# SRQD 11-B

## WINCHESTER DISC CONTROLLER

## USER MANUAL

WEBSTER
COMPUTER
CORPORATION

# SRQD11-B Winchester Controller

## USER MANUAL

Version 1.0

WEBSTER COMPUTER CORPORATION



17 Malvern Street
Bayswater, Victoria 3153
Australia
(03) 729 8444 Telex 36251

First Floor 9-19 Ryde Road
Pymble, New South Wales 2073
Australia
(02) 498 4422 Fax: (02)499 2147

333 Cobalt Way, Suite 106
Sunnyvale, CA 94086
USA
(408) 749 1089 Telex: 172943

93 Carrington Street
Adelaide, South Australia 5000
Australia
(08) 224 3416 Telex: 89762

3464 Pacific Highway
Springwood, Queensland 4127
Australia
(07) 208 1533 Telex: 42438

# TABLE OF CONTENTS

## TABLES



## FIGURES

# CHAPTER ONE

# GENERAL DESCRIPTION

# CHAPTER 1
# SRQD11-B General Description

The Webster SRQD11-B is a dual height Qbus interface to ST506/ST412 compatible 5-1/4 inch Winchester disc drives. Implementing DEC's Mass Storage Control Protocol (MSCP), the SRQD11-B can flexibly couple up tp four disc drives of any size to all standard DEC operating systems without software modification. Comprehensive on-board interactive formatting and diagnostic firmware provides engineering support across the range of LSI-11, MicroVAX, and various non-DEC implementations of the Qbus.

## ST506/ST412 Interface

The Seagate ST506/ST412 interface has become the de facto industry standard for 5-1/4 inch Winchester disc drives. Inexpensive, reliable fixed media units spanning capacities of 2 to 138 megabytes and access times of 20 to 200 milliseconds are now available from 30 to 40 manufacturers, including:

| | | | |
|---|---|---|---|
| AMPEX | CDC | CMI | EVOTEK |
| FUJITSU | IMI | MAXTOR | MEMOREX |
| MICROPOLIS | MINISCRIBE | MITSUBISHI | OLIVETTI |
| PRIAM | RODIME | SEAGATE | SHUGART |
| TANDON | VERTEX | | |

Removable media drives with the ST506/ST412 standard interface are also supported.

## Qbus Interface

Originally introduced in 1975 by Digital Equipment Corporation to support the LSI-11 CPU range, the Qbus architecture has evolved in speed and functionality to the point where it now outperforms most small computer bus systems. The SRQD11-B fully implements all current Qbus enhancements, including block mode transfers and 22-bit addressing, and flexibly supports LSI-11/2, LSI-11/23, LSI-11/73, MicroVAX 1, MicroVAX 2, Motorola 68000 and National 32032 Qbus CPU designs. When used with the MicroVAX I, support is supplied within the controller for "scatter-gather" physical to virtual address translation.

**Block Mode DMA**

When used with block mode memory, the SRQD11-B almost doubles Qbus throughput by interleaving address references with bursts of data, fully conforming with Qbus Block Mode DMA protocol. With non block mode memory, the SRQD11-B reverts automatically to simple DMA.

**MSCP Universal Disc Architecture**

The Webster SRQD11-B communicates with the host through a simple register pair to memory resident 'command packets'. disc geometry factors such as sectors, heads, cylinders and disc capacity are invisible to the host computer. The Webster WQESD accepts 32 bit binary block numbers and converts them to physical disc addresses - allowing any size disc 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.

**Seek Optimisation and Overlap**

Command queueing and overlapped seeks between drives mean the SRQD11-B fulfills data requests in the shortest possible time. The SRQD11-B can queue up to 32 commands - dynamically computing the optimum order of execution to minimise disc head movement and enhance throughput in heavily loaded systems.

**Bad Block Replacement**

Disc surface defects are detected and flagged by the SRQD11-B during formatting and pattern testing, and redirected to a reserved user-specified area at the end of the disc. During all subsequent operations the SRQD11-B simulates fault-free media through transparent bad block replacement.

**Error Checking and Correction**

Error detection and correction guarantees data integrity. A 32-bit ECC polynomial with an 11 bit correction span is added to each disc sector during write operations and verified on each read operation. The SRQD11-B will try up to 10 times to correct an error before reporting the fault to the host system. Non-recoverable errors are virtually eliminated.

**No 'Data Late' Errors**

All data transfers are staged through a single sector buffer to ensure that 'data late' errors do not occur during periods of heavy Qbus traffic. Transfer rate is smoothly adjusted downwards until the bus again becomes available.

**Multi Drive Capability**

The SRQD11-B supports two physical drives with an option of four drives if the Webster SRQD11-X adaptor is used. These adaptors are available in the KSRQD kits listed in chapter 3. The SRQD11-B offers rapid data access times through seek optimisation and overlapping. Three controllers can be installed in one system allowing a total of 12 drives.

**On Board Bootstrap**

A miniature programming plug permits the SRQD11-B to optionally bootstrap at the standard Qbus addresses. There is no need for specially programmed external ROM modules. The SRQD11-B boot supports DU:, DL:, DY: and MS: devices, with multiple units.

**Controls and Indicators**

At the rear edge of the circuit board is a red LED indicator to signal Board Failure and a green LED to signal Access in Progress. An output is provided for off-board indication of Access, and inputs are provided for optional Write Protect switches for the disc drives.

**WOMBAT Utilities**

Webster Omnipotent Mass-storage Builder and Tester (WOMBAT) is an interactive formatting and diagnostic utility totally contained within the SRQD11-B firmware. An on-board serial connection is provided for communication with an ASCII terminal, (KSRQD/T RS232 Terminal Adaptor kit) permitting disc formatting and maintenance operations to be carried out with minimal additional hardware present. Alternatively, the SRQD11-B is able to load a simple console communication program into the host computer's memory.

No external software, media, or program loading device is required in maintenance of the SRQD11-B or its attached disc drives. WOMBAT is always available independently of the host CPU type or the operating system environment.

**WOMBAT Formatter**

WOMBAT initialises a fresh disc drive by writing sector addresses and zero data blocks through the entire recording surface. On invoking the formatter, the user is prompted at the terminal to supply parameters such as numbers of cylinders, heads and sectors, sector interleave factor, bad block replacement capacity, positioner step rate, and shipping zone cylinder. This data is stored twice in reserved areas of track zero during the format process, 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 disc drive.

**WOMBAT Self-diagnostics**

On bootstrap, the SRQD11-B is prompted by the operating system to enter a comprehensive series of controller and disc confidence tests. On failure, a red on board LED is illuminated to highlight the faulty module.

**WOMBAT Interactive Diagnostics**

Terminal oriented engineering utilities contained within the WOMBAT firmware include a down-line loader, an ODT-style firmware debugger, a continuous read/write/seek exerciser, a disc surface pattern tester, a bad block replacement routine and ECC logic verification.

**SRQD11-B Optional Accessory Kits**

The SRQD11-B can be modified to meet application needs or system expansion. Kits can be easily installed to enlarge drive capacity and provide RS232 terminal engineering and maintenance support.
The drive and control cable kits **KSRQD1**, **KSRQD2**, **KSRQD3**, and **KSRQD4** are configured for one to four drives accordingly. The **KSRQD/T** is an RS232 Terminal adaptor for engineering and maintenance support.

# CHAPTER TWO

# CONTROLLER SPECIFICATIONS

**CHAPTER 2**
**SRQD11-B Controller Specifications**

| | |
|---|---|
| Bus interface: | DEC Qbus |
| Transfer mode: | Block mode DMA |
| Memory address capacity: | 4 megabyte (22-bit) |
| Software supported: | All standard DEC Operating Systems |
| Command buffer capacity: | Up to 32 MSCP commands |
| Data buffer capacity: | 512 bytes (one sector) |
| CSR address: | (Plug selectable)<br>LSI-11 17772150<br>17760334<br>17760340<br><br>MicroVAX 1 303FF468<br>303FE0DC<br>FE0E0<br><br>MicroVAX 2 20001468<br>200000DC<br>200000E0 |
| Interrupt vector: | Software selectable |
| Interrupt priority: | Level 4 through 7 (plug selectable) |
| Qbus loads: | 1 DC, 2 AC |
| Drive interface: | Seagate ST506/ST516 |
| Bytes per sector: | 512 |
| Sectors per head: | 17 or 18 |
| Heads per cylinder: | 15 max |
| Cylinders per drive: | 2048 |
| Megabytes per drive: | 283 |
| Drives per controller: | 2 (4 with KSRQD Kits) |
| Access time overhead: | 3 mS (plus drive access time) |
| Single block transfer rate: | 625 Kbyte/sec |
| Full track transfer rate: | 260 Kbyte/sec (depends upon sector interleave factor) |
| Automatic bootstrap: | 17773000 (LSI11 only Plug Selectable) |
| Disc connectors: | 34-way control, 20-way data |
| Power requirement: | 5 volt 2.6 amp typical |
| Physical: | 226mm x 132mm dual height module |
| On-board LED indicators: | RED - board failure<br>GREEN - disc access in progress |
| Output: | Disc access in progress |
| TTL inputs: | Drive 0: Write-Protect switch<br>Drive 1: Write-Protect switch |
| Optional Accessories: | KSRQD/1, KSRQD/2, KSRQD/3, KSRQD/4 drive signal and control cable set for one to four drives respectively<br><br>KSRQD/T RS232 Maintenance Terminal adaptor |

# CHAPTER THREE

# INSTALLATION

# CHAPTER 3
# SRQD11-B Installation

## 3.1 ST506/ST412 Interface Connections

Both the drive signal and control cables consist of a flat or twisted pair cable joining a displacement-type flat cable socket at the controller end, and a displacement-type flat cable PC edge connector at the drive end. The drive controller cable and connector are 34-way types, and the signal cable and sockets are 20-way. Both cables have a maximum length of 6 metres.

