WUSMD

WINCHESTER DISC CONTROLLER

USER MANUAL

WEBSTER
COMPUTER
CORPORATION

WEBSTER COMPUTER CORPORATION

WUSMD Winchester Disc Controller

USER MANUAL

Version 2.1

WEBSTER COMPUTER CORPORATION

USA

2109 O'Toole Avenue, Suite J
San Jose, CA 95131-1303
USA
Voice (408) 945 8054
FAX (408) 945 1832

AUSTRALIA

1270 Ferntree Gully Road
Scoresby, Victoria 3179
AUSTRALIA
Voice (03) 764 1100
Service (03) 764 1156
FAX (03) 764 1179

First Floor 9-19 Ryde Road
Pymble, New South Wales 2073
AUSTRALIA
Voice (02) 498 4422
FAX (02) 499 2147

163 Wharf Street
Spring Hill, Queensland 4000
AUSTRALIA
Voice (07) 832 1899
FAX (07) 832 2302

UNITED KINGDOM

Unit 7 Weltech Centre
Ridgeway
Welwyn Garden City
Hertsfordshire AL72AA
UNITED KINGDOM
Voice (707) 33 6969
FAX (707) 37 3378

TABLE OF CONTENTS

TABLES

FIGURES

Installation Procedure

The installation of the WUSMD is relatively simple. Before the WUSMD can be used the following procedure must be followed.

1. If you are not familiar with the WUSMD and its features, **READ this manual's introductory chapter (Chapter 1).**

2. Select the controller CSR address, power up mode, bootstrap address, media type and interrupt priority by setting the switches located on the controller. See Chapter 2 - Installation.

3. Locate the controller in the Unibus backplane and connect the drive control and data cables.

4. Power-up the system and invoke WOMBAT using one of the methods outlined in Chapter 3 - WOMBAT Utilities. WOMBAT is Webster's interactive diagnostic and formatting program. **The disc(s) cannot be used with the WUSMD until WOMBAT has structured and formatted the disc(s).** Answer the WOMBAT prompts according to the disc structure and system configuration you have selected.

When all these steps have been completed the WUSMD will be ready to use with your computer system.

WEBSTER COMPUTER CORPORATION
WUSMD Winchester Disc Controller

WUSMD Specifications

Bus interface	DEC Unibus
Memory address capacity	256k bytes (18 bits)
Software emulation	DEC MSCP
Command buffer capacity	Up to 32 commands
Disc cache size	1 Megabyte (with parity)
Physical size	226mm x 399mm (Unibus hex)
Power requirement	5V 4A typical 12V 0.4A typical
Temperature	5 - 60 degrees C operating 0 - 66 degrees C store
Drive interface	CDC SMD (Standard, Modified, Enhanced, Extended, SMD-E)
Drive connectors	60 way control, 4 x 26 way data
Number of drives	4
Number of cylinders	4096
Number of heads	16
Max. capacity per drive	1.17 gigabytes
CSR address	Switch selectable
Interrupt vector	Software selectable
Interrupt priority	Level 5 - setable to other levels by changing PC board links
Unibus loads	1 DC, 1 AC
Access time overhead	2 ms (plus drive access time)
Disc transfer rate	3.0 Mbyte/sec maximum (non interleaved)
On board LED indicators	Red: Fatal error indicator Green: Access in progress (one per drive)
TTL outputs	Disc access in progress
TTL inputs	One write protect input per drive
RS232 output	Data transmitted to terminal - 9600 baud
RS232 input	Data received from terminal - 9600 baud

Optional Accessories:

LWUSMD	SMD drive signal and control cable set
KSMD/T	RS232 Maintenance Terminal Adapter kit

Chapter 1
WUSMD General Description

The Webster WUSMD is a high performance hex height interface to SMD compatible disc drives. The WUSMD features a one megabyte cache memory, 24 megabit per second throughput, command queueing, overlapped seeks, and implements DEC's Mass Storage Control Protocol (MSCP).

The WUSMD flexibly couples discs of any size and data rate to all standard DEC operating systems without software modification. Comprehensive on-board interactive formatting and diagnostic firmware provides engineering support across the range of various DEC and non-DEC implementations of the Unibus.

