

SDC-RQD11-EC
MSCP/ESDI
Winchester Disk Controller
Manual

Copyright © Sigma Information Systems

3407 E. La Palma Ave.

Anaheim, CA 92806

MA400740 - REV B, July 1988

Notes

Table of Contents

| | | |
|-----------|--|-----------|
| 1. | General Information..... | 1 |
| 1.1 | Overview..... | 1 |
| 1.2 | Specifications..... | 4 |
| 2. | Installation..... | 7 |
| 2.1 | Switch Settings..... | 7 |
| 2.2 | RS232 Maintenance Terminal Adaptor..... | 11 |
| 2.3 | Connecting Multiple Drives..... | 11 |
| 2.4 | Front Panel Connections..... | 12 |
| 2.5 | On-Board Bootstrap..... | 13 |
| 3. | WOMBAT Utilities..... | 15 |
| 3.1 | Invoking WOMBAT..... | 15 |
| 3.2 | WOMBAT Menu Options..... | 18 |
| 3.3 | Master Menu Options..... | 19 |
| 3.4 | Disk Structure Menu Options..... | 21 |
| 3.5 | Disk Test Menu Options..... | 26 |
| 3.6 | Bad Block Management Menu Options..... | 29 |
| 3.7 | Shadow Options Menu Option..... | 31 |
| 3.8 | WOMBAT Disk Structure..... | 34 |
| 3.9 | Removable Drives..... | 34 |
| 3.10 | Error Recovery Procedures..... | 35 |
| 3.11 | Drive shadowing..... | 35 |
| 3.12 | Computing Sectors Per Track..... | 36 |
| 3.13 | WOMBAT Error Messages..... | 37 |
| 3.14 | WOMBAT Self-Diagnostics..... | 38 |
| 4. | MSCP Programming..... | 39 |
| 4.1 | Overview of MSCP..... | 39 |
| 4.2 | Controller Communications..... | 40 |
| 4.3 | Message Transmission..... | 43 |
| 4.4 | Data Transmission..... | 43 |
| 4.5 | Initialization..... | 44 |
| 4.6 | Registers..... | 45 |
| 4.7 | MSCP Commands..... | 46 |
| 4.8 | Error Handling..... | 47 |
| 4.9 | Fatal Controller Error..... | 47 |
| 5. | Operating Systems..... | 49 |
| 5.1 | Operating Systems Overview..... | 49 |
| 5.2 | RT-11 Operating System..... | 50 |
| 5.3 | RSTS/E Operating Systems (V8.0 and above)..... | 53 |
| 5.4 | RSX-11M Operating Systems (V4.0 and above)..... | 53 |
| 5.5 | RSX-11M-PLUS Operating Systems (V2.1 and above)..... | 56 |
| 5.6 | MicroVAX/MicroVMS Operating System..... | 58 |
| 5.7 | Autoconfigure..... | 60 |

| | | |
|-----------|---|-----------|
| 6. | Cache Operation | 67 |
| 6.1 | Disk Cache | 67 |
| 6.2 | Read Look-ahead | 67 |
| 6.3 | Cache Allocation | 68 |
| 6.4 | Cache Usage | 68 |
| 6.5 | Cache Assignment Algorithm | 68 |
| 6.6 | Cache Operation | 69 |
| 6.7 | Cache Disable | 69 |
| 6.8 | Early Write Notification | 69 |
| | | |
| 7. | ESDI Interface..... | 71 |
| 7.1 | ESDI Interface..... | 71 |
| 7.2 | Control and Data Cables | 71 |
| 7.3 | Command Data | 72 |
| | | |
| 8. | Q-bus Interface | 75 |
| 8.1 | Interrupts | 76 |
| 8.2 | Direct Memory Access..... | 76 |
| | | |
| | Appendix A. Active Front Panel Schematic | 79 |
| | | |
| | Appendix B. Fatal Controller Errors | 80 |

List of Tables

| | | |
|------------|---|----|
| Table 2-1: | Base (CSR) Address - Switch (SW2)..... | 7 |
| Table 2-2: | W-PROMs and CSR Addresses..... | 9 |
| Table 2-3: | Automatic Bootstrap Select - Switch (SW2)..... | 10 |
| Table 2-4: | Drive Configurations..... | 10 |
| Table 2-5: | Interrupt Priority - Switch (SW1)..... | 11 |
| Table 2-6: | RS232 Maintenance Terminal Adaptor..... | 11 |
| Table 2-7: | Passive 10-Pin Front Panel / Maintenance Connector..... | 12 |
| Table 2-8: | CSR Addresses for Bootstrap..... | 14 |
| Table 3-1: | WOMBAT Initialization Procedure..... | 16 |
| Table 3-2: | SDC-RQD11-EC CSR Addresses..... | 16 |
| Table 4-1: | Command Ring Code Descriptions..... | 41 |
| Table 4-2: | Word Envelope Contents..... | 42 |
| Table 4-3: | Initialization Parameters..... | 44 |
| Table 4-4: | Initialization Words..... | 45 |
| Table 4-5: | MSCP Commands..... | 46 |
| Table 4-6: | MSCP Status Code Messages..... | 48 |
| Table 5-1: | Device Names in DEC Operating Systems..... | 50 |
| Table 5-2: | SYSGEN Device Ranking..... | 61 |
| Table 5-3: | Device Registers and Word Boundaries..... | 61 |
| Table 5-4: | Floating Vector Address Device Priority Ranking..... | 62 |
| Table 5-4: | Floating Vector Address Device Priority Ranking..... | 63 |
| Table 5-5: | CSR and Vector Address Example..... | 64 |
| Table 5-6: | Floating CSR Address Assignment..... | 65 |
| Table 7-1: | ESDI Control Cable (J1) Signals..... | 71 |
| Table 7-2: | ESDI Interface Data Cable (J2) Signals..... | 72 |
| Table 7-3: | Command Data Word Structure..... | 72 |
| Table 7-4: | Command (CMD) Data Definition..... | 73 |

List of Figures

| | | |
|-------------|-----------------------------------|----|
| Figure 2-1: | Factory Switch Settings..... | 8 |
| Figure 4-1: | Memory Communications Format..... | 40 |
| Figure 4-2: | Descriptor Format | 40 |
| Figure 4-3: | Message Envelope Format..... | 41 |

1. General Information

1.1 Overview

The SCD-RQD11-EC is a high performance quad height interface to ESDI compatible disk drives. The SCD-RQD11-EC can support up to four ESDI drives and features a one megabyte cache memory, 24 megabit per second throughput, command queuing, overlapped seeks, drive shadowing, and implements DEC's Mass Storage Control Protocol (MSCP).

The SCD-RQD11-EC 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 Q-bus.

The ESDI Interface

The ESDI or Enhanced Small Device Interface is a low cost, high performance interface suitable for smaller high performance hard disk drives.

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.

Four ESDI drives at 3 Megabytes per second

The SCD-RQD11-EC controller will support up to four ESDI interface drives. All four drives can have different storage capacities, speed and data rates. This allows the user to match drive characteristics with applications.

No Sector Interleaving

The SCD-RQD11-EC 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 inter-

face and at maximum speed over the Q-bus without incurring data late errors. Disk and Q-bus transfers are performed simultaneously to minimize access times.

Read Look-ahead

The SCD-RQD11-EC allows the user to program the controller to perform 'look-ahead' reads in anticipation of data requests.

Virtual Units

The SCD-RQD11-EC 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.

Block mode DMA and DMA Throttle

With Block Mode DMA, the SCD-RQD11-EC interleaves address references with bursts of data - almost doubling Q-bus throughput. The SCD-RQD11-EC fully conforms with Q-bus Block Mode DMA protocol. With non block mode memory, the SCD-RQD11-EC 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 Q-bus after 8 words to prevent data loss from other DMA devices.

Drive Shadowing

The SCD-RQD11-EC offers the user the option of drive shadowing. Data integrity is further improved by writing the same data to two drives simultaneously. In some 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 SCD-RQD11-EC 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 SCD-RQD11-EC uses a 48-bit ECC polynomial with an 11-bit correction span for error detection and correction. The SCD-RQD11-EC will try up to 10 times to correct an error before reporting the fault to the host system.

Dynamic Bad Block Replacement

The SCD-RQD11-EC 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 SCD-RQD11-EC 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. The user may also connect an optional front panel which provides on-line, and write protect switches plus drive status/activity LEDs.

MSCP Emulation

The SCD-RQD11-EC 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 SCD-RQD11-EC 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.

Q-bus interface

Originally introduced in 1975 by Digital Equipment Corporation to support the LSI-11 CPU range, the Q-bus architecture has evolved in speed and function to the point where it now outperforms most small computer bus systems. The SCD-RQD11-EC fully implements all current Q-bus enhancements, including block mode transfers, 4-level interrupt structure, 22 bit addressing, and supports all Q-bus processors.