**TABLE 3-1 34 WAY CONTROL CABLE CONNECTIONS**

| Function | Pin |
|---|---|
| SEEK COMPLETE | 8 |
| TRACK 0 | 10 |
| WRITE FAULT | 12 |
| INDEX | 20 |
| READY | 22 |
| WRITE GATE | 6 |
| STEP | 24 |
| DIRECTION IN | 34 |
| DRIVE SELECT 1 | 26 |
| DRIVE SELECT 2 | 28 |
| DRIVE SELECT 3 | 30 |
| DRIVE SELECT 4 | 32 |
| HEAD SELECT 0 | 14 |
| HEAD SELECT 1 | 18 |
| HEAD SELECT 2 | 4 |
| HEAD SELECT 3 | 2 |
| GROUND | 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33 |
| RESERVED | 16 |

**TABLE 3-2 20 WAY DATA CABLE CONNECTIONS**

| Function | Pin |
|---|---|
| +MFM WRITE DATA | 13 |
| -MFM WRITE DATA | 14 |
| +MFM READ DATA | 17 |
| -MFM READ DATA | 18 |
| GROUND | 2, 4, 6, 8, 10, 11, 12, 15, 16, 19, 20 |
| RESERVED | 1, 3, 5, 7, 9 |

## 3.2 SRQD11-B Settings and Adjustments

### Jumper Plug Settings

Miniature movable configuration plugs permit easy selection of base address, automatic bootstrap select, and interrupt priority.

### Base (CSR) Address
### Configuration Plug (A)

| | LSI-11 | MicroVAX 1 | MicroVAX 2 |
|---|---|---|---|
| 1st address | 17772150 | 303FF468 | 20001468 |
| 2nd address | 17760334 | 303FE0DC | 200000DC |
| 3rd address | 17760340 | 303FE0E0 | 200000E0 |

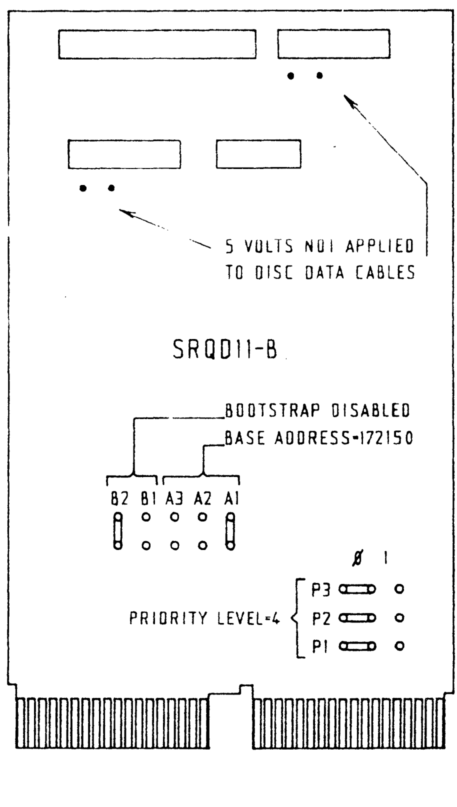
### Automatic Bootstrap Select
### Configuration Plug (B)

| | | |
|---|---|---|
| Position: | B1 | On board bootstrap enabled at address 17773000 |
| | B2 | On board bootstrap disabled |

### Interrupt Priority
### Configuration Plugs (P1-P3)

| | | | |
|---|---|---|---|
| P3,P2,P1: | 000 | Priority level: | 4 |
| | 001 | | 5 |
| | 010 | | 6 |
| | 110 | | 7 |

**FIGURE 3-1 SRQD11-B FACTORY JUMPER SETTINGS**



### 3.3 KSRQD/T RS232 Maintenance Terminal Adaptor

WOMBAT (see Chapter 5) can communicate either through the computer system console via a program loaded into main memory by WOMBAT; or the front panel/maintenance connector. The KSRQD/T RS232 maintenance terminal adaptor allows the controller to be connected to a standard computer terminal. It consists of a 10-way flat cable with a DB25S connector on one end and a 10-way displacement-type flat cable socket on the other. (See Table 3-1).

The communication format is :

ASCII RS232 9600 Baud, 7 Data Bits, 1 Stop Bit, no parity.

Note that if normal disc 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 3-3 KSRQD/T RS232 Maintenance Terminal Adaptor**

| J4 Pins | DB25S Pins | Function |
|---|---|---|
| 7 | 7 | RS232 Enable |
| 8 | 2 | RS232 Input |
| 3 | 3 | RS232 Output |
| 4 | 7 | Ground |

## 3.4 Front Panel Connections

If required, a front panel can be connected to J3, the front panel/maintenance connector. The functions supported are Drive 0: Write-protect switch, Drive 1: Write-protect Switch and Access Light. (See Table 3-4).

**TABLE 3-4 10-WAY FRONT PANEL/MAINTENANCE CONNECTOR**

| x | Front Panel Function | Maintenance Function |
|---|---|---|
| 7 | **Drive 1<br>Write Protect Switch Input. Connecting this input to ground will write-protect the drive. Has an on-board 10K ohm pull-up. | RS232 Enable. |
| 8 | **Drive 0<br>Write-protect Switch Input. Connecting this input to ground will write-protect the drive. Has an on-board 10K ohm pull-up. | RS232 Input. |
| 5 | Access Light<br>This output can be used to drive an Access Light. The levels are :<br>ACCESS : -5V thru 1.5K ohm<br>NO ACCESS : +3.5V @ 5mA max. | RS232 Output<br>RS232 Output Data from WOMBAT.<br>+3.5V to -5V RS232 compatible. |
| 2,4,5,6,9, | Ground | Ground |

** NB : If both switches are closed the controller assumes an KSRQD/T cable is connected and ignores the switch functions - that is the drive will be "on-line" and "write-enabled".

## 3.5 On-Board Bootstrap

For LSI-11 processors only, the controller may be set to provide an on "On Board Bootstrap". By setting a configuration plug the SRQD11-B will implement boot locations 773000 and 773002. These are set up containing :

```
773000/    BR .+1      ;  Branch to next instruction
773002/    BR .        ;  Branch to this instruction
                       ;  i.e. loop waiting for controller
                       ;  to set up boot.
```

These instructions are executed when the LSI-11 is powered-up causing the LSI-11 to loop. The SRQD11-B detects the execution of these instructions and initiates its boot cycle as follows:

1. Initialization code is loaded into host memory at location 1000.

2. A "jump to 1000" instruction (JMP @ #1000) is loaded into location zero and the following message is typed :-

   **SRQD11 Boot V4.3**

3. The controller changes the contents of location 773002 to "CLR PC".

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

   **Booting from DU:**

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

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

   **>**

   The operator may then key in a device (DU,DL,DY,MS) according to the following syntax rules :-

   **> [ DEVICE ] -- [ Controller Number ] -- [ Unit Number ] --[:]**

such that :-

```
         -->DU --- [A,B,C,]--- [0-7] ---- [:]
        ^                                  ^
         |->DL --------------- [0-7] --|
         |                              |
         |->DY --------------- [0-1] --|
         |                              |
         -->MS --- [A,B,C,D] ----------
```

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

   **"? - Device must be DU, DY,DL or MS"**

   **"Unit must be 0 - 7"** - For DU or DL

   **"Unit must be 0 or 1"** - For DY

   **"Controller must be A, B or C"** - For DL

   **"Controller must be A, B, C or D"** - For MS

   **"Boot failure"** - Device unavailable (or not ready if DL)

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

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

**CSR Addresses for SRQD11-B Bootstrap**

```
c$tty    =:    177560    ;console
c$du     =:    172150    ;DU:
c$dua    =:    172150    ;DUA:
c$dub    =:    160334    ;DUB:
c$duc    =:    160340    ;DUC:
c$ms     =:    172520    ;MS:
c$msa    =:    172520    ;MSA:
c$msb    =:    172524    ;MSB:
c$msc    =:    172530    ;MSC:
c$msd    =:    172534    ;MSD:
c$dl     =:    174400    ;DL:
c$dy     =:    177170    ;DY:
```

# CHAPTER FOUR

# SRQD11-B

# PROGRAMMING

# CHAPTER 4
# SRQD11-B Programming

## 4.1 Overview of MSCP

Mass storage control protocol (MSCP) is a message-oriented set of rules by which the SRQD11-B 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 class' driver and a 'communications port' server. The controller layer receives and processes commands which have been formed by the host layer.

The communications port servers handle 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 communication port server monitors all command transmission and response and in the event of port failure or error, initiates recovery procedures.

