HEWLETT PACKARD

RTE-6/VM DVM33/DVN33

Reference Manual

Software Technology Divison
11000 Wolfe Road
Cupertino, CA 95014-9804

Manual Part No. 92084-90025
E0693

Printed in U.S.A. June 1993
Second Edition

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THE MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARs 252.227.7013.

Copyright © 1981-1983, 1985, 1986, 1993 by Hewlett-Packard Company

Printing History

The Printing History below identifies the edition of this manual and any updates that are included. Periodically, update packages are distributed that contain replacement pages to be merged into the manual, including an updated copy of this printing history page. Also, the update may contain write-in instructions.

Each reprinting of this manual will incorporate all past updates; however, no new information will be added. Thus, the reprinted copy will be identical in content to prior printings of the same edition with its user-inserted update information. New editions of this manual will contain new information, as well as all updates.

To determine which manual edition and update is compatible with your current software revision code, refer to the Manual Numbering File or the Computer User's Documentation Index. (The Manual Numbering File is included with your software. It consists of an "M" followed by a five digit product number.)

First Edition	Dec 1981	
Update 1	Jul 1982	Manual Corrections
Update 2	Jan 1983	Include HP 7914P/R Disk
Reprint	Jan 1983	Updates 1 and 2 Incorporated
Update 3	Jan 1985	
Update 4	Jan 1986	
Reprint	Jan 1986	Updates 3 and 4 Incorporated
Second Edition	Jun 1993	Revision 6000

Preface

This instruction manual describes the DVM33 driver, the software interface for devices connected to the HP 12821A Disk Interface. Before attempting to use the driver, you should be familiar with the information contained in the *HP 12821 Disk Interface Manual*, part number 12821-90006, and the *HP CS/80 Instruction Set Programming Manual*, part number 5955-3442. Other related manuals are defined in the *Index to Operating System Manuals*, part number 92084-90001.

To facilitate references to the information, this manual is organized into seven chapters.

Chapter 1 Introduces the DVM33/DVN33 driver and defines the software components and operating environment. This section also contains a functional overview of the driver.

Chapter 2 Defines the DVM33 read and write request operating mode. These include track and sector addressing and block-addressed requests. Special read/write requests such as retrieving DESCRIBE information and track-map information also are described. This chapter also defines the use of the special EXEC call provided to allow a utility program to directly control a CS/80 device.

Chapter 3 Describes the methods for obtaining the device status. The error fields (reject errors, fault errors, access errors and information errors) returned in the status information words are defined together with the most probable causes of the errors.

Chapter 4 Describes the cartridge tape drive (CTD) control requests and defines the disk cache memory scheme used with the CTD cartridge tapes.

Chapter 5 Defines the routines that can be used to directly control the CS/80 and Subset 80 devices. Examples are provided for the disk control routines.

Chapter 6 Provides procedures for generating a CS/80 drive into your system. Procedural steps are provided for generating the Equipment Table (EQT), Device Reference Table (DRT), and Interrupt Table entries. The disk Track Map Table format also is given.

Chapter 7 Defines the DVM33 error message format and lists the most likely causes of I/O errors.

Table of Contents

List of Illustrations

Tables

1

General Introduction

Introduction

Driver DVM33 is the RTE-6/VM operating system software interface for devices connected to the HP 12821A Disk Interface Card. Up to four disk and cartridge tape drives may be driven by a single card. Refer to Table 6-1 for disk model numbers.

Note The HP 9144 CTD is not supported as a boot device. It can be used for other functions, for example, loading media updates or non-system backups.

Refer to the *HP 12821A Disk Interface Installation and Service Manual*, part number 12821-90006, for details of installation and cabling restrictions. A second driver may be used with another HP 12821A card to increase the number of devices that can be connected to the system. This driver, uniquely identified by the system as DVN33, is a copy of DVM33 with components edited as necessary to rename entry points. Functionally, it is identical to DVM33.

Components

The software components of the DVM33 and DVN33 consist of the following modules:

Name	Description
%DVM33	Binary relocatable module of DVM33
%$TM33	Track Map Table (required only if DVM33 is not a system disk driver)
$CSERR	Error reporting module
$DTCLB	Direct Command Library
%DVN33	Binary relocatable module of DVN33

Operating Environment

The operating environment for driver DVM33 (and DVN33) is the RTE-6/VM operating system. The hardware components are the following:

HP 1000 M/E/F-Series Computer
HP 12821A Disk Interface Card
HP Command Set 80 or Subset 80 Disk Drives or
Cartridge Tape Drives

The peripherals are referred to as the Command Set 80 (CS/80) or Subset 80 devices. Each device may contain several units; for example, a 7908 and a CTD may be in the same drive unit with one disk drive controller. A sample disk subsystem configuration is shown in Figure 1-1. Refer to Chapter 6 of this manual and to the HP 12821A Disk Interface Manual for configuration information.

Related Manuals

The following manuals are related to DVM33 and may be ordered from any Hewlett-Packard Sales Office.

Manual Title	Part Number
HP CS/80 Instruction Set Programming Manual	5955-3442
HP 12821A Disk Interface Manual	12821-90006
RTE Operating System Driver Writing Manual	92200-93005
RTE-6/VM Programmer's Reference Manual	92084-90005
RTE-6/VM Utility Programs Reference Manual	92084-90007
RTE-6/VM System Manager's Reference Manual	92084-90009
RTE-6/VM Online Generator Reference Manual	92084-90010

