SDC-RQD11-SC

SMD/MSCP Compatible Disk Controller for LSI-11 and MicroVAX II Manual

Σ Sigma Information Systems

SDC-RQD11-SC

SMD/MSCP Compatible Disk Controller for LSI-11 and MicroVAX II Manual

Sigma Information Systems - Anaheim, California

January 1987

MA400705 REV C

TABLE OF CONTENTS

FIGURES AND TABLES

CHAPTER ONE
SDC-RQD11-SC GENERAL DESCRIPTION

1.1 Overview

The SDC-RQD11-SC is a high performance quad height interface to SMD compatible disk drives. The SDC-RQD11-SC can support one or two SMD drives and features a one megabyte cache memory, 24 megabit per second throughput, command queueing, overlapped seeks, and implements DEC's Mass Storage Control Protocol (MSCP).

The SDC-RQD11-SC flexibly couples disks of any size and data rate to all standard DEC operating systems without software modification. Comprehensive on-board interactive formatting and diagnostic firmware provides engineering support across the range of LSI-11, MicroVAX and various non DEC implementations of the Qbus.

Two SMD Drives up to 3 Megabytes/Second Transfer Rate

The SMD interface is an industry standard for high capacity disk drives. There have been modifications to this standard by various manufacturers. The SDC-RQD11-SC supports Standard, Modified, Enhanced and Extended SMD interfaces with no software or hardware changes. Two drives of any type with transfer rates of up to 24 megabits per second (3 megabytes per second) are supported.

One Megabyte Disk Cache

All data is read from and written into the cache. Data transfers from the cache are approximately 2.5 ms compared to 30 - 38 ms for typical drive access times - up to an 93 percent reduction in access time. The cache memory acts as a track buffer and can provide data at rates of 3 megabytes per second.

No Sector Interleaving

The SDC-RQD11-SC stages all data from the disk drive through the cache memory. This ensures that all data can be transferred at full disk speed over the disk interface and at maximum speed over the Qbus without incurring data late errors. Disk and Qbus transfers are performed simultaneously to minimize access times.

REF:SMDC1.WPS and SMDC2.WPS

Read Look-ahead

The SDC-RQD11-SC allows the user to program the controller to perform 'look-ahead' reads in anticipation of data requests. Whole tracks or more can be read into the cache ensuring that data is ready for the host computer the instant its needed.

Virtual Units

The SDC-RQD11-SC also allows the user to partition each drive into virtual units which are addressed by the host as individual drives. Each virtual unit can be any size up to the size of the entire drive with up to 16 virtual units assigned to each controller. Each virtual unit can be further partitioned under the host operating system.

Block mode DMA and DMA Throttle

With Block Mode DMA, the SDC-RQD11-SC interleaves address references with bursts of data - almost doubling Qbus throughput. The SDC-RQD11-SC fully conforms with Qbus Block Mode DMA protocol. With non block mode memory, the SDC-RQD11-SC automatically reverts to burst mode DMA.

After every 16 word DMA transfer there is a 4 microsecond delay to service any pending interrupt or DMA requests from other devices. If a DMA request occurs a 'DMA throttle' will release the Qbus after 8 words to prevent data loss from other DMA devices.

Drive Shadowing

The SDC-RQD11-SC offers the user the option of drive shadowing. Data integrity is further improved by writing the same data to two drives simultaneously. In certain circumstances greater data throughput may be achieved because the controller can decide which drive has its heads positioned closest to the required data, and schedule a read on that drive.

Seek optimization and Overlap

The SDC-RQD11-SC can queue up to 32 commands. The optimum order of execution is dynamically computed according to the strategy selected by the user. With multiple disk drives seeks are initiated simultaneously, further improving performance.

Error Checking and Correction

The SDC-RQD11-SC uses a 48-bit ECC polynomial with an 11-bit correction span for error detection and correction. The SDC-RQD11-SC will try up to 10 times to correct an error before reporting the fault to the host system.

Dynamic Bad Block Replacement

The SDC-RQD11-SC dynamic bad block replacement and error correction always presents error free 'perfect media' to the host computer. During normal operation the controller dynamically replaces any blocks it detects as bad with an alternative block from a replacement block pool. Blocks with hard errors are replaced but the data in them flagged 'forced error'. This indicates to the host that though the data in these blocks is bad the blocks themselves are now good. All bad block replacement is completely invisible to the host computer.

Statistics Recording

The SDC-RQD11-SC records statistics such as the number of reads and writes, cache hits and misses, and other important information for each drive. The user can interrogate the drive for this information according to application specific performance requirements.

Write Protect

A connector is provided to which the user can connect one write protect switch per drive.

MSCP Emulation

The SDC-RQD11-SC communicates with the host through a simple register pair to memory resident 'command packets'. Disk geometry factors such as sectors, heads, cylinders and disk capacity are invisible to the host computer. The SDC-RQD11-SC accepts 32 bit binary block numbers and converts them to physical disk addresses, allowing any size disk to be fully accessed by any program without software modification.

Supported operating systems include RT-11 version 5; RSX-11M-Plus version 2; TSX-Plus version 5; RSTS/E version 8; MicroVMS version 4; or later versions. Various UNIX versions are also supported.

Qbus interface

Originally introduced in 1975 by Digital Equipment Corporation to support the LSI-11 CPU range, the Qbus architecture has evolved in speed and function to the point where it now outperforms most small computer bus systems. The SDC-RQD11-SC fully implements all current Qbus enhancements, including block mode transfers, 4-level interrupt structure, 22 bit addressing, and fully supports LSI11/2, LSI11/23, LSI11/73 and MicroVAX II Qbus CPU designs.

On-Board WOMBAT Utilities

WOMBAT is an interactive formatting and diagnostic utility contained within the SDC-RQD11-SC firmware. An on-board serial connection allows WOMBAT to be run using an ASCII terminal. This permits disk formatting and maintenance operations to be carried out with minimal additional hardware.

WOMBAT can also load a simple console communication program into the host computer's memory or it can be invoked on system power-up. No external software, media, or program loading device is required in maintenance of the SDC-RQD11-SC or its attached disk drives. WOMBAT is always available independently of the host CPU type or the operating system environment.

WOMBAT Formatter

WOMBAT initializes a fresh disk drive by writing sector addresses and zero data blocks through the entire recording surface. WOMBAT prompts the user at the terminal to supply parameters such as drive geometry (cylinders, heads and sectors) and various other options. This data is stored twice in special reserved areas of track zero and retrieved by a simple homeseek-read sequence at each power-up. No special PROMs or switch settings are required to fully characterize the connected disk drives.

WOMBAT Self Diagnostics

The SDC-RQD11-SC contains a comprehensive set of self diagnostic procedures which are executed automatically on power-up. Failure is indicated by a flashing red LED and a fatal error status which is deposited in the SA register.

WOMBAT Interactive Diagnostics

Terminal oriented engineering utilities contained within the WOMBAT firmware include a continuous read/write/seek exerciser, a disk surface pattern tester and a bad block replacement routine.

Sequential Spin-up

The SDC-RQD11-SC controller will spin-up SMD drives in a sequential manner, providing they have the "motor control" option enabled. This is done to minimize start-up current surge.

1.2 Controller Specifications

Bus Interface (Q-bus):	MicroVAX II, LSI-11/2, -23, -73
Q-bus Loads:	1 DC, 1 AC
Transfer Mode:	Block Mode DMA
Memory Address:	4 megabyte capacity (22-bit)
Software Emulation:	DEC MSCP
Command Buffer:	Up to 32 commands capacity
Disk Cache Size	1 megabyte (with parity)
	2.5 ms for cached reads.
Transfer Rate:	3.0 megabyte per second maximum non-interleaved
Base (CSR) Address:	Jumper selectable
	LSI-11 MicroVAX
	17772150 20001468
	17760334 200000DC
	17760354 200000EC
	17760374 200000FC
	17760414 2000010C
	17760434 2000011C
Vector Interrupt:	Programmable
Interrupt Level:	Jumper selectable. 4, 5, 6 or 7

On-board Bootstrap:	Jumper selectable. Disable or enable at 17773000 or 17771000
Power Requirements:	+5VDC @ 3.5A typical +12VDC @ 0.4A typical
Dimensions:	Standard quad board
LED Indicators:	
Red:	Fatal Error
Green:	Access in Progress
TTL Outputs:	Disk access in progress
TTL Inputs:	One write protect input per drive One online/offline input per drive
RS-232 Output:	Data transmitted to terminal 9600 baud
RS-232 Input:	Data received from terminal 9600 baud
Drive Support:	1 or 2
Interface:	CDC SMD (Standard, Modified, Enhanced, and Extended)
Connectors:	1 34-pin control connector 4 20-pin data connectors 1 10-pin Write Protect front panel connector
Cylinders:	4096 maximum
Heads:	16 (maximum)
Bytes/Sector:	512
Sectors/Track:	255 (maximum)
Cable Kits (Optional):	Single drive configuration (P/N 501017): one 26-pin flat ribbon control cable and one 60-pin flat ribbon data cable. Dual drive configuration (P/N 501018): two 26-pin flat ribbon control cables and one 60-pin daisy chain data cable.
Maintenance Adaptor:	RS-232 Maintenance Terminal Adaptor
(Optional)	kit (P/N 501019)

CHAPTER TWO
INSTALLATION

2.1 Unpacking and Inspection

The SDC-RQD11-SC is shipped in a special packing carton designed to keep the module from vibrating and to give it maximum protection during shipment. The packing carton should be retained in case the unit requires reshipment. The packing carton should contain the following:

P/N 400705 SDC-RQD11-SC quad-wide module

MA400705 Manual entitled "SCD-RQD11-SC SMD, MSCP Compatible Disk Controller for LSI-11 and MicroVAX Systems"

P/N 501017 **Optional** single drive configuration cable kit with one 26-pin data cables and one 60-pin control cable

P/N 501018 **Optional** dual drive configuration cable kit with two 26-pin data cables and one 60-pin control cable daisy chained to second drive (controller-to-drive-to-drive)

P/N 501020 **Optional** dual drive configuration cable kit with two 26-pin data cables and two 60-pin control cables (controller- to-drive and drive-to-drive)

P/N 501019 **Optional** maintenance adaptor

Unpack the SDC-RQD11-SC and visually inspect for physical damage. If any damage has occurred, contact the factory immediately.

2.2 Factory Configurations

The SDC-RQD11-SC is shipped with switch and jumper configurations as shown in Figure 2-1. Verify that these configurations are correct. If other configurations are required, refer to the appropriate paragraphs in this section.

2.3 Jumper Configurations

The SDC-RQD11-SC can be configured with a combination of configuration jumpers. Jumper positions for base (CSR) address, automatic bootstrap selection, interrupt priority, removable media, and CPU type are given in the following paragraphs.

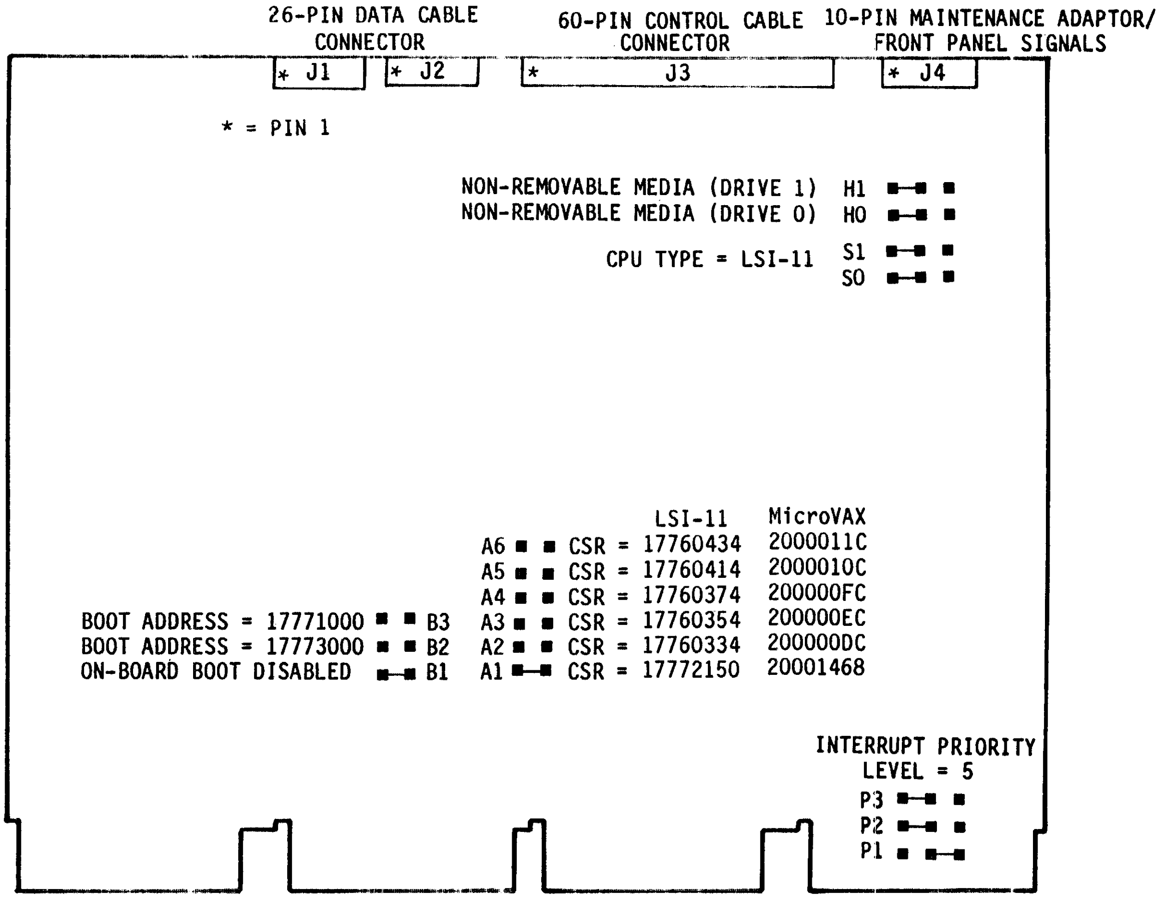
26-PIN DATA CABLE CONNECTOR
60-PIN CONTROL CABLE CONNECTOR
10-PIN MAINTENANCE ADAPTOR/ FRONT PANEL SIGNALS
* J1
* J2
* J3
* J4
* = PIN 1
NON-REMOVABLE MEDIA (DRIVE 1) H1
NON-REMOVABLE MEDIA (DRIVE 0) H0
CPU TYPE = LSI-11 S1
S0
LSI-11
MicroVAX
A6 CSR = 17760434 2000011C
A5 CSR = 17760414 2000010C
A4 CSR = 17760374 200000FC
A3 CSR = 17760354 200000EC
A2 CSR = 17760334 200000DC
A1 CSR = 17772150 20001468
BOOT ADDRESS = 17771000 B3
BOOT ADDRESS = 17773000 B2
ON-BOARD BOOT DISABLED B1
INTERRUPT PRIORITY LEVEL = 5
P3
P2
P1

Figure 2-1: Factory Configuration

2.3.1 Base (CSR) Address

Jumpers A1 through A6 determine the CSR address. Only one jumper should be installed in on of the jumper rows shown in Figure 2-1. Table 2-1 defines the CSR address configurations.