On-Board WOMBAT Utilities

WOMBAT is an interactive formatting and diagnostic utility contained within the SCD-RQD11-EC 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 SCD-RQD11-EC 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 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, using defaults provided by the ESDI drive. 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 SCD-RQD11-EC 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 read from 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 SCD-RQD11-EC controller will spin-up ESDI drives in a sequential manner, providing they have the "motor control" option enabled. This is done to minimize start-up current surge.

1.2 Specifications

| | |
|------------------------|---|
| Bus Interface (Q-bus): | Including MicroVAX II, MicroVAX 3x000, LSI-11/2, -23, -53, -73, -83 |
| 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: | Switch selectable LSI-11 MicroVAX 7772150 2000146 17760334 200000DC 17760354 200000EC 17760340 200000E0 17760444 200000E4 17760450 200000E8 |
| Vector Interrupt: | Programmable |
| Interrupt Level: | Switch selectable. 4, 5, 6 or 7 |
| On-board Bootstrap: | Switch selectable. Disable or enable at 17773000 or 17771000 |
| Power Requirements: | +5VDC @ 3.5A typical |
| Dimensions: | Standard quad board |
| LED Indicators: | |
| Red: | Fatal Error |
| Green: | Access in Progress |
| TTL Outputs: | Disk access in progress |
| RS-232 Output: | Data transmitted to terminal 9600 baud |
| RS-232 Input: | Data received from terminal 9600 baud |
| Drive Support: | 1 to 4 |
| Interface: | ESDI (Enhanced Small Device Interface) Hard and soft sectored drives are supported |
| 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) |

Notes

2. Installation

2.1 Switch Settings

The SDC-RQD11-EC can be configured by selecting the appropriate switch combination. The switch positions for base (CSR) address, automatic bootstrap select, interrupt priority, and CPU type are given below. Note that there is no interrupt vector switch as this is set automatically by the operating system software. The locations and factory configurations for switches are shown in Figure 2-1.

Base (CSR) Address

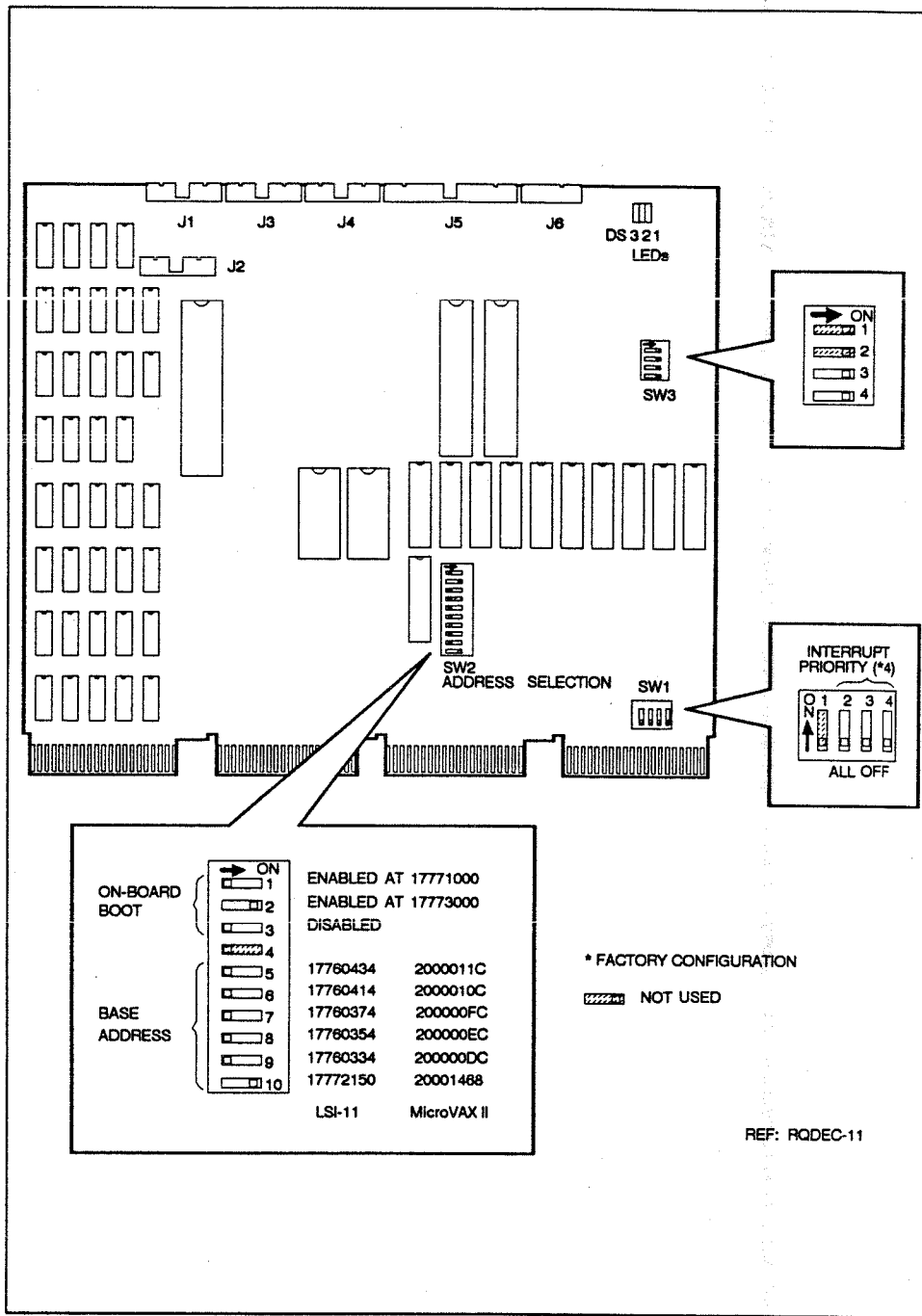
The base CSR address should be selected according to the rules set out in Chapter 5 - Operating Systems. It is important to locate the controller correctly to enable the operating system to identify the device and its type. The following is a table of all the possible CSR addresses for LSI-11 and MicroVAX systems.

Table 2-1:
Base (CSR)
Address - Switch
(SW2)

| *SW2 STATUS | ADDRESS | |
|-------------|----------|-------------|
| | LSI-11 | MicroVAX II |
| SW2-10 - ON | 17772150 | 20001468 |
| SW2-9 - ON | 17760334 | 200000DC |
| SW2-8 - ON | 17760354 | 200000EC |
| SW2-7 - ON | 17760374 | 200000FC |
| SW2-6 - ON | 17760414 | 2000010C |
| SW2-5 - ON | 17760434 | 2000011C |

*NOTE - Only one of the above six switches must be on at any one time.

Figure 2-1:
Factory Switch
Settings



Several different W-PROMs are available when configuring an SDC-RQD11-EC controller into a MicroVAX as the third or higher MSCP controller. The address selected by the MicroVAX depends on the overall configuration and can be determined by using the CONFIGURE option of the SYSGEN program. This allows the user to tell the MicroVAX what is intended to be installed, and in turn tells the user what the CSR (and vector) addresses must be. For the third or higher MSCP controller the number of possibilities far exceeds the 6 possible addresses allowed by the standard W-PROM (W1). Table 2-2 lists the W-PROMs and their respective CSR addresses.

**Table 2-2:
W-PROMs and
CSR Addresses**

| W-PROM NUMBER | PART NUMBER | ADDRESS NUMBER | ADDRESS | |
|------------------|----------------|-------------------|---------|------|
| | | | OCTAL | HEX |
| W1 | 940008-0401 | A1 | 172150 | F468 |
| | | A2 | 160334 | E0DC |
| | | A3 | 160354 | E0EC |
| | | A4 | 160374 | E0FC |
| | | A5 | 160414 | E10C |
| | | A6 | 160434 | E11C |
| W2 | 940008-0402 | A1 | 172150 | F468 |
| | | A2 | 160334 | E0DC |
| | | A3 | 160340 | E0E0 |
| | | A4 | 160344 | E0E4 |
| | | A5 | 160350 | E0E8 |
| | | A6 | 160354 | E0EC |
| W3 | 940008-0403 | A1 | 172150 | F468 |
| | | A2 | 160334 | E0DC |
| | | A3 | 160340 | E0E0 |
| | | A4 | 160344 | E0E4 |
| | | A5 | 160354 | E0EC |
| | | A6 | 160360 | E0F0 |
| W4 | 940008-0404 | A1 | 160360 | E0F0 |
| | | A2 | 160364 | E0F4 |
| | | A3 | 160370 | E0F8 |
| | | A4 | 160374 | E0FC |
| | | A5 | 160100 | E100 |
| | | A6 | 160104 | E104 |

W1 is the standard PROM shipped. It works for nearly all configurations where the SDC-RQD11-EC is the first or second MSCP controller.

W2 allows the controller to be 1st, 2nd, 3rd, 4th, 5th, or 6th controller in the configuration, provided there are no DZV11 MUXes. If there are DZV11 MUXes, then it will allow the controller to be the 1st or second only.

