CDC LMD 9457 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	2	2	2	Head Offset
4	31	37	1F	Sectors per Track
5	4	4	4	Heads
6	622	1156	26E	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	7	7	7	Configuration Bits
10	2	2	2	Split Code
11	1	1	1	Removable Media Flag
12	3080	6010	C08	Gap 0 Parameter
13	5654	13026	1616	Gap 1 Parameter
14	276	424	114	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	4	4	4	Spiral Offset
--		143	63	NOVRAM Checksum

Use a CDC LMD Lark 9457 that is factory configured for 32 sectors.

If configured as shown in Table D-2, the CDC RSD 9710 has a user capacity of 130,784 logical blocks.

Table D-2. CDC RSD 9710 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	32	40	20	Sectors per Track
5	5	5	5	Heads
6	821	1465	335	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	7	7	7	Configuration Bits
10	0	0	0	Split Code
11	1	1	1	Removable Media Flag
12	3080	6010	C08	Gap 0 Parameter
13	5654	13026	1616	Gap 1 Parameter
14	276	424	114	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	2	2	2	Spiral Offset
--		16	E	NOVRAM Checksum

Set the CDC RSD 9710 to 600-byte sectors as shown in Table D-3.

Table D-3. CDC RSD 9710 Sector Setting

Switch	Position	Setting
2^0	0	ON (1)
	1	ON (1)
	2	ON (1)
	3	ON (1)
	4	OFF (0)
	5	OFF (0)
2^{11}	6	OFF (0)
	7	ON (1)
	8	ON (1)
	9	OFF (0)
	10	OFF (0)
	11	OFF (0)

If configured as shown in Table D-4, the CDC 9715-515 has a user capacity of 864,756 logical blocks.

Table D-4. CDC 9715-515 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	51	63	33	Sectors per Track
5	24	30	18	Heads
6	709	1305	2C5	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	2827	5413	B0B	Gap 1 Parameter
14	265	411	109	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		235	9D	NOVRAM Checksum

Set the CDC 9715-515 to 576-byte sectors as shown in Table D-5.

Table D-5. CDC 9715-515 Sector Setting

Switch	Position	Setting
2^0	0	ON (1)
	1	ON (1)
	2	ON (1)
	3	ON (1)
	4	ON (1)
	5	ON (1)
2^{11}	6	ON (1)
	7	ON (1)
	8	OFF (0)
	9	OFF (0)
	10	OFF (0)
	11	OFF (0)

The CDC 9771 XMD has a user capacity of 1,402,832 logical blocks. Table D-6 lists drive parameters for operating systems that can support a single, logical unit of this capacity. For more information, refer to subsection D.3.

Table D-6. CDC 9771 XMD NOVRAM Parameters for a Single Logical Unit

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	86	126	56	Sectors per Track
5	16	20	10	Heads
6	1022	1776	3FE	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	2827	5413	B0B	Gap 1 Parameter
14	265	411	109	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	2	2	2	Spiral Offset
--		363	F3	NOVRAM Checksum

Set the CDC 9771 XMD to 576-byte sectors as shown in Table D-7.

Table D-7. CDC 9771 XMD Sector Setting

Location	Switch	Position	Setting
556	2^0	0	ON (1)
		1	ON (1)
		2	ON (1)
		3	ON (1)
		4	ON (1)
		5	ON (1)
563	2^{11}	6	ON (1)
		7	ON (1)
		8	OFF (0)
		9	OFF (0)
		10	OFF (0)
		11	OFF (0)

The CDC 9771 XMD has a user capacity of 1,402,832 logical blocks. Table D-8 lists drive parameters for operating systems that **cannot** support a single, logical unit of this capacity. For more information, refer to subsection D.3. If configured as shown in Table D-8, each logical drive on the CDC 9771 XMD has a user capacity of 701,416 logical blocks.

Table D-8. CDC 9771 XMD NOVRAM Parameters for Split Logical Units

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	8	10	8	Head Offset
4	86	126	56	Sectors per Track
5	16	20	10	Heads
6	1022	1776	3FE	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	2	2	2	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	2827	5413	B0B	Gap 1 Parameter
14	265	411	109	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	2	2	2	Spiral Offset
--		71	39	NOVRAM Checksum

Set the CDC 9771 XMD to 576-byte sectors as shown in Table D-9.

Table D-9. CDC 9771 XMD Sector Setting

Location	Switch	Position	Setting
556	2^{0}	0	ON (1)
		1	ON (1)
		2	ON (1)
		3	ON (1)
		4	ON (1)
		5	ON (1)
563	2^{11}	6	ON (1)
		7	ON (1)
		8	OFF (0)
		9	OFF (0)
		10	OFF (0)
		11	OFF (0)

If configured as shown in Table D-10 using a drive model with a total of 823 cylinders, the CDS 315 has a user capacity of 528,530 logical blocks. If configured as shown in Table D-10 using a drive model with a total of 845 cylinders, the CDS 315 has a user capacity of 542,742 logical blocks.