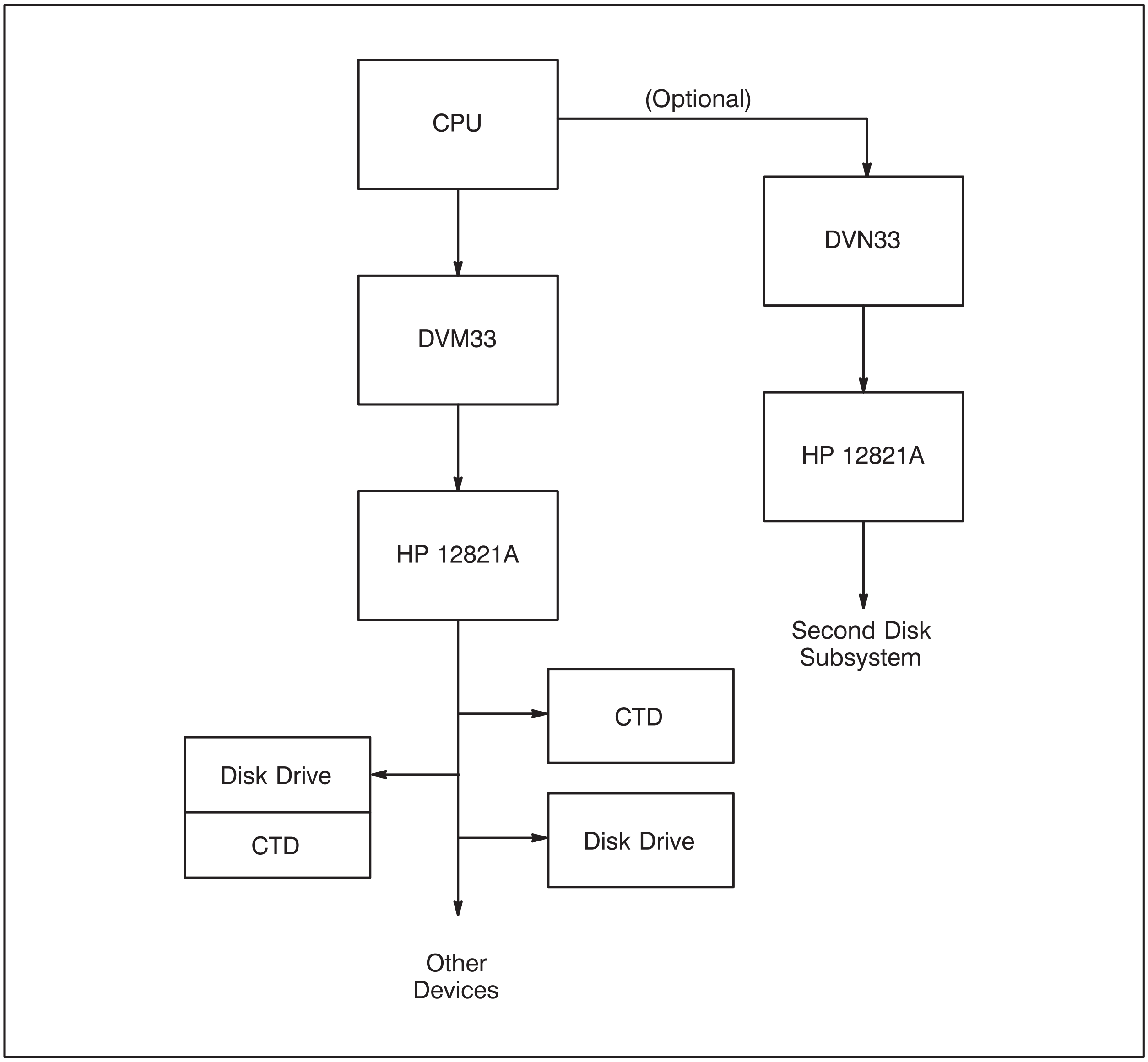

Figure 1-1. Sample Disk Subsystem Configuration

Functional Overview

Driver DVM33 provides the means with which programmers can write application programs to access the CS/80 or Subset 80 disk or cartridge tape drives. Calls can be included in Macro, FORTRAN, or Pascal programs to invoke the driver for disk or tape drive access.

The driver communicates with the disk or tape drives through the HP 12821A Disk Interface Card. A drive is accessed by means of an HP-IB address between 0 and 7, assigned during the operating system generation and set with a switch on the drive.

While all drives accessed by DVM33 have 128-word sectors, the number of sectors per track can vary. The capacities of each drive are given in Chapter 6 of this manual.

Each word (or integer) as used in this manual contains 16 bits. DVM33 supports transfers on 64-word logical sector boundaries. Due to the read-modify-write sequence required for 64-word logical-sector access, performance degradation occurs with read requests beginning on odd sectors, or write requests ending in even sectors.

All devices are addressed in block mode (at the driver level) so that they appear uniform in format. This also allows standard access to all CS/80 and Subset 80 disk devices without concern for track size and number of surfaces.

To optimize disk access time, all CS/80 and Subset 80 disk drives are in “pack” mode. This means that the drive attempts to seek to the next surface before seeking to the next cylinder. Fixed and removable packs are physically separate volumes that are addressed individually. This “pack” mode facility also makes it possible for DVM33 to handle transfers of up to 32k in one operation with no interruptions, thus significantly improving disk access times on RTE systems.

DVM33 has four modes of operations: standard RTE EXEC read and write (the primary operating mode), status request, CTD control, and direct disk control. These operations are briefly described below and detailed in the following chapters.

EXEC Read and Write

The EXEC read and write operation includes the use of pack (cylinder) mode to speed up long transfers (up to 32k in one contiguous transfer). This type of request uses track and sector or block addresses in its EXEC parameters. This mode also uses caching requests through a disk buffer to improve the speed of an integrated CTD.

Status Requests

The status request operation consists of using calls to obtain information about the state or configuration of a device. This includes requests to return status, subchannel size, track map information, and the CS/80 DESCRIBE information.

CTD Control

The CTD control operation consists of using calls to control the operation of a CTD drive. If a cache exists, this operation includes opening and closing the cache, writing file marks, and unloading the tape.

Direct Disk Control

This mode of operation provides calls that allow programs such as SWTCH, FORMC, and the backup utilities to control the disk directly. These requests bypass all checking and mapping of logical to physical subchannels. They are available through a library called $DTCLB.

Optimizing CS/80 Disk Usage

The CS/80 disk drives are low cost, high performance drives. DVM33 takes advantage of the high-speed features in the drive to minimize access times.

CS/80 drives are laid out in pack mode to minimize track seek times. This means that for any volume (fixed or removable pack) the drive attempts to seek to the next surface before moving the head to the next cylinder. Consequently, the use of this mode is forced and surface mode operation is not allowed. The main benefit derived from this mode of operation is the ability to transfer up to 32k words of data in one operation.

The CS/80 drives have built-in error detection and correction algorithms. This ensures that the best possible data is transferred. The system is warned when data blocks should be spared before data is actually lost.

Sparing on CS/80 drives is invoked by a command. When invoked by a program (for example, the FORMC utility), the device controller chooses the optimal sparing algorithm, and places the best guess at the data in a spare area previously not accessible to the user. All spared accesses are transparent and guaranteed to minimize latency time.

Because of the error correction and detection schemes, verify after write is not supported. It is not optimized for fast execution and requires considerable overhead for the device controller. The verify command is provided as part of the formatting process, which should uncover nearly all of the defects on a pack. Refer to the description of FORMC in the *RTE-6/VM Utility Programs Reference Manual*, part number 92084-90007.

The error rate on a CS/80 drive is projected to be better than 1 in 10^{13}. This, in conjunction with the diagnostic testing done by the power-on sequence, helps to ensure a high level of data integrity without the need to verify.

Even with long transfers and pack mode operation, some careful planning of disk subchannels can affect performance tremendously. Care should be taken to place frequently accessed subchannels adjacent to each other on the disk, or on totally different drives. The most frequently used subchannels should be placed near the center of the disk to minimize seek times.

2

Read and Write Requests

General

Read and write requests to DVM33 are the usual mode of operation. These requests are optimized to provide minimum transfer time for requests up to 32k words using DMA. Error correction is done in the disk controller to provide data integrity. Any errors that occur during a disk operation are automatically retried by the disk controller. This relieves the system of unnecessary retry overhead and optimizes the chances that retry will work.

Upon successful completion of the request, the B-Register contains the transmission log in positive words or characters depending on what was requested. The A-Register contains EQT word 5, which is the drive status. Refer to Chapter 7 for fault conditions.

Normal read and write requests use subfunction 0. For read and write requests with nonzero subfunctions, refer to the following sections for descriptions of their operation.

Two special EXEC reads are supported that are normally illegal in the RTE operating system. Both require that the subfunction be set to zero. If an EXEC read requests a TRACK = −1 and SECTOR = 0, the B-Register is returned with the number of tracks in the subchannel and IBUFR(1) contains the number of 64-word logical sectors per track. A read request with TRACK = −1 and SECTOR = −1 returns status information (refer to Chapter 3). All other negative tracks and sectors will abort the program.

Track and Sector Requests

The usual mode of operation for the driver is track and sector addressing. In this mode, the calling program passes track and sector addresses to the driver. This address is then converted to the block address on the disk, according to the specifications in the track map table. Requests that do not start on an even sector incur a latency, as the driver must buffer the data. Write requests should be multiples of 128 words beginning on an even sector and ending in an odd sector. Requests that do not start on an even sector and end in an odd one will incur a latency in order to do a read-modify-write that preserves the rest of the block. Sectors that are not filled by a write will be filled with random data.