Four SMD drives up to 3 megabytes/second transfer rate

The SMD interface is an industry standard for high capacity disc drives. There have been modifications to this standard by various manufacturers which the WUSMD supports with no software or hardware changes: Standard, Modified, Enhanced, Extended and SMD-E. Four drives of any type with transfer rates of up to 24 megabits per second (3 megabytes per second) are supported.

One Megabyte Disc Cache

The WUSMD stages all data from the disc drive through the cache memory. Data transfers from the cache are approximately 3 ms compared to 20 - 30 ms for typical drive access times - up to an 80 percent improvement in access time.

No Sector Interleaving

The WUSMD stages all data from the disc drive through the cache memory. This ensures that all data can be transferred at full disc speed over the disc interface and at maximum speed over the Unibus without incurring data late errors. Disc and Unibus transfers are performed simultaneously to minimise access times.

Read Look-ahead

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

Virtual Units

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

Block mode DMA and DMA Throttle

When used with block mode memory, the Webster WUSMD interleaves address references with bursts of data - almost doubling Unibus throughput. The WUSMD fully conforms with Unibus Block Mode DMA protocol. With non block mode memory, the WUSMD automatically reverts to simple DMA.

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

Drive Shadowing

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

Seek Optimisation and Overlap

The Webster WUSMD can queue up to 32 commands. The optimum order of execution is dynamically computed according to the strategy selected by the user. With multiple disc drives seeks are initiated in parallel to further improving performance.

Error Checking and Correction

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

Dynamic Bad Block Replacement

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

Statistics Recording

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

Write Protect

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

MSCP Emulation

The Webster WUSMD 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 WUSMD 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.

Unibus Interface

The Digital Equipment Corporation's Unibus is a high speed communications path that links system components and peripheral devices. The WUSMD fully implements all current Unibus enhancements, including block mode transfers, 18 bit addressing, and fully supports PDP-11/84, PDP-11/44 and PDP-11/24 Unibus CPU designs, as well as VAX 11/750, VAX 11/780, VAX 11/785 and VAX 8600 32-bit CPUs.

Sequential Spin-up

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

On-Board WOMBAT Utilities

Webster Omnipotent Mass Builder and Tester (WOMBAT) is an interactive formatting and diagnostic utility totally contained within the WUSMD 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 minimal additional hardware present.

WOMBAT can also load a simple console communication program into the host computer's memory or it can be invoked on system power-up. No external software, media, or program loading device is required in maintenance of the WUSMD 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 a test pattern through the entire recording surface. WOMBAT prompts the user at the terminal to supply parameters such as drive geometry (cylinders, heads and sectors) and various other options. This data is stored twice in special reserved areas of track zero and retrieved by a simple homeseek-read sequence at each power-up. No special PROMs or switch settings are required to fully characterize the connected disc drives.

WOMBAT Self Diagnostics

The WUSMD contains a comprehensive set of self diagnostic procedures which are executed automatically on power-up. Failures are reported via a flashing red LED. WOMBAT examines the device registers then reports the nature of the fault to the console.