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

In addition to relieving the host-resident layer of disc-specific data, the disc controller and disc together provide the host with "clean" data.The disc drive handles some positioner errors entirely by itself and perform certain error-recovery operations under direction of the disc 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:

1. The header area containing interrupt identification words.

2. A variable-length section containing the response (receive) and command (send) lists, organised into ring buffers.

The following diagram shows the format of the memory communications area.

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

### Command and Response Rings

Command and response lists are each organised into 'rings' of 32 bit descriptors. The length of each ring is determined by the relative speeds with which the host and controller generate and process messages. The host sets the ring lengths at initialization time.

Descriptor Format:

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

| 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.<br><br>When the controller returns ownership to the host it sets F=1 to indicate that it has completed action on the descriptor.<br><br>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. The interrupt will occur only if this descriptor causes a ring transition and if transition interrupts were enabled during initialization.<br><br>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 follows:

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

<u>Word</u> <u>Envelope Contents</u>

0 Message length, in bytes.

For commands, this length is equal to the size of the command (in bytes), beginning with [text+0].

For responses, the host sets the length equal to the size of the response buffer (in bytes), beginning with <text+0>. Before actual transmission of a response, the controller reads the field length in the message envelope. If the controller's response is longer than the response buffer, the controller will fragment its response into as many response buffers as necessary.

The controller sets the resulting value into the message length field. The host must therefore keep re-initializing the value of this field for each proposed response. If a controller's responses are less than or equal to 60 bytes, then the controller need not check the size of the response slot.

1 Connection Id

Identifies the connection serving as a source of, or destination for, the message in question.

2 Message Type

The following response ring message types are implemented:

| | |
|---|---|
| MSGMNT | Maintenance packet (diagnostic) |
| MSGCRD | Credit notice (ignored) |
| MSGDAT | Datagram packet. |
| MSGSEQ | Sequential packet |

3 Credit field

Gives a credit value (usually one) associated with the message. This mask, in response packets, is added to the controller's credit field to give the number of commands-in-progress. So while Word 1 is always 1 for the command ring, this is not the case for response rings.

## 4.3 Message Transmission

### Command Transmission

When the ownership bit (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

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

## 4.4 Data Transmission

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

## 4.5 Initialization

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

Initialization Process

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

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

**Initialization Parameters**

| Word | Contents |
|---|---|
| 0 | Command and Response ring sizes, interrupt enable and vector.<br><br>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.<br><br>The host reads an echo of the ring lengths from SA, and then writes into SA the low-order portion of the ring base address. |

2 High order address of communications area, bits 0-14.

The interrupt vector address and the master interrupt arming signal are echoed in SA. The host then writes the high order portion of the ring base address to SA along with a signal that conditionally triggers an immediate test of the polling functions of the controller.

3 Burst transfer control, last failure flag, and the 'GO' bit.

The controller tests the ability of the Qbus to perform DMA transfers. If successful, the controller zeros the entire communications area, and then signals the host that initialization is complete.

## 4.6 Registers

The programmable registers contained on the SRQD11-B 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. While DEC's controller always performs an initialization when the IP register is written to, the SRQD11-B responds to the following initialization words :

| Word (octal) | Function |
|---|---|
| 250 | Call WOMBAT |
| 251 | Call DEBUG (if implemented) |
| 252 | Read block zero into host computer memory at location zero (simple boot procedure) |
| Anything else | Initialization |

**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

The following commands are supported by the SRQD11-B controller.

| Command | Function |
|---|---|
| Access | Reads data from the specified unit. |
| Abort | Guarantees that referenced MSCP command will complete within the controller timeout period. |
| Available | If specified unit is on-line, returns it to the unit-available state. If specified unit is currently in the unit-available state, this command has no affect. |
| Compare Host Data | Reads data from the disc 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 disc, 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 disc, from the host memory. |

## 4.8 Error Handling

High data integrity is achieved by the controller through a 32 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.

**MSCP Status Code Messages**

| | |
|---|---|
| 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 disc data to the host data. |
| Controller Error | The SRQD11-B 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 SRQD11-B controller found some field in the command to be in error. |
| Success | The command was successfully completed. |
| Unit Available | The SRQD11-B controller is not on line, but it can accept an On Line command from the host. |
| Unit Offline | The SRQD11-B 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. |

### 4.9 Fatal Controller Error

If a fatal error is detected when the controller is initialized, the error LED is lit, and the fatal error status set in the SA register.

| ERROR CODES (octal) | DESCRIPTION |
|---|---|
| 100004 | RAM test failure |
| 100005 | ROM checksum failure |
| 100011 | No drive |
| 100100 | Disc unformatted |
| 100101 | Disc unstructured |
| 100103 | No RCT table |
| 100103 | No FCE table |

A full description may be found under Section 5.4 : WOMBAT ERROR messages.

# CHAPTER FIVE

## WOMBAT

## UTILITIES

# CHAPTER 5
# SRQD11-B WOMBAT Utilities

Webster Omnipotent Mass Builder and Tester (WOMBAT) utilities provide a controller resident means of formatting, testing and maintaining the drive and controller sub-systems.

Interactive communication with WOMBAT may be achieved by connecting a 9600-baud terminal to the SRQD11-B as described in Chapter 3. When no terminal is connected directly to the SRQD11-B, WOMBAT will download a communication program into main memory which establishes a connection with the user console. This allows disc testing and diagnostics may be undertaken from the user console.

## 5.1 Starting up WOMBAT

WOMBAT is invoked by depositing a special data pattern in the SRQD11-B using Micro ODT. (Refer to the appropriate CPU manual for a detailed description of its ODT.) The procedures for invoking WOMBAT on LSI-11, MicroVAX I and MicroVAX II are given below. WOMBAT can be stopped by simply re-booting the system in all cases.

**NOTE** 'CSR' denotes one of the following addresses depending upon CPU and number of contollers installed. Each controller must have a separate address.

| | LSI-11 | MicroVAX I | MicroVAX II |
|---|---|---|---|
| 1st address | 17772150 | 303FF468 | 20001468 |
| 2nd address | 17760334 | 303FE0DC | 200000DC |
| 3rd address | 17760340 | 303FE0E0 | 200000E0 |

**LSI-11 CPU**

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 the bus reset that a micro ODT would cause) |

## MicroVAX I CPU

Halt the processor.

| | | | |
|---|---|---|---|
| **D/P/W** | **'CSR'** | **AC** | (ask WOMBAT to load the communications program into memory) |
| **S** | **400** | | (start the program) |

## MicroVAX II

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

When it halts :-

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

## 5.2 WOMBAT Menu Options

When WOMBAT is invoked it will display a version number and ask for the disc drive number on which to perform its operations. On input of the drive number, WOMBAT will immediately try to read the disc and assess if it has a valid format. If the disc is unformatted the keyboard bell is sounded and a message is printed under the drive select menu.

```
***WOMBAT Version 4.3***
 Drive 0:  [Ready    ]
 Drive 1:  [Not Ready]
 Select Drive   [0-1]:
 Disc unformatted
```

Upon the selection of the disc drive WOMBAT proceeds to the Master Menu from which the required options are selected.

***** Master Menu *****

1 **Structure Disc**
2 **Test Disc**
3 **Manage Bad Blocks**
4 **Initialize Controller**
5 **Position Head On Shipping Zone**

Select an option by typing 'option number' followed by a carriage return (RETURN). Options 4 and 5 are single function options, while accessing options 1, 2 or 3 will provide you with a further sub-menu of options as detailed below.To return to the Master Menu while in a sub-menu, enter RETURN at an option prompt.

*** Option 1 : Disc Structure Menu ***

1 **Create Disc Structure**
2 **Format Disc**
3 **Write Disc Structure**
4 **Update HDR blocks**
5 **Display Disc Structure**

*** Option 2 : Test Disc Menu ***
**(! Means all data on Disc destroyed)**

1 Read all Disc (preserves all data)
2 **! Write Disc !**
3 **! Pattern Test !**
4 **! Random Writes !**
5 **ECC Validation (preserves all data)**
6 **Display Error Statistics**
7 **Zero Error Statistics**

***** Option 3 : Bad Block Management Menu *****

1. **Manually Replace Bad Block**
2. **Automatically Replace Bad Block from Error Statistics**
3. **Display Replaced Bad Blocks**

***** Option 4 : Initialize Controller *****

This forces the controller to perform its initialization functions : RAM clear and test, ROM test, and checks on the disc and its structure. It then restarts WOMBAT, allowing the user to again specify the drive number.

**Warning : The Error Statistics Table is cleared.**

***** Option 5 : Position Head on Shipping Zone *****

When this option is selected, the head is positioned on the default shipping zone cylinder number which is 340 unless an alternate zone is provided by the manufacturer.(See drive data sheet.) This provides a safeguard for data areas on the disc.
When the head is positioned on the specified zone, the following message is displayed:

**Disc Heads now on SHIPPING zone!**
**This disc drive may now be safely removed.**

## DISC STRUCTURE MENU

### Create Disc Structure

When you connect a new disc to the SRQD11-B controller, this option MUST be the FIRST selected as you cannot do anything else until the disc structure is specified.

WOMBAT has a completely 'soft' disc structure. That is, it assumes nothing about the various disc parameters: cylinders, heads, and other pertinent factors - such as whether the disc is removable. This option allows these parameters to be specified.