Block Addressed Requests

Block addressing is supported to allow a program to access a disk or CTD as a contiguous segment of 128-word blocks for disk or 512-word blocks for CTD.

For disk requests, the subchannel is seen as a group of blocks ranging from 0 to n with n=(#tracks * #blocks-per-track)−1. The block address is a double word with the high word in IADD1 and the low word in IADD2.

For CTD requests, the block address is a 512-word block on the tape. The double-word address (IADD1 and IADD2) is passed directly to the tape if a subfunction of 1 is specified. If a subfunction of 2 is specified, the request is passed through a cache on the disk before it goes to the CTD. Cached mode is transparent to the program and, if enabled, increases the performance significantly. Refer to Chapter 4 of this manual for a discussion of the caching scheme.

Macro:

```
EXT EXEC
    .
    .
    .
JSB EXEC
DEF RTRN
DEF ICODE
DEF ICNWD
DEF IBUFR
DEF IBUFL
DEF IADD1
DEF IADD2
```

where:

RTRN = Return address
ICODE = Request code
- 1 = Read
- 2 = Write

ICNWD = Control word
- Bits 0−5 = LU number of subchannel
- Bits 6−10 = Control Subfunction:
  - 0 = Standard Read or Write
  - 1 = Block Addressing
  - 2 = Block Addressing, cached access (CTD)
  - 3 = Block Addressing, Write File Mark (CTD)
  - 21B = Return Describe Information
  - 22B = Return Track Map Information
  - 23B = Direct Control Mode

IBUFR = Starting address of data buffer
IBUFL = Length of transfer; positive number of words or negative number of characters (rounded to even number)
IADD1 = Track number in standard addressing mode (high order word of block number in Block mode)
IADD2 = Sector number in standard addressing mode (low order word of block number in Block mode)

FORTRAN:

```
CALL EXEC (ICODE, ICNWD, IBUFR, IBUFL, IADD1, IADD2)
```

Figure 2-1. Read and Write Calling Sequence, Macro and FORTRAN

Special Read and Write Requests

Two special read requests return information about the disk or CTD. The first, a read with a subfunction of 22B, will return information from the track map table. The second call, a read with a subfunction of 21B, will return the CS/80 DESCRIBE information for the address pointed to by the LU in the EXEC call.

A third type of request, a write with a subfunction of 3, is used to write a file mark on a CTD. This subfunction is related to the use of the CTD and its cache. Refer to the section on CTD usage for particulars of when to write the file mark.

A fourth type of request, subfunction 23B, allows a program to directly control the disk subsystem. The format of the calls is dependent on the function being performed. These calls are described in Chapter 5 of this manual.

Retrieving the Track Map Table

A read with a subfunction of 22B returns information from the track map table. If a length of 8 words is requested, the driver returns the track map table for the subchannel of the LU requested. If a length of 513 words is requested, the driver returns the entire track map table. The format of the table is shown in Chapter 6.

Retrieving the DESCRIBE Information

A read request with a subfunction of 21B will return the CS/80 (Subset 80) DESCRIBE information for the unit pointed to by the track map for the requested LU. The call returns 37 bytes, so a length of at least 37 bytes must be specified. Refer to the *CS/80 Instruction Set Programming Manual*, part number 5955-3442, for details of the information returned by DESCRIBE.

Write File Mark

The Write-File-Mark subfunction posts the contents of the CTD disk buffer, if necessary, and writes a file mark at the block number given in the IADD1 and IADD2 parameters. If the CTD is cached, the cache is posted to the CTD if it has been written on, and then the file mark is written at the block address passed. The cache is left cleared after the request. If the CTD is not cached, the file mark is written at the block address passed in IADD1 and IADD2. Length and buffer address parameters are ignored. File marks use an entire block, which cannot contain any data.

Direct CS/80 Command Calls

A special EXEC call is provided to allow a utility program to directly control a CS/80 or Subset 80 device. This call permits direct device control and access to functions not provided by the normal read and write sequences. This is especially useful for formatting disks, sparing bad blocks and performing diagnostic functions.

All known disk commands are provided in a library, $DTCLB, that sets up the parameters to the special EXEC call. This special call is of the following form:

```
CALL EXEC [ICODE, ICNWD, IBUFR, IBUFL, ITYPE]
```

where:

`ICODE` = EXEC Function = 1 (the type of call is specified by ITYPE)

`ICNWD` = Control word:
Bits 0 - 5 = LU that points to a DVM33 subchannel on the bus
Bits 6 - 10= 23B to indicate special mode

`IBUFR` = Control and Data Buffer:

Word 1 = Primary listen, HP-IB address, ATN, odd parity (440B + HP-IB address w/odd parity)

Word 2 = Command, ATN, odd parity (745B for type 0 - 3)
(762B for type 4 - 6)

Words 3 - 59 = Command buffer for disk protocol.
Command specific.
Last command has sign bit set (LBO).

For words 1 - 59, commands are in low byte; control (ATN, LBO) are in the high byte.
ATN − attention (for commands) − 400B.
LBO − last byte out flag, data tagged with EOI − 100000B.

Word 60 = QSTAT value:
0 = normal completion
1 = hard error
2 = power on state
−1 = timeout on request

Word 61 = User data area

`IBUFL` = Expect data transfer length in positive words or negative bytes

`ITYPE` = Transaction type:
0 = command-report (QSTAT)
1 = command-outbound data-report (QSTAT) write
2 = command-inbound data-report (QSTAT) read
3 = report (QSTAT)
4 = transparent command-report (QSTAT) read
5 = transparent command-outbound data-report (QSTAT)
6 = transparent command-inbound data-report (QSTAT)

This mode makes no checks on any boundary conditions. The user is in complete control of the disk subsystem and may issue any type of request to any device on the bus.

On return, the A-Register is not significant. The B-Register contains the transmission log in a positive number of words. There are no error conditions.

EXEC Function and Subfunction Codes

The EXEC function and subfunction codes are shown and described in Table 2-1.

Table 2-1. EXEC Function and Subfunction Codes

Function	Sub-Function	Description

1	0	Read Request, Track and Sector addresses
		IADD1 = −1 and IADD2 = 0 Subchannel size
		#Tracks in Breg, Sectors per Track in BUFFR(1)
		IADD1 = −1 and IADD2 = −1 Dynamic Status Request
		10 or 20 words in BUFFR, depending on length

	1	Block Addressing
	2	Block Addressing, Cached Requests (CTD only)
	21B	Return Describe Information
	22B	Return Track Map Table
	23B	Direct Protocol Control mode
2	0	Write Request, Track and Sector Addresses
	1	Block Addressing
	2	Block Addressing, Cached Requests (CTD only)
	3	Write EOF, post and Close Cache (CTD only) write File Mark at address in IADD1 and IADD2
	23B	Direct Protocol Control mode
3	1	Clear Cache (CTD only) reset cache and check write protect
	2	Post and Close Cache (CTD only) writes cache to tape if necessary
	3	Unload tape (CTD only) causes tape to rewind and unload

3

Status Request

EQT Status

There are two types of status available to DVM33: EQT status and dynamic status. An EXEC 13 request returns EQT word 5 and, optionally, EQT word 4 and DRT word 2. Normally, these words indicate device type and provide a hint as to the type of error. This is not a dynamic request, but reflects the last request to the driver. The EXEC 13 status request has the form:

```
CALL  EXEC(13,ICNWD,ISTA1[,ISTA2[,ISTA3]])
```

where:

`ICNWD` = Logical Unit number

`ISTA1` = EQT word 5 (see Table 3-1)

`ISTA2` = EQT word 4 (optional)

`ISTA3` = DRT word 2 (optional)

Refer to the *RTE-6/VM Programmer's Reference Manual*, part number 92084-90005, Appendix C, for a description of EQT 5, EQT 4, and DRT 2.