Table 2-1 Base (CSR) Address (Jumper A1-A6)

*JUMPER IN	ADDRESS LSI-11	ADDRESS MicroVAX II
1st (A1)	17772150	20001468
2nd (A2)	17760334	200000DC
3rd (A3)	17760354	200000EC
4th (A4)	17760374	200000FC
5th (A5)	17760414	2000010C
6th (A6)	17760434	2000011C

*Only one jumper can be installed at one time.

2.3.2 Automatic Bootstrap Selec

The automatic bootstrap is for LSI-11 processors only. If this option is used then CPU type must be set to zero. (See Figure 2-4.)

Table 2-2 Automatic Bootstrap (Jumpers B1-B3)

*JUMPER IN	FUNCTION
B1	On board bootstrap disabled
B2	On board bootstrap enabled at address 17773000
B3	On board bootstrap enabled at address 17771000

*Only one jumper can be installed at one time.

2.3.3 Interrupt Priority

The interrupt priority can be set for level 4, 5, 6 or 7 as shown in Figure 2-2.

Figure 2-2: Interrupt Priority (Jumpers P1-P3)

```
!-----------------------------------------------------------------!
!                                                                 !
!      ------------------INTERRUPT LEVEL------------------        !
!      4                5                 6                7      !
!                                                                 !
! P3 o---o   o    P3 o---o   o     P3 o---o   o    P3 o   o---o   !
! P2 o---o   o    P2 o---o   o     P2 o   o---o    P2 o   o---o   !
! P1 o---o   o    P1 o   o---o     P1 o---o   o    P1 o---o   o   !
!                                                                 !
!-----------------------------------------------------------------!
```

2.3.4 Removable Media

Each drive attached to the SDC-RQD11-SC can be configured for operation with either removable or fixed media as shown in Figure 2-3.

Figure 2-3: Removable Media (Jumpers H0,H1)

```
!------------------------------------------------------!
!                                                      !
!    MEDIA TYPE          DRIVE 0          DRIVE 1      !
!                                                      !
!    FIXED MEDIA      H0 o---o   o     H1 o---o   o    !
!  REMOVABLE MEDIA    H0 o   o---o     H1 o   o---o    !
!                                                      !
!------------------------------------------------------!
```

2.3.5 CPU Type

The SDC-RQD11-SC can be configured for opertion with LSI-11 or MicroVAX II CPUs as shown in Figure 2-4. When using the SDC-RQD11-SC in the MicroVAX II the CPU jumper can be configured as "LSI-11" or "undefined," but not as "no CPU."

Figure 2-4: CPU Type (Jumpers S0, S1)

```
!-------------------------------------------------------------!
!                                                             !
!   TYPE 0          TYPE 1          TYPE 2          TYPE 3    !
!   LSI-11         UNDEFINED       UNDEFINED        NO CPU    !
!                                                             !
! S1 o---o   o    S1 o---o   o    S1 o   o---o    S1 o   o---o !
! S0 o---o   o    S0 o   o---o    S0 o---o   o    S0 o   o---o !
!                                                             !
!-------------------------------------------------------------!
```

2.4 Front Panel/RS-232 Maintenance Connector

Connector J4 can be used either to supply front panel signals to a user The defined device, or to connect an ASCII terminal for maintenance purposes.

2.4.1 Maintenance Terminal Adaptor

The maintenance terminal adaptor allows the controller to be connected to an ASCII terminal. It consists of a 10-pin flat cable with a DB25S connector on one end and a 10-pin insulation displacement type flat cable socket on the other. The communication format is :

ASCII RS232 9600 Baud, 8 Data Bits, 1 Stop Bit, no parity.

Note that if normal disk access is attempted with this cable connected to a terminal, garbage will appear on the terminal due to the shared RS232 Output/Access Light Function. This is normal.

Table 2-3: RS-232 Maintenance Adaptor Pinouts

-PIN NUMBERS-		
J4	DB25S	Function
7	7	RS-232 Enable
8	2	RS-232 Input
3	3	*RS-232 Output
4	7	Ground

*From WOMBAT - +5V to -5V compatible.

2.4.2 Front Panel Connections

If required, a front panel can be connected to J4, the front panel/maintenance connector. The functions supported are listed in Table 2-4.

Table 2-4: Front Panel Connector

PIN	FRONT PANEL FUNCTION
7	Drive 1 Write Protect Input. Connecting this input to Ground will write-protect the drive. Has an on-board 180 ohm pull-up.
8	Drive 0 Write Protect Input. Connecting this input to Ground will write-protect the drive. Has an on-board 180 ohm pull-up.
1	Drive 1 Online/Offline Switch Input. Connecting this input to Ground will prevent access to the drive. Has an on-board 22k ohm pull-up.
10	Drive 0 Online/Offline Switch Input. Connecting this input to Ground will prevent access to the drive. Has an on-board 180 ohm pull-up.
3	Access Light. Indicating access to Drive 0, Drive 1 or Cache. This output can be used to drive an access light. The levels are: ACCESS: -5V through 1.5k ohm NO ACCESS: +4.5 @ 20mA max.
5	TTL signal indicating access to Drive 0. Low true.
6	TTL signal indicating access to Drive 1. Low true.
2.4.9	Ground

2.5 Drive Cabling

The SDC-RQD11-SC can simultaneously control two SMD drives which can be of different size, speed and data rate. Drive cabling is illustrated in Figure 2-5. Data cables are connected in radial fashion to edge connectors on the SDC-RQD11-SC. The control cable is connected to drive 0 and daisy chained between each drive (if a second drive is installed) with the last drive being terminated.

Pre-configured data and control cable kits (Section 2-1) for one or two drives can be supplied to minimize difficulties when installing the SDC-RQD11-SC or when adding additional drives.

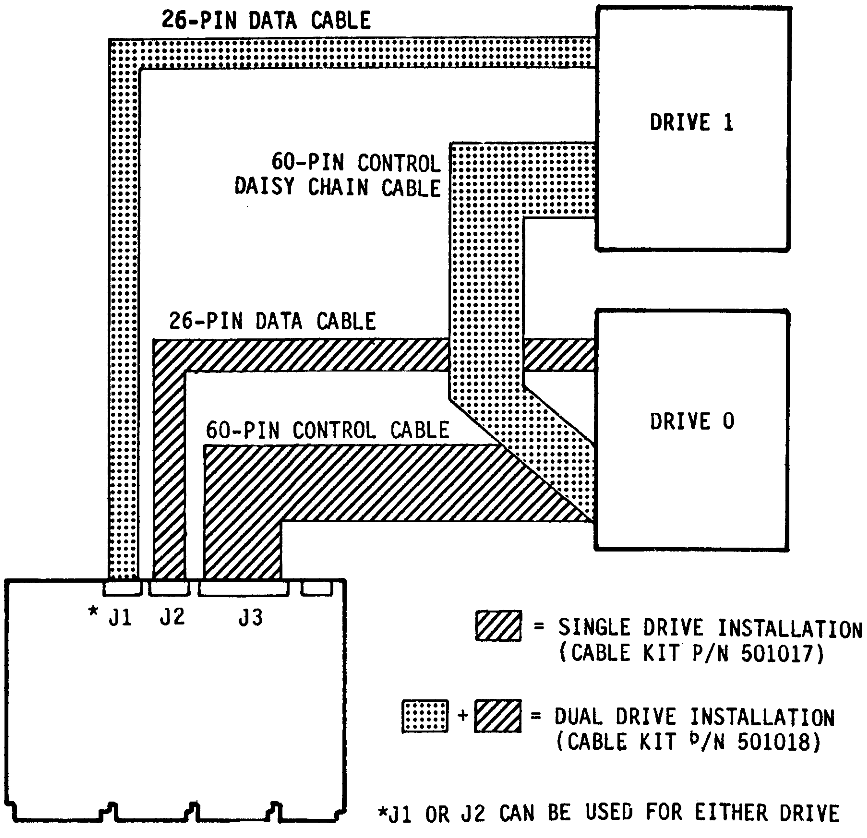

Figure 2-5: Drive Cabling

2.6 On-Board Bootstrap (LSI-11 only)

If the SDC-RQD11-SC bootstrap is enabled the following occurs:

1. Initialization code is loaded into host memory at location 2000.
2. A "jump to 2000" instruction (JMP @ #2000) is loaded into location zero.
3. The controller changes the contents of location 773002 to "CLR PC" and the following message is typed:.

   BOOT V2.x (x = firmware revision level)

4. The controller allows approximately 2 seconds for the operator to strike any key on the keyboard. If no key is struck the boot types:

   Booting from DUO:

   The boot then sets up registers and waits for that device's boot block to be read into memory starting at location zero.

5. If any key is struck by the operator within two seconds the boot prompts with:

   ```
   >
   ```

   The operator may then key in a device DU, DL, DY, MS or W. Note that device W will invoke WOMBAT. By further specifying A, B, or C after W, WOMBAT on controller A, B or C will be invoked. For example, WB will invoke WOMBAT on controller B.

   The following is the syntax of the SDC-RQD11-SC bootstrap procedure:

   ```
   > [Device] [Controller Number] [Unit Number]:
   ```

where:

[Device]	[Controller Number]	[Unit Number]
DU	[A,B,C]	[0-7]:
DL		[0-7]:
DY		[0-1]:
W	[A,B,C]	
MS	[A,B,C,D]	

[ ] = options. If no options are specified, defaults = A, 0.

Examples:

DU1:	for the second drive on the first DU controller
DUB2:	for a drive on the second DU controller
DY1:	for the second RX02 floppy disk drive.
W:	for WOMBAT
WB:	for WOMBAT on the second controller (B)

6. If any error occurs, a message from the following set is printed and the boot re-prompts for step five.

"? - Device must be DU, DY,DL or MS"	
"Unit must be 0 - 7"	For DU or DL
"Unit must be 0 or 1"	For DY
"Controller must be A, B or C"	For DL
"Controller must be A, B, C or D"	For MS
"Boot failure" - Device unavailable	(or not ready if DL)

7. The controller reads the boot block, block zero, from the specified device into the host memory and then waits to be initialized.

8. The host computer commences execution of the instructions in the boot block.

Table 2-5 CSR Addresses for Bootstrap

DEVICE	ADDRESS
Console	17777560
DU:	17772150
DUA:	17772150
DUB:	17760334
DUC:	17760354
DUD:	17760374
DUE:	17760414
DUF:	17760434
MS:	17772520
MSA:	17772520
MSB:	17772524
MSC:	17772530
MSD:	17772534
DL:	17774400
DY:	17777170

STARTING WOMBAT

CHAPTER THREE
SDC-RQD11-SC WOMBAT UTILITIES

3.1 INITIALIZING INVOKING Starting Up WOMBAT

WOMBAT provides a controller resident means of formatting, testing and maintaining the drive and controller subsystem. All WOMBAT functions are menu driven and are designed to simplify the process of structuring, formatting and testing drives.

WOMBAT can be invoked using any of the following methods :-

(a) By PRESSING selecting the '**W**' option during the system bootstrap operation ~~if the~~ SDC-RQD11-SC boot is enabled. This allows disk testing and diagnostics to be performed from the user console. The console link is formed by a communication program which WOMBAT downloads into main memory.

(b) By entering 250_8 (LSI-11) or 254_8 (MicroVAX II) INTO THE CSR from the console terminal using ODT. This allows disk testing and diagnostics to be performed from the user console. The console link is formed by a communication program which WOMBAT downloads into main memory.

(c) By connecting a 9600 baud auxiliary terminal to the SDC-RQD11-SC Maintenance Connector and entering 260_8 INTO THE CSR from the console terminal using ODT. This allows disk testing and diagnostics to be performed from an auxiliary terminal which communicates directly with on-board WOMBAT utilities.

(d) By connecting a 9600 baud auxiliary terminal to the SDC-RQD11-SC Maintenace Connector and configuring the switches on the PCB to automatically run WOMBAT. (CPU independent.)

Runs WOMBAT CAN independently of, or without, a CPU for controller testing or engineering purposes if necessary. First disable the controller bootstrap (Table 2-2) and set the CPU type to "No CPU" (Table 2-4). Connect a 9600 baud terminal to the Front Panel connector and position the controller in the backplane. WOMBAT will be invoked automatically on power-up or by pressing reset. If no CPU is present then the backplane must be correctly terminated and a bus intialization signal (BINIT) must be generated.