When you attach a new disc to the controller, you must go through the process of setting up the disc structure, and then finally writing that structure on to the disc. You do this by:

1. Creating the disc structure.

2. Formatting the disc (the structure is still in the SRQD11-B memory).

3. Testing the disc, if you are unsure how reliable it is. At this stage you may still write to the disc as the structure is still in the SRQD11-B memory.

4. If the disc is not suspect and does not have errors on blocks zero or one, write the disc structure to it. If the disc has errors on blocks zero or one, you cannot attach and use it with the SRQD11-B controller.

5. If necessary, do bad block management.

When you have set up the disc structure, formatted, and written the structure to disc, you may let the host operating system use the disc.

## Example of a disc's Structure

The following example is representative of a particular 20 Mbyte disc. You would enter the correct figures relevant to your disc. All numeric values are decimals unless otherwise stated. Do not enter the decimal point to signify a decimal number as this is automatically returned by WOMBAT.

All figures shown in bold type are generated and displayed by WOMBAT.

**TABLE 5-1 SAMPLE WOMBAT DISC FORMAT**

```
-----------------------------------------------------------------
Size, Cylinders:                      [306.]?
Heads:                                [8.]?
Sectors per Track:[17-18]             [18.]?
Interleave Factor:[2-6]               [2.]?
Replacement Cylinders:                [6.]?
Head Step Rate:                       [0.]?
Seek Optimisation: 0-None, 1-Nearest, 2-Elevator, 3-Forward  [2.]?
Command Queue Size:[1-32]             [8.]?
Optimisation Strategy:[0-3]           [1.]?
Fairness Count:[1-255]                [25.]?
Shipping Zone Cylinder:               [340.]?
Media Type:(AAAnn)                    [WCC31.]?
Removable Media Y/N                   [Y]?
Serial Number:(Octal)                 [0]?
Unit Number:[0-255]                   [0]?

USR area: 43038 blocks



'RETRN' to continue:                  (Returns the Disc Structure Menu)

-----------------------------------------------------------------
```

## Field Descriptions

The drive's data sheet is required for reference as WOMBAT needs to know some of the specifications. It is important that the details are accurately transcribed to ensure that WOMBAT runs effectively. (Refer to Appendix A for a table of typical disc structure parameters used.) An existing or incorrect parameter can be altered by pressing RETURN until the field to be altered is displayed and then entering the new parameter.

| | |
|---|---|
| Size, Cylinders | The total (decimal) number of cylinders on the disc. You will find this information on the drive data sheet. |
| Heads | The number of read/write data heads (other than servo heads) that the drive has, as stated on the data sheet. |
| Sectors per Track | The number of 512-byte sectors (17 or 18). You can determine this figure from information available on the data sheet. If the rotational speed (rpm) is specified as 3,600 +/- 0.1% (or less) then enter 18. Some drives may specify 1.0%. In these cases it is mandatory to use 17 because of the risk of sector overlap. If a number other than 17 or 18 is entered, WOMBAT displays '?Invalid' and repeats the prompt. |
| Interleave Factor [2-6] | The SRQD11-B cannot directly transfer data between the Qbus memory and the disc. It must stage it via local RAM. Therefore an interleave factor of at least two is needed for the sectors on a track to get maximum transfer rate during a single revolution. The process involves the controller renaming the physical sector addresses. Some guidelines are: |
| Factor 2 | Suitable for fast memory with block mode capability, including DEC MSV11-PK, MSV11-PL, and Webster SMSV11-P memory. |
| Factor 3 | Necessary for slower, non block mode memory. |
| Factor 4/5/6 | The rest of these were provided in case some application code could benefit. |

CAUTION: You must match the interleave factor to the memory speed. A too low interleave factor will cause the disc to run much more slowly. If an invalid interleave factor is entered, WOMBAT will display '? **Invalid**' and repeat the prompt.

**Replacement Cylinders**

This allocates room for the bad block replacement table. Allow at least 6 cylinders. When a structure is created on a formatted disc, the allocation is made as follows:

2 tracks for RCT (Replacement Control Table)
1 track for WRK (Controller work area)
1 track for FCE (Forced error table)
remainder for BAD (Bad block replacement area)

If you do not allow enough space serious problems may occur.

**Head Step Rate**

Specify the rate at which the disc heads can step. Zero is default, and is used for all discs with buffered stepper motor or voice coil head positioners. Parameters 14 & 15 are ultra-high speed step rates are currently supported only by Maxtor drives, giving a significant performance improvement when used with them.

| Parameter | Step Rate |
|---|---|
| 0 | about 35 usecs |

Other valid parameters for non-buffered steppers are:

| | |
|---|---|
| 1 | 0.5 ms |
| 2 | 1.0 ms |
| 3 | 1.5 ms |
| 4 | 2.0 ms |
| 5 | 2.5 ms |
| 6 | 3.0 ms |
| 7 | 3.5 ms |
| 8 | 4.0 ms |
| 9 | 4.5 ms |
| 10 | 5.0 ms |
| 11 | 5.5 ms |
| 12 | 6.0 ms |
| 13 | 6.5 ms |
| 14 | 3.6 uS |
| 15 | 16.0 uS |

**Command Queue Size**

The MSCP protocol allows the controller to stack a number of commands; this parameter allows you to specify the size of the command stack.
Please realize that the larger the stack, the more overhead the disc controller incurs when it scans it; 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 acceptable to most operating systems.

Optimisation Strategy — The seek optimisation strategy can be either:

0-None
1-Nearest
2-Elevator
3-Forward

This is a user preference feature. WOMBAT displays '? Invalid' if the number entered is out of range, and repeats the prompt.

Explanation :

None — No optimisation done. First request found executed. Warning : This may not be the next sequential request.

Near — Nearest cylinder strategy selects the request that is closest to the current cylinder.

Elevator — This processes requests like an elevator - as it moves in one direction along the disc until it reaches the last request in that direction. This means that "Elevator" favours the centre of the disc, as it passes it twice as often as the periphery.

Forward — This processes requests from the lowest cylinder number to the highest; like "Elevator" except in only one direction. This is the generally recommended strategy for most purposes, and will yield an approximately 2:1 improvement in apparent performance in random access applications.

Note that optimisation is only effective if the host operating system supports multiple accesses. RT-11, TSX-Plus, and RSX-11M normally do not without provision of special device handlers.

Fairness Count — The fairness count relates to disc commands. A reasonable count for normal use would be around 25. If the number entered is not within the range 1 - 255, WOMBAT displays the message 'Number too large', and the cursor moves to the next line. Enter a valid number.
This count determines the number of times an I/O request will be passed over by seek optimisation before it is executed. 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 optimisation strategy is in effect.
This count has no effect if no optimisation is selected.

Shipping Zone Cylinder

WOMBAT is setup to use cylinder 340, being standard for a number of disc drives. Refer to the drive data sheet to ensure that this is appropriate for your type of drive.

Media Type

This field allows the media type to be specified. The MSCP protocol returns - as part of unit status when a "Get Unit Status" command is issued - a 5 character media type.
As a default, WOMBAT sets up "WCCnnn", where nnn is the size of the drive in megabytes. To change this, enter 1 to 3 alphabetic characters and 2 digits, e.g. RD52, to emulate DEC's 31 megabyte Winchester.
This field is displayed by some operating systems when you enquire about the type of drive. For example, RSX-11M-PLUS responds to a "DEV DU:" command with :
"DU0: Public Mounted Loaded Label = RSX11MPBL15 Type = WCC31"

Removable Media

This field reports to the operating system whether a removable cartridge Winchester disc has been installed. This is necessary to prevent the operating system from reporting an error when a removable disc is off-line.

Serial Number

The MSCP protocol returns - as part of its response when an "on-line" command is issued - a 32-bit volume serial number. WOMBAT defaults this field to zero.
To change this, enter, in octal, the desired serial number.
This field is used, for example, by RSX-11M-PLUS, when you initialize a disc with the "INI DU:" command. It sets up the volume serial number.

Unit Number

Some operating systems, notably RSX-11M-Plus, require unique unit numbers for drives, even if they are attached to

different controllers.
A "Unit Number" in the range 0-255 may be specified and the controller will then only respond to that unit number (i.e. DU1: if you answered "1"). This parameter may be changed at any time and the disc structure modified by invoking "Update HDR blocks".

USR Area — The size, in blocks, of the user area on the disc, as reported to the host operating system. The user area is the ONLY portion of the disc seen by the operating system.

WOMBAT will then compute but not report where various necessary areas on the disc are to go.

HDR — Header track, containing the disc structure on the first two sectors. Because the controller does not know the structure on the disc, the only known place is at the beginning. The physical block displays as -17 or -18, depending on the number of sectors per track, and is at cylinder 0, head 0, sector 0. The user area is offset by the number of sectors per track (17 or 18) and commences at cylinder 0, head 1, sector 0.

USR — User area starts on the second track of the disc - the start of the operating system visible area.

RCT — Replacement control table start.

WRK — One track of controller work area, for staging information while mapping out a bad sector in a track, etc.

FCE — Area to hold table for blocks which have been forced into error by the operating system. Data written with the forced error indicator set will be preserved until the next time the block is written. Forced errors are recognized by the controller as deliberate and never reported.

