

**SRQD11-A Winchester Disc Controller**

**USER MANUAL**

**Version 1.0**

**WEBSTER COMPUTER CORPORATION**

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

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

TABLE OF CONTENTS

CHAPTER 1	SRQD11-A General Description . . . . .	1
CHAPTER 2	SRQD11-A Controller Specifications . . . . .	5
CHAPTER 3	SRQD11-A Installation . . . . .	6
	3.1 ST506 Interface Connections . . . . .	6
	3.2 SRQD11-A Settings and Adjustments . . . . .	7
	3.3 SRQD11-A Optional Accessories . . . . .	9
	3.4 On-Board Bootstrap . . . . .	11
CHAPTER 4	SRQD11-A Programming . . . . .	12
	4.1 Overview of MSCP . . . . .	12
	4.2 Controller Communications . . . . .	13
	4.3 Message Transmission . . . . .	16
	4.4 Data Transmission . . . . .	17
	4.5 Initialization . . . . .	17
	4.6 Registers . . . . .	18
	4.7 MSCP Commands . . . . .	19
	4.8 Error Handling . . . . .	21
	4.9 Fatal Controller Error . . . . .	22
CHAPTER 5	SRQD11-A WOMBAT Utilities . . . . .	23
	5.1 Starting up WOMBAT . . . . .	23
	5.2 Setting up the Disc Structure . . . . .	25
	5.3 WOMBAT Menu Options . . . . .	25
	5.4 WOMBAT Error Messages . . . . .	39
	5.5 WOMBAT Self Diagnostics . . . . .	40
CHAPTER 6	SRQD11-A Functional Description . . . . .	41
	6.1 Microprogrammed Sequencer . . . . .	42
	6.2 Qbus Interface . . . . .	43
	6.3 Disc Interface . . . . .	44
	6.4 I/O Port . . . . .	44
CHAPTER 7	SRQD11-A Circuit Diagrams . . . . .	46

CHAPTER 1  
SRQD11-A General Description

The Webster SRQD11-A is a dual height Qbus interface to ST506 compatible 5-1/4 inch Winchester disc drives. Implementing DEC's Mass Storage Control Protocol (MSCP), the SRQD11-A flexibly couples any size disc to all standard DEC operating systems without software modification. Comprehensive on-board interactive formatting and diagnostic firmware provides engineering support across the range of LSI11, MicroVAX, and various non-DEC implementations of the Qbus.

ST506 Interface

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

AMPEX	ATASI	CDC	CMI
DISCRON	EVOTEK	FUJITSU	IMI
MAXTOR	MEMOREX	MICROPOLIS	MINISCRIBE
MITSUBISHI	OLIVETTI	OTARI	PRIAM
RODIME	SEAGATE	SHUGART	TANDON
VERTEX			

Qbus Interface

Originally introduced in 1975 by Digital Equipment Corporation to support the LSI11 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-A fully implements all current Qbus enhancements, including block mode transfers and 22-bit addressing, and flexibly supports LSI11/2, LSI11/23, LSI11/73, MicroVAX 1, MicroVAX 2, Motorola 68000 and National 32032 Qbus CPU designs.

Block Mode DMA

When used with block mode memory, the SRQD11-A 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-A reverts automatically to simple DMA.

## MSCP Universal Disc Architecture

The SRQD11-A communicates with the software through a simple register pair to memory resident command packets. Disc geometry factors such as sectors, heads and cylinders are invisible to the host computer as the SRQD11-A accepts 32-bit binary block numbers, converting them to physical disc addresses. Disc capacity is inherently communicated back and never assumed by the software, therefore any size disc may be fully accessed without software modification. Supported operating systems include RT-11 version 5, RSX-11M plus version 2.1, TSX-plus version 4, RSTS/E version 8, Micro-VMS and UNIX.

## Seek Optimisation

Queueing of up to 32 commands is permitted within the SRQD11-A. The optimum order of execution of these packets is dynamically computed 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-A during formatting and pattern testing, and redirected to a user-specified reserved area at the end of the disc. During all subsequent operations the SRQD11-A simulates fault-free media through transparent bad block replacement.

## Transparent Read Retry

A 16-bit cyclic redundancy word is added to each disc sector on writing, and verified during read. On miscompare, the SRQD11-A will automatically repeat the read operation up to 10 times before reporting an unrecoverable error to the operating system.

## Data Buffer

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

### Multi Drive Capability

While only one drive may be connected to each SRQD11-A controller, up to three controllers may coexist in one system. This configuration permits simultaneous read, write and seek operations on up to three disc drives, vastly outperforming single-minded multiple drive per controller arrangements.