W3 is much the same as W2 except that it cannot be the 4th controller if there are no DZV11's. It can be the 3rd if there are DZV11's.

W4 allows the controller to be the 7th through 12th if there are no DZV11's installed.

**Table 2-3:
Automatic
Bootstrap Select -
Switch (SW2)**

| *SW2 STATUS | BOOTSTRAP FUNCTION |
|-------------|--|
| SW2-3 - ON | On board bootstrap disabled |
| SW2-2 - ON | On board bootstrap enabled at address 17773000 |
| SW2-1 - ON | On board bootstrap enabled at address 17771000 |

*NOTE - Only one of the above three switches must be on at any one time. Switch 4 is unused.

Drive Configurations

Switch SW3 defines the drive configurations as shown below:

**Table 2-4:
Drive
Configurations**

| POSITION 2 | POSITION 3 | POSITION 4 | SW3 FUNCTION |
|------------|------------|------------|---------------------------------|
| On | On | On | All drives fixed media |
| On | Off | On | Drive 0 removable |
| On | On | Off | Drives 0 & 1 removable |
| On | Off | Off | Drives 0, 1 & 2 removable |
| Off | On | On | Drives 0, 1, 2 & 3 removable |
| Off | Off | Off | WOMBAT only - via on-board port |

Position 1 is unused and should be left ON

Interrupt Priority

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. Table 2-5 shows the interrupt priority.

Table 2-5:
Interrupt Priority -
Switch (SW1)

| SW1-2 | SW1-3 | SW1-4 | PRIORITY |
|-------|-------|-------|----------|
| ON | ON | ON | 4 |
| ON | ON | OFF | 5 |
| ON | OFF | ON | 6 |
| OFF | OFF | ON | 7 |

Note - SW1-1 is unused.

2.2 RS232 Maintenance Terminal Adaptor

The optional RS232 maintenance terminal adaptor allows the controller to be connected to an ASCII terminal. It consists of a 10-pin socket on a 10-conductor cable that is terminated with either a male DB25 connector (P/N 700756-00XX) or a female DB25 connector (P/N 700756-01XX).

The communication format is :

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

Note that if a VT220 terminal is used the communication format must be set-up for space parity rather than no parity.

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 2-6:
RS232
Maintenance
Terminal Adaptor

| J6 Pins | DB25S Pins | Function |
|---------|------------|--------------|
| 7 | 7 | RS232 Enable |
| 8 | 2 | RS232 Input |
| 3 | 3 | RS232 Output |
| 4 | 7 | Ground |

2.3 Connecting Multiple Drives

The SDC-RQD11-EC supports up to four ESDI drives each of which can be of different size, speed and data rate. The procedure for connecting multiple drives is simple. Data cables are connected in radial fashion to connectors on the SDC-RQD11-EC. There is no particular order of connection. The control cable is daisy chained between each drive with the last drive being terminated.

Pre-configured data and control cable kits for one to four drives can be supplied to minimize difficulties when installing the SDC-RQD11-EC or when adding additional drives.

Cables should be checked for correct installation by ensuring that Pin 1 on the connector aligns with Pin 1 on the cable. Pin 1 is generally indicated by the arrow mark on the connector shroud. If there is no shroud Pin 1 is the top right pin with the connectors facing you. Pin 1 on the cable is usually indicated by a colored stripe on the appropriate cable.

Correct drive termination is also necessary. The disk drive technical literature should be consulted for the appropriate termination procedure.

2.4 Front Panel Connections

If required, a front panel can be connected to J6, the front panel/maintenance connector. There are three possible options for the front panel - an active panel, a passive panel or no panel. The active front panel provides write protection for all drives, an online/offline button and drive ready indicators. The passive panel provides write protection and access light for four drives. (See Table 2-6).

**Table 2-7:
Passive 10-Pin
Front Panel /
Maintenance
Connector**

| PIN NUMBER | MAINTENANCE FUNCTION | FRONT PANEL FUNCTION |
|------------|----------------------|------------------------------|
| 3 | RS232 Output | Activity Light. |
| 6 | | Drive 0 Access Light |
| 5 | | Drive 1 Access Light |
| 8 | RS232 Input | Drive 0 Write Protect Input. |
| 7 | RS232 Enable | Drive 1 Write Protect Input. |
| 10 | | Drive 2 Write Protect Input. |
| 2,4,9 | Ground | Ground |

The correct panel-type must be selected in the WOMBAT Disk Structure Menu (Section 3.4). Appendix A contains the schematic for the active front panel.

NOTE: The schematic below is provided so that customers can build an active front panel. The digital circuitry must be separately powered. An active front panel PCBA is available from Sigma.

2.5 On-Board Bootstrap

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

1. On initialization location 773000 responds to the CPU fetch instruction with a BR . + 2 (400) instruction to address 773002 where the CPU loops on a BR . instruction (777).
2. A "jump to zero" instruction (JMP @ #0 (137, 000)) is loaded into locations zero and two.
3. The controller program forces the CPU to start executing at location zero by changing the contents of location 773002 to CLR#PC (5007).
4. Bootstrap code is loaded into host memory at location 2000 and the rest of memory is cleared.
5. The controller changes location 2 from zero to 2000 changing the JMP @ #0 to JMP @ #2000, starting the execution of the loaded bootstrap program and the following message is typed:
BOOT Vx.x.
 where x.x is the firmware revision level.
6. The controller boot program allows approximately 2 seconds for the operator to strike any key on the keyboard. If no key is struck the boot types:

Booting from DU0:

The bootstrap program reads in that device's boot block starting at location zero.

7. 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, C, D, E or F after W, WOMBAT on controller A, B, C, D, E or F will be invoked. For example, WB will invoke WOMBAT on controller B.

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

> [DEVICE] -- [Controller Number] -- [Unit Number] -- [:] such that:
 --> DU-[A, B, C, D, E, F]-[0-7]----[:]

| |
|------------------------------|
| -> DL-----[0-7]-- |
| -> DY-----[0-1]-- |
| -> W-----[A, B, C]----- |
| --> MS-----[A, B, C, D]----- |

8. 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)

9. The selected controller is commanded to read the boot block (block zero) from the specified device into the host memory and then waits to be initialized. The host computer commences execution of the instructions in the boot block. The devices "assumed" by the SDC-RQD11-EC bootstrap are detailed in Table 2-7.

**Table 2-8:
CSR Addresses
for Bootstrap**

| DEVICE | ADDRESS |
|---------|----------|
| CONSOLE | 1777560 |
| 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: | 1777170 |

3. WOMBAT Utilities

3.1 Invoking 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 by any of the following methods:

- a. By selecting the 'W' option during the system bootstrap operation if the SDC-RQD11-EC 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 (LSI-11) or 254 (MicroVAX II) 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-EC Maintenance Connector and entering 260 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-EC Maintenance Connector and configuring the switches on the PCB to automatically run WOMBAT. (CPU independent.)

Runs WOMBAT 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 2-3). 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 initialization 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.

- e. By depositing 272 into the 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
Procedure**

| ACTION | | CONDITIONS | CONTROL |
|---------------|--------------|-------------|---------------------------------------|
| *OCTAL | *HEX | | |
| 000250 | 00AB | 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 |
| | SW2-3 ON | CPU | Aux Terminal on Maintenance Connector |
| | SW3-3,-4 OFF | Independent | |

*These codes are deposited into the 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-EC.

**Table 3-2:
SDC-RQD11-EC
CSR Addresses**

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

LSI-11 CPU

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

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 Q-bus access to memory) |
| D/L | 20088008 | 80000002 | (set-up the appropriate Q-bus map entry) |
| D/W | 2000xxxx | AC | (ask WOMBAT to load the communications program into memory) |
| S | 400 | | (start the program) |

where xxxx is defined as follows:

| PROM | ADDRESS SELECTION | | | | | |
|------|-------------------|------|------|------|------|------|
| | A1 | A2 | A3 | A4 | A5 | A6 |
| W2 | 1468 | 00DC | 00E0 | 00E4 | 00E8 | 00EC |
| W3 | 1468 | 00DC | 00E0 | 00E4 | 00EC | 00F0 |
| W4 | 00F0 | 00F4 | 00F8 | 00FC | 0100 | 0104 |

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-EC

WOMBAT Version: 2.x

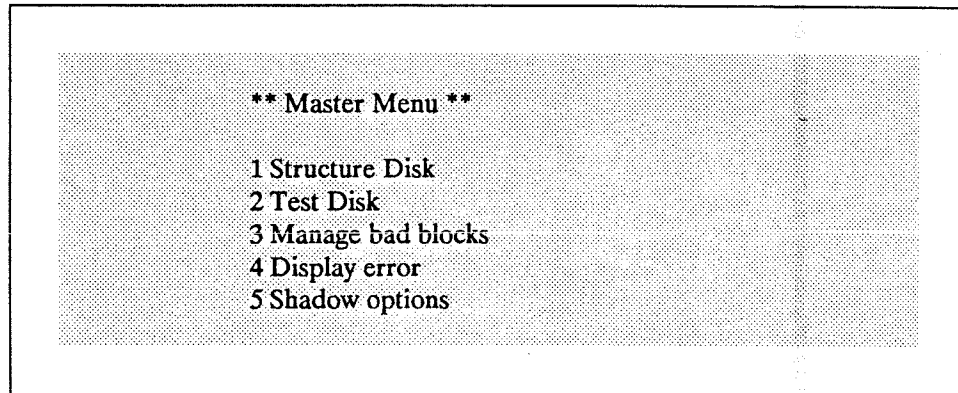
where x is the
firmware revision level.

| 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.