BAD — The remainder of the disc, used for replacements for bad blocks.

End Disc at Block — The end of the disc, as a physical block number.

## Format Disc

The disc structure is now set up in the SRQD11-B controller's local memory. Before the structure can be written out to disc, the disc must be formatted.

Formatting destroys ALL INFORMATION on the disc, and this includes previous bad block allocations recorded in the Replacement Control Table. Take note that if a disc is restructured and not reformatted, then previously allocated bad block numbers could create an error when the structure is written to disc. The error message '**Disc Error Writing Disc Structure at NNNNN**' would be displayed. NNNNN represents a bad block number allocated prior to the last formatting procedure.

When you select this option WOMBAT displays:

> ***** Warning: all information on this disc will be destroyed, including any bad block mapping.**
>
> **Is this what you want to do [Y/N]?**

Ensure that you do in fact want to format the disc and enter Y. If you enter N, the Disc Structure Menu is returned.

While the disc is being formatted WOMBAT displays the mapped cylinder counter which commences from zero and increments by 64, thus showing you when the formatting operation is nearing completion. The final cylinder number (the parameter you entered when structuring the disc) is not displayed, and when formatting is complete the message '* Disc Formatted *' is displayed and the disc Structure Menu is returned.

## Write Disc Structure

When you set up the disc structure, it is only recorded in the controller's local memory. This option writes the structure on the disc, where it is read by either the controller program or WOMBAT, to set up the disc parameters. Once this procedure is complete the host operating system may use the disc. If an override structure is written to disc and the disc has not been reformatted, an error could occur if there were blocks previously marked as bad. (Refer "BAD BLOCK MANAGEMENT MENU").

No information is displayed during the write process and when finished, the Disc Structure menu is returned.

**Update HDR Blocks**

This option allows you to just update the disc structure as recorded in the HDR blocks. If you, in the "Create Disc Structure" option, have ONLY altered any of the following parameters:

Command Queue Size
Optimisation Strategy
Fairness Count
Shipping Zone Cylinder
Media Type
Serial Number
Unit Number

You may update the HDR blocks to include these changed parameters without destroying any data on the disc. A good example of this is if you are testing various disc optimisation strategies.
WARNING : Do not alter any other parameter and try to use this option; you must "Write Disc Structure" to update all the disc structures in this case.

No information is displayed during the update process, and when finished, the Disc Structure Menu is returned.

**Display Disc Structure**

This is an optional sequence which may be used at any time to review parameters entered during a 'Create Disc Structure' procedure.

**TEST DISC MENU**

A disc 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.

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

1. If an ASCII 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 Disc Menu options are returned. If tests are run from an ASCII terminal attached to the controller, beware of system activity on the host computer as Qbus initializations will cause the disc 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. The codes are defined as follows:

BB Bad Block; block marked as bad.

ID Id error; sector header cannot be found on disc.

ECC Error Checking and Correction; will display whether the error is correctable or non-correctable. Data may still be read even if an error persists.

**Read all Disc**

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.

**Write Disc**

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

```
Block: 32040    ID
Pass:1. Errors: 1.
Pass:2. Errors: 1.
```

Displayed error count is cumulative until the test is terminated.

**Pattern Test**

This test writes a worst case pattern to each block along with the block number. It does one write and 10 read and compare passes, where it checks that it is reading the right block and that the data pattern is correct. This test reports any errors in the standard format as shown above.

**Random Writes**

As for Pattern test on USR area of disc, then performs 5000 random writes, reporting every 1000 writes for timing purposes. Finally it does a check of all data on the disc again. This test reports:

```
Initializing disc ...
 5000 Random Writes
1000.
2000.
```

**ECC Validation**

This test deliberately corrupts data in a block and then tests ECC correction. No user data on the disc is destroyed.

**Display Error Statistics**

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

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

**Zero Error Statistics**

Zeroes the error statistics table & redisplays Test Menu options.

## BAD BLOCK MANAGEMENT MENU

Bad blocks are replaced by marking them as bad on the disc (i.e. flagging them as "BAD" in the sector header), and recording the block number in the Replacement Control Table, pointing at a replacement block in the BAD area of the disc.

When the disc structure is written, a zeroed replacement table is written to the RCT. Therefore, any bad blocks on the disc will exist without a replacement block so that any testing of the disc will then report them as "BAD" blocks (BB).

Bad block management cannot be done until the structure has been written to the disc.

### Manually Replace Bad Block

This option prompts operator entry of an arbitrary block to be replaced as "bad".
If you enter a block number that is higher than the total number of blocks available, or enter an illegal non-numeric block number, then WOMBAT displays the error message '* Failed *'.

### Automatically Replace Bad Block from Error Statistics

This option replaces all bad blocks discovered during any or all of the three tests, subsequently displayed in the 'Display Error Scan Statistics' function. Any block with 10 or more errors will be marked as bad and replaced with a block in the BAD area.

### Display Replaced Bad Blocks

Displays all blocks in the Replacement Control Table and gives the total number of blocks replaced.

## 5.3 WOMBAT Error Messages

**No Drive**
Initialization tried to read the disc structure and failed. Check that the disc is connected correctly to the controller.

**Disc Unstructured**
Initialization can read the disc but cannot identify either copy of the data in the header blocks.

**Cannot read RCT Table**
Cannot read either copy of the RCT Table because of disc errors or the structure is not as expected.

**Format error at block**
An error was encountered while trying to format the disc at the specified block number. Since this is the first thing you do to a disc, it normally means that either the disc is not working at all, or the disc is not connected properly (e.g. wrong cables) to the controller. If the message occurs at other times the disc drive is probably very sick, or intermittent.

**No Disc Structure**
An attempt has been made to display a disc structure on a disc that has not been structured.

**Cannot replace this block**
The block number entered is not within the bounds of the user area of the disc.

**No RCT Table**
Initialization could read neither of the two copies it keeps of the RCT (Replacement Control Table). Though you will destroy all information about replaced bad blocks, try writing the disc structure again. If errors persist in the RCT area, you may have to move the RCT to another cylinder on the disc. If the writing of the disc structure is successful this time, you can recover any replaced bad blocks by just performing a read test, which will bring up any unreplaced bad blocks as hard (10 retries) BB (Bad Block) errors. You may then replace them by using the "Automatically Replace Bad Blocks From Error Statistics" option.

**No FCE Table**
Initialization cannot locate the "Forced Error Table". Comments for "No RCT Table" apply.

**RCT error**
WOMBAT found the answer invalid when it computed the address of the RCT on disc. Check and respecify the bad block replacement table size in cylinders.

**Disc error writing disc structure at: (block)**

For some reason, WOMBAT cannot write the structure details on to the disc. Test the disc thoroughly with the "Test disc" menu. If that works properly, move the RCT and try again. If it still fails, you may have a bad disc or controller. Try component swapping.

**Command Timeout**
When all commands are executed, a counter is started to stop them "hanging". This counter has expired. If this is the first time the disc has been used, is the disc connected properly? Are the cables correct? Is there power to the disc? If the disc has a select light, is it on? Try powering off and on. Try component (disc and controller) swapping.

**Fatal - Respecify Structure**
An error has been detected in the disc structure specified; check and re-enter.

## 5.4 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 disc drive and its structure

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

# CHAPTER SIX

# OPERATING SYSTEMS

# CHAPTER 6
# SRQD11-B Operating Systems

The following discussion is intended as a supplement to DEC operating system resources and aims to aid the user of the SRQD11-B in understanding how different operating systems integrate new devices. This information will help the user of the controller plan the installation and in choosing the appropriate bus addresses and interrupt vectors for the the disc subsystem.For a complete description the DEC system documentation should be consulted.

## 6.1 Operating Systems Overview

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

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

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

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

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

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

### MSCP Devices

The Webster SRQD11-B is an MSCP (Mass Storage Control Protocol) type device. All MSCP-type devices contain two registers that are visible to the Qbus I/O page. They are the Initialisation and Polling (IP) register and the Status and Address (SA) register.

## Qbus Addresses

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

## Vector Addresses

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

## Device Names

Table 6-1 is a list of device names for five operating systems. Two controller and device names are given to indicate the numbering scheme.

**TABLE 6-1 DEVICE NAMES IN DEC OPERATING SYSTEMS**

| OPERATING SYSTEM | CONTROLLER 1st: | 2nd: | DRIVE 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 |

## 6.2 RT-11 Operating System

### Installation of a Single Controller

A single controller is installed at the Qbus address of 17772150 (Octal) where RT-11 will find and then install the handler for that device. It is not necessary to run sysgen for a single controller. One of the pre-generated monitors provided with the RT-11 distribution kit can be used. To properly implement disc 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 the 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 disc partitioning is carried out consistently on each bootstrap or power-up.

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

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

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

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

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

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

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

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

| | | | |
|---|---|---|---|
| SET DU CSR | = | 17772150 | (DEFAULT) |
| SET DU CSR2 | = | 17772154 | |
| SET DU VECTOR | = | 154 | (DEFAULT) |
| SET DU VEC2 | = | 300 | |

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

**Disc Partitioning**

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 discs any physical disc can be partitioned into is calculated by dividing the selected block size of each logical disc into the total capacity of the the disc 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 discs.