WOMBAT Interactive Diagnostics

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

Chapter 2
WUSMD Installation

2.1 WUSMD Switch Settings

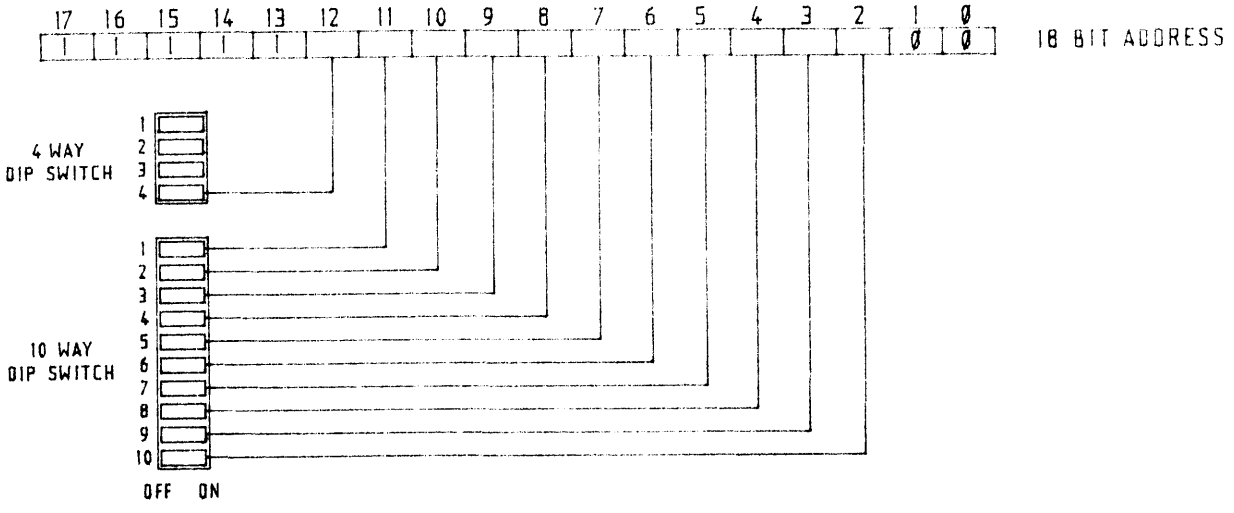
The WUSMD can be configured by selecting the appropriate switch combination. The switch positions for base (CSR) address, on-board bootstrap select, power up option and removable media are given below. Note that there is no interrupt vector switch as this is set automatically by the operating system software.

Base CSR Address

The base CSR address should be selected according to the rules set out in Chapter 5 - Operating Systems. It is important to locate the controller correctly to enable the operating system to identify the device and its type.

Switches 1 to 10 of the 10-way DIP switch and switch 1 of the 4-way DIP switch are used to set the desired base CSR address anywhere within the I/O page. Each switch corresponds to a Unibus address bit. The following figure shows how the switches relate to the 18-bit Unibus address.

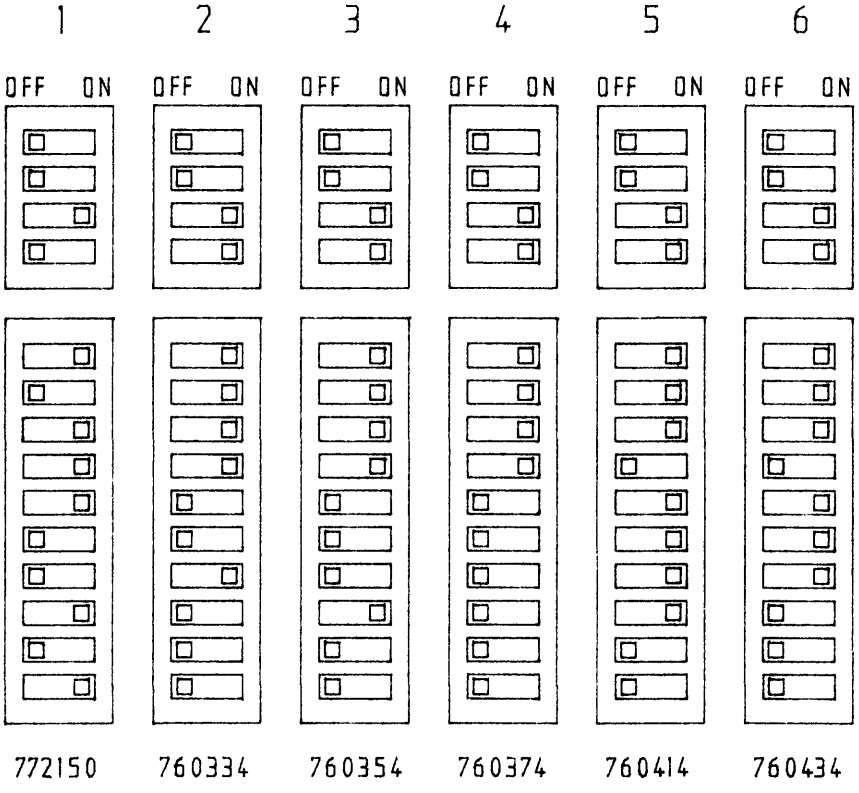
FIGURE 2-1 CSR ADDRESS SELECTION



NOTE - Setting a switch to the OFF position assigns a one to the corresponding address bit.