### On Board Bootstrap

A miniature programming plug permits the SRQD11-A optionally to bootstrap on power-up, obviating the need for a specially programmed external ROM module.

### 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 and Online control switches.

### WOMBAT Utilities

Webster Omnipotent Mass-storage Builder and Tester (WOMBAT) is a set of interactive formatting, diagnostic and debug utilities totally contained within the SRQD11-A firmware. An on board serial connection is provided for communication with an ASCII terminal, permitting disc formatting and maintenance operations to be carried out with no other hardware present. Alternatively, the SRQD11-A is able to seek out and communicate directly with the system console terminal over the Qbus.

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

### 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-A 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 TECO-like disk control and manipulation language, a continuous read/write/seek exerciser, a disc surface pattern tester, and a bad block replacement routine.

CHAPTER 2  
SRQD11-A Controller Specifications

Bus interface:	DEC Qbus
Transfer mode:	Block mode DMA
Memory address capacity:	4 megabyte (22-bit)
Software emulation:	DEC Mass Storage Control Protocol
Command buffer capacity:	Up to 32 MSCP commands
Data buffer capacity:	512 bytes (one sector)
CSR address:	172150, 54, 60 plug selectable
Interrupt vector:	Software selectable
Interrupt priority:	Level 4 through 7 plug selectable
Qbus loads:	1 DC, 2 AC
Drive interface:	Seagate ST506
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:	173000, 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 GREEN - disc access in progress
Output:	Disc access in progress
TTL inputs:	Write Protect switch Online switch

CHAPTER 3  
SRQD11-A Installation

3.1 ST506 Interface Connections

34-way Control

<u>Function</u>	<u>Pin</u>
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

20-way Data

<u>Function</u>	<u>Pin</u>
+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-A 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)

Position: A1	Base (CSR) Address: 172150
A2	172154
A3	172160

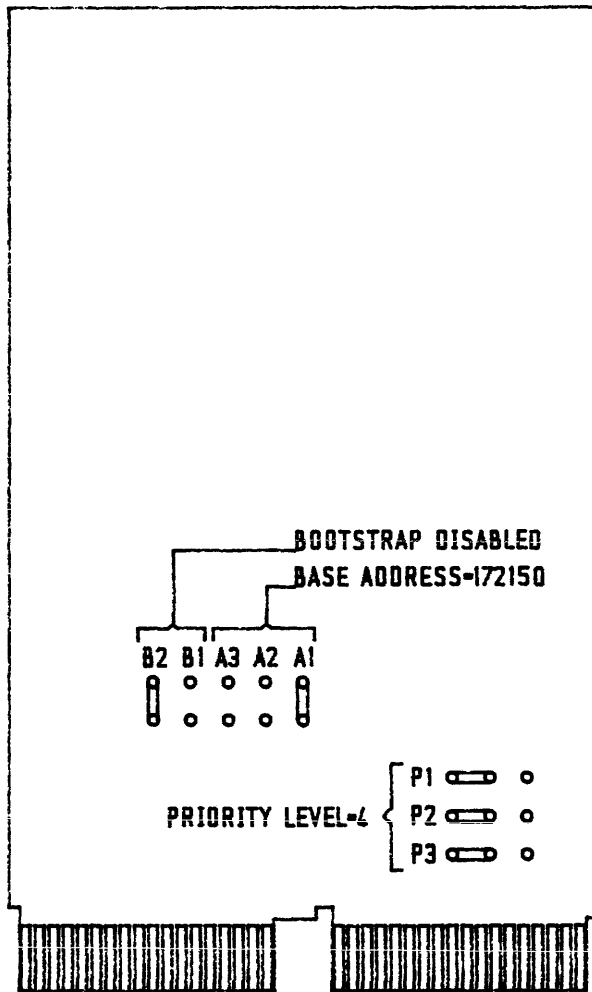
#### Automatic Bootstrap Select Configuration Plug (B)

Position: B1	On board bootstrap enabled at address 173000
B2	On board bootstrap disabled

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

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

Factory Jumper Settings



### 3.3 SRQD11-A Optional Accessories

ST506/LSRQD11-A Drive Signal and Control Cable Set

MBSRQD11-A : RS232 maintenance terminal adaptor

#### 3.3.1 ST506/LSRQD11-A Drive Signal & Control Cable Set

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 36-way types, and the signal cable and sockets are 20-way. Both cables have a maximum length of 6 metres.

#### 3.3.2 Optional Front Panel Connections

If required, a front panel can be connected to J3, the front panel/maintenance connector. The functions supported are On-line Switch, Write-protect Switch and Access Light. (See Table I).