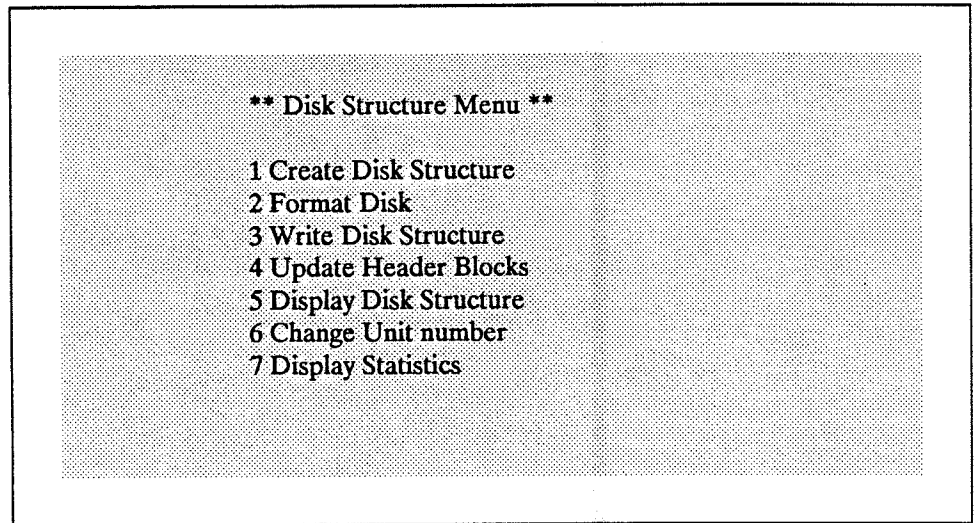
Select an option by typing the option number followed by RETURN. Options 1 through 3 and 5 will provide sub menus while option 4 displays the last controller detected fatal error. To return to the master menu from a sub menu type RETURN.

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.

3.3 Master Menu Options

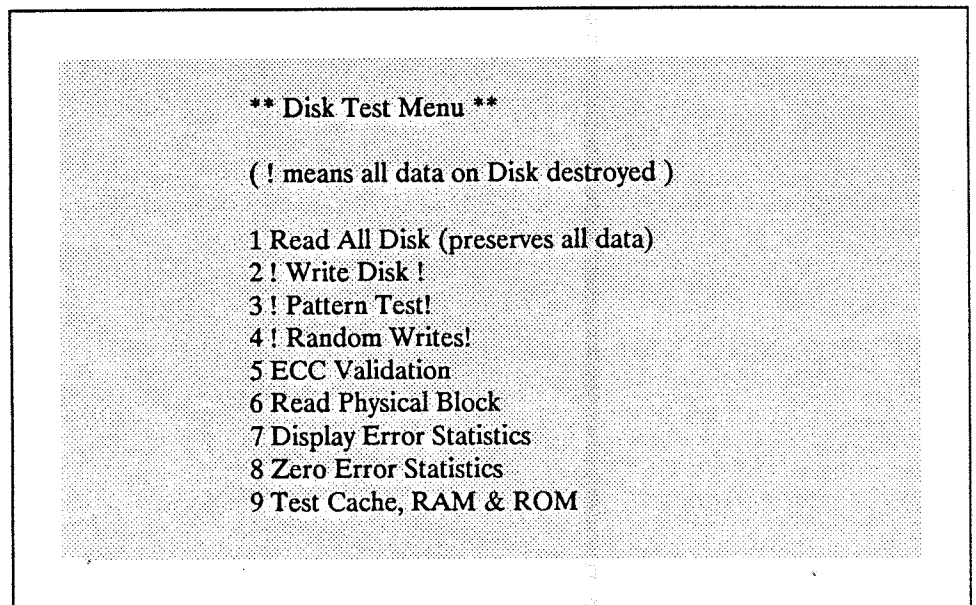
Option 1 - Structure Disk

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



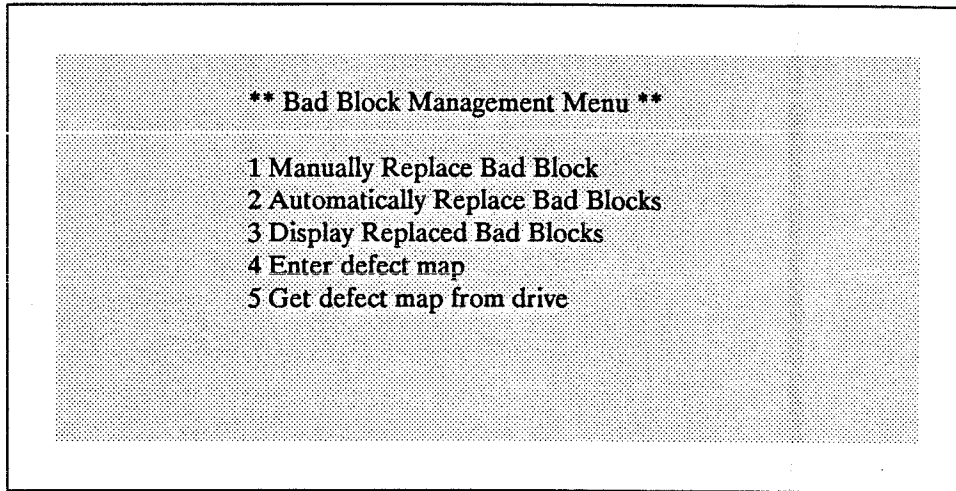
Option 2 - Test Disk

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



Option 3 - Manage Bad Blocks

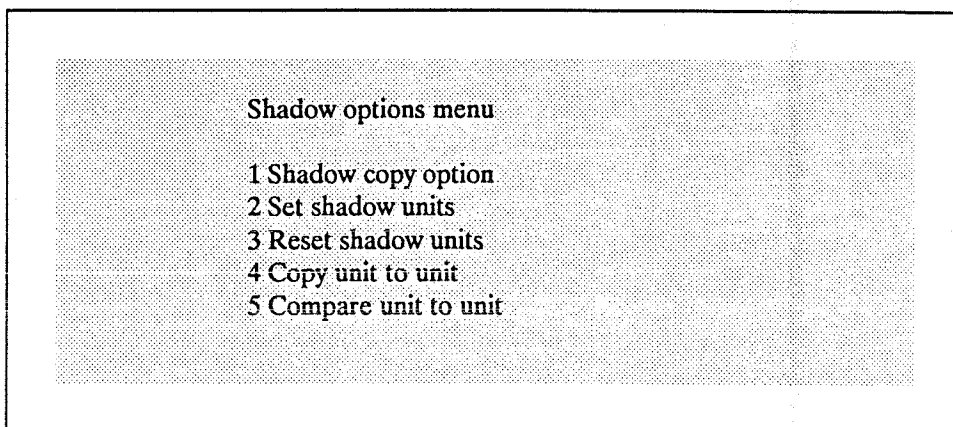
Selecting this option causes a sub menu to be displayed as follows:

**Option 4 - Display Error**

Selecting this option causes WOMBAT to display a message which explains the most recent controller detected fatal error. The occurrence 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:



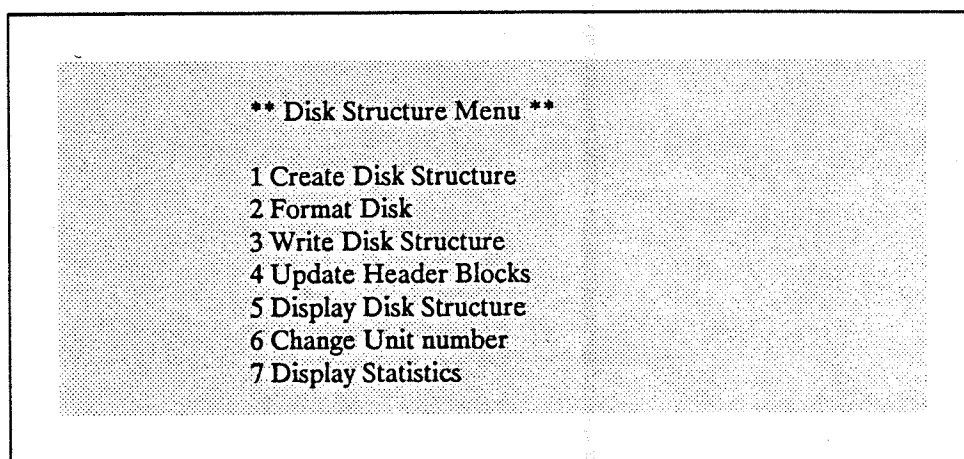
Setting Up a New Disk

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.