The following figure shows WUSMD CSR address examples.

FIGURE 2-2 CSR ADDRESS EXAMPLES



NOTE - The top three switches of the 4-way dip switch select the boot option. The third switch on as shown selects a boot address of 771000.

On-Board Bootstrap

The on-board bootstrap is provided to allow systems to boot from the controller. The bootstrap procedure is given in Section 2.5. The bootstrap must be disabled before being installed in a VAX.

Switches 1 to 3 of the 4-way DIP switch allow selection of the bootstrap options as given below.

TABLE 2-1 BOOTSTRAP SELECT

SWITCH NUMBER			FUNCTION
1	2	3	
OFF	OFF	OFF	On board boot disabled
ON	OFF	OFF	On board boot at address 773000
OFF	ON	OFF	On board boot at address 774000
OFF	OFF	ON	On board boot at address 771000

Power Up Mode

Switches 1 and 2 of the 8-way DIP switch allow different power up options as shown below.

TABLE 2-2 POWER UP MODE

SWITCH NUMBER	SETTING	FUNCTION AT POWER UP
1	ON	Not DP8466 Rev G
	OFF	DP8466 Rev G
2	ON	Normal Power Up
	OFF	Start WOMBAT on Power Up

Switch 1 is set to correspond to the version of disk controller chip used on the WUSMD. This switch is correctly set at the factory, so there should be no need to check its setting unless the disk controller chip is changed.

If it is necessary to check the setting of this switch, firstly locate the disk controller chip on the WUSMD. It is a large 48 pin device located at U88.

If the part number is "DP8466AN-20" or "DP8466AN-25", then it is a "Rev G" chip and switch 1 should be set to the "OFF" position.

Any chip that does NOT have the "A" immediately after the DP8466 in the part number is NOT a "Rev G" chip, and switch 1 should be set in the "ON" position.

Removable Media

Switches 3 to 6 of the 8-way DIP switch allow you to specify fixed or removable media for each drive.

TABLE 2-3 REMOVABLE MEDIA

SWITCH NUMBER	FUNCTION
3	Drive 3 Fixed/Removable
4	Drive 2 Fixed/Removable
5	Drive 1 Fixed/Removable
6	Drive 0 Fixed/Removable

NOTE - Switch ON specifies Fixed Media.
Switch OFF specifies Removable Media.

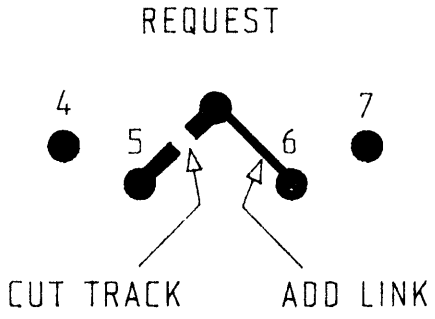
WARNING - If Removable Media has been specified the drive must be powered up, otherwise the controller will "hang".

Interrupt Priority Level

The WUSMD interrupt priority is permanently configured to level 5 by printed circuit board traces. This setting will suit most systems but can be changed, if desired, to any other level by cutting traces and adding wire links. The area of the board where the changes are made is identified by REQUEST and GRANT silkscreen labels near the centre of the edge connector.