#### 3.3.3 MBSRQD11-A RS232 Maintenance Terminal Adaptor

WOMBAT (see Chapter 5) can communicate either through the computer system console or the front panel/maintenance connector. If pin 7 on this connector is grounded, the latter will be the case.

The communication format is :

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

The MBSRQD11-A RS232 maintenance terminal adaptor allows simple connection 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 I).

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 I - 10-WAY FRONT PANEL/MAINTENANCE CONNECTOR

<u>Pin</u>	<u>Front Panel Function</u>	<u>Maintenance Function</u>
7	**On-line Switch Input. Connecting this input to ground will send the drive off-line. Has an on-board 220 ohm pull-up.	RS232 Enable. Connecting this input to ground enables WOMBAT to use this connector for communication.
9	**Write-protect Switch Input. Connecting this input to ground will write-protect the drive. Has an on-board 10K ohm pull-up.	RS232 Input. RS232 Input Data to WOMBAT.
5	Access Light This output can be used to drive an Access Light. The levels are : ACCESS : -5V thro' 1.5K ohm NO ACCESS : +3.5V @ 5mA max.	RS232 Output RS232 Output Data from WOMBAT. +3.5V to -5V RS232 compatible.
4,6,8,10	Ground	Ground

\*\* NB : If both switches are closed the controller assumes an MBSRQD11-A cable is connected and ignores the switch functions - that is the drive will be "on-line" and "write-enabled".

MBSRQD11-A RS232 Maintenance Terminal Adaptor

<u>10-way Connector Pins</u>	<u>DB25S Pins</u>	<u>Function</u>
7	7	RS232 Enable
9	2	RS232 Input
5	3	RS232 Output
4	7	Ground

### 3.4 On-Board Bootstap

For LSI processors only, the controller by means of a configuration plug, may be set up to provide an "On Board Bootstrap". It does this by implementing 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, which are executed when the LSI-11 boots, cause the computer to loop. The controller 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.
3. The controller changes the contents of location 773002 to a "CLR PC". This causes the host computer to execute the initialization code, which sets up registers and waits for the disc's boot block to be read into memory starting at location zero.
4. The controller reads the boot block, block zero, into host memory. It then waits to be initialized.
5. The host computer commences execution of the instructions in the boot block.

## CHAPTER 4 SRQD11-A Programming

### 4.1 Overview of MSCP

Mass storage control protocol (MSCP) is the message-oriented set of rules by which the SRQD11-A controller module communicates with the host system. This protocol allows the host to simply send message requests for 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 two independent layers (host and controller), where changes can be made within one layer without affecting the other. Within this framework, commands and data are transmitted from one layer to the other, through the controller, in the form of message packets

In the host layer, the computer runs users' applications programs that make demands on the mass storage (disc) medium. The controller layer's various functions ensure that the host layer is able to read or write data, without error, at its own speed, and when it wishes. The disc layer receives the data, stores it as long as necessary, and makes it available to the controller on command.

The host layer uses two layers of software to accomplish input/output operations. The first sub-layer includes a mass-storage class driver which constructs the message packets in order to perform I/O functions, such as reading and writing, and on the same level, a diagnostics and utilities class driver which constructs message packets in the diagnostics and utilities protocol. The other sub-layer is the controller driver which passes the message packets along the bus between the host and controller.

The controller layer includes routines that receive messages from or transmit messages to the host. Its other functions concern the disc, which include controlling head motion, accepting commands from the controller, reporting status to the controller, and reading and writing.

The mass-storage class driver handles all message exchange between the operating system and any mass-storage device of a specific class, which means that any size disc may be fully accessed without software modification. The disc drive itself contains a parameter table of all of its own characteristics, such as geometry and retry counts for error handling. At system startup, this information is passed to the controller so that it may manage operation of that particular disc configuration.