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

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

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

**Sample Disc Partitioning Procedure**

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

(a) $\frac{245,412}{65,535} = 3.74$ ( 4 logical units )

(b) $\frac{204,800}{65,535} = 3.12$ ( 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=2 PORT=0
```

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

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

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

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

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

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

### Installing a single controller

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

### Installing multiple MSCP controllers

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

1. Invoke SYSGEN.

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

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

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

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

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

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

4. Indicate the number of devices that are installed.

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

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

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

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

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

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

**Vector and Qbus Addresses**

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

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

**Drive Configuration**

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

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

Count each RX50 drive as two drives, these contain two 5.25 inch floppy discs. 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 (Disc partitions) for the disc 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 minimise system overhead and is the recommended and default value.

6. The type of disc 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 SRQD11-B, indicate that there is one RD51 for each logical disc drive.

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

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

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

### Installing a Single Controller

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

### Installing Multiple Controllers

To add the SRQD11-B 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 disc 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 disc 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 discs. 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 (Disc partitions) for the disc 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 initialisation 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 minimise 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 minimise system overhead and is the recommended and default value.

## 6.6 MicroVMS Operating Systems

The first SRQD11-B controller is located at the standard bus address of 17772150 (octal) and the second in floating address space. The MicroVMS SYSGEN utility can determine the Qbus and interrupt vector addresses for any of the I/O devices installed on the bus. MicroVAX/MicroVMS must be running in order to use this utility. The Qbus and interrupt vector addresses can be determined manually if access to a running system is not possible.

### Using MicroVAX/MicroVMS SYSGEN

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

1. Login and run the SYSGEN utility.

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

   The SYSGEN> prompt indicates that the program is ready.

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

   SYSGEN> **SHOW/CONFIGURATION<return>**

   and get :-

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

* Indicates a floating vector or address.

Sysgen lists the devices already installed on the Qbus by logical name. Devices with floating bus and vector addresses should be noted if it is intended to re-install them with the SRQD11-B controller. Floating bus addresses will be larger than 760000 (octal). Floating interrupt vectors will be larger than 300 (octal).

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

SYSGEN> **CONFIGURE<return>**
DEVICE>

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

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

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

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

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

DEVICE> **^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 SRQD11-B by selecting the correct pin configuration on the PCB.

5. If an address other than that selected for the SRQD11-B by CONFIGURE command is desired, CONNECT statements must be entered into the SYSCONIF.COM file. SYSCONIF.COM can only be accessed through the system manager's account SYS$MANAGER. The correct syntax is given in the DEC MicroVMS SYSGEN documentation.

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

## 6.7 Autoconfigure

### Overview

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 6-2 ). The presence or absence of floating bus and interrupt vector address devices will affect the assignment of addresses to other floating vector devices.

**TABLE 6-2 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 6.6 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 6-3 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 6-3 DEVICE REGISTERS AND WORD BOUNDARIES**

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

## Vector Addresses and Autoconfiguration

Devices are assigned vector addresses in order of rank commencing at 300 (octal) up to 777 (octal). Extra devices of the same type are assigned consecutive vector addresses according to the number of vectors required and starting boundaries for each device type. Table 6.4 shows the order of assignment.