Table D-10. CDS 315 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	34	42	22	Sectors per Track
5	19	23	13	Heads
6	821/843	1465/1513	335/34B	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	2827	5413	B0B	Gap 1 Parameter
14	265	411	109	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		241/337	A1/DF	NOVRAM Checksum

Set the CDS 315 to 576-byte sectors as shown in Table D-11.

Table D-11. CDS 315 Sector Setting

Switch	Position	Setting
SW2	1	ON (1)
	2	ON (1)
	3	OFF (0)
	4	ON (1)
	5	ON (1)
	6	ON (1)
	7	OFF (0)
	8	OFF (0)
SW3	1	OFF (0)
	2	OFF (0)
	3	OFF (0)
	4	ON (1)

If you are using 845 cylinders, drive switch SW3-8 must be ON.

If configured as shown in Table D-12, the Fujitsu M2351A has a user capacity of 787,156 logical blocks.

Table D-12. Fujitsu M2351A NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	47	57	2F	Sectors per Track
5	20	24	14	Heads
6	840	1510	348	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	4112	10020	1010	Gap 1 Parameter
14	268	414	10C	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	0	0	0	Spiral Offset
--		116	4E	NOVRAM Checksum

Set the Fujitsu M2351A to 587-byte sectors as shown in Table D-13.

Table D-13. Fujitsu M2351A Sector Setting

Location	Jumpers
BC7	3-4, 5-6, 10-11, 12-13
BD7	3-4, 6-7, 9-10, 13-14
BE7	3-4, 5-6, 10-11, 13-14
BF7	3-4, 6-7, 10-11, 13-14
AE7*	3-4, 6-7, 9-10*

* This location selects the SMD-0 interface.

The Fujitsu M2361A has a user capacity of 1,122,116 logical blocks. Table D-14 lists drive parameters for operating systems that can support a single, logical unit of this capacity. For more information, refer to subsection D.3.

Table D-14. Fujitsu M2361A NOVRAM Parameters for a Single Logical Unit

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	67	103	43	Sectors per Track
5	20	24	14	Heads
6	840	1510	348	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	4112	10020	1010	Gap 1 Parameter
14	268	414	10C	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		67	37	NOVRAM Checksum

Set the Fujitsu M2361A to 593-byte sectors as shown in Table D-15.

Table D-15. Fujitsu M2361A Sector Setting

Switch	Position	Setting
2^1	1	OFF (0)
	2	OFF (0)
	3	OFF (0)
	4	OFF (0)
	5	OFF (0)
	6	OFF (0)
	7	ON (1)
	8	OFF (0)
2^0	1	OFF (0)
	2	ON (1)
	3	OFF (0)
	4	ON (1)
	5	OFF (0)
	6	OFF (0)
	7	OFF (0)
	8	ON (1)

The Fujitsu M2361A has a user capacity of 1,122,116 logical blocks. Table D-16 lists drive parameters for operating systems that **cannot** support a single, logical unit of this capacity. For more information, refer to subsection D.3. If configured as shown in Table D-16, each logical drive on the Fujitsu M2361A has a user capacity of 560,924 logical blocks.

Table D-16. Fujitsu M2361A NOVRAM Parameters for Split Logical Units

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	10	12	A	Head Offset
4	67	103	43	Sectors per Track
5	20	24	14	Heads
6	840	1510	348	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	2	2	2	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	4112	10020	1010	Gap 1 Parameter
14	268	414	10C	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		270	B8	NOVRAM Checksum

Set the Fujitsu M2361A to 594-byte sectors as shown in Table D-17.

Table D-17. Fujitsu M2361A Sector Setting

Switch	Position	Setting
2^1	1	OFF (0)
	2	OFF (0)
	3	OFF (0)
	4	OFF (0)
	5	OFF (0)
	6	OFF (0)
	7	ON (1)
	8	OFF (0)
2^0	1	OFF (0)
	2	ON (1)
	3	OFF (0)
	4	ON (1)
	5	OFF (0)
	6	OFF (0)
	7	ON (1)
	8	OFF (0)

If configured as shown in Table D-18, the Fujitsu M2333 has a user capacity of 548,194 logical blocks.

Table D-18. Fujitsu M2333 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	67	103	43	Sectors per Track
5	10	12	A	Heads
6	821	1465	335	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	4112	10020	1010	Gap 1 Parameter
14	268	414	10C	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		256	AE	NOVRAM Checksum

Set the Fujitsu M2333 to 594-byte sectors as shown in Table D-19.