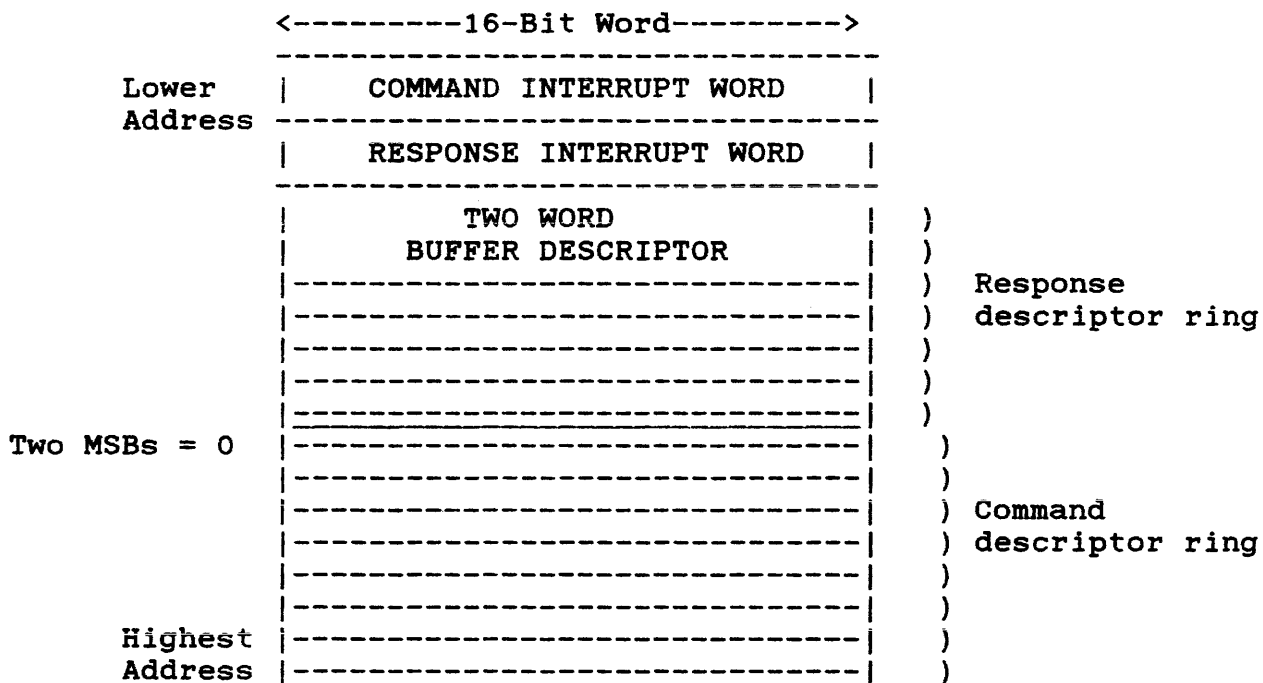
In addition to relieving the host-resident driver of disc-specific data, the controller and disc together provide the host with "clean" data. This implies data for which all necessary error detection, correction, and recovery have already been done. The disc drive handles some positioner errors entirely by itself and performs certain error-recovery operations under direction of the controller.

#### 4.2 Controller Communications

The host designates an area of memory to be used as a communications area. 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) rings, organised into ring buffers.

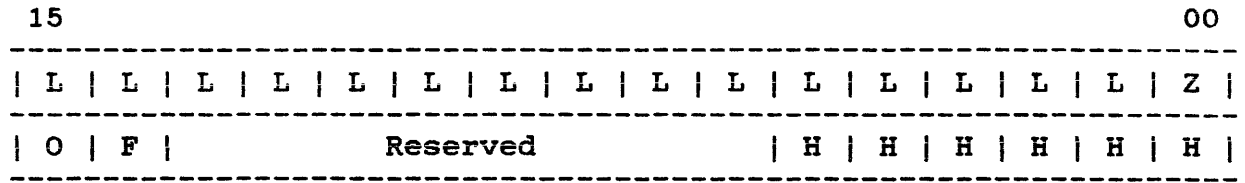
The following diagram shows the format of the memory communications area.



#### Command and Response Rings

Command and response rings are each organised into a ring 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:



Code    Description

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

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

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

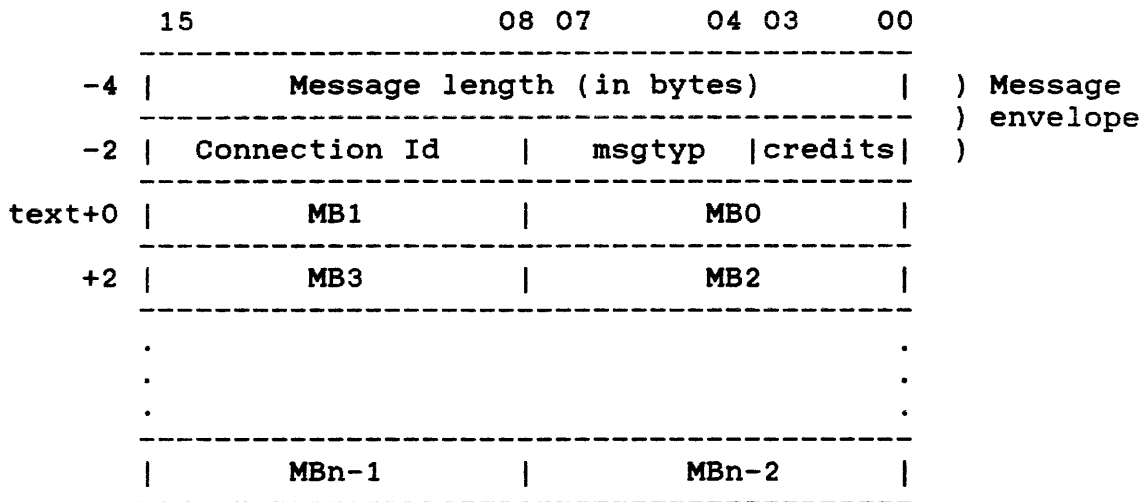
The controller always sets F=1 when returning a descriptor to the host, so if a host wishes to override ring transition interrupts it must always clear F when passing ownership of a descriptor to the controller.