3.4 Disk Structure Menu Options



Option 1 - Create Disk Structure

The Create Disk Structure option must be performed when a new disk is connected to the SDC-RQD11-EC. 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: The number of cylinders on the drive. There is normally no need to change this value. Note that this will be one less than the number of cylinders specified by the manufacturer - this is to preserve the manufacturer recorded defect list on the last cylinder. You may override this option and recover the last cylinder, but the manufacturers defect list will be destroyed during formatting.

Heads: The number of heads on the drive is displayed. There is normally no need to change this value.

Full sectors/track: Hard sectored drives report the correct value here. For soft sectored drives, the required value will have to be computed and entered. See section 3.12 - Computing 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 either:

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 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.

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 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 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. 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 "on-line" command is issued. WOMBAT defaults this field to zero. To change this enter the desired 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 confirmation: 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.
- Front Panel Type:** This option allows the correct selection of front panel type. See 2.4 Front Panel Connections.
- 0 - None
 - 1 - Passive
 - 2 - Active

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.

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.

3.5 Disk Test Menu Options

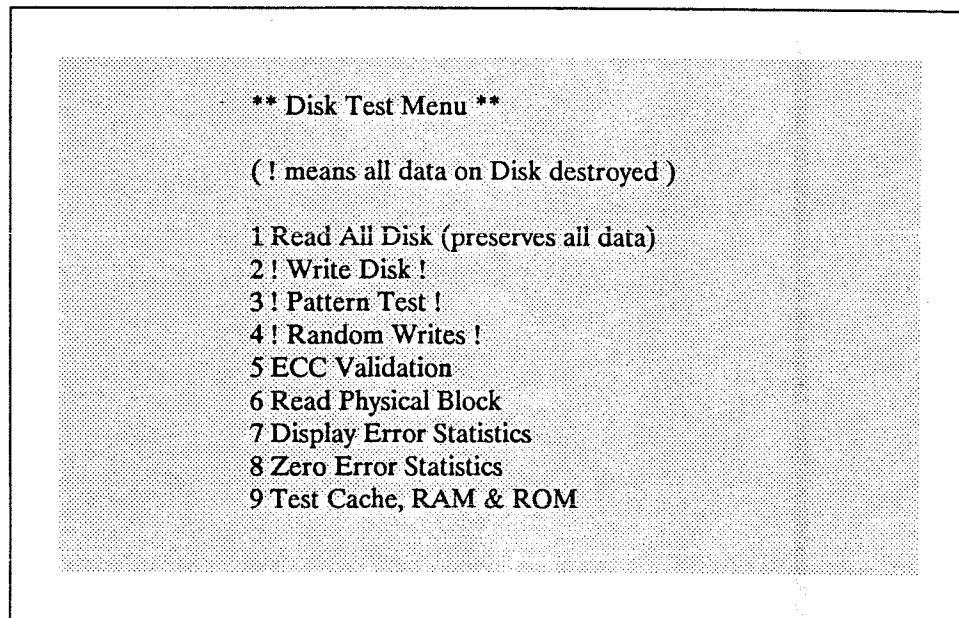
A disk can be tested after it has been formatted and before the structure is written to it. Testing does not overwrite the HDR or RCT blocks. The disk structure must be written to the disk before bad blocks can be replaced.

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

1. If an RS232 serial port terminal is attached to the controller, press BREAK.
2. 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 Q-bus 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.



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.

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-EC.

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 @.