To resume normal operation the configuration switches must be reset as required. Note that setting the CPU type to 3 will make the controller completely unavailable to the host CPU.

STARTING WOMBAT

(e) By depositing 272_8 into the (CSR) IP register a user-written communication routine can communicate directly with WOMBAT. This call does not outload any communication routine into the host memory.

Table 3-1 summarizes the procedures that invoke WOMBAT.

TABLE 3-1 WOMBAT INITIALIZATION PROCEDURES

ACTION		CONDITIONS	CONTROL
*OCTAL	*HEX		
000250	00A8	ODT	LSI-11 Console
000254	00AC	ODT	MicroVAX Console
000260	00B0	ODT	Aux Terminal on Maintenance Connector
000272	00BA	ON-LINE	User Communication Program
W		Via Boot	System Console
Boot Disabled CPU Type 0		CPU Independent	Aux Terminal on Maintenance Connector

*These codes are deposited into the (CSR) IP register (Section 4.6).

The procedures for invoking WOMBAT on LSI-11, and the MicroVAX II are given below. WOMBAT can be stopped by simply re-booting the system.

In each case input the appropriate CSR address for the controller to be accessed where indicated by 'CSR'. Table 3-2 lists the possible CSR addresses for the SDC-RQD11-SC.

TABLE 3-2 SDC-RQD11-SC CSR ADDRESSES

LSI-11 CSR	MicroVAX II CSR
17772150	20001468
17760334	200000DC
17760354	200000EC
17760374	200000FC
17760414	2000010C
17760434	2000011C

LSI-11 CPU — ~~TO START~~ STARTING WOMBAT

The following details the procedure for invoking WOMBAT on an LSI-11 CPU system using console ODT.

Halt the processor.

'CSR' /	000000	**250**	(ask WOMBAT to load the communications program)
R7 /	XXXXXX	**2000**	(set up the program start address)
RS /	000000	**340**	(set PSW to block interrupts)
P			(now start the program without a bus reset.)

MicroVAX II — ~~TO START~~ STARTING WOMBAT

The following details the procedure for invoking WOMBAT in a MicroVax II using ODT.

Halt the CPU at the end of its start-up diagnostics by turning on the "halt enable" switch at the back of the CPU.

When it halts :-

D/P/W	**20001F40**	**20**	(enable QBUS access to memory)
D/L	**20088008**	**80000002**	(set-up the appropriate QBUS map entry)
D/W	**'CSR'**	**AC**	(ask WOMBAT to load the communications program into memory)
S	**400**		(start the program)

WOMBAT MENU OPTIONS

3.2 WOMBAT Menu options

When WOMBAT is invoked it will display an announcement and then print a list of all drives and units and prompts for the drive number on which to perform operations.

```
      SDC-RQD11-SC   WOMBAT Version: 2.x    where x is the firm-
                                              ware revision level.
```

```
UNIT  DRIVE  OFFSET  SIZE    WRITE   STATUS
 0     0       34    20000   LATE    AVAIL

DRIVE CYLS   HEADS   SECTORS BLOCKS  MTYPE  OPT     FAIR    STATUS
 0    1224    5      34      291312  FIXED  NONE    24      SPUN UP
Drive number :
```

Enter the drive number (zero on a single-drive system). WOMBAT will then display the Master Menu options.

```
      ** Master Menu **
      1   Structure Disk
      2   Test Disk
      3   Manage bad blocks
      4   Display error
      5   Shadow options
```

Select an option by typing the option number followed by RETURN. Options 1 through 3 and 5 will provide sub menus while option 4 displays the last controller detected fatal error. To return to the master menu from a sub menu type **RETURN**.

To exit from the master menu to the announcement (to select a different drive) type **RETURN**. WOMBAT will not allow you to do this before verifying whether the disk structure data has been written to disk. 'NO' is the default value.

MASTER MENU OPTIONS

3.3 Master Menu Options

Option 1 - Structure Disk

Selecting this option causes a sub menu to be displayed as follows:

```
** Disk Structure Menu **

1   Create Disk Structure
2   Format Disk
3   Write Disk Structure
4   Update Header Blocks
5   Display Disk Structure
6   Change Unit number
7   Display Statistics
```

Option 2 - Test Disk

Selecting this option causes a sub menu to be displayed as follows:

```
** Disk Test Menu **

( ! means all data on Disk destroyed )

1   Read All Disk (preserves all data)
2   ! Write Disk !
3   ! Pattern Test !
4   ! Random Writes !
5   ECC Validation
6   Read Physical Block
7   Display Error Statistics
8   Zero Error Statistics
9   Test Cache, RAM & ROM
```

Option 3 - Manage Bad Blocks

Selecting this option causes a sub menu to be displayed as follows:

```
** Bad Block Management Menu **

1   Manually Replace Bad Block
2   Automatically Replace Bad Blocks
3   Display Replaced Bad Blocks
4   Enter defect map
```

DISPLAY ERROR

Option 4 - Display Error

Selecting this option causes WOMBAT to display a message which explains the most recent controller detected fatal error. The occurence of an error is indicated by the controller hanging with the red LED flashing. If no error has occurred this option produces a meaningless error message.

Option 5 - Shadow Options

Selecting this option causes a sub menu to be displayed as follows:

Shadow options menu

1 Shadow copy option
2 Set shadow units
3 Reset shadow units
4 Copy unit to unit
5 Compare unit to unit

Setting Up a New Disk

The procedure for structuring a new disk is as follows:

1. Create disk structure.

2. Format the disk.

3. Write the disk structure.

4. Replace bad blocks using manufacturer's media defect map.

5. Pattern test the disk.

6. Replace bad blocks detected by pattern test.

Once these 6 steps have been undertaken the host operating system may use the disk.

DISK STRUCTURE MENU OPTIONS

3.4 Disk Structure Menu Options

**** Disk Structure Menu ****

1 Create Disk Structure
2 Format Disk
3 Write Disk Structure
4 Update Header Blocks
5 Display Disk Structure
6 Change Unit number
7 Display Statistics

Option 1 - Create Disk Structure

The Create Disk Structure option must be performed when a new disk is connected to the SDC-RQD11-SC. This allows the various disk geometry, controller wide tuning parameters and virtual unit structure to be specified. The virtual unit structure allows a single large drive appear to the host operating system as multiple drives.

This option enters an interactive question and answer dialogue which specifies the disk structure. WOMBAT displays either the current or the default value for a parameter and gives you the option of accepting or changing it to a new value. To accept the displayed value, hit RETURN. To change it, type in the new value followed by RETURN. If WOMBAT detects improper values it will issue a warning.

The create disk structure dialogue is divided into three parts.

(a) The drive structure specification, which describes the physical geometry of the drive.

(b) The unit structure specification, which is executed once for each virtual unit defined where the size of each unit and unit specific parameters is described.

(c) The controller wide tuning parameters, where read lookahead and command queue size are specified.

Drive structure specification

Cylinders: Enter the number of cylinders the drive has.

Heads: Enter the number of heads that the drive has.

Full sectors/track: Compute and enter the required value described in section 3.12 - Computing Sectors Per Track.

Short sectors/track: A short sector on each track will result if the number of full sectors does not completely fill a track. WOMBAT will render the sector invisible to the controller by writing an illegal header on the sector. If the sector is too

GAP + PLOSYNC +HEADER

DISK STRUCTURE MENU OPTIONS

small WOMBAT will be unable to perform this operation and errors will consistently occur during disk testing. In this case it will be necessary to reduce by one the number of sectors per track.

Track spiralling: Track spiralling improves disk performance on data transfers over more than one track. Sector zero on each track is offset by a nominated factor to allow head select and positioning before sector zero on the next track is reached. The recommended factor is four.

Optimization: The seek optimization strategy can be:

0 (None) - No optimization done. First request found executed. This may not be the next sequential request.

1 (Nearest) - Selects request that is closest to the to the current cylinder.

2 (Elevator) - Processes requests as it moves in one direction along the disk until it reaches the last request in that direction. This means that "Elevator" favors the center of the disk, as it passes it twice as often as the periphery.

3 (Forward) - This processes requests from the lowest cylinder number to the highest in one direction only.

Note that optimization is only effective if the host operating system supports multiple accesses. RT-11 and TSX plus do not support optimization without a special device handler.

Interface Type: Enter the correct value for the SMD interface type being used. It is possible to have a different SMD interface for each drive connected to the controller. The values are:

0 - Standard
1 - Modified
2 - Enhanced
3 - Extended

Fairness count: The fairness count determines the number of times an I/O request will be passed over by the controller's seek optimization setting before it is executed. A reasonable count for normal use would be around 25. Every time a request is

DISK STRUCTURE MENU OPTIONS

passed over its fairness count is decremented. When that count reaches zero that request will be selected, no matter what optimization strategy is in effect.This count has no effect if no optimization is selected.

This completes the drive structure specification. WOMBAT next prompts for a unit number. Type the unit number of the next unit to be defined, then hit RETURN. When the unit is completely defined, WOMBAT will again prompt for a unit number. If there are no more units to define hit RETURN. WOMBAT will then proceed to the controller wide parameter definition. Note that if no units are defined, the operating system will not see anything attached to the controller.

Unit Structure Specification

Unit Number: WOMBAT next prompts for a unit number, which allows the user to define the unit size for partitioning into logical units. If the default unit size is used, one logical unit is defined. The options for "Unit Number" follow:

Unit size: If an existing unit number is specified WOMBAT will display the size in blocks. If a new unit number is specified WOMBAT will display the size in blocks of the first unallocated disk area it finds beginning at the start of the disk. On a new disk this will be the entire user area. This can be changed to a smaller value if necessary. To delete an existing unit, specify zero for this field.

Media type: This field is displayed by some operating systems when you enquire about the type of drive. As a part of unit status when a "Get Unit Status" command is issued the MSCP protocol returns a 5 character media type (default is DURA81). The first two characters must be 'DU'- for example - DURD54. To change this enter 1 to 5 alphabetic characters and 2 digits, e.g. RD52, to emulate DEC's 31 megabyte Winchester. For example, RSX-11M-PLUS responds to a "DEV DU:" command with : "DU0: Public Mounted Loaded Label = RSX11MPBL15 Type = RA81"

Serial number: The MSCP protocol returns a 32-bit volume serial number as a part of its response when an "online" command is issued. WOMBAT defaults this field to zero. To change this enter the desired

DISK STRUCTURE MENU OPTIONS

serial number. This field is used, for example, by RSX-11M-PLUS, when a disk is initialized with the "INI DU:" command. It sets up the volume serial number.

Host write Confirm: This specifies if the controller is to notify the host that a write request has been completed. Enter 1 if the host is to be notified when the data is in the cache. The data will be written to the disk later. Enter 2 if the host is to be notified when the data has actually been written to the disk.

This completes the unit structure definition. Now WOMBAT enters the final part of the dialogue which specifies the controller wide tuning parameters.

Controller Wide Tuning Parameters

Read lookahead: This is a feature of the cache which allows the controller to read a specified number of sectors in addition to those requested by the host. Enter the minimum number of blocks you wish the controller to read for any request. For example if a value of 4 is specified, when the host asks for a single block to be read, the next 3 blocks will automatically be read into the cache. If the host subsequently asks for one of these 3 blocks then the request can be honored immediately from cache. If the host requests a transfer equal to or larger than the read lookahead size then this parameter will have no effect.

Cache Enable: This parameter enables or disables the cache. A value of 0 disables the cache and a value of 1 enables the cache. The cache cannot be selectively enabled or disabled for a particular drive or unit number.

Command queue: This parameter allows you to specify the number of commands the controller can stack. The controller will then attempt to optimize the order in which they are executed. Large command queue stacks incur considerable overhead and will degrade controller performance. Note also that some operating systems (RSX-11M-Plus 2.1B is a good example) have a maximum limit for the size of the stack. The default size [8.] is a good compromise and is acceptable to most operating systems.

DISK STRUCTURE MENU OPTIONS

Front Panel Type: This option allows the correct selection of front panel type. This option is normally set to "1." See Section 2.4.2 - Front Panel Connector.

0 - None
1 - Passive

This completes the disk structure definition. WOMBAT now checks the tables for consistency and returns to the disk structure menu.

Option 2 - Format Disk

WOMBAT asks you to confirm this drastic action as it will destroy ALL data that resides on the disk. WOMBAT will then initiate a two pass formatting operation. During the first pass WOMBAT creates all the sectors on the disk, write a sector headers which contain the sector number, the head number, and the cylinder number, as well as preambles and sync bytes, followed by a 2 byte data field. During the second pass WOMBAT writes a test pattern to each sector, preparing the disk for read testing. WOMBAT then writes the disk structure onto the reserved areas.

Option 3 - Write Disk Structure

WOMBAT will ask you to confirm this drastic action as any existing disk structure will be destroyed. WOMBAT will then write the new structure onto special reserved areas of track zero. The data is recorded twice for improved recoverability. A total of 6 blocks is written on track zero. After the structure has been written, the drive's replacement block table is zeroed. If there were any replaced blocks recorded there, they will be lost. However they will still be marked as replaced and will generate hard errors during a read operation.

Option 4 - Update Header Blocks

This is similar to Write disk structure except that the replacement block table is not written, thus preserving any blocks which may have been replaced. This option is used after changes to the disk structure such as changing unit numbers or redefining the virtual units. Unless the header blocks have been updated, the changes are not recorded on disk.

DISK STRUCTURE MENU OPTIONS

Option 5 - Display Disk Structure

WOMBAT displays the structure of the currently selected drive in a form similar to the create disk structure dialogue. This is useful for checking that the newly created structure is correct.