Table 3-1. Format of Status Bits, EQT Word 5

Bit	Meaning
0	Severe Error.
1	Channel Errors. Set for any Reject or DMA length error. If this bit is set, the severe bit (bit 0) is also set.
2	Not ready. Unit not ready for access. If this bit is set, the severe bit is also set.
3	Fault. A Fault error has occurred. If this bit is set, the severe bit is also set.
4	Uninitialized Media. If this bit is set, the severe bit is also set.
5	EOF/EOV. The severe bit is set only for end of volume.
6	Unrecoverable data/data-recoverable marginal error. The severe bit is set only for unrecoverable data)
7	Write protected volume. The severe bit is set only if the write failed.

Dynamic Status Request

A special EXEC call is provided to retrieve all of the status information returned by the controller. This information contains only the target address after a normal completion. Be aware that RTE-6/VM does not guarantee that another request affecting status was not received after the faulty operation and before the status request. DVM33 automatically requests and preserves status after an error return from the controller.

The request requires a length of at least 10 words to retrieve the status in the driver, or a length of at least 20 words for both the status in the driver and the current status words in the controller. The dynamic status request has the following form:

```
CALL  EXEC(ICODE,ICNWD,IBUFFR,IBUFFL,IOPT1,IOPT2)
```

where:

`ICODE` = Request code = 1 for read

`ICNWD` = Control word. Bits 5 - 0 = logical unit

`IBUFFR` = Buffer for status date

`IBUFFL` = 10 for status length (status in driver) 20 for status length (driver status and current drive status)

`IOPT1` = −1

`IOPT2` = −1

CS/80 Device Status

CS/80 devices return 10 words of status information ordered by severity, with the most serious errors shown in the first words of status. The 10 words are:

word 1 − identification field: status unit, volume
word 2 − reject errors (highest severity)
word 3 − fault errors
word 4 − access errors
word 5 − information errors
word 6 - 10 − parameter area for words 2 through 5

Identification Field

The identification field format is:

VVVVUUUUSSSSSSSS

where:

VVVV = volume number for this status
UUUU = unit number for this status
SSSSSSSS = other unit requesting service (−1 = none)

Reject Errors Field

The reject errors field format is:

< x x 1 x x 2 3 4 5 6 7 x 8 x x x >

where:

1 = Channel Parity Errors (20000B)
A channel command was received without odd parity. Generally indicates a programming error.

2 = Illegal Opcode (2000B)
An unrecognizable opcode was received. Generally a programming error.

3 = Module Addressing (1000B)
Illegal Unit or Volume was received.

4 = Address Bounds (400B)
The target address exceeded the bounds for this device.

5 = Parameter Bounds (200B)
A parameter other than unit, volume or address is out of bounds for this device.

6 = Illegal Parameter (100B)
A parameter field has the wrong length for the opcode preceding it.

7 = Message Sequence Error (40B)
The message sequence has been violated. This may be a programming error; that is, an execution message was required, but none was sent, or something was lost and the request should be retried.

8 = Message Length Error (10B)
The length of the execution message did not match the length command. A programming error or a bus fault has occurred.

Fault Errors Field

The fault errors field format is:

< x 1 x 2 x x 3 x 4 x 5 6 7 x 8 9 >

where:

1 = Cross Unit Error (40000B)
An error has occurred during a copy data operation. The units involved are listed in the parameter area.

2 = Controller Fault (10000B)
A hardware controller fault has occurred. Run the diagnostic or call your local HP Customer Engineer.

3 = Unit Fault (1000B)
A hardware unit fault has occurred. Run the diagnostic or call your local HP Customer Engineer.

4 = Diagnostic Result (200B)
The hardware failed a diagnostic test, indicated in the parameter field. Run the diagnostic or call your local HP Customer Engineer.

5 = Release for Operator Request (40B)
Release is required by the drive to service an operator request; that is, load or unload. Programs should grant or deny this request as appropriate for the application.

6 = Release Required for Diagnostic (20B)
Release is required for diagnostics initiated from the front panel. This request must be granted to complete a CTD load, but may be denied if necesary.

7 = Release for Internal Maintenance (10B)
Release is required so that the drive can update its logs. If release is not granted, error information may be lost.

8 = Power Fail (2B)
Power to the drive failed, or state was lost.

9 = Release Completed (1B)
The drive has returned from automatic release.

Access Errors Field

The access errors field format is:

< 1 2 3 4 5 6 x x 7 8 x 9 10 x x x >

where:

1 = Illegal Parallel Operation (100000B)
The operation requested cannot be executed in parallel with other operations in progress. This error should not occur on RTE as no parallel operations are permitted.

2 = Uninitialized Medium (40000B)
The medium is unformatted in this device, or an unusable medium has been loaded. The medium should be formatted before further access is attempted.

3 = No Spare Areas Available (20000B)
No spare areas are left for the spare block command. Either the medium should be replaced or the drive should be serviced to determine if more spare areas can be recovered.

4 = Not Ready (10000B)
The selected unit is not ready for access. The medium may not be loaded or the device may not be online.

5 = Write Protect (4000B)
The device is write protected. This error only occurs on commands which write to the device.

6 = No Data Found (2000B)
A block accessed during read has not been written. This generally occurs on tape that has not been initialized.

7 = Unrecoverable Data Overflow (200B)
The last transaction had more than one unrecoverable data error. Address of the first error is in the parameter area.

8 = Unrecoverable Data Error (100B)
The controller was unable to read data at address in parameter area. The data cannot be totally recovered, although reading with maximum retries will get the controller's best guess.

9 = End of File (20B)
An end of file mark was encountered during the read. The address in the parameter area is the address of the block after the end-of-file mark.

10 = End of Volume (10B)
The last access attempted to access past the end of the volume. The parameter area contains the address of the last block on the volume.

Information Errors Field

The information field format is:

< 1 2 3 4 5 x x 6 x 7 8 9 x 10 x x >

where:

1 = Release for Operator Request (100000B)
The controller requests release to allow load/unload, save/restore. Release may be ignored or denied if necessary. System will grant release during idle time if a release request occurs.

2 = Release for Diagnostic Request (40000B)
Release for diagnostic generated by front panel or self test. If ignored, system will grant release during idle time.

3 = Release for Internal Maintenance (20000B)
Request release to update error logs or perform maintenance. If release is not granted, system will release during idle time.

4 = One Spare Left (10000B)
There is only one spare block left on this volume. The medium should be backed up and replaced or serviced to see if spares can be recovered.

5 = Data Overrun (4000B)
During a transfer, a latency occurred when the system could not accept data as fast as the device can send. No data was lost.

6 = Auto Sparing Invoked (400B)
A defective block was automatically spared. No data was lost.

7 = Recoverable Data Overflow (100B)
The last request generated more than one recoverable data error. The first error is in the parameter area.