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.
```

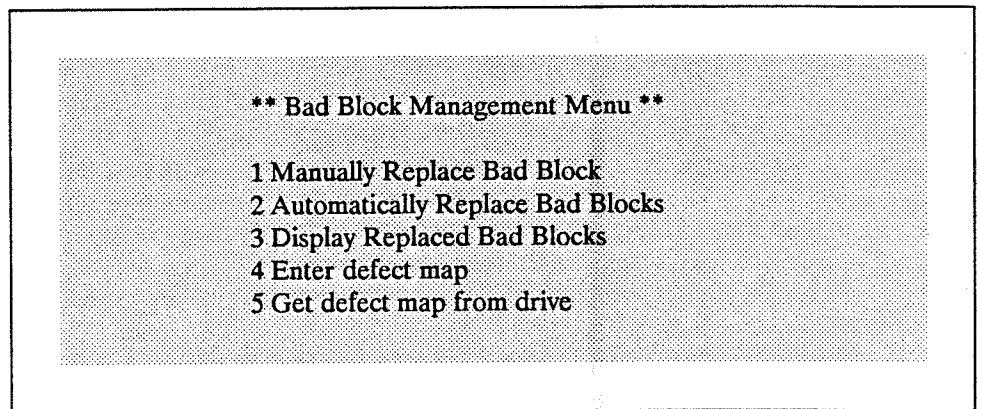
Option 8 - Zero Error Statistics

Zeroes the error statistics table & redisplay 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



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.

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 in decimal

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, RETURN or CTRL-C if no more.

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.

Option 5 - Get Defect Map from Drive

This option tells the controller to find then read the defect map information written on the drive by the manufacturer. It will then list the defective blocks by head number and ask for confirmation of replacement. It is important that the displayed data be checked, as it may have been corrupted by previous formatting/overwriting. WOMBAT does do some checks, but relies partially on the operator checking the validity of the list, in particular, the fields underlined below.

There should be a printed copy of the defect map supplied with the drive and this should be used for checking.

ESDI defect map for drive n

Soft/Hard sectored drive, xxx bytes/sector

Defect map read error, head x -Displayed if the defect map is unusable.

Then, for each head on the drive:

Defect map read: Date - mm-dd-yy (Fujitsu mm-dd-yy)

NOTE

The date code in brackets (Fujitsu mm-dd-yy) is displayed because early model Fujitsu drives recorded the date incorrectly.

The date code should match the date on the printed Defect List supplied with the drive.

Head x

Defect list for head x

Cyl. xxx, bytes past index xxx, length xxx

This is displayed for each error.

The Defect Map for each head has the head number recorded. If WOMBAT detects a mismatch it reports it and asks if you wish to proceed when all the defects have been displayed,

Do you wish to proceed with replacement for head x?

Only reply "Y" if the list appears correct. A defect at cylinder 20,000, 30,000 bytes past the index, and of length 0 is a possible (but patently wrong) display.

The defects are again displayed as they are replaced :

Cyl. xxx, bytes past ind. xxx, length xxx: block xxx, byte xxx

When this is complete, the message

Defect map replacement complete for head x

is displayed, and the data for the next head processed.

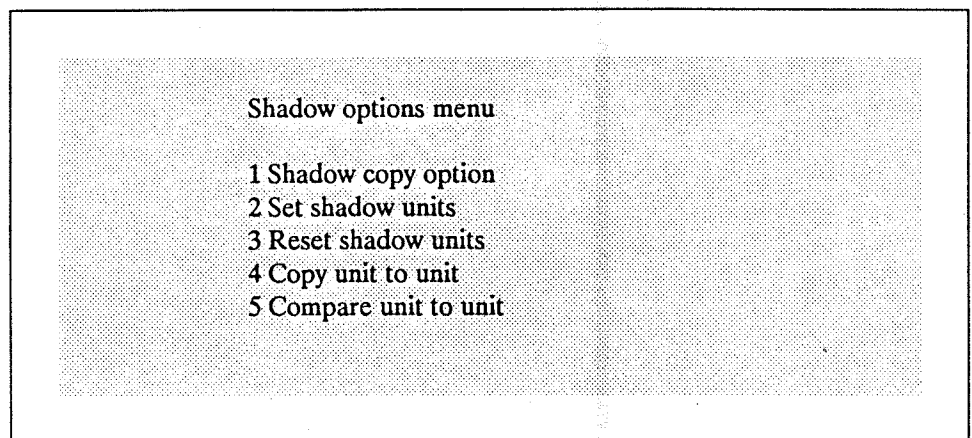
3.7 Shadow Options Menu Option

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.

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.



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 brought 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. |

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.

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.

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 Removable Drives

The SDC-RQD11-EC supports removable drives. SW3 defines which drives are removable, and they may only be drives 0, 1, 2, or 3. (See section 2-1.) The SDC-RQD11-EC makes certain changes to the way the removable drives operate. The important change is that you may 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.

3.10 Error Recovery Procedures

During normal operation, the SDC-RQD11-EC checks every disk transfer for errors. When an I/O error is detected, the SDC-RQD11-EC 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 reseek 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.

3.11 Drive shadowing

The intent of this feature is to provide a continuous automatic backup of important data. The normal case would be 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, then 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 will only offer 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 duplicating the garbage. 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.

If the primary does fail, then you will want to recover the data from the shadow. To do this WOMBAT must be run. First, 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. You determine the number of bytes per sector by the following calculation:

$$\begin{aligned} \text{Bytes per sector} &= K + (\text{PLO} \times 2) \\ \text{where:} \quad K &= 566 \text{ for hard sectored drives} \\ &= 557 \text{ for soft sectored drives} \end{aligned}$$

PLO is the number of bytes of PLO sync required by the drive. Obtain this figure from the drive manual.

e.g. Maxtor EXT4125, PLO sync field is 26 bytes

$$\begin{aligned} \text{bytes/sector} &= 557 + (26 \times 2) \\ &= 609 \text{ bytes per sector} \end{aligned}$$

This drive has 20808 bytes/track minimum

$$\begin{aligned} \text{sectors/track} &= 20808 / 609 \\ &= 34.17 \end{aligned}$$

so choose 34 sectors per track

e.g. Hitachi DK512 PLO sync field is 11 bytes

$$\begin{aligned} \text{bytes/sector} &= 566 + (11 \times 2) \\ &= 588 \text{ bytes per sector} \end{aligned}$$

This drive has 20832 bytes/track minimum

$$\begin{aligned} \text{sectors/track} &= 20832 / 588 \\ &= 35.42 \end{aligned}$$

so choose 35 sectors per track

3.13 WOMBAT Error Messages

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

| | |
|---------------------------------|--|
| 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 occurred 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. |
| 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 Appendix B, Fatal Controller Errors.

4. MSCP Programming

4.1 Overview of MSCP

Mass storage control protocol (MSCP) is a message-oriented set of rules by which the SDC-RQD11-EC 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.

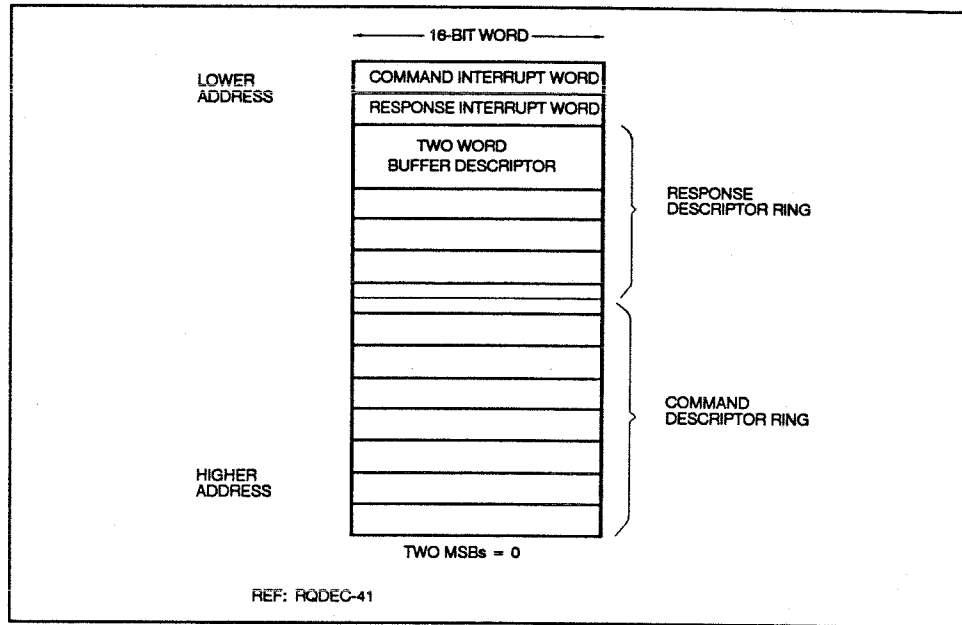
Disk drive parameters are transparent to both the host and controller resident layers of MSCP. The disk drive passes factors such as disk geometry, storage capacity or error retry counts to the disk controller on system start-up.

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.

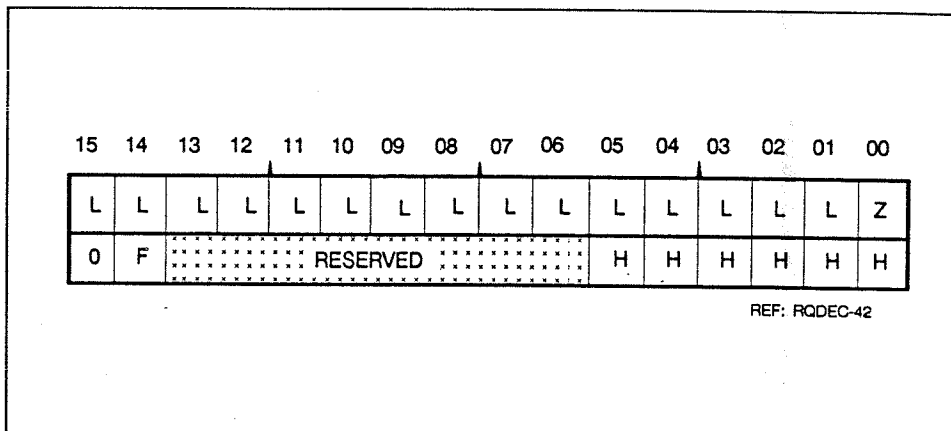
Figure 4-1:
Memory Communications Format



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



**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.

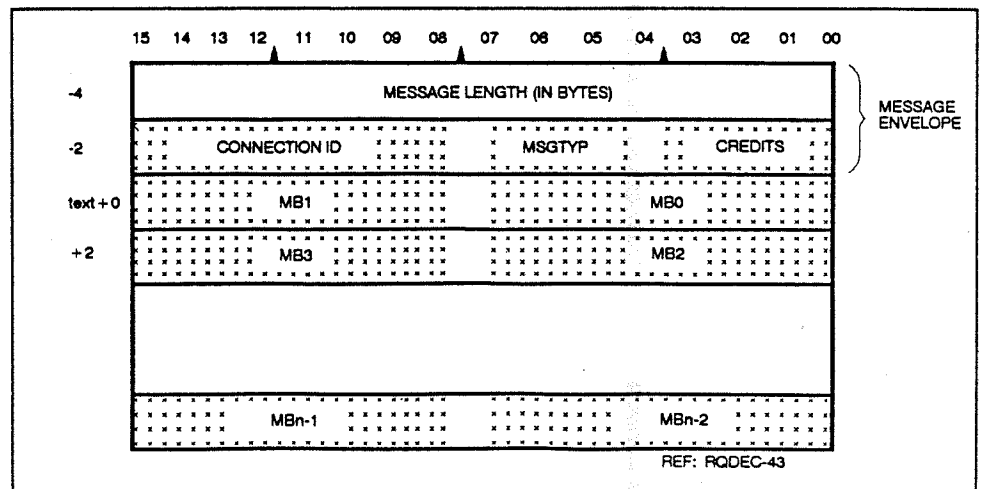
The controller always sets F=1 when returning a descriptor to the host, so if a host wishes to override ring transition interrupts it must always clear F when passing ownership of a descriptor to the controller.

O 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**



**Table 4-2:
Word Envelope
Contents**

| WORD | ENVELOPE CONTENTS | | | | | | | | |
|--------|---|--------|---------------------------------|--------|-------------------------|--------|------------------|--------|-------------------|
| 0 | <p data-bbox="639 323 919 348">Message length, in bytes.</p> <p data-bbox="639 390 1289 449">For commands, this length is equal to the size of the command (in bytes), beginning with [text + 0].</p> <p data-bbox="639 491 1295 705">For responses, the host sets the length equal to the size of the response buffer (in bytes), beginning with +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.</p> <p data-bbox="639 747 1299 905">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.</p> | | | | | | | | |
| 1 | <p data-bbox="639 940 802 966">Connection Id</p> <p data-bbox="639 1008 1295 1062">Identifies the connection serving as a source of, or destination for, the message in question.</p> | | | | | | | | |
| 2 | <p data-bbox="639 1104 802 1129">Message Type</p> <p data-bbox="639 1171 1234 1226">The following response ring message types are implemented:</p> <table data-bbox="639 1268 1166 1394"> <tr> <td data-bbox="639 1268 769 1293">MSGMNT</td> <td data-bbox="802 1268 1166 1293">Maintenance packet (diagnostic)</td> </tr> <tr> <td data-bbox="639 1297 769 1323">MSGCRD</td> <td data-bbox="802 1297 1062 1323">Credit notice (ignored)</td> </tr> <tr> <td data-bbox="639 1327 769 1352">MSGDAT</td> <td data-bbox="802 1327 997 1352">Datagram packet.</td> </tr> <tr> <td data-bbox="639 1356 769 1381">MSGSEQ</td> <td data-bbox="802 1356 997 1381">Sequential packet</td> </tr> </table> | MSGMNT | Maintenance packet (diagnostic) | MSGCRD | Credit notice (ignored) | MSGDAT | Datagram packet. | MSGSEQ | Sequential packet |