Option 6 - Change Unit Number

It is sometimes necessary to change a unit number in order to resolve a duplicate unit number or to satisfy operating system requirements. This method is a safe and simple way of doing so. WOMBAT prompts for a unit number on the current drive, and then for the new number for that unit. **NOTE:** For the change to take effect, the header blocks must be updated using option 4 above.

Display Statistics

Statistics about disk and cache usage are maintained, and recorded on the disk periodically. They are displayed as :

```
                        Controller statistics report

# of commands             xxxx
# of reads                xxxx  # of cache hits xxxx (xxx%)

# of writes               xxxx

                        Drive statistics report

Drive #                 Soft errors   Re-vectors   Blocks replaced
xxx                         xxxx         xxxx           xxxx

Drive #                 Seek distance # of seeks   Seek errors
xxx                         xxxx         xxxx           xxxx
```

Commands is the number of MSCP commands issued.
Re-vectors is the number of accesses to replaced blocks.
Seek distance is the total seek distance, in cylinders.
Blocks replaced is the number of blocks dynamically replaced by the controller during normal operation, rather than through WOMBAT.

Reset Counters is then asked. "Y" will reset them to zero.

DISK TEST MENU OPTIONS

3.5 Disk Test Menu Options

A disk can be tested after it has been formatted and before the structure is written to it. Testing does not overwrite the HDR or RCT blocks. The disk structure must be written to the disk before bad blocks can be replaced.

All tests continue indefinitely until aborted by one of the following methods:

1. PRESS <BREAK> AT THE AUXILARY PORT TERMINAL
   If an RS232 serial port terminal is attached to the controller~~, press BREAK.~~
2. PRESS <CTRL/C> AT THE COMPUTER CONSOLE TERMINAL
   If WOMBAT is running from the Console terminal~~, type CTRL/C.~~

When a test is aborted the Test Disk Menu options are returned. If tests are run from an RS232 terminal attached to the controller, beware of system activity on the host computer as Qbus initializations will cause the disk controller firmware to re-initialize and so leave WOMBAT.

All tests give 10 retries on an error, reporting every error by displaying the block number and an error code.

```
** Disk Test Menu **

( ! means all data on Disk destroyed )

1   Read All Disk (preserves all data)
2   ! Write Disk !
3   ! Pattern Test !
4   ! Random Writes !
5   ECC Validation
6   Read Physical Block
7   Display Error Statistics
8   Zero Error Statistics
9   Test Cache, RAM & ROM
```

Option 1 - Read All Disk

This test reports any read errors. Successful operation will be reported in the following format:

```
Pass: 1. Errors: 0.
Pass: 2. Errors: 0.
```

This function does not destroy any information.

DISK TEST MENU OPTIONS

Option 2 - Write Disk

This test reports any write errors while writing a test pattern to the whole disk. ALL INFORMATION on the disk, excepting HDR and RCT blocks, is DESTROYED. Errors are displayed in the standard format:

```
Block: 32040    (Error message)
Pass:1. Errors: 1.
Pass:2. Errors: 1.
```

The displayed error count is cumulative until the test is terminated.

Option 3 - Pattern Test

A test pattern is written to each block, including the replacement blocks. WOMBAT does one write and 10 read passes. This test reports any errors in the standard format as shown above.

Option 4 - Random Writes

This test writes 5000 blocks at random locations in the user area of the disk. It then reads the entire disk to determine if any of the writes caused an error. This test is designed to test the head positioning and selecting logic of the drive.

Option 5 - ECC Validation

The ECC test uses a special reserved block on track zero for testing. It first proves that it can successfully correct an 11 bit error and then proves that it cannot correct a 12 bit error. This test checks the ECC logic within the SDC-RQD11-SC.

Option 6 - Read Physical Block

WOMBAT prompts for a block number anywhere on the disk. It then converts that block number into a physical address consisting of cylinder, head, and sector, and displays these values in hex and ASCII. Then it reads that sector and displays a message indicating the success or failure of the read. The same physical block can be re-read by typing \ instead of a block number. The block's replacement block can be read by typing **@**.

DISK TEST MENU OPTIONS

Option 7 - Display Error Statistics

Displays the error statistics gathered by any of the above disk testing options in the following format:

```
** Error Statistics **
Block   Number (of errors)
32040     1.
Blocks in error: 1.
```

Option 8 - Zero Error Statistics

Zeroes the error statistics table & redisplays Test Menu options.

Option 9 - Test Cache, RAM & ROM Option

This test continuously writes test patterns throughout the entire cache and reads them back testing for veracity. A separate part of the test automatically checks that the parity logic is functioning correctly by forcing incorrect parity and checking that an error occurred. The cache pattern tests use special microcode instructions which iteratively read and write large blocks of cache memory at high speed. The Static RAM is also tested and the code PROM is Checksummed.

3.6 Bad Block Management Menu Options

```
** Bad Block Management Menu **

1  Manually Replace Bad Block
2  Automatically Replace Bad Blocks
3  Display Replaced Bad Blocks
4  Enter defect map
5  Get defect map from drive
```

Option 1 - Manually Replace Bad Blocks

WOMBAT prompts for a block number within the user area of the disk. Then it marks the specified block as bad and allocates a replacement block for it.

Option 2 - Automatically Replace Bad Blocks

WOMBAT searches the error statistics table, which is compiled by the read, write, and pattern tests, for blocks whose error count exceeds three. Any such blocks are marked as bad on the disk and replacement blocks are allocated for them.

DISK TEST MENU OPTIONS

Option 3 - Display Replaced Bad Blocks

WOMBAT reads the Replacement Control Table and displays the logical block numbers of any blocks recorded there.

Option 4 - Enter Defect Map

This option prompts for the drive manufacturer's defect map information as follows :

Enter defect map for drive n

(Enter all values as decimal!!!)

CTRL-C to exit at any time.
RETURN to back up 1 prompt.

Bytes per sector: The number of bytes per sector as set up in the drive switches (hard sector drives), or calculated by following the procedure in section 3.12 Computing Sectors per Track.

Cylinder: Cylinder number of defect.

Head: Head number of defect.

Bytes past index: Location past index of defect.

Bit length of defect: Length of defect in BITS.

This data is then used to compute the address of a block on the disk. If it does not match a block on the specified disk track the error message "!! Beyond last sector" is produced. This may mean that the defect is located beyond the last data sector on the track, or that the entered data was wrong. WOMBAT then calculates the block number, displays it and replaces it. The "Cylinder:" prompt is then repeated.

3.7 Shadow Options Menu Option

This feature shadows, i.e. keeps two copies, of a 'logical unit'. When the disk controller and drives are first powered up, any unit (and a physical disk may be broken up into a number of 'logical units') that is shadowed has its contents copied to its shadow unit. Thereafter any update of that unit will cause the controller to update the shadow unit as well. When reading from a shadowed unit, the drive with its heads nearest the required data is used. This helps to keep the drive shadowing overhead down, although in normal circumstances, with writes consisting about 10% of reads, there is a performance penalty.

SHADOW OPTIONS MENU

Shadowing consists of two operations: the initial copy of the entire primary unit to the shadow unit; and then the updating of both primary and shadow units on every disk write. Updating of both copies takes place as a matter of course, while the initial copying can happen at various times depending on the 'Shadow copy option' selected.

The main reason for drive shadowing is RELIABILITY. When reading, any error detected will cause the controller to use the data on the other drive. This means that, besides the controller having extensive error recovery facilities, there is a redundant backup of all data on the shadowed unit without any user programming or operating system overhead.

Drive shadowing is completely controlled from the 'Shadow options' menu.

Shadow options menu

1 Shadow copy option
2 Set shadow units
3 Reset shadow units
4 Copy unit to unit
5 Compare unit to unit

Option 1 - Shadow Copy Options

The first step is to select how the unit (the 'Primary unit') is to be copied to its 'Shadow unit'. There are 4 options :

0 - WOMBAT Only This is the default. With this selected the primary is only copied to its shadow when you invoke it in the Copy Unit to Unit option.

1 - On power up This causes the copying to be done whenever the drive(s) are first powered up, or WOMBAT has been invoked.

2 - Not ready/ready The copying is to be done whenever the controller detects the shadow unit going from a not ready to ready condition - e.g. a removable drive being bought on-line.

3 - Power up or Not ready/ready A combination of 1 and 2 above.

The use of copy type 0 only is recommended at this time. It is considered that the other options have too high a probability of copying bad data from the primary unit over good backup data on the shadow unit.

SHADOW OPTIONS MENU

Option 2 - Set Shadow Units

Shadow units are logically connected to primary units by this option which asks :

Primary unit: The unit to be shadowed.

Shadow unit: The unit number of the shadow.

The shadow unit will not be available to any operating system as it appears as 'unit undefined'. It must be EXACTLY the same size as the primary unit.

Now copy the primary to its shadow by using the **Copy unit to unit** option.

You may shadow a unit on the same physical drive, though you lose both reliability, by having the data on only one drive, and performance, by having the disk heads move to write all data twice.

If you have selected a shadow copy type of 1 or 3, whenever the controller is powered up and its first 'MSCP Initialize' sequence executed for the primary unit (the unit you wanted shadowed), the contents of this unit are copied to its shadow, or if WOMBAT has been invoked. The time this takes obviously depends upon the number of blocks on the unit, but we have found that a third of a megabyte per second is a good guide. While this is happening user I/O may take place, but will be made slower by the 'shadow copy' taking place.

WARNING:- a potentially serious problem exists with the automatic copy of data from the primary to the shadow at power up (shadow copy types 1 or 3). If the primary has been corrupted, or the data is invalid, IT WILL DESTROY YOUR BACKUP. If you have trouble with your primary unit use WOMBAT to change it from a shadow primary. You may even want to change its unit number via the 'Change Unit number' option, and then bring in the shadow unit as its replacement. Please realize that the controller MUST assume the validity of the primary unit at the first initialization after power up if you have selected one of the power up copy types.

SHADOW OPTIONS MENU

Option 3 - Reset shadow units

This option allows disabling of an assigned shadow unit :

Primary unit: The primary unit.

Shadow unit: The unit number of the shadow to be disconnected from the primary.

This shadow unit will now be available to the operating system.

Option 4 - Copy Unit To Unit

To copy a primary unit to its shadow :

Primary unit The number of the unit you want copied.

Shadow unit The unit you want the data copied to.

The copy will take place. Data is transferred at about twenty megabytes a minute

Option 5 - Compare Unit to Unit

This option allows the comparison of the data on a primary and a shadow unit.

Primary unit One of the units.

Shadow unit The other.

If any data errors, or data compare mismatch, is found, the block number and type of error will be reported. The comparison proceeds at about two megabytes a minute.

3.8 WOMBAT Disk Structure

WOMBAT records the logical structure of the drive on track 0. All of track 0 is reserved for this and other testing purposes. The user area begins at the next block after track 0. This is the same as the number of sectors per track. The user area extends to the block before the beginning of the Replacement Control Table (RCT). The size of the RCT is fixed and is always 2 tracks. The 2 tracks are accessed by different heads and contain identical data. All the blocks from the end of the second RCT track to the end of the media are reserved for replacement blocks.

WOMBAT DISK STRUCTURE

The disk area reserved for RCT and replacement blocks is always an integral number of cylinders. The number of cylinders required is always computed from the total number of cylinders on the drive. This reserves a constant proportion of the media for replacement blocks. The amount reserved is approximately 0.1% of the total formatted capacity. Option 4 of the master menu will report the formatted capacity of each drive. You can determine the user area easily. It will be reported as the size of the first unit configured on that drive. If there is more than 1 unit, then the sum of their sizes is the total user area, assuming there are no unallocated areas on the disk.

All block numbers in WOMBAT are physical block numbers beginning at the first sector of the first head of the first cylinder of the drive, which is defined as block zero. The last block on the drive is block n-1 where n is the total number of blocks on the drive. Therefore, the first block of the user area is not block zero. Its block number is the same as the number of sectors per track, as track 0 on the drive is reserved. Option 4 of the master menu will display both the size and the offset (starting block number) of each unit defined. Using these figures you can determine the exact position and extent of any unit.

3.9 Drives with Removable Media

Some manufacturers offer drives with removable media. The SDC-RQD11-SC detects when such a drive is connected and makes certain changes to the way it operates. The important changes are that you can only have one virtual unit on a drive whose media can be removed. That unit must have the same unit number as the physical drive number. The reason for this is that it is necessary to determine the unit number even though the media may be removed. This means that the unit number cannot be recorded on the media for these drives. WOMBAT automatically detects these drives and prevents you from creating incorrect units on them.

ERROR RECOVERY PROCEDURES

3.10 Error Recovery Procedures

During normal operation, the SDC-RQD11-SC checks every disk transfer for errors. When an i/o error is detected, the SDC-RQD11-SC enters a special error recovery procedure to attempt to provide the host with 'perfect' media.

The first method used to recover the data is simply to try the operation again. If this succeeds, the host is guaranteed to receive good data. This is repeated until a threshold is reached, at which time the second error recovery procedure is initiated. The second procedure recalibrates the drive and reseeks to the block in error in an attempt to correct any positioner errors that may have prevented data recovery. After the reseek, the operation is again retried until the retry threshold is reached. If any of these retries succeeds then the host is guaranteed good data. If the retry threshold is reached after reseeking, then the final error recovery procedure is attempted. This procedure is ECC. The ECC bits appended to each block are used in an attempt to correct the error. If successful, the host is guaranteed good data.

If the data is successfully recovered by retrying, then the number of retries necessary is checked. If it exceeds the retry soft error limit then that block is dynamically replaced, on the assumption that it is in the process of gradual failure and will get worse. The known good data is written to the replacement block and the host is notified of success.