8 = Marginal Data (40B)
The last request generated an error which was recovered with difficulty. The block should be spared before it becomes uncorrectable.

9	=	Recoverable Data Error (20B) The last request generated an error which was recovered by retry or error correction. No data was lost.
10	=	Maintenance Track Overflow (4B) The maintenance track on the device has overflowed. The error and fault logs are full. The drive should be serviced to recover this information before it is lost.

Parameter Field

The Parameter Field contains the parameters appropriate to the most serious error seen. This area contains the target or fault address, except for the following errors:

Spare Block	After the spare block, the parameter area contains the address of the beginning of the affected area in the first 3 words. The last 2 words contain the length, in bytes, of the affected field.
Cross Unit	The parameter area contains a list of units that experienced errors. Each unit is 1 byte long, and the list is terminated by a byte of 377B.
Diagnostic Results	The first 3 words of the parameter area contain the diagnostic numbers which failed. Each number is 1 byte long.

4

Cartridge Tape Drive Control Requests

General

Driver DVM33 provides a set of requests for controlling the Cartridge Tape Drive (CTD). These control requests allow a program to make use of the disk buffer for the CTD, write file marks on the tape, and unload the tape from the drive. These calls and a brief explanation of how to use the CTD are given in this chapter.

Using the Cartridge Tape Drive

The Cartridge Tape Drive is a high capacity device. It requires that data be transferred in a continuous streaming mode, with or without buffering. Non-buffered control, read, and write requests go directly to the CTD. Buffering involves creating a disk buffer, called the disk cache, at generation time. The disk cache is 64k bytes long in an area reserved on the associated disk drive. This area is used only for the CTD disk buffering. If the Cartridge Tape Device supports caching, that is, the CTD shares a controller with the disk drive, buffering should be used.

Buffering improves transfer times to CTD by an order of magnitude for transfers over 8k bytes. For Cartridge Tapes, primarily Subset 80 devices, that do not support caching (that is, have their own dedicated controller), non-buffered transfers with immediate reporting option should be used. Immediate reporting allows the device contoller and driver to process the pending request without waiting for completion of the current I/O write to the CTD. This feature takes advantage of large internal buffers inside the device and approaches a streaming mode of operation, reducing delay time due to start/stop requirements of the tape mechanism.

Caching Scheme

If a program is to use the CTD, it should lock the LU of the CTD to itself to protect the integrity of the disk buffer and tape. At this point it is necessary to initialize the disk buffer and cache pointers to a known state. This is accomplished by issuing EXEC calls to the CTD LU, clearing the cache. This will clear all the cache pointers and check the CTD for write protected and certified tapes. It is possible now to read and write to the cache. The cache is set up for forward reference; that is, the blocks in the cache are a duplicate of a contiguous set of blocks on the CTD, beginning with the first block accessed. This set of blocks is accessed from the disk buffer until the buffer is overflowed, closed, or has a file mark written to it.

At this point, if the buffer has been written into, it is posted and a new buffer is brought in if necessary. If an end-of-file mark is encountered on a read, it will show up when that block is accessed. This process of reading and writing continues until all the desired data is transferred. The program should then close the cache to ensure that the last set of blocks written is posted and the cache is reset. The program also has the option of writing an end-of-file mark on the tape, and it may also programmatically unload the CTD. The close, write-file-mark, and unload functions are all implemented as EXEC control calls.

Clear Cache

Cache clearing is accomplished with a subfunction code of 1 in an EXEC control call. The subfunction codes are explained in Chapter 2 of this manual. The control call allows a program to clear the CTD cache before the CTD is accessed. If the cache is not cleared, it is possible to overwrite random data on the CTD. This call also checks the CTD for write-protected and certified cartridges. This status is returned in the A-Register (EQT word 5) regardless of whether or not caching is used.

Close Cache

Cache closing is accomplished with a subfunction code of 2 in the EXEC control call. This call is used to terminate cleanly the use of the disk buffer for the CTD. This call posts the disk buffer to the CTD if the buffer has been written and leaves the cache in a clear state. This call guarantees that blocks written into the disk buffer are moved to the CTD. If no cache is present, driver DVM33 takes no action and returns to the program.

Unload Tape

Unloading tape from a cartridge tape drive can be done programmatically with a subfunction code of 3 in an EXEC control call. This call rewinds the tape so that the cartridge can be removed. It can be used for both cached or uncached CTD; and DVM33 returns to the programs in either case.

An example program using the CTD control calls is shown on the following page.

```
FTN7X
      INTEGER CTD,DISK,TRACK,SECTR,LENTH
      INTEGER BLOCK(2),BUFFR(8192),SIZE(2)
      INTEGER*4 DBLCK,DSIZE
      EQUIVALENCE (DBLCK,BLOCK(1)),(DSIZE,SIZE(1))
   .
   .
   .
C
C    SET UP THE CTD, DISK LUS, AND INITIALIZE LENGTH AND BLOCK
C
      CTD = 10
      DISK = 20
      LENTH = 8192
      DBLCK = 0
      SECTR = 0
C
C   FIND OUT HOW LONG THE TAPE IS  (BUFFR(17) AND BUFFR(18))
C       REFER TO THE CS80 REF. MANUAL FOR DESCRIBE FIELDS
C
      CALL EXEC(1,CTD+2100B,BUFFR,-37,0,0)
      SIZE(1) = BUFFR(17)
      SIZE(2) = BUFFR(18)
C
C   CLEAR THE CTD CACHE
C
      CALL EXEC(3,CTD+100B)
C
C    MOVE DATA FROM DISK TO CTD BY WAY OF THE CACHE
C
      DO 10 TRACK = 1,1000
      CALL EXEC (1,DISK,BUFFR,LENTH,TRACK,SECTR)
C
      CALL EXEC (2,CTD+200B,BUFFR,LENTH,BLOCK(1),BLOCK(2))
   10 DBLCK = DBLCK + LENTH/512
C
C
C     CLOSE THE CACHE NOW AND WRITE A FILE MARK
C
      DBLCK = DBLCK + 1
      CALL EXEC (2,CTD+300B,BUFFR,0,BLOCK(1),BLOCK(2))
C
C     UNLOAD THE TAPE FOR THE USER
C
      CALL EXEC (3,CTD+300B)
    .
    .
    .
```

5

Direct Disk Control

Control Function Routines

Direct disk control can be done through a group of routines contained in a command library, $DTCLB. The control functions available and the associated routine are listed below.

Command	Routine
Cancel	XCNCL
Channel Independent Clear	XCICL
Cold Load Read	XCOLD
Complementary Commands	XCOMP
Copy Data	XCOPY
Describe	XDESC
Initialize Media	XINMD
Initiate Diagnostic	XDIAG
Initiate Utility	XUTIL
Locate & Read	XLCRD
Locate & Verify	XLCVF
Locate & Write	XLCWR
Read Loopback Test	XRLPB
Release	XRELS
Release denied	XRELD
Request status	XRQST
Selected Device Clear	XSDCL
Spare Block	XSPRE
Unload	XUNLD
Write File Mark	XFMRK
Write Loopback Test	XWLPB

Disk Control Routines CALL Format

The calling format for the direct disk control routines is as follows:

```
CALL  routine(LU,IADDR,ICOMP,IBUF)
```

where:

LU	Target LU. Gets routine to the correct driver.
IADDR	HP-IB Address of CS/80 drive.
ICOMP	A 24-word array containing parameters for all the possible complementary commands. ICOMP is a static buffer, not changed by $DTCLB routines. The meaning of each word is described in Table 5-1.
IBUF	An array. The first 60 words are reserved for use by the routine in building a command. The use of word 61 and beyond is dependent upon the particular routine. On return, words 58 through 60 contain the following:

IBUF(58)	A-Register contents
IBUF(59)	B-Register contents
IBUF(60)	QSTAT

If IBUF(60) is a 1, the full status is returned in IBUF(2) through IBUF(11). The QSTAT associated with the request status command is returned in IBUF(1).