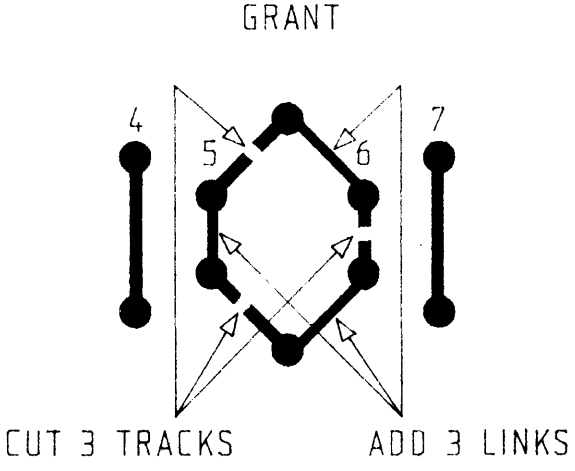
As an example, to change the interrupt priority from level 5 to 6, cut the track under the REQUEST label which currently connects to Request level 5. Add a wire link connecting to Request level 6 as shown in Figure 2-3.

FIGURE 2-3 EXAMPLE OF INTERRUPT PRIORITY CHANGE (PART A)



Cut two tracks connecting to level 5 under the GRANT label and cut the track connecting level 6. Add three wire links as shown in Figure 2-4.

FIGURE 2-4 EXAMPLE OF INTERRUPT PRIORITY CHANGE (PART B)



Note : The Request and Grant levels must be the same.

2.2 KESDI/T RS232 Maintenance Terminal Adaptor

The KESDI/T RS232 maintenance terminal adaptor allows the controller to be connected to an ASCII terminal. It consists of a 10-way flat cable with a DB25S connector on one end and a 10-way insulation displacement type flat cable socket on the other.

The communication format is :

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

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

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 2-4 KESDI/T RS232 MAINTENANCE TERMINAL ADAPTOR

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

2.3 Connecting Multiple Drives

The WUSMD can simultaneously support four SMD drives which can be of different size, speed and data rate. The procedure for connecting four drives is simple. Data cables are connected in radial fashion to edge connectors on the WUSMD. There is no particular order of connection. The control cable is daisy chained between each drive with the last drive being terminated.

Pre-configured data and control cable kits for one, two, three or four drives can be supplied to minimise difficulties when installing the WUSMD or where adding additional drives (LWUSMD/1, LWUSMD/2, LWUSMD/3, LWUSMD/4).

Cables should be checked for correct installation by ensuring that Pin 1 on the edge connector aligns with Pin 1 on the cable. Pin 1 is

generally indicated by the arrow mark on the edge connector shroud. If there is no shroud Pin 1 is the top right pin with the edge connectors facing you. Pin 1 on the cable is usually indicated by a coloured stripe on the appropriate cable.