If the data is successfully recovered by the ECC algorithm then the block is dynamically replaced on the assumption that the block has developed a hard error. The corrected data is written to the replacement block and the host is notified of success.

If the data cannot be recovered by any of these means then the block is assumed to be bad. The block is dynamically replaced, and is written with forced error status. This will cause forced error status to be returned whenever the block is read, telling the host that while the block is good, the data in the block is bad. When the block is written, the forced error status will be removed and from then on the block will be good. When a block is dynamically replaced with forced error status, the host is notified of a forced error on that block.

DRIVE SHADOWING

3.11 Drive shadowing

This feature provides continuous automatic backup of important data. The normal case is to have two identical drives with a single unit on each. The primary unit must be of identical size to the shadow. Once the primary/shadow pair has been defined, the shadow unit will become invisible to the host. All writes directed to the primary will also be written to the shadow. If a read is directed to the primary, the controller will attempt to redirect that read to whichever drive of the pair has its heads positioned closest to the requested data.

This feature only offers protection against the primary unit failing. It cannot protect, as a normal backup can, against a rogue program which writes garbage on the disk. The garbage will simply be copied onto the shadow unit as well. It is advisable, although not mandatory, that the shadow unit is on a different drive to the primary if the maximum protection is to be gained.

Data can be recovered from the shadow if the primary fails by running WOMBAT. Disable the shadow unit with the "Reset shadow units" option of the Shadow Options Menu. Then (optionally) change the unit numbers of the two units with the "Change unit number" option of the Disk Structure Menu. Once the data has been recovered the primary unit can be serviced and re-installed in the system.

3.12 Computing Sectors Per Track

To compute the number of sectors per track first determine the number of bytes per track from the drive manual. Divide this figure by the number of bytes per sector. The controller requires at least 593 bytes per sector. The drive switches should be set to provide the appropriate number of sectors per track. In some cases this will result in a short sector at the end of the track. If the drive has a short sector WOMBAT must be told when structuring the disk.

For example the number of full sectors per track for a PRIAM 806 Winchester with an SMD interface would be calculated as follows:

The unformatted capacity per track is 20,160 bytes. This is divided by 593 giving the number of sectors per track.

$$\frac{20{,}160}{593} = \textbf{33.996 sectors per track}$$

In this case there will be 33 full sectors with one short sector.

COMPUTING SECTORS PER TRACK.

For a CDC 9715 Winchester, the unformatted capacity of each track is 30,240 bytes.

$$\frac{30{,}240}{593} = 50.994 \text{ bytes sectors per track}$$

In this case there will be 50 full sectors with one short sector.

THE SHORT SECTOR MUST BE AT LEAST 50 BYTES LONG. (SEE SECTION 3.4 "SHORT SECTORS/TRACK:")

3.13 WOMBAT Error Messages

The following is a list of the error messages displayed by WOMBAT.

sector too short
READ OR WRITE IS ACTIVE INDEX OR SECTOR PULSE OCCURS.

sector not found
Indicates that the sector asked does not exist or cannot be located.

drive fault
Indicates that the drive is faulty and that service is necessary.

drive timed out
Indicates that the drive has failed to complete an operation.

data field error
Indicates that bad data exists in a sector.

controller fault
Indicates controller failure. Service will be necessary.

block marked as bad
Indicates that the block has been flagged as bad. The controller will refer to the RCT for a replacement block.

data late
Data is lost due to internal overflow in controller memory before transmission over the bus to host.

forced error
A forced error occurs when a good block has bad data. The block is flagged forced error until good data is written to the block.

seek error
An error has occured on a seek operation.

rct full
The replacement control table is full. An error of this kind indicates that the disk has too many errors to be serviceable.

WOMBAT ERROR MESSAGES

rct read error — Fatal error which indicates the controller cannot read the RCT.

rct write error — Fatal error which indicates the controller cannot write the location of replacement blocks.

illegal sector specified — A sector with an illegal number has been specified.

illegal block number — A block with an illegal number has been specified.

non existent drive — A drive has been specified which does not exist.

unit table full — The unit table is full. The maximum number of units is 16. To create a new unit an existing unit must be undefined.

no drive selected — A drive must be selected before WOMBAT can perform any operation. Select any valid drive.

non-existent unit — The unit selected has not been defined.

disk structure write error — WOMBAT is unable to write the structure to the disk. This is a fatal error indicating that the drive is not serviceable.

3.14 WOMBAT Self-Diagnostics

Initialization procedures

A common initialization procedure exists for both WOMBAT and the MSCP firmware. It performs :

a RAM integrity test
a ROM checksum
various checks on the disk drive and its structure

The errors which can result from this are described under Section 4.9 Fatal Controller Errors.

CHAPTER FOUR
MSCP PROGRAMMING

4.1 Overview of MSCP

Mass storage control protocol (MSCP) is a message-oriented set of rules by which the SDC-RQD11-SC controller module and the host system communicate. This protocol allows the host to send message requests for data reads or writes to the controller and receive response messages back from the controller. The host does not concern itself with details such as device type, media geometry, media format, or error recovery.

All software and hardware functions are partitioned into independent 'host' and 'controller' layers. Each layer consists of a high-level I/O driver and a communications server. The controller layer receives and processes commands which have been formed by the host layer.

The communications server handles all communications protocol between the I/O layers, leaving the I/O system free to process data requests. Communications between host and controller are carried out on the I/O bus without having to generate processor interrupts. The host's communications server monitors all command transmission and response and in the event of failure or error, initiates recovery procedures.

Disk drive parameters are transparent to both the host and controller resident layers of MSCP. The disk drive passes factors such as disk geometry, storage capacity or error retry counts to the disk controller on system start-up.

In addition to relieving the host of disk-specific data, the disk controller and disk provide the host with "clean" data. The disk drive handles some positioner errors entirely by itself but performs certain error-recovery operations under direction of the disk controller.

4.2 Controller Communications

The host designates an area of memory to be used as a communications area between itself and the controller. This area is made up of two sections a header area containing interrupt identification words and a variable-length section containing the response (receive) and command (send) lists, organized into ring buffers. Figure 4-1 shows the memory communications format.

FIGURE 4-1   MEMORY COMMUNICATIONS FORMAT

```
                <---------16-Bit Word--------->
                --------------------------------
Lower           !    COMMAND INTERRUPT WORD    !
Address         --------------------------------
                !    RESPONSE INTERRUPT WORD   !
                --------------------------------
                !           TWO WORD           !  )
                !       BUFFER DESCRIPTOR      !  )
                !------------------------------!  )  Response
                !------------------------------!  )  descriptor ring
                !------------------------------!  )
                !------------------------------!  )
                !------------------------------!  )
Two MSBs = 0    !------------------------------!   )
                !------------------------------!   )
                !------------------------------!   ) Command
                !------------------------------!   ) descriptor ring
                !------------------------------!   )
                !------------------------------!   )
Highest         !------------------------------!   )
Address         !------------------------------!   )
```

Command and Response Rings

Command and response lists are organized into 'rings' of 32 bit descriptors. The length of each ring is determined by the speed with which the host and controller generate and process messages. The host sets the ring lengths at initialization time. Figure 4-2 describes the Descriptor Format while Table 4-1 details the code descriptions.

FIGURE 4-2   DESCRIPTOR FORMAT

```
15                                                              00
-----------------------------------------------------------------
! L ! L ! L ! L ! L ! L ! L ! L ! L ! L ! L ! L ! L ! L ! L ! Z !
-----------------------------------------------------------------
! O ! F !             Reserved              ! H ! H ! H ! H ! H ! H !
-----------------------------------------------------------------
```

TABLE 4-1 COMMAND RING CODE DESCRIPTIONS

CODE	DESCRIPTION
Z	Is zero, as envelope address (text+0) is word-aligned. The controller will always assume that Bit 00 is set to zero.
L	Low-order envelope address.
H	High-order envelope address.
F	Flag bit.

When the controller returns ownership to the host it sets F=1 to indicate that it has completed action on the descriptor.

When the controller acquires ownership of a descriptor from the host, F=1 indicates that the host is requesting a ring transition interrupt. If F=0, the host is not requesting a ring transition interrupt. An interrupt will occur only if this descriptor causes a ring transition and if transition interrupts were enabled during initialization.

The controller always sets F=1 when returning a descriptor to the host, so if a host wishes to override ring transition interrupts it must always clear F when passing ownership of a descriptor to the controller.

0 Ownership bit. Set to 0 if owned by the host or 1 if owned by the controller. Interlocks the descriptor against premature access by either party.

Message Packets

The command or response descriptor points to word (text+0) of a 16-bit word-aligned message envelope formatted as shown in Figure 4-3. Table 4-2 describes the word envelope contents.

FIGURE 4-3 MESSAGE ENVELOPE FORMAT

```
           15                08 07     04 03   00
           ----------------------------------------
      -4 !      Message length (in bytes)       !  ) Message
           ----------------------------------------  ) envelope
      -2 !  Connection Id    !  msgtyp  !credits!  )
           ----------------------------------------
  text+0 !        MB1        !        MB0       !
           ----------------------------------------
      +2 !        MB3        !        MB2       !
           ----------------------------------------
         .                                      .
         .                                      .
         .                                      .
           ----------------------------------------
         !       MBn-1       !       MBn-2      !
           ----------------------------------------
```

TABLE 4-2 WORD ENVELOPE CONTENTS

WORD	ENVELOPE CONTENTS
0	Message length, in bytes. For commands, this length is equal to the size of the command (in bytes), beginning with [text+0]. For responses, the host sets the length equal to the size of the response buffer (in bytes), beginning with <text+0>. Before actual transmission of a response, the controller reads the field length in the message envelope. If the controller's response is longer than the response buffer, the controller will fragment its response into as many response buffers as necessary. The controller sets the resulting value into the message length field. The host must therefore keep re-initializing the value of this field for each proposed response. If a controller's responses are less than or equal to 60 bytes, then the controller need not check the size of the response slot.
1	Connection Id Identifies the connection serving as a source of, or destination for, the message in question.
2	Message Type The following response ring message types are implemented: MSGMNT Maintenance packet (diagnostic) MSGCRD Credit notice (ignored) MSGDAT Datagram packet. MSGSEQ Sequential packet
3	Credit field Gives a credit value (usually one) associated with the message. This mask, in response packets, is added to the controller's credit field to give the number of commands-in-progress. So while Word 1 is always 1 for the command ring, this is not the case for response rings.

4.3 Message Transmission

Command Transmission

When the ownership bit (0) of a command ring descriptor is equal to 1, it means that the host has filled the descriptor and is releasing it to the controller. When the ownership bit (0) resets to zero, it means that the controller has emptied the command ring descriptor and is returning ownership of the descriptor to the host.

To ensure that the controller sees every command, the host must read the IP register whenever it inserts a command in the command ring. This forces the controller to poll the command if it was not already accessing the command ring.

Response Transmission

When the ownership bit (0) of a response ring descriptor is equal to zero, it means that the controller has filled the descriptor and is releasing it to the host. When the ownership bit (0) sets to 1 it means that the host has emptied the response ring descriptor and is returning ownership of the descriptor to the controller. Just as the controller must poll for commands, so must the host poll for responses.

Interrupts

The transmission of a message will result in a host interrupt from the controller under the following circumstances.

1. During the initialization process (open a 'connection').

2. When the command ring buffer transitions from 'full' to 'not full'. This interrupt means that the host may place another command in the command ring.

3. When the response ring buffer transitions from 'empty' to 'not empty'. This interrupt means that there is a response for the host to process.

4. When a fatal controller error is detected and an interrupt can be generated. These are:

    Failure to become Qbus master for data transfer
    Failure to become Qbus master for interrupt
    Failure to access I/O page registers or communication area
    Qbus parity error detected.

4.4 Data Transmission

In the command ring, the descriptor points to a command packet. Within the command packet is a buffer descriptor which contains a pointer and a byte or word count. The buffer descriptor points to the data buffer which holds data transfers. The data is moved by the controller into or out of the buffer as DMA transfers to/from Qbus addresses.

4.5 Initialization

The purpose of initialization is to identify the parameters of the host-resident communications region to the controller, provide a confidence check of controller integrity, and bring the controller online to the host.

Initialization Process

This paragraph describes the activity within the SA register during an initialization process. This is dependent on whether SA is being read or written.

By moving 4000 into IP, the controller initializes and passes back the 'step' response in SA. Then, the initialization parameters are written into SA. There are 4 words of initialization, and the controller must reflect each step by the appropriate step response, which is also returned in SA. The initialization parameters are set out below in Table 4-3.

TABLE 4-3 INITIALIZATION PARAMETERS

WORD	CONTENTS
0	Command and Response ring sizes, interrupt enable and vector. The host writes into SA the lengths of the rings, whether interrupts are to be armed, and if so, the address of the interrupt vector. The controller then runs a complete internal integrity check and signals either success or failure.
1	Low order address of communications area, ie., ring buffer address. The host reads an echo of the ring lengths from SA, and then writes into SA the low-order portion of the ring base address.
2	High order address of communications area, bits 0-14. The interrupt vector address and the master interrupt arming signal are echoed in SA. The host then writes the high order portion of the ring base address to SA along with a signal that conditionally triggers an immediate test of the polling functions of the controller.
3	Burst transfer control, last failure flag, and the 'GO' bit. The controller tests the ability of the Qbus to perform DMA transfers. If successful, the controller zeros the entire communications area, and then signals the host that initialization is complete.

4.6 Registers

The programmable registers contained on the SDC-RQD11-SC are the Initialize and Poll register (IP) and the Status and Address register (SA).