Table 5-1. Complementary Command Array (ICOMP)

Word	Meaning
1	Unit number
2	Volume
3	Address Mode: 0 = Block; 1 = 3 vector
4	Target address
5	Target address
6	Target address
7	Set Block Displacement: −1 = No block displacement; 1 = Valid block displacement in words 8 and 9
8	Block displacement
9	Block displacement (Note: Block displacement is a signed 6-byte integer. The first two bytes will be filled with the sign bit.)
10	Set length flag: −1 = No length; 1 = Valid length in 11 and 12
11	Length
12	Length
13	Burst Size (Set Bit 15 to tag bursts with EOI)
14	RPS Time 1
15	RPS Time 2
16	Number of read retries
17	Set status mask flag: −1 = No status mask; 1 = Valid status mask in 18−22
18	Status mask
19	Status mask
20	Status mask
21	Status mask
22	Set release: −1 = No Set release
23	Set address return mode: 0 = Block; 1 = 3 vector
24	Set device specific options

Complementary Commands

To use a complementary command, simply set the appropriate word or words in the ICOMP array. All other words should be set to −1. Not all complementery commands are compatible with all commands. Refer to the *CS/80 Instruction Set Programming Manual* for specifics.

The use of word 61 or greater in IBUF is routine dependent. Usage by these routines is described below.

Routine	Usage			
XCNCL	Not used			
XCICL	Not used			
XCOLD	Return area for read data			
XCOMP	Not used			
XCOPY	Source:		Destination:	
	IBUF(61)	Unit	IBUF(67)	Unit
	IBUF(62)	Volume	IBUF(68)	Volume
	IBUF(63	Address Mode	IBUF(69)	Address Mode
	IBUF(64)	Address	IBUF(70)	Address
	IBUF(65)	Address	IBUF(71)	Address
	IBUF(66)	Address	IBUF(72)	Address
XDESC	Return area for DESCRIBE data			
XFMRK	Not used			
XFORM	IBUF(61) = Option IBUF(62) = Interleave Factor			
XINTP	IBUF(61) = Option			
XDIAG	IBUF(61) = Not used IBUF(62) = Not used IBUF(63) = Length in bytes of parameter string IBUF(64) - IBUF(66) Parameter string left justified			
XLCRD	Return area for read data			
XLCVF	Not used			
XLCWR	Source area for write data			
XRELS	Not used			
XRELD	Not used			
XRLPB	IBUF(61 - 62) Loopback Length in bytes On Return: Return area for Loopback data			
XRQST	Return area for Status information			
XSDCL	Not used			
XSPRE	Parameter field			
XUNLD	Not Used			
XUTIL	IBUF(61) Utility number IBUF(62) Type of execution message 0 = No execution message 1 = Device receives message 2 = Device sends message IBUF(63) Length in bytes of parameter string IBUF(64) - IBUF(67) Parameter string left justified. IBUF(68) Length of execution message in bytes IBUF(69) - IBUF(*n*) Execution message if IBUF(62)=1 (where *n* depends on the length of the execution message) On return: IBUF(61) - IBUF(*m*) Contains execution message if execution message was sent or received (*m* depends on length of message)			
XWLPB	IBUF(61 - 62) Loopback Length in bytes IBUF(63 - *n*) Source for Loopback data (where *n* depends on length of data)			

Examples

Following are examples using the direct disk control routines.

XLCRD/XLCWR/XLCVF/XSPRE Usage

Assuming you have an LU and HP-IB address, the following is a FORTRAN calling sequence to read data from a disk starting at block 100 for a length of 2 (disk) blocks (512 bytes) and write it to block 200 of the same disk. A verify is then issued to ensure that the data at block 200 can be read.

Note that IBUF must be at least 188 words long; 60 words for XLCRD and 128 words for the data from the disk.

In this example, note that only those values that change for the next call need be changed in the ICOMP array. In the call to XLCWR, the only change is in the block number from the previous call. In the call to XLCVF, the same area that was just written is also being verified, so no change is neccessary in the ICOMP array.

If a series of blocks is to be read, equate ICOMP(5) and ICOMP(6) to a 4-byte integer and keep incrementing the integer by one (or more) for each succesive read. Using four of the available six bytes is sufficient to cover the maximum block address possible.

The spare command is very similar to the locate and read command in terms of calling sequence. To spare block 100, you would go through the same sequence as for XLCRD. Setting the length in ICOMP would have no effect, however, on the spare command.

```
C               CLEAR ICOMP ARRAY
      DO 10 I=1,24
         ICOMP(I)=-1
   10 CONTINUE
         .
         .
         .
C
C               DO THE READ
       ICOMP(1)=0              !UNIT 0
       ICOMP(2)=0              !VOLUME 0
       ICOMP(3)=0              !BLOCK ADDRESSING
       ICOMP(4)=0              !3 WORD
       ICOMP(5)=0              !   BLOCK ADDRESS
       ICOMP(6)=100            !             OF 100
       ICOMP(10)=1             !INDICATE LENGTH IS SET
       ICOMP(11)=0             !DOUBLE WORD
       ICOMP(12)=512           !   NUMBER OF BYTES
       CALL XLCRD(LU,IADDR,ICOMP,IBUF)
C
C              DO WRITE
       ICOMP(6)=200                !NEW BLOCK NUMBER
       CALL XLCWR(LU,IADDR,ICOMP,IBUF)
C
C              VERIFY BLOCK WRITTEN
       CALL XLCVF(LU,IADDR,ICOMP,IBUF)
           .
           .
```

XINMD Usage

The following will format disk unit 0, volume 0, retaining all factory and field spares and setting a block interleave factor of 1. Refer to the *RTE-6/VM Utility Programs Reference Manual*, part number 92084-90007, for a description of interleaving.

```
          DO 10 I=1,24
             ICOMP(I)=-1
       10 CONTINUE
                .
                .
                .
          ICOMP(1)=0                !UNIT 0
          ICOMP(2)=0                !VOLUME 0
          IBUF(61)=0                !KEEP FACTORY & FIELD SPARES
          IBUF(62)=1                !INTERLEAVE
C
          CALL XINMD(LU,IADDR,ICOMP,IBUF)
```

XRELS/XRELD Usage

To release unit 0:

```
          DO 10 I=1,24
             ICOMP(I)=-1
       10 CONTINUE

               .
               .
               .
          ICOMP(1)=0                    !UNIT 0
          CALL XRELS(LU,IADDR,ICOMP,IBUF)
```

To deny release, the sequence is exactly the same but the call is to XRELD.

XRQST/XDESC Usage

Status and describe information requests have similar calling sequences, however, the set volume command has no effect on the request status command. To get the status of unit 0, the sequence would be:

```
        DO 10 I=1,24
           ICOMP(I)=-1
    10  CONTINUE
          .
          .
          .
        ICOMP(1)=0                    !UNIT 0
        CALL XRQST(LU,IADDR,ICOMP,IBUF)
```

Status is returned (in packed format) in IBUF(61) and on. IBUF(59) has the contents of the B-Register, which should be the number of bytes transferred (this should be 20).