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

**Message Packets**

The command or response descriptor points to word (text+0) of a 16-bit word-aligned message envelope formatted as follows:





Word Envelope Contents

0 Message length, in bytes.

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

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

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

1 Connection Id

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

2 Message Type

The following response ring message types are implemented:

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

### 3 Credit field

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

## 4.3 Message Transmission

### Command Transmission

When the ownership bit (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.

The host writes into SA the lengths of the rings, whether interrupts are to be armed, and if so, the address of the interrupt vector. The controller then runs a complete internal integrity check and signals either success or failure.

- 1 Low order address of communications area, ie., ring buffer address.

The host reads an echo of the ring lengths from SA, and then writes into SA the low-order portion of the ring base address.

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

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

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

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

#### 4.6 Registers

The programmable registers contained on the SRQD11-A 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-A responds to the following initialization words :

<u>Word (octal)</u>	<u>Function</u>
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-A controller.

<u>Command</u>	<u>Function</u>
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 error detection, recovery, and transparent bad block replacement. A cyclic redundancy check (CRC) allows a read operation to be performed up to 10 times before a hardware error is reported to the operating system.

##### 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-A controller detected an internal error, but is able to continue processing its outstanding commands.
Data Error	Data could not be read or written due to CRC errors, "Header Not Found", or due to a sector being read whose forced error bit was set.
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-A controller found some field in the command to be in error.
Success	The command was successfully completed.
Unit Available	The SRQD11-A controller is not on line, but it can accept an On Line command from the host.
Unit Offline	The SRQD11-A 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.

<u>ERROR CODES</u> (octal)	<u>DESCRIPTION</u>
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 5  
SRQD11-A 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-system.

Interactive communication with WOMBAT may be achieved by connecting a 9600-baud terminal to the SRQD11-A as described in Chapter 3. When used this way, no additional hardware or software is required (other than the drive, terminal, SRQD11-A and +5V. power supply) to perform drive and controller initialization and maintenance under WOMBAT.

When no terminal is connected directly to the SRQD11-A, WOMBAT will automatically search for and use the console terminal at address 177560 on the Qbus.

5.1 Starting up WOMBAT

WOMBAT is invoked by depositing the special data pattern 250 (octal) into the SRQD11-A control and status register. When used with any LSI-11 CPU, WOMBAT may be started by entering a small program via console ODT : -

1000/	106427	Line feed
1002/	340	"
1004/	12737	"
1006/	250	"
1010/	172150	"
1012/	777	Return
1000G		

A similar procedure is adopted when using non-LSI-11 CPUs.

The following is a PDP-11 MACRO listing of the WOMBAT starting program. This program will run under RT-11. Unfortunately it is far more complex to do the same functions under RSX or RSTS, so we recommend that in these cases you key the program in under console ODT as above.

DUTALK MACRO V05.00 Wednesday 08-Aug-84 14:03 Page 1

```
;Talk to the SRQD11-A controller and get WOMBAT to run.  
;It will communicate with the console terminal unless a  
;terminal is plugged into the controller's terminal port,  
;in which case it will use that terminal.
```

```
CSR=      172150                ;Controller's CSR address  
  
001000    106427 start: mtps #340        ;make us uninterruptable  
          000340  
001004    012737      mov  #250, @CSR    ;invoke WOMBAT  
          000250  
          172150  
001012    000777      br   .            ;and hang so it can run  
                                           ;(if we don't hang there  
                                           ;will be contention for  
                                           ;the terminal if WOMBAT's  
                                           ;terminal is the console).  
  
          .end start
```

Since this program "hangs" the computer, you must reboot to stop both it and WOMBAT and commence operations again.

When WOMBAT is invoked it will immediately try to read the disc and assess if it has a valid structure. The structure incorporates such factors as the number of cylinders, number of heads, and other details relating to the disc. This information is always written on the first two sectors of the first track of the disc, which makes this area and the remainder of track zero permanently unavailable to the user. Once the disc is structured, WOMBAT can access this information without having to know anything about the disc. When a disc does not have a valid structure a warning message is displayed:

```
ID  
ID  
* Warning - disc is not formatted  
* Warning - disc has no valid structure
```

You cannot perform any function with WOMBAT until the disc has first been structured (refer paragraph 5.2). A previously structured disc will automatically return the Master Menu when WOMBAT is invoked.