Initialize and Poll Register (IP)

The host begins the initialization sequence by either issuing a bus initialize or by using the IP initialize operation. Any write to that address will cause an initialization of the controller. When read while the controller is operating, it causes the controller to initiate polling. The SDC-RQD11-SC responds to the 16 bit initialization words as set out in Table 4-4.

TABLE 4-4 SDC-RQD11-SC INITIALIZATION WORDS

NUMBER octal	 hex	FUNCTION	PROCESSOR
000250	00A8	WOMBAT	PDP/11
000254	00AC	WOMBAT	MicroVAX II
000260	00B0	WOMBAT	On board maintenance port
000272	00BA	WOMBAT	User communication program

Status and Address register (SA)

The SA register consists of a set of two registers, the SA read register and the SA write register.

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 controller.

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

4.7 MSCP Commands

TABLE 4-5 SDC-RQD11-SC MSCP COMMANDS

COMMAND	FUNCTION
Access	Reads data from the specified unit.
Abort	Guarantees that referenced MSCP command will complete within the controller timeout period.
Available	If specified unit is on-line, returns it to the unit-available state. If specified unit is currently in the unit-available state, this command has no affect.
Compare Host Data	Reads data from the disk and compares it with the data in the host buffer.
Erase	Writes zeros to the specified logical blocks on the unit. (No data is accessed from the host).
Get Command Status	Reports on the status of a specified command by returning a number that reflects the command's progress.
Get Unit Status	Reports on the status of a specified unit.
On Line	Places the specified unit on line, if possible.
Read	Reads data starting from the specified logical block on the disk, into host memory.
Set Controller Characteristics	Sets host-settable controller characteristics.
Set Unit Characteristics	Sets host-settable unit characteristics.
Write	Writes data starting at the specified logical block on the disk, from the host memory.

4.8 Error Handling

High data integrity is achieved by the controller through a 48 bit ECC (error checking and correction) polynomial with an 11 bit correction span. ECC will first try to read or write a block up to 10 times before attempting to correct the error. If error correction fails a non-recoverable error is reported. Table 4-7 details the MSCP status code messages.

4.9 Fatal Controller Error

If a fatal error is detected when the controller is initialized, the red error LED flashes and the fatal error status is set in the SA register. Table 4-6 describes fatal controller errors.

TABLE 4-6 FATAL CONTROLLER ERRORS

ERROR		DESCRIPTION
octal	hex	
100004	8004	RAM test failure
100005	8005	ROM checksum failure
100011	8009	No drive
100100	8040	Disk unformatted
100101	8041	Disk unstructured

Running WOMBAT and selecting the Display Error option will give an appropriate error message. A full description may be found under Section 3.17 - WOMBAT ERROR messages.

TABLE 4-7 MSCP STATUS CODE MESSAGES

MESSAGE	MEANING
Command Aborted	The current command was aborted before it could be completed normally.
Compare Error	While performing a Compare command, a Discrepancy was found while comparing the disk data to the host data.
Controller Error	The SDC-RQD11-SC controller detected an internal error, but is able to continue processing its outstanding commands.
Data Error	An error was detected in the reading or writing of data. ECC attempts to read or write data up to 10 times. If the error persists correction is attempted. If correction fails the error is reported.
Drive Error	A drive-related error was detected (such as a seek failure).
Media Format Error	Indicates that the media mounted on the unit was incorrectly formatted.
Host Buffer Access Error	Reports bus timeouts and parity errors during data transfers. (Applies only to the data portion of an MSCP command).
Invalid Command	The SDC-RQD11-SC controller found some field in the command to be in error.
Success	The command was successfully completed.
Unit Available	The SDC-RQD11-SC controller is not on line, but it can accept an On Line command from the host.
Unit Offline	The SDC-RQD11-SC controller is not on line, and it cannot be brought on line.
Write Protected	A Write or Erase command was attempted to a unit that is logically write-protected.

CHAPTER FIVE
SDC-RQD11-SC Operating Systems

The following discussion is intended to supplement DEC operating system resources and aims to aid the user of the SDC-RQD11-SC in understanding how different operating systems integrate the device. This information will help the user of the controller plan the installation and in choosing the appropriate bus addresses and interrupt vectors for the the disk subsystem. For a complete description the DEC system documentation should be consulted.

5.1 Operating Systems Overview

In order to install any new device in a computer, the host operating system must be informed of the device's existence and where to find that device. In DEC operating systems this can be done in one of the following ways :-

(a) The device can be manually connected using CONNECT or Configure statements.

(b) The operating system can be informed about the peripheral device during an interactive SYSGEN.

(c) The operating system can poll the device I/O address space.

Any of these methods will accomplish the desired result. The host system will be alerted to the device's existence, type, address and interrupt vectors.

Method (a) creates a command file that is executed on power-up. Method (b), interactive sysgen, creates a configuration file that the operating system accesses on power-up. Method (c) is referred to as 'autoconfigure'. RT-11 does not use autoconfigure but references standard bus addresses where it expects to find a device. All DEC operating systems try to follow the same set of rules but there are differences. These are discussed below.

MSCP Devices

The SDC-RQD11-SC is an MSCP (Mass Storage Control Protocol) type device. All MSCP-type devices contain two registers that are visible to the Qbus I/O page. They are the Initialization and Polling (IP) register and the Status and Address (SA) register.

Qbus Addresses

The standard Qbus address of 17772150 (Octal) is used by all of the operating systems described in this manual as the address of the first controller on the host system. The IP register, CSR address, Qbus address and the base address all refer to the same register.

Vector Addresses

Many operating systems choose vector addresses automatically. If an operating system requires manual input of vector addresses they are programmed into the controller during the initialization process.

Device Names

Table 5-1 is a list of device names for five operating systems. Two controller and device names are given to indicate the numbering scheme.

TABLE 5-1 DEVICE NAMES IN DEC OPERATING SYSTEMS

OPERATING	**CONTROLLER**		**DRIVE**	
	1st:	**2nd:**	**1st:**	**2nd:**
RSTS/E	RU0,	RU1	DU0,	DU1
RSX-11M	---	---	DU0,	DU1
RSX-11M-PLUS	DUA	DUB	DU0,	DU1
RT-11	Port0,	Port1	DU0,	DU1
VAX/VMS	PUA,	PUB	DUA0,	DUA1

5.2 RT-11 Operating System

Installation of a Single Controller

A single controller is installed at the Qbus address of 17772150 (Octal) where RT-11 will find and then install the handler for that device. It is not necessary to run sysgen for a single controller. One of the pre-generated monitors provided with the RT-11 distribution kit can be used. To properly implement disk partitioning, the system start-up file (STARTx.COM) must be modified.

Installation of Multiple Controllers

There are two valid methods that can be used to install multiple controllers. Either by modifying the MSCP handler, which is described in the RT-11 Software Support Manual or by performing a Sysgen. The following procedure describes the SYSGEN technique with user input marked in **boldface** type.

1. Initiate SYSGEN:

   IND SYSGEN <return>

2. The system will then prompt the user by asking questions. The first concerns the use of a start-up command file when booting.

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

   The start-up file performs two main functions. These specify the additional controller addresses and ensure that disk partitioning is carried out consistently on each bootstrap or power-up.

3. Select the device DU: as the MSCP device when prompted for Disk Options.

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

4. Inform the system of the number of controllers to be installed.

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

   RT-11 refers to individual MSCP controllers on the host as ports. Each port has its own Qbus and vector addresses.

5. All other devices in the host computer configuration have to be specified. After completing this step, indicate that there are no more devices by entering a period (.).

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

6. Using the SET CSR keyboard command, specify the address of all the MSCP controllers. These must be added to the system start-up file STARTx.COM. The 'x' indicates the monitor to be used - S for single job, **F** for foreground/background, and **X** for extended memory. The command file must be edited to include the following statements :-

   ```
   SET DU CSR      =   17772150  (DEFAULT)
   SET DU CSR2     =   17760334
   SET DU VECTOR   =   154       (DEFAULT)
   SET DU VEC2     =   160
   ```

   The second device can be at any unused address on the Qbus I/O page supported by the pin settings on the controller. The vector address can be any unused address in the vector page. No default statements are required.

Disk Partitioning Under RT11

Drives with capacities greater than 65,535 blocks (33.5 Mbytes) cannot be handled by RT-11 unless they are partitioned into smaller segments. Each partition can be smaller than 65,535 blocks if desired but there is a maximum of eight logical devices per physical drive. Each logical drive will be addressed by RT-11 as an independent physical drive.

The assignment names of each logical drive must be placed in the start-up command file to ensure that the drives are partitioned consistently and automatically each time the system is booted. The following is an outline of the procedure used to determine the number of logical drives to be assigned to each physical drive.

1. Decide on the drive configuration to be used. The logical unit number (LUN) and data storage capacity in logical blocks of each logical drive must be known. The controller plug settings must correspond to the bus address selected.

2. The total number of logical disks any physical disk can be partitioned into is calculated by dividing the selected block size of each logical disk into the total capacity of the the disk unit. Round the result to the nearest whole number. The last partition can be less than the maximum size selected. This number equals the number of logical disks.

3. STARTx.COM must now be edited to include the logical names of each partition. The format of each statement is :-

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

   where 'n' is the logical device name, 'y' is the unit number, 'x' is the partition number, and 'z' is the controller number. This must be done for each partition on each drive, including drives that have only one partition.

Sample Disk Partitioning Procedure

The following is an example of the disk partitioning procedure for a drive of 245,412 blocks and a drive of 204,800 blocks. It has been decided to partition the drives into logical units of 65,535 blocks.

```
(a)     245,412
        -------     =    3.74  ( 4 logical units )
         65,535

(b)     204,800
        -------     =    3.12  ( 4 logical units )
         65,535
```

Dividing the unit capacities by 65,535 and rounding the result to the nearest whole number gives the number of logical units. If the remainder is very small (under 800 blocks) then it would be advisable to round the figure down rather than up to the next highest number. This may avoid problems with partitions that are too small to be practicable.

Logical names can then be assigned to the partitions beginning with DU0 on controller unit 0 and modifying the start-up file to include the assignments.

```
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=1 PART=0 PORT=0
SET DU5 UNIT=1 PART=1 PORT=0
SET DU6 UNIT=1 PART=2 PORT=0
SET DU7 UNIT=1 PART=2 PORT=0
```

5.3 RSTS/E Operating Systems (V8.0 and above)

RSTS/E can support two MSCP type controllers. The first is located at the standard Qbus address (17772150 octal) while the second can be located in floating address space. However, the recommended address for the second controller is 17760334. A controller must be located at the standard Qbus address to be a bootstrap device.

A program called INIT.SYS scans the system on power-up. INIT.SYS references a user-specified table located in the currently installed monitor. To alter the autoconfigure algorithm, the HARDWARE sub-option of INIT.SYS is used. This modifies the configuration table and allows an MSCP controller to be placed at any address on the I/O page. If a new monitor is installed then the table must be reset.

Controllers are assigned vector addresses and programmed by INIT.SYS during initialization.

Warning: RSTS/E supports disks of a maximum size of 1,048,576 blocks. Larger drives must be broken up into multiple smaller virtual units. At a later date RSTS/E may support larger disks, refer to the RSTS/E Software Despatch for details.

5.4 RSX-11M Operating Systems (V4.0 and above)

The RSX-11M SYSGEN program is an interactive program that builds a complete and running RSX-11M system for a particular hardware configuration. RSX-11M SYSGEN supports autoconfigure. This program detects MSCP type controllers located at standard Qbus addresses. Additional controllers must be manually attached to the system according to the procedure outlined below. The procedure is fully outlined in the RSX-11M System Generation and Configuration Guide.

Installing a single controller

A single controller is installed at the standard Qbus address of 17772150 (Octal). Autoconfigure can then be used to connect peripheral devices.

Installing multiple MSCP controllers

For two controllers manual initialization must be undertaken. The following procedure will connect the devices to the operating system :-

1. Invoke SYSGEN.

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

2. Indicate that AUTOCONFIGURE has to be used by answering **Y** (Y) to the following :-

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

3. Indicate that the autoconfigure results are not to be overriden. Answer **N** (no) to the following :-

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

   Continue to answer the SET-UP questions as required then continue onto the TARGET CONFIGURATION section. Target configuration defaults for the first group of questions should be accurate because autoconfigure was used.

4. Indicate the number of devices that are installed.

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

Enter the value the correct value of two. The period (.) terminates the device input operation.

The questions over the next four sections - HOST CONFIGURATION,EXECUTIVE OPTIONS, TERMINAL DRIVER OPTIONS, and SYSTEM OPTIONS - should be answered appropriately.

5. After answering the above sections it is necessary to define the PERIPHERAL OPTIONS for the controllers on the system. The questions will be asked once for each controller. The abbreviated form of controller "contr" is used.

The first prompt is for the interrupt vector address, Qbus address, the number of DU-type disk drives, the number of command rings, and the number of response rings. There is no default value for the number of disk drives.

```
*   DU contr 0 [D:154,17772150,,4,4]
*   154,17772150,3,4,4<return>
```

Vector and Qbus Addresses

The standard vector address for MSCP controllers is 154 (octal). Any unused vector between 300 (octal) and 774 (octal) can be allocated for the second unit.

The standard Qbus address of 17772150 (octal) is used for the first controller, while the second can be 17760334 (octal) or in floating LSI-11 address space.

Drive Configuration

The following is a list of DEC manufactured drives that are DEC operating system compatible. Non-DEC drives must be compatible with those listed below.

If in doubt consult the manufacturer's specifications to verify compatibility.

```
*RX50
*RD51
*RD52
*RC25
*RA60
*RA80
*RA81
```

Count each RX50 drive as two drives, these contain two 5.25 inch floppy diskettes. The RC25 has both fixed and removable media and should also be counted as two drives.

The configuration of the drives and the logical arrangement (disk partitions) for the disk sub-system is programmed by WOMBAT.