Table D-19. Fujitsu M2333 Sector Setting

Switch	Position	Setting
2	1	OFF (0)
	2	OFF (0)
	3	OFF (0)
	4	ON (1)
	5	OFF (0)
	6	ON (1)
	7	OFF (0)
3	1	OFF (0)
	2	ON (1)
	3	OFF (0)
	4	OFF (0)
	5	OFF (0)
	6	OFF (0)
	7	OFF (0)

Table D-20. Fujitsu 2298 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0*	0	0	Head Offset
4	68	31	19	Sectors per Track
5	16	20	10	Heads
6	1022	1776	3FE	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0*	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	4112	10020	1010	Gap 1 Parameter
14	268	414	10C	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		204	84	NOVRAM Checksum

* A single drive is too large for RSX11M, and RSTS/E must use a split logical. The recommended head offset is 8, with a split code of 2.

Set the Fujitsu 2298 to 586-byte sectors.

Table D-21. Fujitsu 2321/22 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	33	41	21	Sectors per Track
5	5/10	5/12	5/0A	Heads
6	821	1465	335	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	4112	10020	1010	Gap 1 Parameter
14	268	414	10C	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		321/243	D1/A3	NOVRAM Checksum

Set the Fujitsu 2321/22 to 586-byte sectors.

Table D-22. NEC 2352 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	62	76	3E	Sectors per Track
5	19	23	13	Heads
6	758	1366	2F6	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	2827	5413	B0B	Gap 1 Parameter
14	265	411	109	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		111	49	NOVRAM Checksum

Table D-23. CDC 9720 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	51	63	33	Sectors per Track
5	10	10	A	Heads
6	1215	2277	4BF	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	2827	5413	B0B	Gap 1 Parameter
14	265	411	109	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		50	28	NOVRAM Checksum

Set the CDC 9720 to 255 servo clocks per sector, with a 806 kHz servo clock.

Table D-24. CDC 9715-340 NOVRAM Parameters

Word	Dec	Oct	Hex	Description
1	1	1	1	Number of Drives
2	1	1	1	Type Code
3	0	0	0	Head Offset
4	32	40	20	Sectors per Track
5	24	30	18	Heads
6	711	1307	2C7	Cylinders
7	1	1	1	Spare Sectors per Track
8	2	2	2	Spare Cylinders
9	6	6	6	Configuration Bits
10	0	0	0	Split Code
11	0	0	0	Removable Media Flag
12	259	403	103	Gap 0 Parameter
13	2827	5413	B0B	Gap 1 Parameter
14	265	411	109	Gap 2 Parameter
15	0	0	0	Cylinder Offset
16	1	1	1	Spiral Offset
--		214	8C	NOVRAM Checksum

D.3 SPECIAL CONSIDERATIONS FOR LARGE CAPACITY DRIVES

Some DEC operating systems have limits on the number of blocks that can be supported by an individual device. If you are using a large capacity drive, such as a CDC 9771 XMD, you must be aware of these special considerations when entering the drive configuration parameters.

If your application uses MicroVMS, the operating system automatically sizes the device during the disk initialization process. You do not need to make any adjustments to the drive configuration parameters.

However, if your application uses either the RSX-11M, RSX-11M-PLUS, or RSTS/E operating systems, you must ensure that the number of blocks on the drive does not exceed the operating system limit for individual devices. The limit for individual devices under RSX-11M and RSX-11M-PLUS is 1,044,480 blocks; the limit for RSTS/E is 1,048,576 blocks.

If the user area on the formatted drive is larger than the number of blocks supported by the operating system, you must either split the drive into logical units of an acceptable size or limit the drive capacity.

To determine if the drive capacity falls within the operating system limits for individual devices, use the following procedure:

1. Determine the approximate number of usable blocks on a formatted drive by multiplying the number of sectors per track (not including spares) by the number of data heads by the number of cylinders (not including spares).

   For a CDC 9771, 86 sectors per track multiplied by 16 data heads equals 1376; 1376 multiplied by 1022 cylinders equals 1,406,272 blocks. (Data taken from Table D-10.)

2. Compare the number of usable blocks to the number of blocks allowed for an individual device. The limit for individual devices under RSX-11M and RSX-11M-PLUS is 1,044,480 blocks; the limit for RSTS/E is 1,048,576 blocks.

   The number of usable blocks on a CDC 9771, which has approximately 1,406,272 blocks, is greater than the number of blocks allowed for an individual device under the RSX-11M, RSX-11M-PLUS, and RSTS/E operating systems.

3. If the number of usable blocks is greater than the limit for your operating system, you may either limit the drive capacity, or split the drive into logical units that do not exceed the operating system limits.

   The number of usable blocks for a CDC 9771 exceeds the limits for the RSX-11M, RSX-11M-PLUS, and RSTS/E operating systems. For an example of a logical split, refer to Table D-10.

4. If the number of usable blocks is less than the limit for your operating system, the operating system will support the entire drive capacity as one device.