## 5.2 Setting up the Disc Structure

When you attach a 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-A 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-A 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-A 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.

The instructions for these procedures and the other available options, are described in the next paragraph. For consistency and clarity, each option is described within its Menu set.

## 5.3 WOMBAT Menu Options

You communicate with WOMBAT by selecting the required option from the master menu set which is given below :

\*\*\* 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 Display Error Statistics
- 6 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.

Warning : Error Statistics Table 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 attach a new disc to the SRQD11-A 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. This option allows these parameters to be specified.

WOMBAT first displays its name and version. This is used as an identifier when the disc structure is read to ensure that the various routines accessing the disc agree about the structure format. The warning message simply highlights the fact that the disc is not structured.

If a disc has been previously structured, WOMBAT displays -

```
*Disc Characteristics *  
Created by: WOMBAT Version 3.0
```

You may override any characteristic with different data, or press RETURN to accept the current parameter and display the next line.

If you incorrectly type in any one of the 10 parameters, simply press the RETURN key enough times to bypass the remaining parameter fields, until the 'Disc Structure Menu' is returned. Select this option again and enter the correct data.

### 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.

```
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  
Command Queue Size:[1-32] [8.]?  
Optimisation Strategy:[0-3] [1.]?  
Fairness Count:[1-255] [25.]?  
Shipping Zone Cylinder: [340.]?  
Media Type:(AAAnn) [WCC31.]?  
Serial Number:(Octal) [0]?  
HDR at Block: -18  
USR at Block: 0  
RCT at Block: 43038  
WRK at Block: 43074  
FCE at Block: 43092  
BAD at Block: 43110  
End Disc at Block: 44045

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. Please refer to Appendix A at the end of this manual for a table of typical disc structure parameters used by other manufacturers. Please realise that WOMBAT trusts you. Entering nonsensical data will result in chaos. Press the RETURN key after each entry and the next prompt will display.

**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-A 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.

<u>Parameter</u>	<u>Step Rate</u>
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	7.0 ms
15	7.5 ms

**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"
```

**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.

WOMBAT will then compute where various necessary areas on the disc are to go, and reports their positions:

**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.

USR Area        The size, in blocks, of the user area on the disc, as reported to the host operating system.

NB: The user area is the ONLY portion of the disc that the host operating system can see.

## Format Disc

The disc structure is now set up in the SRQD11-A 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

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 Disk 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.

CRC Cyclic redundancy error; miscompare of the 16-bit cyclic redundancy word on a write or read verification sequence.