MSCP Ring Buffers

Command and response ring buffers which MSCP establishes in main memory also have to be specified. RSX-11M supports a maximum of eight rings. A value of four will minimize system overhead and is the recommended and default value.

6. The type of disk drives on each controller must now be specified.

```
*DU contr 0 unit 0. is an RA60/80/81/RC25/RD51/rx50
[D:RA81] RD51<return>
```

For the RQDX1, indicate that there is a RD51 and two RX50 drives. For the SDC-RQD11-SC, indicate that there is one RD51 for each logical disk drive.

RSX-11M must have contiguous unit numbers which must be the same as those reported by the controller during initialization.

Warning: Versions of RSX-11M prior to 4.2C support disks of a maximum size of 1,044,480 blocks. Larger drives must be broken up into multiple smaller virtual units.

5.5 RSX-11M-PLUS Operating Systems (V2.1 and above)

As with RSX-11M an interactive SYSGEN will build a complete running version of RSX-11M-Plus for a particular hardware configuration. RSX-11-Plus supports autoconfigure and will detect the first controller located at the standard Qbus address. Additional controllers must be installed manually.

Installing a Single Controller

A single controller is installed at the standard Qbus address of 17772150 (octal) using autoconfigure to connect the peripherals. The procedure is fully outlined in the RSX-11M-Plus System Generation and Configuration Guide.

Installing Multiple Controllers

To add the SDC-RQD11-SC to the system configuration use the Add a Device option of SYSGEN or do a complete SYSGEN. The Add a Device procedure is described below :-

1. Invoke SYSGEN

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

2. Answer **N** (no) to the following questions to indicate that only a subset of the SYSGEN procedure is wanted :-

```
        * SU120 Do you want to do a complete SYSGEN?
                [Y/N D:Y]: N<return>

        * SU130 Do you want to continue a previous SYSGEN
                from some point? [Y/N D:Y]: N<return>
```

3. Indicate that a specific module of SYSGEN is required by answering **Y** (yes) to the following :-

```
* SU150 Do you want to do any individual sections
        of SYSGEN? [Y/N D:Y]: Y<return>
```

4. Select the Add a Device option of SYSGEN by typing the letter **H**.

```
* SU160 Which sections would you like to do?
        [S R:0.-15.]: H<return>
```

SYSGEN now asks questions about the type and number of controllers to be installed in the system. There is one question for each controller supported. Type **0** (zero) until the prompt for UDA-type devices appears.

5. Specify the number of MSCP devices when asked by typing :-

```
* CP3004  How many MSCP disk controllers do you
          have? [D R:0.-63. D:0.] 2<return>
```

6. Give the total number of drives on each controller installed on the system.

```
* CP3008  How many MSCP disk drives do you have?
          [D R:0.-n. D:1.] 5<return>
```

The following is a list of DEC manufactured drives that are DEC operating compatible. Non-DEC drives must be compatible with those listed below. If in doubt consult the manufacturer's specifications to verify compatibility.

*RX50
*RD51
*RD52
*RC25
*RA60
*RA80
*RA81

Count each RX50 drive as two drives, these contain two 5.25 inch floppy disks. The RC25 has both fixed and removable media and should also be counted as two drives.

The configuration of the drives and the logical arrangement (disk partitions) for the disk sub-system is programmed by WOMBAT.

7. SYSGEN then asks the user to specify controllers for each drive.

```
* CP3044  To which DU controller is DU0:
          connected? [S R:1-1]: A<return>
```

This question is repeated until the number of MSCP drives has been exhausted. RSX-11M-Plus must have contiguous unit numbers and be the same as those reported by the controller during initialization or errors will occur. Use A as the primary and B as the alternate controller.

8. Enter the Vector Address for each controller.

```
* CP3068 Enter the vector address of DUA
         [O R:-774 D:154]
```

The standard vector address for MSCP controllers is 154 (octal). Any unused vector between 300 (octal) and 774 (octal) can be allocated for the second unit.

9. Enter the CSR address for each controller.

```
* CP3076 What is its CSR address?
         [0 R:1.-8. D:4.] 4<return>
```

The standard CSR address 17772150 (octal) is used for the first controller, while the second can be 17772154 (octal) or in floating CSR address space.

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

```
* CP3076 Enter the number of command rings for
         DUA [D R:1.-8. D:4.] 4<return>
```

RSX-11M-Plus supports a maximum of eight command rings. A value of four will minimize system overhead and is the recommended and default value.

11. Specify the number of response rings for each MSCP controller.

```
* CP3076 Enter the number of response rings for
         DUA [D R:1.-8. D:4.] 4<return>
```

RSX-11M-Plus supports a maximum of eight response rings. A value of four will minimize system overhead and is the recommended and default value.

Warning: Versions of RSX--M Plus prior to 3.0C support disks of a maximum size of 1,044,480 blocks. Larger drives must be broken up into multiple smaller virtual units.

5.6 MicroVMS Operating Systems

The first SDC-RQD11-SC controller is located at the standard bus address of 17772150 (Octal) and the second in floating address space. The MicroVMS SYSGEN utility can determine the Qbus and interrupt vector addresses for any of the I/O devices installed on the bus. MicroVAX/MicroVMS must be running in order to use this utility. The Qbus and interrupt vector addresses can be determined manually if access to a running system is not possible.

Using MicroVAX/MicroVMS SYSGEN

The following is an outline of the MicroVMS SYSGEN procedure to determine Qbus and Interrupt vector addresses. This procedure requires system manager privileges.

1. Login and run the SYSGEN utility.

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

   The SYSGEN> prompt indicates that the program is ready.

2. Obtain a list of the devices currently installed on the MicroVAX Qbus by typing :-

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

   and get :-

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

* Indicates a floating vector or address.

Sysgen lists the devices already installed on the Qbus by logical name. Devices with floating bus and vector addresses should be noted if it is intended to re-install them with the SDC-RQD11-SC controller. Floating bus addresses will be larger than 760000 (octal). Floating interrupt vectors will be larger than 300 (octal).

3. Execute the configure command. This will determine the Qbus and Vector addresses that autoconfigure will expect for each device type.

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

Specify the devices to be installed on the bus by typing their Qbus names. Under MicroVAX/MicroVMS the device name for MSCP-type controllers is **UDA**.

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

The device name is separated from the number of devices by a comma. The number of devices is specified in decimal.

Devices with floating addresses or vectors are not affected by devices with fixed addresses or vectors. Only devices with floating addresses or vectors need be specified.

4. When all the devices have been specified enter a control-Z.

```
DEVICE> CTRL-Z
```

The addresses and vectors of the devices entered will be listed in the following manner :-

```
Device: UDA    Name: PUA CSR: 772150  Vector: 154   Support: yes
Device: UDA    Name: PUB CSR: 760334* Vector: 300*  Support: yes
Device: DHV11  Name: TXA CSR: 760500* Vector: 310*  Support: yes
```

* Denotes floating bus and interrupt vector addresses. Floating CSR addresses must be programmed into the SDC-RQD11-SC by selecting the correct pin configuration on the PCB.

5. If an address other than that selected for the SDC-RQD11-SC by CONFIGURE command is desired, CONNECT statements must be entered into the SYSCONIF.COM file. SYSCONIF.COM can only be accessed through the system manager's account SYS$MANAGER. The correct syntax is given in the DEC MicroVMS SYSGEN documentation.

   The STARTUP.COM or UVSTART.COM command files in the main system account, SYS$SYSTEM must not be altered.

5.7 Autoconfigure

Autoconfigure is a utility program that finds and identifies I/O devices in the I/O page of system memory. Most devices have a fixed bus address reserved for them. When the computer is bootstrapped autoconfigure polls those addresses - specifically the console status register (CSR) which is usually the first register of the block.

A block of addresses is reserved when a device is detected. The size of the block is determined by the number of registers the device uses. Autoconfigure then looks to the next CSR address space for that same type of device. If there are no other devices of that type autoconfigure looks to the next valid CSR address. Autoconfigure expects an eight byte block to be reserved for each device not installed in the system. An empty block tells autoconfigure to look to the next valid address space.

Devices with no fixed address are assigned addresses from floating CSR address space. This may be necessary if there are several of the same device in the system. Floating address space is in the vicinity of 76000 to 763776 of the bus I/O page. Devices can also have floating interrupt vector addresses. Floating CSR and interrupt vectors must be assigned in specific sequences depending on the rank of the device (see Table 5-2). The presence or absence of floating bus and interrupt vector address devices will affect the assignment of addresses to other floating vector devices.

TABLE 5-2 SYSGEN DEVICE RANKING

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 treated as two DZ11s.

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

[3] The first two devices of this type have fixed addresses while extra devices have floating addresses.

An eight byte gap must also be reserved in floating address space for each device type not currently installed in the system. This gap must start on the proper boundary. See Table 5.6 for an example of gap placement.

A device's CSR address is determined on word boundaries according to the number of bus accessible registers the device has. The relationship of word boundaries and device registers is set out in Table 5-3 Autoconfigure only inspects for a device type at one of the possible device boundaries. For instance, autoconfigure will not look for a DMZ32 which has 16 registers at an address that ends in 20.

TABLE 5-3 DEVICE REGISTERS AND 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

Vector Addresses and Autoconfiguration

Devices are assigned vector addresses in order of rank commencing at 300 (octal) up to 777 (octal). Extra devices of the same type are assigned consecutive vector addresses according to the number of vectors required and starting boundaries for each device type. Table 5-4 shows the order of assignment.

The boundaries in the modulus column indicate where vector addresses are assigned. If the modulus is 10 the first vector address for that device must end with a zero (XX0). If the modulus is 4 the first vector must end with with either a zero or four (XX0, XX4).

Vector addresses can only end on an address of four or zero i.e. modulo 4 boundaries (XX0, XX4). If a device has two vectors the first must start on a modulo 10 boundary. Using 350 as a starting point the vectors will be 350 and 354.

TABLE 5-4 FLOATING VECTOR ADDRESS DEVICE PRIORITY RANKING

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 5-4 FLOATING VECTOR ADDRESS PRIORITY RANKING

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] KL11 or DL11 have fixed vectors when used as a console.

[2] The first device has a fixed vector all subsequent device of the same type have a floating vector.

[3] ML11 is a Mass Bus device which connects to the Qbus or Unibus via a bus adaptor.

System Configuration Example

An example of a system configuration is shown in Table 5-5. The configuration includes both fixed and floating addresses and vectors.

TABLE 5-5 CSR AND VECTOR ADDRESS EXAMPLE

Controller	Vector	CSR
1 UDA50	154	772150
1 DZ11	300	760100
1 UDA50	310	760334
2 DHV11	320	760520
	330	760520

Table 5-6 shows the computed CSR addresses and gaps for floating devices.

TABLE 5-6 FLOATING CSR ADDRESS ASSIGNMENT

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	760330
	DEUNA	Gap	760340
----->	UDA50 (SDC-RQD11-SC)		772334[1]
----->	UDA50 (SDC-RQD11-SC)		760354
		Gap	760360
	DMF32	Gap	760400
		Gap	760440
	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] indicates a fixed address device

NOTES

CHAPTER SIX
SDC-RQD11-SC CACHE OPERATION

6.1 SDC-RQD11-SC Disk Cache

The SDC-RQD11-SC implements a disk cache which is designed to facilitate larger and faster data transfers between disk and the host by reducing the time wasted on positioner operations. Even with fast drives 98% of disk time for continuous random access to single sectors of data is taken by positioner operations. The SDC-RQD11-SC offers at least an 80% improvement in access times by reducing the number of disk accesses required. The SDC-RQD11-SC cache is implemented as one megabyte of dynamic RAM.

6.2 Read Look-ahead

The SDC-RQD11-SC allows the user to program the controller to perform read look-ahead in anticipation of impending data requests. The optimum look-ahead value can only be determined within system and application parameters but can range from 0 to 255 blocks. A value of zero will disable the feature. The default value is four.

The anticipated hit ratio for the SDC-RQD11-SC cache is 90% although this can be reduced depending upon the nature of the data accessed. Because most user programs write and read data sequentially there is a high probability that in one fetch operation the controller will be able to satisfy several sequential data reads without the need for further disk accesses.

The cache has been designed to maximise the probability of finding the target data over a range of sequential and non-sequential reference patterns while minimising cache misses and controller overhead.

6.3 Cache allocation

Cache memory is used to hold the disk cache blocks, a cache map and fixed buffers for special usage. Data from the disk or main memory is stored in blocks at addresses determined by the cache assignment algorithm. Their contents and location are recorded in the cache map.

The cache map consists of a 4-byte entry for each cache block. The cache map is indexed by the cache block number and contains the address (drive number, logical block) of the current occupant together with flags (locked, valid, primary copy not written, shadow copy not written).

Fixed buffers are assigned for RCT buffers (1 per drive) and a single block buffer for disk management I/O. The location and size of all cache variables are held in RAM.

6.4 Cache usage

All disk I/O is done via the cache. A set of cache blocks must be assigned for all transfers and continuous disk operations must be done via contiguous cache blocks. Disk and Qbus transfers are performed simultaneously.

6.5 Cache Assignment Algorithm

The SDC-RQD11-SC cache implements a contention based hashing algorithm to determine block replacement. A given disk block has a fixed cache address calculated as follows :-

(a) Get the remainder of the logical block number modulo the number of cache blocks.

(b) Bias this by a fixed offset which is a function of the drive number. (This is so that the same logical block on 2 disks have a different cache block number).

A disk block also has an alternative cache address calculated by biasing it by approximately half the number of cache blocks. The alternate cache block is only used for compare operations.