| MSGMNT | Maintenance packet (diagnostic) | | | | | | | | |
| MSGCRD | Credit notice (ignored) | | | | | | | | |
| MSGDAT | Datagram packet. | | | | | | | | |
| MSGSEQ | Sequential packet | | | | | | | | |
| 3 | <p data-bbox="639 1430 769 1455">Credit field</p> <p data-bbox="639 1497 1289 1656">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.</p> | | | | | | | | |

4.3 Message Transmission

Command Transmission

When the ownership bit (O) 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 (O) 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

This forces the controller to poll the command if it was not already accessing the command ring.

When the ownership bit (O) 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 (O) 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 Q-bus master for data transfer
- Failure to become Q-bus master for interrupt
- Failure to access I/O page registers or communication area
- Q-bus 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 Q-bus 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 Q-bus 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-EC 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-EC responds to the 16 bit initialization words as set out in Table 4-4.

Table 4-4:
Initialization Words

| NUMBER | | FUNCTION | PROCESSOR |
|--------|------|----------|----------------------------|
| octal | hex | | |
| 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:
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 effect. |
| 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.

Running WOMBAT and selecting the Display Error option will give an appropriate error message. A description of WOMBAT Error Messages maybe found in Appendix B.

**Table 4-6:
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-EC 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-EC controller found some field in the command to be in error. |
| Success | The command was successfully completed. |
| Unit Available | The SDC-RQD11-EC controller is not on line, but it can accept an On Line command from the host. |
| Unit Offline | The SDC-RQD11-EC 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. |

5. Operating Systems

The following discussion is intended to supplement DEC operating system resources and aims to aid the user of the SDC-RQD11-EC 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 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 next.

MSCP Devices

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

Q-bus Addresses

The standard Q-bus 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, Q-bus 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 Q-bus 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 Q-bus 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 Q-bus 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 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:

$$\text{SET DUn UNIT=y PART=x PORT=z}$$
 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.

$$\begin{array}{r} \text{(a)} \quad 245,412 \\ \text{-----} \\ \quad 65,535 \end{array} = 3.74 \text{ (4 logical units)}$$

$$\begin{array}{r} \text{(b)} \quad 204,800 \\ \text{-----} \\ \quad 65,535 \end{array} = 3.12 \text{ (4 logical units)}$$

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=3 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 Q-bus 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 Q-bus 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 Dispatch 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 Q-bus 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 Q-bus 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 overridden. 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, Q-bus 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 Q-bus 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 Q-bus 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.

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-EC, 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-11M-Plus supports autoconfigure and will detect the first controller located at the standard Q-bus address. Additional controllers must be installed manually.

Installing a Single Controller

A single controller is installed at the standard Q-bus 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-EC 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?
[O 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 MicroVAX/MicroVMS Operating System

The first SDC-RQD11-EC 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 Q-bus 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 Q-bus 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 Q-bus and Interrupt vector addresses. This procedure requires system manager privileges.

1. Log in 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 Q-bus 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 Q-bus 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-EC 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 Q-bus 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 Q-bus 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-EC by selecting the correct pin configuration on the PCB.

5. If an address other than that selected for the SDC-RQD11-EC 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 SYSS\$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, SYSS\$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 | XXXXX00, XXXXX20, XXXXX40, XXXXX60 |
| 9 thru 16 | XXXXX00, XXXXX40 |

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 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 |

continued on next page...

**Table 5-4:
Floating Vector
Address Device
Priority Ranking**

| Rank | Device | Number of Vectors | Octal Modulus |
|------|------------------|-------------------|---------------|
| 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 |
| 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 Q-bus 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-EC) | | 772334 ¹ |
| ---- | UDA50 (SDC-RQD11-EC) | | 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

6. Cache Operation

6.1 Disk Cache

The SDC-RQD11-EC 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-EC offers at least an 80% improvement in access times by reducing the number of disk accesses required. The SDC-RQD11-EC cache is implemented as one megabyte of dynamic RAM.

6.2 Read Look-ahead

The SDC-RQD11-EC 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-EC 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 maximize the probability of finding the target data over a range of sequential and non-sequential reference patterns while minimizing 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 Q-bus transfers are performed simultaneously.

6.5 Cache Assignment Algorithm

The SDC-RQD11-EC 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 describes the cache operation algorithm:

Read:

Examine cache for data required.

If all data in cache

Transfer data from cache to Q-bus

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 Q-bus to cache and flag as valid

Perform write

Unlock cache

6.7 Cache Disable

The SDC-RQD11-EC 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-EC 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.

NOTE: 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

7. ESDI Interface

7.1 ESDI Interface