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

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

**TABLE 6-4 FLOATING VECTOR ADDRESS DEVICE PRIORITY RANKING**

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

continued on next page...

**TABLE 6-4 FLOATING VECTOR ADDRESS PRIORITY RANKING**

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

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

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

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

## System Configuration Example

An example of a system configuration is shown in Table 6-5. The configuration includes both fixed and floating addresses and vectors.

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

**TABLE 6-6 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 (SRQD11-B) | | 772154[1] |
| -----> | UDA50 (SRQD11-B) | | 760334 |
| | | 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

# CHAPTER SEVEN

# FUNCTIONAL DESCRIPTION

# CHAPTER 7
# SRQD11-B Functional Description

## GENERAL DESCRIPTION

The heart of the SRQD11-B is a microprogrammed sequencer combined with an 8-bit RALU, (Register and Arithmetic Logic Unit) to form a high speed processor which performs the following functions :-

1. Implement the MSCP register pair - the Initialization and Polling Register (IP) and the Status and Address Register (SA). Responds to the LSI-11, MicroVAX 1 and MicroVAX 2 CPUs' access of these registers.

2. If the Auto Boot option is enabled, responds to accesses of the first two Boot locations (17773000 and 17773002), and initiates the automatic bootstrap function. (LSI-11 CPUs only.)

3. Generate Qbus interrupt requests and control the vector transfer phase of the interrupt cycle.

4. Generate Qbus DMA requests and control the transfer of data between local buffer memory and Qbus main memory. Use block mode DMA transfers if the main memory will support them, otherwise use conventional burst mode DMA. During DMA transfers detect main memory parity and timeout errors.

5. Implement a microprocessor-like instruction and register set. Fetch and execute these instructions from an on-board EPROM. This PROM contains software modules which provide an ODT style debug, a set of disc-formatting, testing and bad block management routines, and the high-level MSCP protocol implementation.

Other major functional sections of the SRQD11-B are the Qbus interface and the ST506 Winchester disc interface.

The Qbus interface consists of the necessary logic to support Data In (DIN), Data Out (DOUT) and Read Modify Write (DATIO) bus cycles from the LSI-11 and MicroVAX CPUs. It also supports Interrupt Request cycles, DMA request cycles and DMA transfer cycles with 22-bit addressing in both block mode and burst mode protocols.

The disc interface is implemented almost entirely by the WD2010-05 Winchester disc controller chip. MFM data separation, Drive and Head select are the only functions performed externally to this chip. Appropriate differential line drivers and receivers complete the disc interface.

A dual-purpose I/O port is provided at the 10-way connector, J3. An ASCII terminal may be connected to this port in order to run the on-board diagnostic software, or the port may be connected to external status indicators and push buttons to implement disc write protect and access enable functions.

## DETAILED DESCRIPTION

### 7.1 Microprogrammed Sequencer

The sequencer consists mainly of a set of 512 x 8 registered PROMS which stores a set of microcode routines, each of which performs a specific function within the SRQD11-B. Routines are initiated by the P PROM at J-5. The inputs of the P PROM are connected to various system states such that it can recognize when a particular routine should be executed.

The outputs of the P PROM are connected to the Next Address bus of the sequencer, uA1 - uA8. When appropriate input conditions arise, the P PROM will output the starting address of the required sequencer routine onto the Next Address bus. On the next system clock edge (SYSCLK), the P PROM will be disabled, and the Next Address PROM (N PROM) will be enabled by the TRAP signal. The N PROM at H5 has its inputs and outputs connected to the next address bus, allowing it to use the current address on this bus to look up what the next address should be. This new address is placed on the Next Address bus on the next SYSCLK edge, thus entering the second step of the routine.

On each SYSCLK edge, the N PROM provides each successive address of the routine being executed. At the end of the routine the TRAP signal is switched on, returning control of the Next Address bus to the P PROM.

The least significant bit if the Next Address bus, uA0, can be made to assume the state of any one of the seven system signals connected to the 8-input multiplexer at K4. The sequencer controls this multiplexer via the B PROM at J4, so that at any step in a routine the next address will be conditionally odd or even depending on the state of the selected input of K4. This mechanism allows the sequencer to make decisions based on various system states, and to take the appropriate action.

Five other PROMS - the E,A,F,C and S PROMS - are connected to the Next Address bus, each producing eight output signals which control the rest of the logic on the board.

If there are no pending requests at the input of the P PROM the sequencer enters an instruction fetch routine. This routine places the contents of a program counter, stored in the ALU, into the address latch made up of D3,E3,F3,G3. The output of this latch addresses the EPROM at D4 and RAM at F4. The sequencer then asserts the FETCH signal, which places the output of the EPROM (or RAM) on to the Next Address bus via octal buffer B5. This starts up a microcode routine which performs the function specified by the value in the EPROM; thus the instruction is executed.

Data manipulation and storage is performed by the 8-bit Register and Arithmetic Logic Unit (RALU) made up of the two 2901C bit slice chips at A4 and B4.