6.6 Cache Operation

The following decribes the cache operation algorithm :-

```
Read:
       Examine cache for data required.

           If all data in cache
           Transfer data from cache to Qbus

       else

           Assign cache (lock it and wait if locked already)
           Perform read
           Unlock cache and flag as valid

Write:
       Assign cache (lock it and wait if locked already)
       Transfer from Qbus to cache and flag as valid
```

```
        Perform write
        Unlock cache
```

6.7 Cache Disable

The SDC-RQD11-SC cache can be disabled for performance evaluation, engineering, and diagnostic application requirements by selecting that option in WOMBAT. The cache cannot be selectively disabled for a particular drive.

6.8 Early Write Notification

The SDC-RQD11-SC implements early write notification where data to be written to disk is retained in the cache and the host is issued a write complete notification. The controller will then write the data to the disk at the most convenient time.

Early Write Notification should be disabled when saving the boot block on the system volume. If it is not disabled, a subsequent reset instruction will clear the cache memory and the boot data will not be written to disk.

It should be noted that in the event of system failure any data residing in the cache will be lost. The early write notification can be disabled by invoking WOMBAT and selecting the appropriate option.

NOTES

CHAPTER SEVEN
SMD INTERFACE

7.1 SMD Interface

The Storage Module Drive (SMD) interface is an industry standard for high capacity disk drives. The basic interface consists of a control cable (J3), data (read/write) cable (J1 and J2), and a ground cable. The SDC-RQD11-SC uses flat-ribbon cables for connection to J1 through J3.

The control cable connects each drive to the host in either radial or daisy-chained configuration. The maximum cable length is 30 metres. The control cable is terminated on the last drive. The data (read/write) cable connects to each drive in a radial fashion only and has a maximum length of 15 metres. The data (read/write) cable is terminated at the last drive. Ground cabling should establish the host, drives and cabinets at the same safety ground reference.

The SMD interface has been modified by various manufacturers. The SDC-RQD11-SC supports the following types of SMD interface with no software or hardware changes. All the modified versions of the SMD interface implement the minimum set of requirements of interchangeability and compatibility for SMD devices but with degrees of flexibilty.

Standard	1024	Cylinder address capability
	1	Byte status available
	9	Control functions
Modified	4096	Cylinder address capability
	4	Byte status available
	12	Control functions
Enhanced	2048	Cylinder address capability
	1	Byte status available
	10	Control functions
Extended	1M	Cylinder address capability
	16	Byte status available
	19	Control functions

Pin assignments for the various SMD interface types are listed in Appendix A. The SMD documentation should be consulted for a detailed explanation of the specific function details of each SMD interface type.

The data (Read/Write) cable is common to all of the SMD interface types. Pin assignments are shown below.

Table 7-1 SMD Standard Control Cable Signals

SIGNAL NAME	SOURCE	SIGNAL PIN	
		+	-
DEVICE SELECT ENABLE	HOST	44	43
DEVICE SELECT 0	HOST	46	45
DEVICE SELECT 1	HOST	48	47
DEVICE SELECT 2	HOST	52	51
DEVICE SELECT 3	HOST	54	53
TAG 1	HOST	2	1
TAG 2	HOST	4	3
TAG 3	HOST	6	5
TAG BUS 0 (B0)	HOST	8	7
TAG BUS 1 (B1)	HOST	10	9
TAG BUS 2 (B2)	HOST	12	11
TAG BUS 3 (B3)	HOST	14	13
TAG BUS 4 (B4)	HOST	16	15
TAG BUS 5 (B5)	HOST	18	17
TAG BUS 6 (B6)	HOST	20	19
TAG BUS 7 (B7)	HOST	22	21
TAG BUS 8 (B8)	HOST	24	23
TAG BUS 9 (B9)	HOST	26	25
INTERFACE ENABLE	HOST	28	27
STATUS 0 (UNIT READY)	DEVICE	38	37
STATUS 1 (ON CYLINDER)	DEVICE	34	33
STATUS 2 (SEEK ERROR)	DEVICE	32	31
STATUS 3 (FAULT)	DEVICE	30	29
STATUS 4 (WRITE PROTECTED)	DEVICE	56	55
STATUS 5 (ADDRESS MARK)	DEVICE	40	39
STATUS 6 (INDEX)	DEVICE	36	35
STATUS 7 (SECTOR)	DEVICE	50	49
BUSY	-	42	41
	-		
PICK	-	57	
SEQUENCE	-	58	
SPARE	-	60	59

Table 7-2 SMD Extended Control Cable Signals

SIGNAL NAME	SOURCE	SIGNAL PIN	
		+	-
DEVICE SELECT ENABLE	HOST	44	43
DEVICE SELECT 0	HOST	46	45
DEVICE SELECT 1	HOST	48	47
DEVICE SELECT 2	HOST	52	51
DEVICE SELECT 3	HOST	54	53
TAG 1	HOST	2	1
TAG 2	HOST	4	3
TAG 3	HOST	6	5
TAG 4	HOST	60	59
TAG BUS 0 (B0)	HOST	8	7
TAG BUS 1 (B1)	HOST	10	9
TAG BUS 2 (B2)	HOST	12	11
TAG BUS 3 (B3)	HOST	14	13
TAG BUS 4 (B4)	HOST	16	15
TAG BUS 5 (B5)	HOST	18	17
TAG BUS 6 (B6)	HOST	20	19
TAG BUS 7 (B7)	HOST	22	21
TAG BUS 8 (B8)	HOST	24	23
TAG BUS 9 (B9)	HOST	26	25
INTERFACE ENABLE	HOST	28	27
STATUS 0 (UNIT READY)	DEVICE	38	37
STATUS 1 (ON CYLINDER)	DEVICE	34	33
STATUS 3 (FAULT)	DEVICE	30	29
STATUS 4 (WRITE PROTECTED)	DEVICE	56	55
STATUS 5 (ADDRESS MARK)	DEVICE	40	39
STATUS 6 (INDEX)	DEVICE	36	35
STATUS 7 (SECTOR)	DEVICE	50	49
BUSY	-	42	41
	-		
ROTATIONAL POSITIONING (INDEX)	-	57	
ROTATIONAL POSITIONING (SECTOR)	-	58	

Table 7-3 SMD Modified Control Cable Signals

SIGNAL NAME	SOURCE	SIGNAL PIN	
		+	-
DEVICE SELECT ENABLE	HOST	44	43
DEVICE SELECT 0/TAG BUS 10 (B10)	HOST	46	45
DEVICE SELECT 1	HOST	48	47
DEVICE SELECT 2	HOST	52	51
DEVICE SELECT 3 / TAG 5	HOST	54	53
TAG 1	HOST	2	1
TAG 2	HOST	4	3
TAG 3	HOST	6	5
TAG 4	HOST	60	59
TAG BUS 0 (B0)	HOST	8	7
TAG BUS 1 (B1)	HOST	10	9
TAG BUS 2 (B2)	HOST	12	11
TAG BUS 3 (B3)	HOST	14	13
TAG BUS 4 (B4)	HOST	16	15
TAG BUS 5 (B5)	HOST	18	17
TAG BUS 6 (B6)	HOST	20	19
TAG BUS 7 (B7)	HOST	22	21
TAG BUS 8 (B8)	HOST	24	23
TAG BUS 9 (B9)	HOST	26	25
INTERFACE ENABLE	HOST	28	27
STATUS 0 (UNIT READY)	DEVICE	38	37
STATUS 1 (ON CYLINDER)	DEVICE	34	33
STATUS 2 (SEEK ERROR)	DEVICE	32	31
STATUS 3 (FAULT)	DEVICE	30	29
STATUS 4 (WRITE PROTECTED)	DEVICE	56	55
STATUS 5 (ADDRESS MARK)	DEVICE	40	39
STATUS 6 (INDEX)	DEVICE	36	35
STATUS 7 (SECTOR)	DEVICE	50	49
BUSY	-	42	41
	-		
PICK	-	57	
SEQUENCE	-	58	

SMD Enhanced Control Cable Signals

SIGNAL NAME	SOURCE	SIGNAL PIN	
		+	-
DEVICE SELECT ENABLE	HOST	44	43
DEVICE SELECT 0	HOST	46	45
DEVICE SELECT 1	HOST	48	47
DEVICE SELECT 2	HOST	52	51
DEVICE SELECT 3	HOST	54	53
TAG 1	HOST	2	1
TAG 2	HOST	4	3
TAG 3	HOST	6	5
TAG BUS 0 (B0)	HOST	8	7
TAG BUS 1 (B1)	HOST	10	9
TAG BUS 2 (B2)	HOST	12	11
TAG BUS 3 (B3)	HOST	14	13
TAG BUS 4 (B4)	HOST	16	15
TAG BUS 5 (B5)	HOST	18	17
TAG BUS 6 (B6)	HOST	20	19
TAG BUS 8 (B8)	HOST	24	23
TAG BUS 9 (B9)	HOST	26	25
TAG BUS 10 (B10)	HOST	60	59
INTERFACE ENABLE	HOST	28	27
STATUS 0 (UNIT READY)	DEVICE	38	37
STATUS 1 (ON CYLINDER)	DEVICE	34	33
STATUS 2 (SEEK ERROR)	DEVICE	32	31
STATUS 3 (FAULT)	DEVICE	30	29
STATUS 4 (WRITE PROTECTED)	DEVICE	56	55
STATUS 5 (ADDRESS MARK)	DEVICE	40	39
STATUS 6 (INDEX)	DEVICE	36	35
STATUS 7 (SECTOR)	DEVICE	50	49
BUSY	-	42	41
	-		
PICK	-	57	
SEQUENCE	-	58	

Table 7-5 SMD Read/Write Cable Signals

SIGNAL NAME	SOURCE	SIGNAL PIN		
		+	-	GND
WRITE DATA	HOST	14	15	13
WRITE CLOCK	HOST	12	11	10
SERVO CLOCK	DEVICE	2	3	1
READ CLOCK	DEVICE	8	9	7
SELECTED	DEVICE	17	18	16
SEEK END	DEVICE	20	19	21
INDEX MARK	DEVICE	22	23	24
SECTOR MARK	DEVICE	26	25	
READ DATA	DEVICE	6	5	4

CHAPTER EIGHT
QBUS INTERFACE

8.1 Qbus Interface

All data, address and control information transfers between the processor and disk controller are carried out over the Qbus. The SDC-RQD11-SC supports all current Qbus functions including block mode DMA, 22 bit addressing, 4-level position independent interrupt structure, all LSI-11 CPU's and MicroVAX II.

The Qbus consists of 42 bidirectional and 2 unidirectional signal lines wired into the backplane assembly. These are grouped into the following categories :-

* Sixteen multiplexed data/address lines - BDAL<15:00>

* Two multiplexed address/parity lines - BDAL<17:16>

* Four extended address lines - BDAL<21:18>

* Six data transfer control lines - BBS7L, BDINL, BDOUTL, BRPLYL, BSYNCL, BWTBTL

* Six system control lines - BHALTL, BREFL, BEVNTL, BINITL, BDCOKL, BPOKL

* Ten interrupt control and direct memory access control lines - BIAKOL, BIAKIL, BIRQ4L, BIRQ5L, BIRQ6L, BIRQ7L, BDMGOL, BDMRL, BSACKL, BDMGIL

Communication is asynchronous, allowing devices with differing data rates to share the bus. A strict master/slave protocol avoids the need for synchronising clock pulses by implementing handshaking and other control signals between I/O devices.

8.2 Interrupts

Interrupt priority for the SDC-RQD11-SC is switch selectable on the PCB. The recommended priority setting is five. In order to service LSI-11 and LSI-11/2 CPU's the SDC-RQD11-SC automatically outputs level four interrupts despite switch priority selections.

Interrupts suspend program execution while the processor starts the device service routine at a vector address input from the requesting device.

Interrupts are serviced according to device priority. Device priority can be determined in two ways. These are termed 'Position Defined' and 'Distributed' arbitration. Positioned Defined arbitration gives priority to those devices which are electrically closest to the processor. Distributed arbitration implements priority according to the priority levels set on the device hardware. When devices with equal priority generate an interrupt, the processor gives preference to the device which is electrically closest. A previous bus transaction must have been completed before another can be commenced.

The interrupt protocol has three phases :-

1. Interrupt Request Phase. The interrupt enable bit in the status register is set and interrupt request lines are asserted according to priority settings.

2. Interrupt Acknowledge and Priority Arbitration Phase. The processor detects the request and checks if any other device with higher priority is requesting an interrupt. If there are no devices with higher priority seeking an interrupt the processor acknowledges the interrupt.

3. Interrupt Vector Transfer Phase. The device outputs vector address bits to the processor which then enters the device service routine.

8.3 Direct Memory Access

The SDC-RQD11-SC supports both normal and block mode Direct Memory Access (DMA). During a DMA transfer the processor passes mastership of the bus to the controller.

During block mode DMA transfer the SDC-RQD11-SC has a four microsecond delay after every 16 words to service any pending interrupt or DMA requests from other devices. The SDC-RQD11-SC also detects DMA requests from other devices and will implement a 'DMA Throttle' after eight words. This prevents data loss from other DMA devices which may also share the Qbus.

The SDC-RQD11-SC interleaves address references with bursts of data during DMA. Because the starting memory address is asserted only once every sixteen data words so data throughput is almost doubled.

DMA protocol consists of three phases :-

1. Bus Mastership Acquisition Phase. The SDC-RQD11-SC requests control of the bus. The processor arbitrates the request then initiates the transfer of bus mastership.

2. Data Transfer Phase. The processor provides the controller with the following information utilising MSCP - block number on the disk, the number of bytes to transfer, and address in main memory, and if the operation is a read or write.

3. Bus Mastership Relinquish Phase. Bus mastership is relinquished after completing or aborting the data transfer cycle.

For a detailed description of the Qbus the appropriate DEC manual should be consulted.