The Enhanced Small Device Interface is an industry standard developed by the ESDI Committee to provide a low cost interface suitable for smaller high performance memory devices. The ESDI interface consists of a 34-pin control cable and a 20-pin data cable.

7.2 Control and Data Cables

The control cable allows for a daisy chain connection of up to four drives with the last drive being terminated. The data cable must be attached in radial fashion. The maximum cable length is 3 meters (9.8 feet). Tables 7-1 and 7-2 provide control and data cable details.

Table 7-1:
ESDI Control
Cable (J1) Signals

| SIGNAL NAME | SIGNAL PIN | GROUND PIN |
|---|---------------|---------------|
| -HEAD SELECT 2 ³ | 2 | 1 |
| -HEAD SELECT 2 ² | 4 | 3 |
| -WRITE GATE | 6 | 5 |
| -CONFIG/-STATUS DATA | 8 | 7 |
| -TRANSFER ACK | 10 | 9 |
| -ATTENTION | 12 | 11 |
| -HEAD SELECT 2 ⁰ | 14 | 13 |
| -SECTOR/-BYTE CLOCK/-ADDRESS MARK FOUND | 16 | 15 |
| -HEAD SELECT 2 ^{1*} | 18 | 17 |
| -INDEX | 20 | 19 |
| -READY | 22 | 21 |
| -TRANSFER REQ | 24 | 23 |
| -DRIVE SELECT 1 | 26 | 25 |
| -DRIVE SELECT 2 | 28 | 27 |
| -DRIVE SELECT 3 | 30 | 29 |
| -READ GATE | 32 | 31 |
| -COMMAND DATA | 34 | 33 |

*Unused but terminated or held false. Ground all odd pins

Table 7-2:
ESDI Interface
Data Cable (J2)
Signals

| SIGNAL NAME | SIGNAL PIN | GROUND PIN |
|---|------------|------------|
| -DRIVE SELECTED | 1 | |
| -SECTOR/-BYTE CLOCK/-ADDRESS MARK FOUND | 2 | |
| -SEEK COMPLETE | 3 | |
| -ADDRESS MARK ENABLE | 4 | |
| -RESERVED FOR STEP MODE | 5 | 6 |
| + WRITE CLOCK | 7 | |
| -WRITE CLOCK | 8 | |
| -CARTRIDGE CHANGED | 9 | |
| + READ REF CLOCK | 10 | 12 |
| -READ REF CLOCK | 11 | 12 |
| +NRZ WRITE DATA | 13 | 15 |
| -NRZ WRITE DATA | 14 | 16 |
| +NRZ READ DATA | 17 | 19 |
| -NRZ READ DATA | 18 | 19 |
| -INDEX | 20 | |

The four head select lines $2^0 - 3$ allow selection of each individual read/write head with HEAD SELECT 0 being the least significant line. Heads are numbered 0 through 15. Head 0 is selected for removable media drives, Head 0 being selected when all HEAD SELECT lines are high (inactive). The SELECT HEAD GROUP command allows for the addressing of more than 16 heads. Head addressing is continuous from 0. Data can be written to or read from the disk when the WRITE GATE or READ GATE signals are active (low).

7.3 Command Data

COMMAND DATA consists of 16 information bits of serial data plus parity. The command data word structure is set out in Table 7-3 below. Flow control is through the handshake signals TRANSFER REQ and TRANSFER ACK with the MSB being transmitted first. The parity bit is odd.

Table 7-3:
Command Data
Word Structure

| MOST SIGNIFICANT BIT | | | | | | | | | | LEAST SIGNIFICANT BIT | | | | | | |
|----------------------|----|----|----|---------------|----|---|---|-----------|---|-----------------------|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | P |
| CMD FUNCTION | | | | CMD MODIFIER | | | | ALL ZEROS | | | | P | | | | |
| CMD FUNCTION | | | | CMD PARAMETER | | | | | | | | | | | P | |

BIT P: PARITY (ODD)

Command Data bits 15 through 12 in combination with the Command Modifier bits 11 through 8 define a variety of ESDI functions. The Command Modifier bits are appended to each of the command functions where applicable. Table 7-4 shows the meaning of the various bit combinations. Detailed explanations of these functions can be found in the ESDI Interface Standard documentation.

**Table 7-4:
Command (CMD)
Data Definition**

| CMD FUNCTION BIT | CMD FUNCTION DEFINITION | CMD MODIFIER APPLICABLE BITS 11-8 | CMD PARAMETER APPLICABLE BITS 11-0 | STATUS CONFIG. DATA RETURNED TO CONTROLLER |
|------------------------|---|--|---|---|
| 0 0 0 0 | SEEK | NO | YES | NO |
| 0 0 0 1 | RECALIBRATE | NO | NO | NO |
| 0 0 1 0 | REQUEST STATUS | YES | NO | YES |
| 0 0 1 1 | REQUEST CONFIG | YES | NO | YES |
| 0 1 0 0 | SELECT HEAD GROUP(OPTIONAL) | NO | YES | NO |
| 0 1 0 1 | CONTROL | YES | NO | NO |
| 0 1 1 0 | DATA STROBE OFFSET | YES | NO | NO |
| 0 1 1 1 | TRACK OFFSET | YES | NO | NO |
| 1 0 0 0 | INITIATE DIAGNOSTICS (OPTIONAL) (OPTIONAL) | NO | NO | NO |
| 1 0 1 0 | RESERVED | - | - | - |
| 1 0 1 1 | RESERVED | - | - | - |
| 1 1 0 0 | RESERVED | - | - | - |
| 1 0 0 1 | RESERVED | - | - | - |
| 1 1 1 0 | RESERVED | - | - | - |
| 1 1 1 1 | RESERVED | - | - | - |

Notes

8. Q-bus Interface

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

The Q-bus 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:00
- * Two multiplexed address/parity lines - BDAL:16
- * Four extended address lines - BDAL: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 synchronizing clock pulses by implementing handshaking and other control signals between I/O devices.

8.1 Interrupts

Interrupt priority for the SDC-RQD11-EC 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-EC 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.

8.2 Direct Memory Access

The SDC-RQD11-EC 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-EC has a four microsecond delay after every 16 words to service any pending interrupt or DMA requests from other devices. The SDC-RQD11-EC 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 Q-bus.

The SDC-RQD11-EC 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.

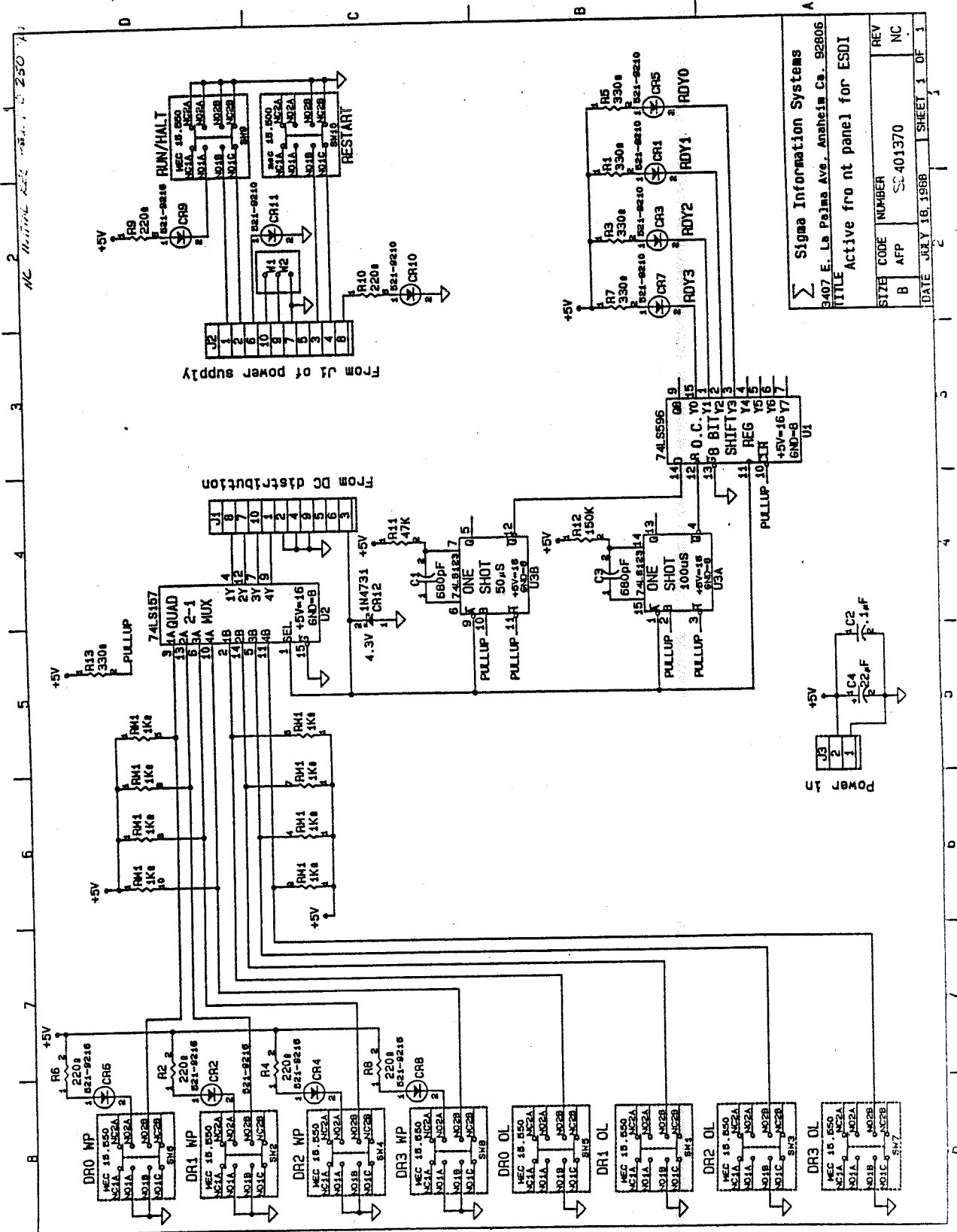
DMA protocol consists of three phases:

1. **Bus Mastership Acquisition Phase.** The SDC-RQD11-EC 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 utilizing 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 Q-bus the appropriate DEC manual should be consulted.

Notes

A. Active Front Panel Schematic



B. Fatal Controller Errors

| ERROR | | DESCRIPTION | ERROR | | DESCRIPTION |
|--------|------|-------------------------------|--------|------|----------------------------------|
| OCTAL | HEX | | OCTAL | HEX | |
| 1004xx | 81xx | No operation requested | 1200xx | A0xx | ECC fail |
| 1010xx | 82xx | Q-bus error | 1204xx | A1xx | Bad version |
| 1014xx | 83xx | Sector not found | 1210xx | A2xx | Command data parity |
| 1020xx | 84xx | Drive fault | 1214xx | A3xx | Interrupt fault |
| 1024xx | 85xx | Write protect | 1220xx | A4xx | Invalid command |
| 1030xx | 86xx | Drive timeout | 1224xx | A5xx | Write track off |
| 1034xx | 87xx | Data field error | 1230xx | A6xx | Vendor unique status |
| 1040xx | 88xx | Controller fault | 1234xx | A7xx | Write fault |
| 1044xx | 89xx | Block marked as bad | 1240xx | A8xx | Removable media change |
| 1050xx | 8Axx | Data late | 1244xx | A9xx | No command complete |
| 1054xx | 8Bxx | Forced error | 1250xx | AAxx | Drive insane |
| 1060xx | 8Cxx | Seek error | 1254xx | ABxx | RAM test fail |
| 1064xx | 8Dxx | Replacement table full | 1260xx | ACxx | ROM test fail |
| 1070xx | 8Exx | Replacement table read error | 1264xx | ADxx | Cache test fail |
| 1074xx | 8Fxx | Replacement table write error | 1270xx | AExx | Cache unlock error |
| | | | 1274xx | AFxx | Q-bus error - get host message |
| 1100xx | 90xx | Illegal sector | 1300xx | B0xx | Q-bus error - clear host rings |
| 1104xx | 91xx | Illegal block | 1304xx | B1xx | Buffers; Free buffer count error |
| 1110xx | 92xx | Illegal function | 1310xx | B2xx | Queue empty - unfork |
| 1114xx | 93xx | Illegal cylinder | 1314xx | B3xx | Queue empty - wait |
| 1120xx | 94xx | Not used | 1320xx | B4xx | Queue empty - pause |
| 1124xx | 95xx | Sector too short | 1324xx | B5xx | Adjust fail |
| 1130xx | 96xx | Illegal Q-bus address | 1330xx | B6xx | Cache parity |
| 1134xx | 97xx | Illegal SRAM address | | | |
| 1140xx | 98xx | Illegal CPU | | | |
| 1144xx | 99xx | No drive | | | |
| 1150xx | 9Axx | Unit table full | | | |
| 1154xx | 9Bxx | Drive under | | | |
| 1160xx | 9Cxx | Not used | | | |
| 1164xx | 9Dxx | No such unit | | | |
| 1170xx | 9Exx | Structure write | | | |
| 1174xx | 9Fxx | No replacement block | | | |