## 7.2 Qbus Interface

Data and address information is transferred between the SRQD11-B and the Q-bus via the 2908 bus transceivers A7,B7,C7 and D7.The bus transceivers D6 and E7 are used to drive the high order address lines during Q-bus DMA, giving full 22-bit addressing capacity. Incoming 16-bit data appear on DAL0 - DAL15, and are multiplexed on to the 8-bit data bus via the 74LS257s at A6 and B6 under the control of microcode signals DAL/ and HILO.

Incoming addresses also appear on DAL0-DAL15 and are decoded by the W PROM at C6. This PROM produces four outputs, three of which correspond to the three base address options of the SRQD11-B, while the fourth output is asserted when the bootstrap addresses 173000 or 173002 appear at the input. The address and boot option links select the appropriate outputs from the W PROM to be presented to the P PROM via address latch K6 and synchronising latch K5. These inputs, along with DAL1, RBS7 and RDIN provide the P PROM with sufficient information to initiate the appropriate microcode routine.

If the **auto-boot** option is enabled, the SRQD11-B behaves in the following way : -

When the LSI-11 accesses location 173000, the SRQD11-B responds with 000400. This is a branch instruction which forces the LSI-11 to fetch the next instruction from 173002. The SRQD11-B then responds with 000777, also a branch instruction, which forces the LSI-11 to continue to fetch from 173002. While the LSI-11 is "hung" at 173002, the SRQD11-B uses its DMA capability to take control of the Qbus, and then follows the sequence detailed in Section 3.4 to complete the boot process.

16-bit data to be transferred to the Q-bus are loaded into storage latches in the bus transceivers from the 8-bit data bus under the control of microcode signals LDBRL/ and LDBRM/. Addresses are loaded in a similar manner except that the high order 6-bits of a 22-bit address are loaded by the microcode signal LDBRH/. Once the bus transceivers have been loaded with the appropriate information, the bus drivers are enabled by signals BUSEN/ and BUSENH/.

When the SRQD11-B has to interrupt the LSI-11 it asserts the signal DEVIRQ/ at H4 pin 9. This signal asserts BIRQ4L on the Q-bus, and depending on the configuration of the interrupt priority links P1,P2, and P3, may also assert BIRQ5L, BIRQ6L or BIRQ7L. The LSI-11 responds with BDINL followed by BIACKIL, which, provided that no higher priority device is requesting an interrupt, will cause the INTR signal at pin 6 of J6 to be asserted. The INTR signal is connected via K5 to the P PROM and causes the interrupt vector transfer microcode routine to be executed.

The SRQD11-B initiates DMA transfers by asserting the DMR signal at H4 pin 6. This signal asserts BDMRL on the Q-bus. The LSI-11 responds with BDMGIL which asserts the DMAGR signal at G6 pin 8. This signal is connected to the P PROM and initiates the DMA transfer routine in the microcode. This routine asserts the TSACK signal to become Q-bus master, then proceeds to transfer data over the Q-bus using the block mode protocol, provided this is supported by the memory, as indicated by the BREFL signal. If block mode is not supported then the SRQD11-B reverts to normal burst mode.

## 7.3 Disc Interface

The Winchester disc controller chip at K2 performs most of the functions necessary to control the disc, including :

1. Generation of step and direction signals for seeking.
2. MFM encoding for writing data to disc.
3. With the help of the 8465 data separator at H2, MFM decoding for reading data from disc.
4. Sector header recognition.
5. ECC (error checking and correction)
6. Transfer of disc data to and from the RAM at F4.

The WD2010 contains a register set and implements a set of high level disc commands which allow such operations as "read sector", "write sector", "format track" etc. Under microcode control, these commands and registers are made directly available to the firmware in the EPROM at D4 to effect all necessary disc operations.

Disc head select and drive select functions are implemented by the octal latch F1 under the control of microcode signal DHSEL/.

The 8465 data separator uses analog phase locked loop circuitry to recover clock information from the MFM read data stream from the disc. This clock signal is used by the WD2010 to decode the MFM data. The data separator also detects missing clocks in the MFM data, and generates the MCD signal which is used by the WD2010 to determine when sector header information is being received.

## 7.4 I/O Port

The I/O port serves two main functions. When running WOMBAT it can serve as an RS232 port to an ASCII terminal. This function is enabled by grounding the drive WRITE PROTECT 1/ input. If this input is left open then WOMBAT will try to communicate with the system console over the Qbus at address 17777560.

The signal from pin 12 of H4 drives the emitter of the 2N3638 transistor via the green access indicator LED. The collector of this transistor provides the necessary $\pm$ 3V swing at the ACCESS/ output to drive an RS232 terminal.

The negative supply rail required for this purpose is provided by the 7660 voltage inverter at J3.

RS232 input is received at the WRITE PROTECT 0/ input and connected to G3 pin 8 via a suitable clipping circuit.

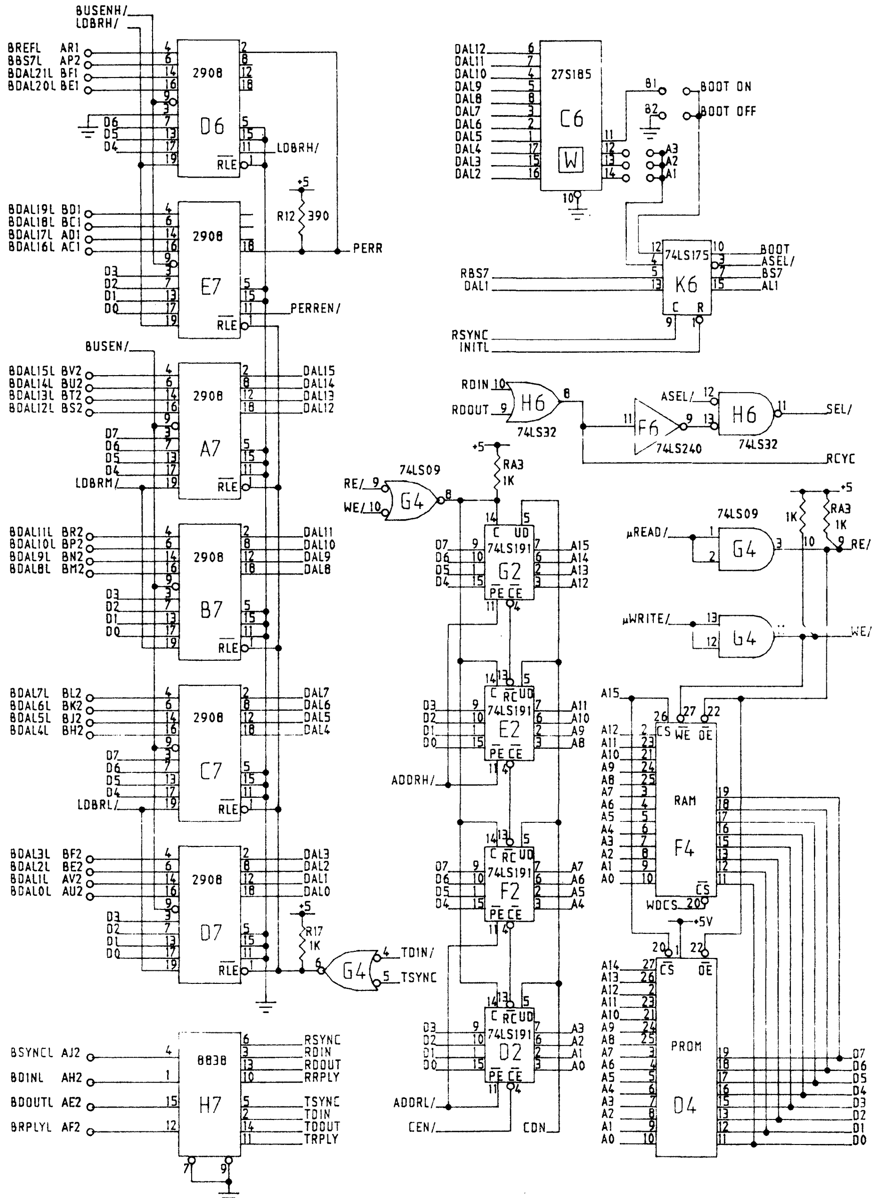
In normal operation (i.e. without WOMBAT running) these signals may be left unconnected. The WRITE PROTECT 1/ and WRITE PROTECT 0/ signals for the disc drives can also be connected through switches to ground, providing a write protect function on the drives.

The access signal can be connected to a lamp or LED via a suitable driver circuit to indicate drive access. The ACCESS/ signal is low when the drive is being accessed.

# CHAPTER EIGHT

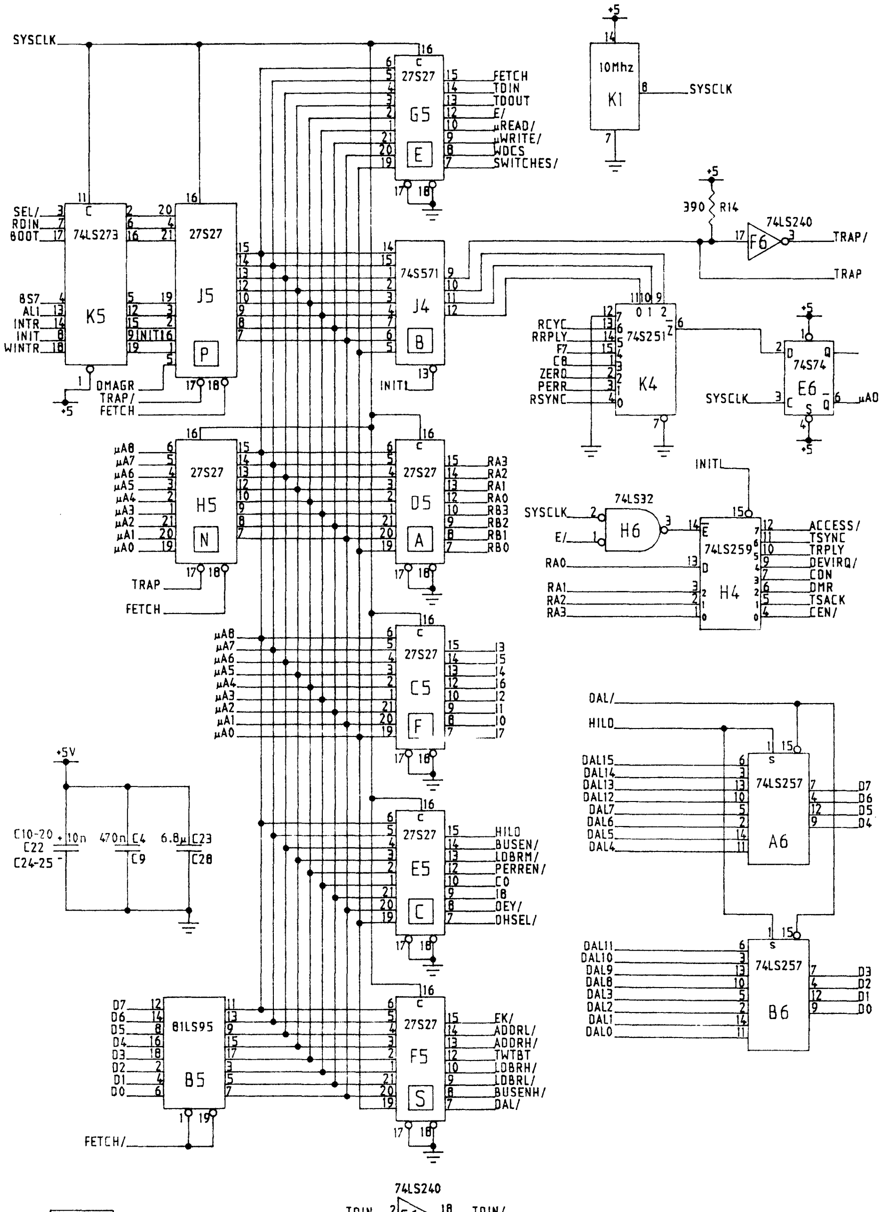
## SRQD11-B

## CIRCUIT DIAGRAMS



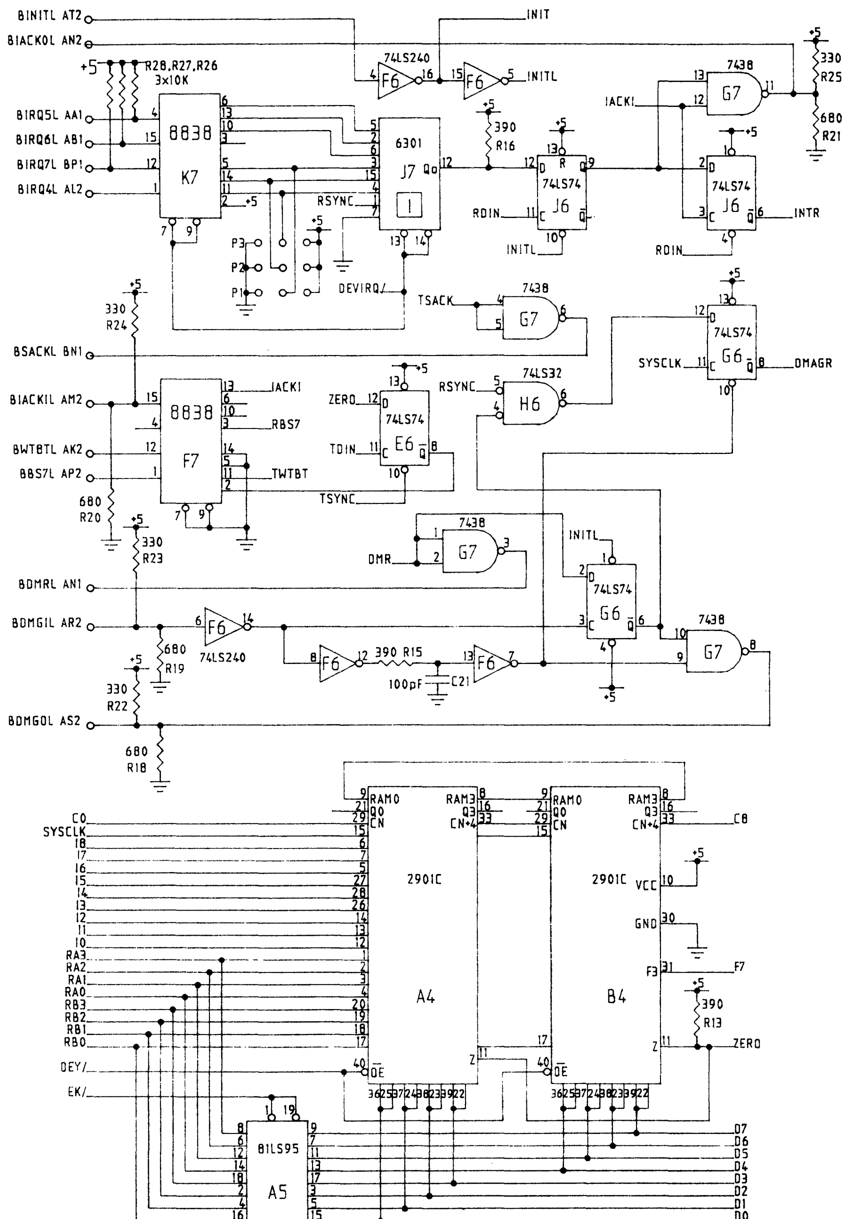

WEBSTER
COMPUTER
CORPORATION

SRQD11-B
WINCHESTER DISC CONTROLLER
SHEET 1 OF 4




WEBSTER
COMPUTER
CORPORATION

SRQD11-B
WINCHESTER DISC CONTROLLER
SHEET 2 OF 4

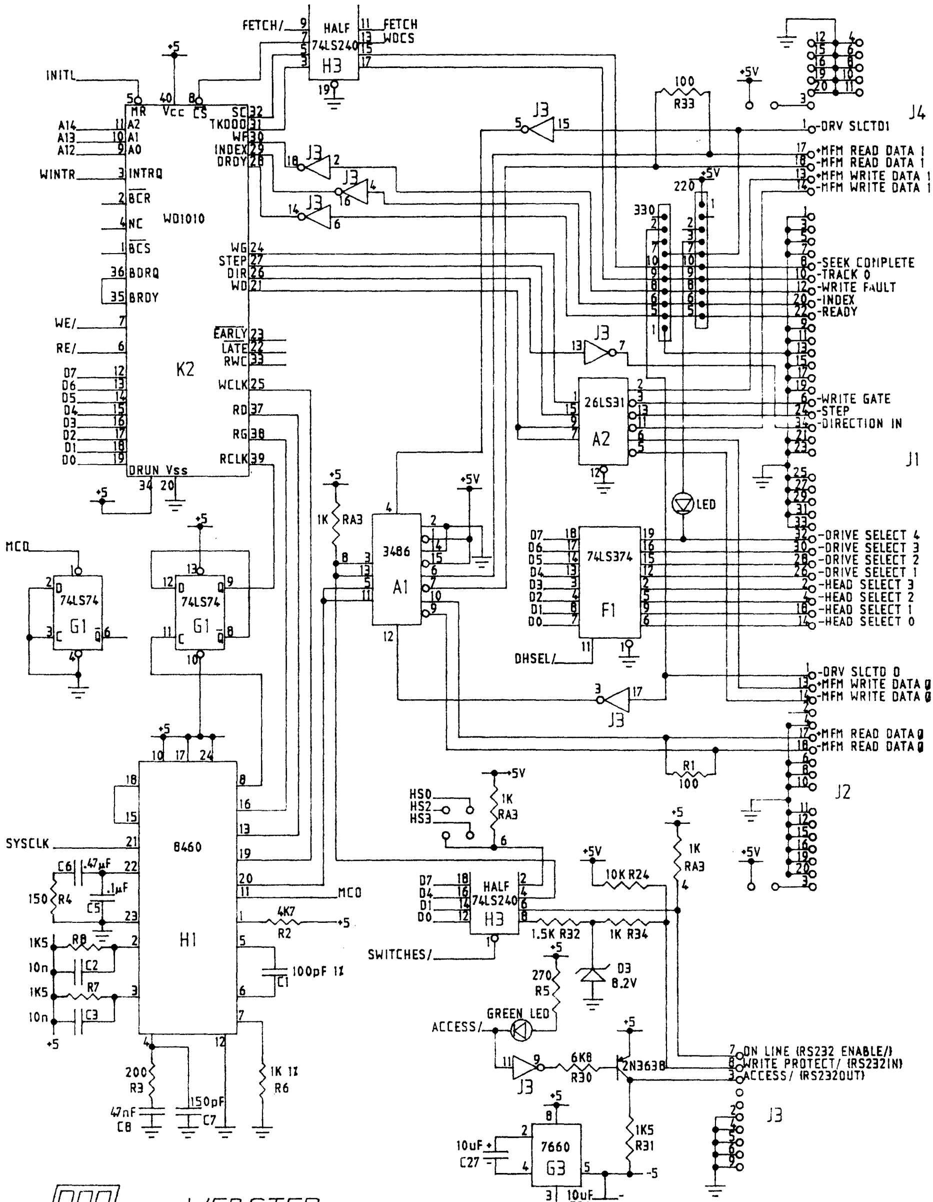



WEBSTER COMPUTER CORPORATION

SRQD11-B

WINCHESTER DISC CONTROLLER

SHEET 3 OF 4




WEBSTER
COMPUTER
CORPORATION

SRQD11-B
WINCHESTER DISC CONTROLLER
SHEET 4 OF 4