### 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  020 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.
```

### 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 & redisplay Test Menu options.

## **BAD BLOCK MANAGEMENT MENU**

Bad blocks are replaced by marking them as bad on the disc (by 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.



#### 5.4 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.

## 5.5 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 6  
SRQD11-A Functional Description

GENERAL DESCRIPTION

The heart of the SRQD11-A is a microprogrammed sequencer, which, combined with an 8-bit RALU, forms a high speed processor which performs the following functions :

1. Implement the MSCP register pair - the Initialization and Polling Register and the Status and Address Register. Respond to the LSI-11 CPU's access of these registers.
2. If the Auto Boot option is enabled, respond to accesses of the first two Boot locations (17773000 and 17773002), and initiate the automatic bootstrap function.
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-A 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. 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 WD1010-05 Winchester disc controller chip. MFM data separation and 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

### 6.1 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-A. 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 of 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.

## 6.2 Qbus Interface

Data and address information is transferred between the SRQD11-A 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-A, 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-A behaves in the following way : -

When the LSI-11 accesses location 173000, the SRQD11-A responds with 000400. This is a branch instruction which forces the LSI-11 to fetch the next instruction from 173002. The SRQD11-A 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-A 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-A 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-A 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-A reverts to normal burst mode.

### 6.3 Disc Interface

The WD1010 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 8460 data separator at H2, MFM decoding for reading data from disc.
4. Sector header recognition.
5. CRC generation and checking.
6. Transfer of disc data to and from the RAM at F4.

The WD1010 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 8460 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 WD1010 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 WD1010 to determine when sector header information is being received.

### 6.4 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 ONLINE 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/ 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.

Alternatively the ONLINE signal can be connected through a switch contact to ground to control access to the drive. A closed contact will inhibit access. The WRITE PROTECT/ signal can also be connected through a switch to ground, providing a write protect function on the drive.

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.

ST506 is a trademark of Seagate Technology.

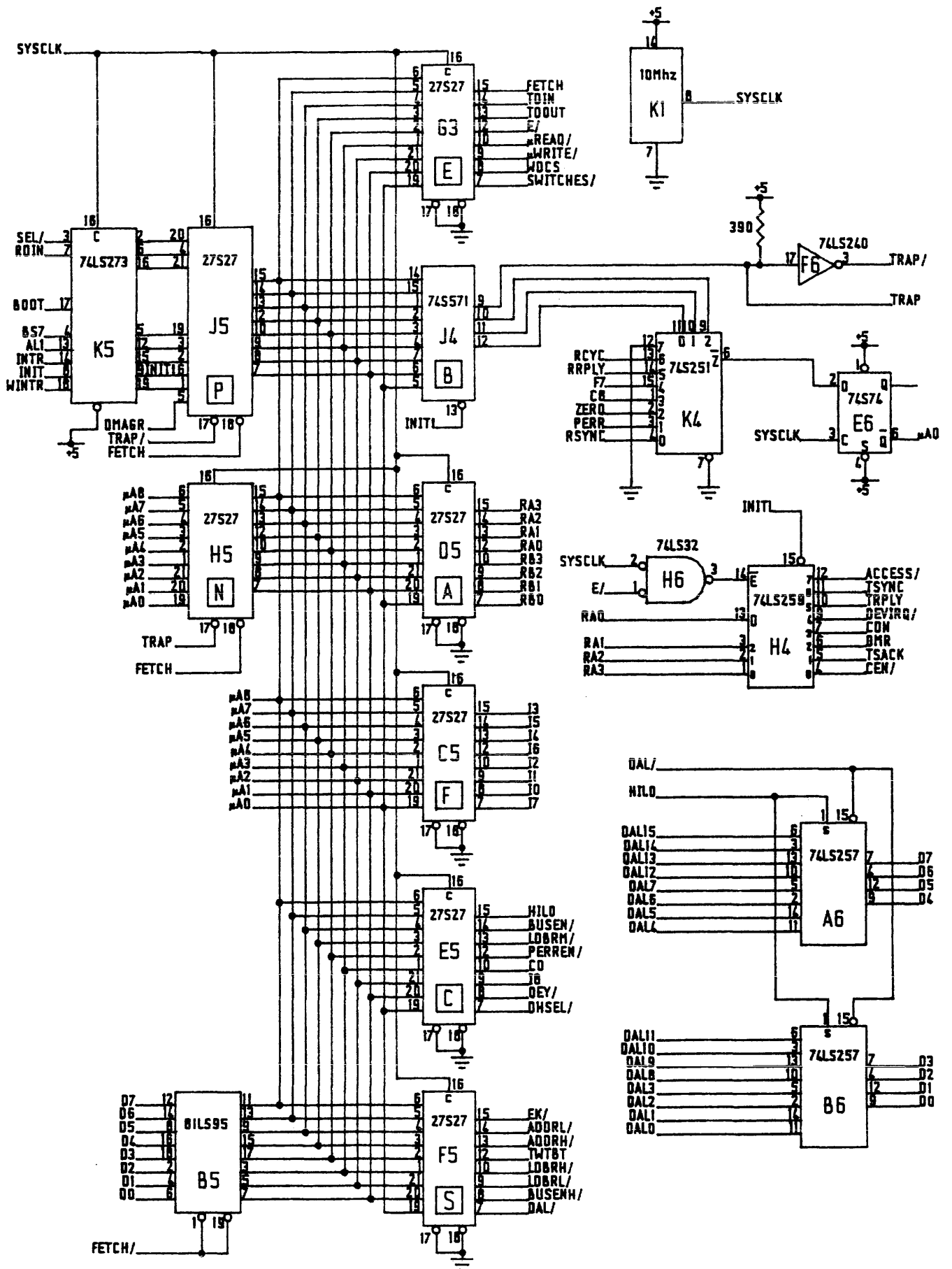
DEC, MSCP, LSI11, MicroVAX, RT-11, RSX-11M plus, RSTS/E, Micro VMS, and Qbus are trademarks of Digital Equipment Corporation.

TSX-plus is a trademark of S & H Computer Systems.

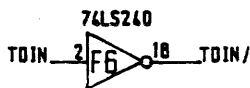
SRQD11-A and WOMBAT are trademarks of Webster Computer Corporation.





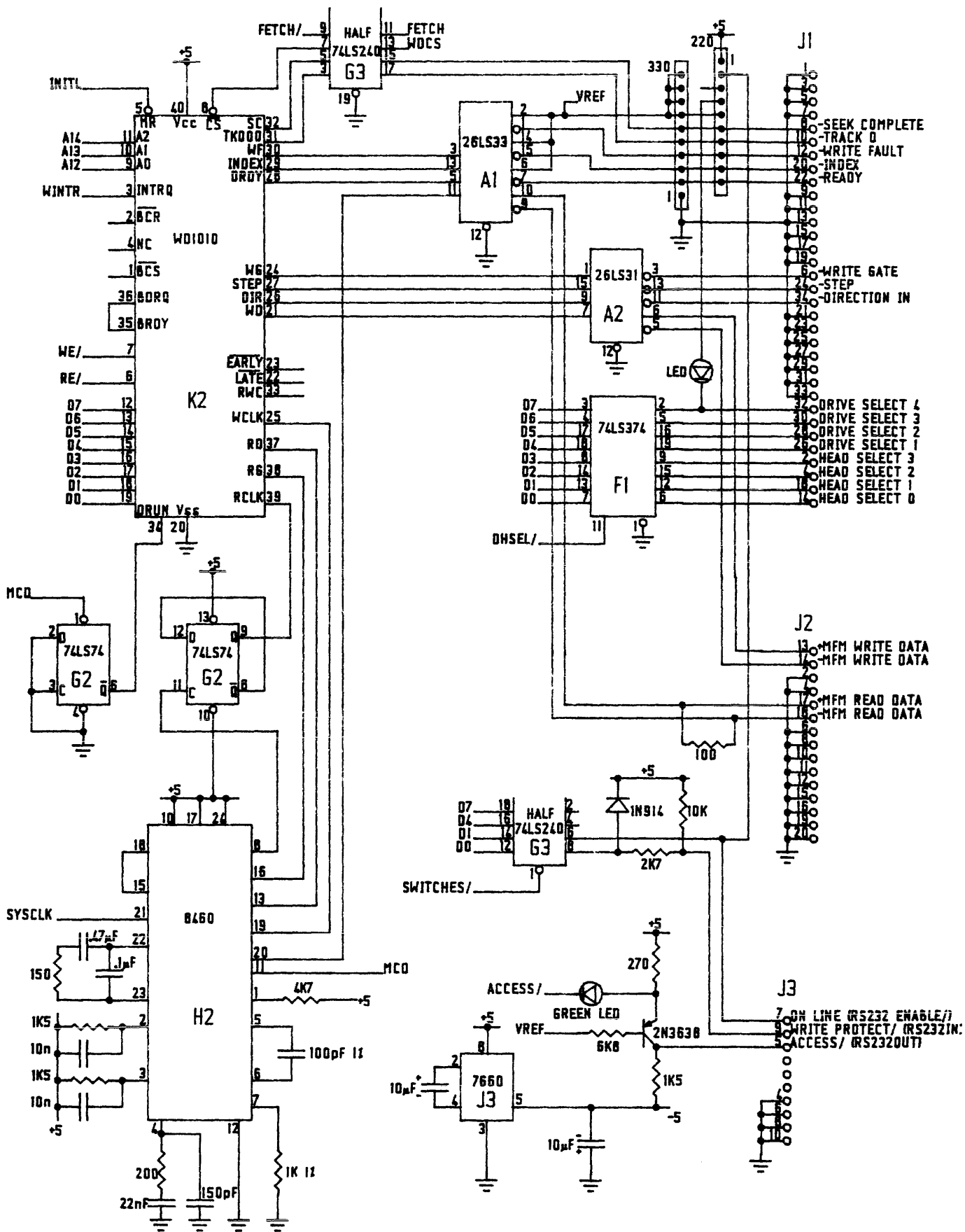


WEBSTER  
COMPUTER  
CORPORATION



SRQ01-A  
WINCHESTER DISC CONTROLLER  
SHEET 2 OF 4





WEBSTER  
COMPUTER  
CORPORATION

SRQD11-A  
WINCHESTER DISC CONTROLLER  
SHEET 4 OF 4

WEBSTER COMPUTER CORPORATION  
SRQD11-A Winchester Disc Controller

APPENDIX A

Table of Typical Disc Structure Parameters

MANUFACTURER	MODEL	NO. CYL.	NO. HEADS	NO. SECTS	REPL. CYL.	STEP RATE	SHIPPING CYL.
CDC	9415-5 WREN	697	5	17	7	0	-
HITACHI	DK503-2	306	4	17	6	0	320
MAXTOR	XT-1140	918	15	18	8	0	-
MINISCRIBE	3012	612	2	17	12	0	656
OTARI	D526	306	8	18	6	0	340
PRIAM	504	755	11	17	10	0	-
TANDON	TM503	306	6	17	6	0	-
RODIME	204	320	8	18	4	0	340
VERTEX	V170	987	7	18	7	0	-