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

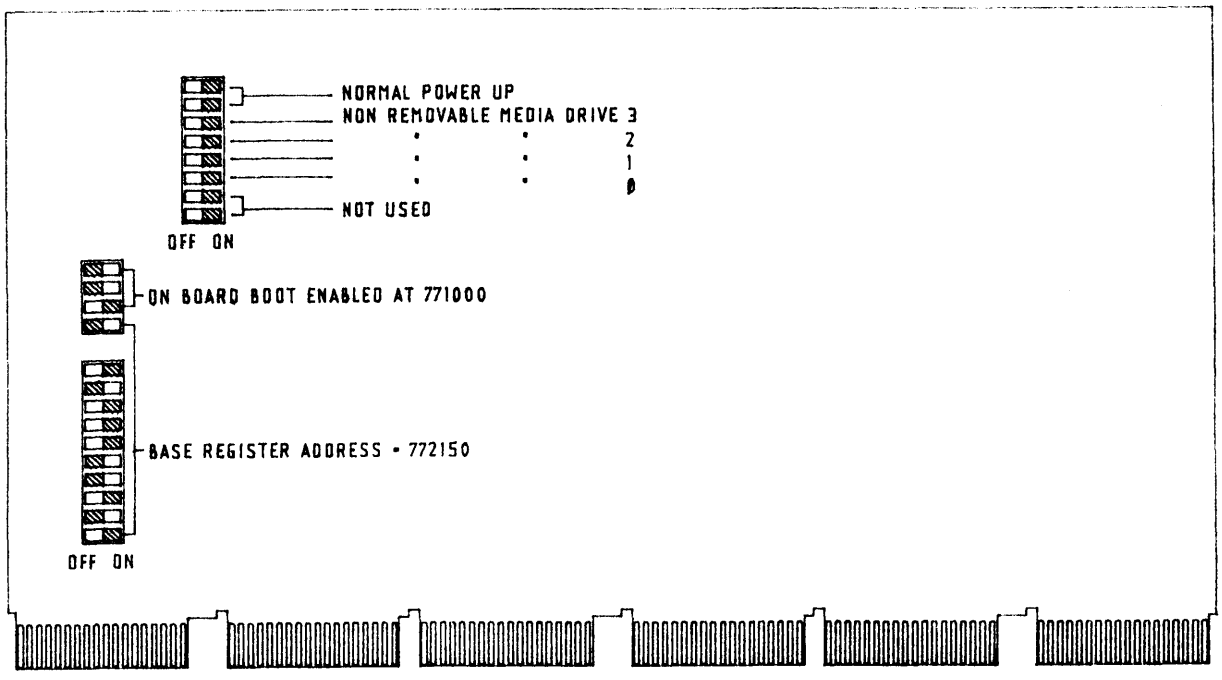
2.4 Front Panel Connections

If required, a front panel can be connected to J1, the front panel/maintenance connector. There are three possible options - an active panel, a passive panel or no panel. The active panel provides write protection for all drives, an on/offline button and a drive ready indicator. The passive panel only provides write protection and access light. The active panel must contain some digital circuitry and be separately powered to support these functions. Webster Computer Corporation can provide full details on request.

TABLE 2-5 PASSIVE 10-WAY FRONT PANEL/MAINTENANCE CONNECTOR

Pin No.	Front Panel Function	Maintenance Function
7	Drive 1 Write Protect Input Connecting this input to Ground will write-protect the drive. Has an on-board 330 ohm pull-up.	RS232 Enable
8	Drive 0 Write Protect Input Connecting this input to Ground will write-protect the drive. Has an on-board 22k ohm pull-up.	RS232 Input
1	Drive 3 Write Protect Input Connecting this input to Ground will write-protect the drive. Has an on-board 330 ohm pull-up.	
10	Drive 2 Write Protect Input Connecting this input to Ground will write-protect the drive. Has an on-board 330 ohm pull-up.	
3	Access Light. Indicating access to any drive or Cache. This output can be used to drive an access light.The levels are: ACCESS -5 v through 1.5k ohm NO ACCESS : +4.5 @ 20mA max.	RS232 Output Data
5	TTL signal indicating access to Drive 0. Low true.	
6	TTL signal indicating access to Drive 1. Low true.	
2,4,9	Ground	Ground

FIGURE 2-5 WUSMD FACTORY SWITCH SETTINGS



2.5 Bootstrap Procedure (PDP-11 only)

If the WUSMD bootstrap is enabled the following occurs :-

1. On initialisation location 773000 responds to the CPU fetch instruction with a BR .+2 (400) instruction to address 773002 where the CPU loops (BR . (777)).

2. A "jump to zero" instruction (JMP @#0) is loaded into location zero.

3. The controller program forces the CPU to start executing at location zero by changing the contents of location 773002 to "CLR PC".

4. Bootstrap code is loaded into host memory at location 2000 and the rest of memory is cleared.

5. The controller changes location 2 from zero to 2000 changing the JMP @#0 to JMP @#2000, starting the execution of the loaded bootstrap program and the following message is typed :-.

BOOT V [version number].

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

Booting from DU0:

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

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

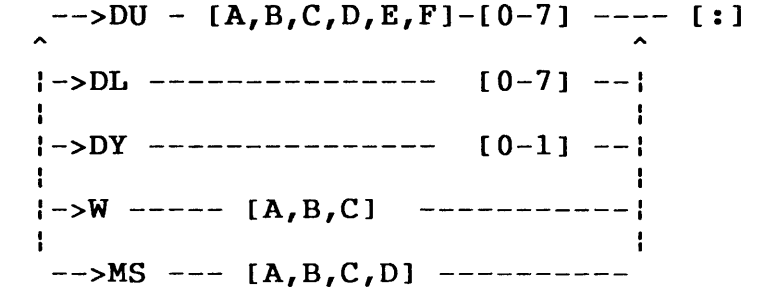
>

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

The following is the syntax of the WUSMD bootstrap procedure :-

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

such that :-



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

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

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

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

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

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

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

9. The selected controller is commanded to read the boot block (block zero) from the specified device into the host memory and then waits to be initialised. The host computer commences execution of the instructions in the boot block. The devices "assumed" by the WUSMD bootstrap are detailed in Table 2-6.