To issue a DESCRIBE command, you would also set ICOMP(2) to the volume number. The description is returned (in packed format) in IBUF(61) and on. IBUF(59) contains the number of bytes returned.

XCOPY Usage

Use the following sequence to copy 10 blocks (2560 bytes) on unit 0, volume 0 starting at block 100, to unit 1, volume 0, block 200.

```
      DO 10 I=1,24
       ICOMP(I)=-1
10    CONTINUE
         .
         .
         .
      ICOMP(1) =15                 !TALK TO CONTROLLER UNIT
      ICOMP(10)=1                  !INDICATE LENGTH IS SET
      ICOMP(11)=0                  !DOUBLE WORD
      ICOMP(12)=2560               !       NUMBER OF BYTES
 C
 C        SET FROM UNIT/VOL/ADDR
      IBUF(61)=0                   !UNIT 0
      IBUF(62)=0                   !VOLUME 0
      IBUF(63)=0                   !ADDR MODE IS BLOCKS
      IBUF(64)=0                   !6 BYTE
      IBUF(65)=0                   !   BLOCK
      IBUF(66)=100                 !      ADDRESS
C
C         SET DEST UNIT/VOL/ADDR
      IBUF(67)=1                   !UNIT 1
      IBUF(68)=0                   !VOLUME 0
      IBUF(69)=0                   !ADDR MODE IS BLOCKS
      IBUF(70)=0                   !6 BYTE
      IBUF(71)=0                   !   BLOCK
      IBUF(72)=200                 !     ADDRESS
C
C          COPY DATA
       CALL XCOPY(LU,IADDR,ICOMP,IBUF)
```

Note that the XCOPY routine returns no data (except, of course, the QSTAT and A- and B-Register returns). What is placed in IBUF(61) onwards remains intact after the call. If you wish to do a series of copy commands you could increment the block addresses in IBUF(70) through IBUF(72) and IBUF(64) through IBUF(66) by the appropriate number, then go back and call XCOPY again.

XCOMP Usage

XCOMP sends only complementary commands in the command message. This has the effect of permanently changing the disk parameters, as opposed to sending the complementary commands with another command (which causes them to be in effect only for that command sequence). The following sequence illustrates the use of XCOMP:

```
      DO 10 I=1,24
         ICOMP(I)=-1
   10 CONTINUE
        .
        .
        .
C
C          CHANGE SELECTED DISK PARAMETERS
      ICOMP(7)=1            !FLAG FOR BLOCK DISP
      ICOMP(8)=0            !DOUBLE WORD
      ICOMP(9)=10           !   BLOCK DISP OF 10 BLOCKS
      ICOMP(13)=1024        !SET BURST SIZE TO 1024 BYTES
      CALL XCOMP(LU,IADDR,ICOMP,IBUF)
```

The parameters set in the above will be in effect for all subsequent commands to the disk that do not send their own values for them in complementary commands. The parameters will remain set until another call to XCOMP or until the disk is powered on or cleared.

6

Configuration Information

General

This chapter contains configuration information for DVM33. Useful information concerning this process may be found in the following manuals:

RTE-6/VM Online Generator Reference Manual, part number 92084-90010
RTE-6/VM System Manager's Manual, part number 92084-90009
RTE-6/VM Programmer's Reference Manual, part number 92084-90005
RTE-6/VM Utility Programs Reference Manual, part number 92084-90007

These manuals will help you generate and configure a system with CS/80 drives. Follow the procedures in this chapter to generate a CS/80 drive into your system.

Program Input Phase

During the program input phase of the generation process, load the driver, error module and track map table along with the other I/O drivers by making the following entry:

```
Prog. Input Phase
      .
      .
RE, %DVM33
RE, %$TM33
RE, $CSERR
```

RE, %$TM33	*Only relocate the track map table if DVM33 is not the system disk driver.*
RE, $CSERR	*This is the extended error reporting module, a memory-resident program. Its presence is not required, but it provides extremely useful information to service personnel. For more information on the 10 words of device status returned, see Chapter 3 of this manual, the System Generation Response Preparation section of the RTE-6/VM System Manager's Reference Manual, and the I/O Status EXEC 13 section of the RTE-6/VM Programmer's Reference Manual.*

Table Generation Phase

During the table generation phase make the following entries:

1. Equipment Table Entry [EQT]

   An EQT entry must be made for the EQT you wish to have pointing to DVM33.

```
Equipment Table Entry
   .
   .
EQT n?
sc, DVM33, D, T=x
```

   where:

   n is the EQT entry number
   sc is the select code of the HP 12821A interface card (CS/80 devices only; ICD disks require a separate card)
   x is the timeout value chosen for the disk subsystem

2. Device Reference Table [DRT] entry

   A Device Reference Table entry must be made for each subchannel to be accessed by the system.

```
Device Reference Table
   .
   .
lu = EQT ?
n,m
```

   where:

   lu is a LU number
   n is the EQT entry number associated with the LU
   m is the EQT subchannel number associated with the LU

Note All disk LU numbers must be less than 64. RTE-6/VM allows 64 subchannels per EQT.

---

3. Interrupt Table Entry

   An interrupt table entry must be defined for the HP 12821A interface card:

```
Interrupt Table
   .
   .
sc, EQT, n
```

   where:

   sc is the select code of the HP 12821A card (CS/80 devices only)
   n is the EQT number from step 1

If you want a second CS/80 / HP 12821 subsystem in your system, follow the steps above for another select code and EQT using %DVN33.

The track map tables can be built by editing the source file, &$TM33, or by following the track map table format given in this chapter.

Note To create a track map for %DVN33, copy &$TM33 into a new file named &$TN33. Edit this new file, changing its entry points to $TN33.

Device Timeout

The device timeout value is normally set at generation time in the EQT entry. This is described in the *RTE-6/VM System Manager's Manual*, part number 92084-90009. The timeout can also be set interactively using the "TO" command, described in the *RTE-6/VM Quick Reference Guide*, part number 92084-90003.

The device timeout value should be set depending on whether or not CTD operations occur on the device. Some CTD operations may take 20 to 60 seconds. If the system does not include any CTD, a shorter timeout may be used. A typical timeout value is 2 seconds.

If, however, the CS/80 device is an HP 7933 or HP 7935, the drive performs automatic head alignment. During the alignment, the drive may return a not-ready status for up to 2 seconds. For these drives, the timeout value should be slightly greater than 2 seconds.

If the device includes a CTD, the timeout value should be set to allow for the longest CTD seek time expected (about 20 seconds for a short tape and 60 seconds for a long tape).

Track Map Table Format

The track map table format is shown on the following page. The first word of the table contains the negative number of subchannels. Each subchannel definition is eight words long; the subchannel definitions are in ascending order.

Disk Track Map Table Format

	15 ... 8	7 ... 0
word 0	reserved	HP-IB Address
word 1	Unit number	Volume number
word 2	High word	of starting block number
word 3	Middle word	of starting block number
word 4	Low word	of starting block number
word 5	No. of tracks	
word 6	No. of sectors per track	
word 7	Reserved	

CTD Track Map Table Format

	15 ... 8	7 ... 0
word 0	1 | I | reserved	HP-IB Address
word 1	CTD Unit number	CTD Volume
word 2	c | Disk cache unit	Disk Volume
word 3	High word	starting block of cache on disk
word 4	Low word	starting block of cache on disk
word 5	d | Reserved	
word 6	Reserved	
word 7	Reserved	

where: 1 = set indicates a CTD.
c = set indicates a cached CTD. 0 for noncached CTD.
d = dirty bit set by driver. Reserved
I = CTD supports immediate reporting (see Chapter 4).

Reserved words must be set to zero.

CS/80 Devices

The common characteristics of the CS/80 devices are shown in Table 6-1.

Table 6-1. CS/80 Disk Drive Characteristics

Model	Capacity (Mbytes)	Transfer Rate (Kbytes)	Number of Surfaces	Number of Cylinders	Sectors per Track (128 words)	Number of Blocks	Access Time (ms)
HP 7907A	41	600	2/unit	627	64	80,256/unit	44
HP 7908A	16	561	5	370	35	64,750	58
HP 7911A	27	835	3	572	64	109,824	43
HP 7912A	64	835	7	572	64	256,256	43
HP 7914A	132	1000	7	1152	64	516,096	41
HP 7933/35	404	1000	13	1321	92	1,579,916	35
HP 7936	307	1000	7	1396	123	1,201,956	31
HP 7937	571	1000	13	1396	123	2,232,204	31
HP 7941/42	24	625	3	968	32	92,928	48
HP 7945/46	55	625	7	968	32	216,832	48
HP 7957A	81	853	5	1013	63	319,095	42
HP 7957B	81	875	4	1269	63	319,788	28
HP 7958A	130	853	8	1013	63	510,552	42
HP 7958B	152	875	6	1572	63	594,216	28
C2200	335	1840	8	1449	113	1,309,840	27
C2203	670	1840	16	1449	113	2,619,792	27
CTD	67/16	35				* 65,408/16,352	

* Number of blocks on a CTD depends on the cartridge tape size.

Example Generation

The following portion of an RTE-6/VM generation list file shows how to generate an HP 9122D into your system. In this example, the system disk is an HP 7908 in select code 12. The HP 9122D is configured as one LU per floppy and it shares the HP 12821 interface card with the HP 7908. User responses are in lowercase.

```
SYSTEM DISK MODEL?
cs80

CONTROLLER SELECT CODE?
12

DEVICE (MODEL,HP-IB ADDR,UNIT VOLUME)?
ctd,0,1,0
SUBCHANNEL 0 ASSIGNED

DEVICE (MODEL,HP-IB ADDR UNIT,VOLUME)?
7908,0,0,0

64750     BLOCKS REMAINING (STARTING AT 0        )
SUBCHANNEL 1  (TRACKS,BLOCKS/TRACK)?
400,48

45550    BLOCKS REMAINING (STARTING AT 19200     )
SUBCHANNEL 2  (TRACKS,BLOCKS/TRACK)?
943,48

286      BLOCKS REMAINING (STARTING AT 64464     )
SUBCHANNEL 3  (TRACKS,BLOCKS/TRACK)?
ctd,0

30       BLOCKS REMAINING (STARTING AT 64720     )
SUBCHANNEL 3 (TRACKS,BLOCKS/TRACK)?
-30,1

DEVICE (MODEL,HP-IB ADDR,UNIT,VOLUME)?
9122,1,0,0            * HP-IB address 1, unit 0

NUMBER OF BLOCKS ON DEVICE?
2464                  * total number of blocks (left unit)

2464      BLOCKS REMAINING (STARTING AT 0        )
SUBCHANNEL 3  (TRACKS,BLOCKS/TRACK)?
154,16                * 154 tracks, 16 blocks/track

DEVICE  (MODEL,HP-IB ADDR,UNIT,VOLUME)?
9122,1,1,0            * HP-IB address 1, unit 1

NUMBER OF BLOCKS ON DEVICE?
2464                  * total number of blocks(right unit)
```

```
2464        BLOCKS REMAINING (STARTING AT 0        )
SUBCHANNEL 4   (TRACKS,BLOCKS/TRACK)?
154,16                 * 154 tracks, 16 blocks/track

DEVICE (MODEL,HP-IB ADDR,UNIT,VOLUME)?
/E

SYSTEM SUBCHNL?
   .
   .
   .
EQUIPMENT TABLE ENTRY

EQT 0001?
12,dvm33,d,t=10000      * EQT 01 System Disk
   .
   .
   .
DEVICE REFERENCE TABLE

   .
   .
   .
002 = EQT #?
1,1                     * LU 2 7908 disk

003 = EQT #?
1,2                     * LU 3 7908 disk

004 = EQT #?
1,0                     * LU 4 7908 CTD
   .
   .
   .
044 = EQT #?
1,3                     * LU 44 9122D Left Unit

045 = EQT #?
1,4                     * LU 45 9122D Right Unit
   .
   .
   .
INTERRUPT TABLE

   .
   .
   .
12,EQT,1                * 7908 and 9122 disk
   .
   .
   .
```

This procedure is not specific to a single CS/80 device since DVM33 treats all devices as a sequence of blocks. If RT6GN does not know how many blocks are on a device, it will prompt with: “NUMBER OF BLOCKS ON DEVICE?”.

7

Error Conditions

General

This chapter defines the DVM33 error message format and lists the most likely causes of I/O errors.

IO07 Error

Illegal request, driver rejects call; program aborted. Reasons:

a. Requested subchannel out of range. Subchannel associated with requested LU is not defined in track map table.

b. Request too large. Requested starting address + length overflowed subchannel.

c. Requested negative track (except track = −1 for read) or track too large.

d. Requested sector was negative or larger than number of sectors per track (except sector= −1 which is used for status requests).

Track Error

Unrecoverable error was encountered.

Error Format:

```
TRnnnnnn EQTeq Usu S(or U)
SCODE sc EQTeq SUBCH su
ADDRESS a  QSTAT q
ssssss ssssss ssssss ssssss
ssssss ssssss ssssss ssssss
ssssss ssssss
```

where:

nnnnnn	=	logical track number
eq	=	EQT number
su	=	subchannel on EQT
`S` (or `U`)	=	system (`S`) or user (`U`) request
sc	=	select code of device
a	=	HP-IB address of device
q	=	Error reporting message from device
ssssss	=	10 words of CS/80 status returned from the device after the error

The last 5 lines of error status information are returned through a memory-resident program, CSERR, and displayed only on the system console. If this program is not generated in, these messages will not appear. If this program is generated in and the messages do not appear, a channel error has occurred which indicates problems with cabling. Consult the *HP 12821A Interface Manual* for cabling restrictions.

This error is caused by any condition that prevents the transfer from successfully completing. This is not a fatal error. The B-Register contains the track in error or the low order block for block addressed requests. The A-Register contains EQT word 5 status, which indicates the error.

IO NOT READY Error

Error Format: `IONR L`*nn* `E`*xx* `S`*yy* *zzz*

where:
nn = LU number
xx = EQT number
yy = Subchannel number
zzz = Status byte from EQT word 5

This error is caused by:

a. An attempt to access a non-existent or offline device.

b. Device controller is offline to service due to:
   − pack dismount.
   − head realignment.
   − mounting or dismounting a CTD.
   − a push-button backup or restore.

c. Device did not respond in specified timeout period or within 5 seconds if no timeout was specified.

d. A hardware fault taking the controller offline.

Conditions a, b, and c will self-correct when the device recovers and/or completes the offline operation.

Index

R

S

T

U

W