TABLE 2-6 CSR ADDRESSES FOR WUSMD BOOTSTRAP

DEVICE	ADDRESS
CONSOLE	177560
DU:	172150
DUA:	172150
DUB:	160334
DUC:	160354
DUD:	160374
DUE:	160414
DUF:	160434
MS:	172520
MSA:	172520
MSB:	172524
MSC:	172530
MSD:	172534
	0
	0
	0
DY:	177170

NOTE: When starting WOMBAT the letter "W" may be followed by any letter available for "DU".

Chapter 3
WUSMD WOMBAT Utilities

3.1 Starting Up WOMBAT

WOMBAT (Webster Omnipotent Mass Builder and Tester) provides a con troller resident means of formatting, testing and maintaining the drive and controller subsystem. All WOMBAT functions are menu driven and are designed to simplify the process of structuring, formatting and testing drives.

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

(a) By selecting the '**W**' option during the system boot-strap operation if the WUSMD boot is enabled.
The letter "W" may be followed by any letter allowed for "DU" to use any of the available CSRs.

(b) By connecting a 9600 baud terminal to the WUSMD Front Panel connection and depositing the appropriate data pattern using ODT.

(c) By connecting a 9600 baud terminal to the WUSMD and configuring the switches on the PCB to automatically run WOMBAT.

(d) By depositing a special data pattern in the WUSMD using ODT from the system console.

We recommend using option (a) if you are running a PDP-11. You should always, for PDP-11's, have the optional boot enabled at 171000. You may then start the boot by starting the CPU at 171000. You may then boot from any supported device or run WOMBAT by entering 'W'.

Use option (d) if you are running a VAX. It enables you to configure any discs on the controller from the system console. See the procedure for your VAX set out below.

WOMBAT can be run independently of, or without a CPU for controller testing or engineering purposes if necessary. First disable the controller bootstrap and set the power-up mode to 3. (See Table 2-1 and 2-2). Connect a 9600 baud terminal to the Front Panel connector and position the controller in the backplane. WOMBAT will be invoked automatically on power-up or by pressing reset. If no CPU is present then the backplane must be correctly terminated and a bus intialisation signal (BINIT) must be generated.

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

Table 3-1 lists the octal codes used to invoke WOMBAT using console ODT.

TABLE 3-1 WOMBAT INITIALISATION CODES

NUMBER octal	NUMBER hex	FUNCTION	PROCESSOR
000250	00A8	WOMBAT	PDP-11
000254	00AC	WOMBAT	VAX
000260	00B0	WOMBAT	On board port

Set out below are the common CSR addresses used for MSCP compatible controllers. Note that the WUSMD controller allows ANY CSR addresses to be set up, but the CSR's in the table below are the ones expected by DEC operating systems.

TABLE 3-2 COMMON WUSMD CSR ADDRESSES

ADDRESS	VAX11/750 UBI	VAX11/750 SUB	VAX11/780 (first Unibus adapter)
772150	FFF468	FBF468	2013F468
760334	FFE0DC	FBF0DC	2013E0DC
760354	FFE0EC	FBF0EC	2013E0EC
760374	FFE0FC	FBF0FC	2013E0FC
760414	FFE10C	FBF10C	2013E10C
760434	FFE11C	FBF11C	2013E11C

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

In each case input the appropriate CSR address for the controller to be accessed where indicated by 'CSR'.

PDP-11 CPU

The following details the procedure for invoking WOMBAT on a PDP-11 CPU system using console ODT. You must 'reboot' to exit WOMBAT.

Halt the processor.

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

Unfortunately some PDP-11's, such as the 11/04 and 11/34, do not implement a version of console ODT that enables you to start the WOMBAT communication routine without a bus reset. A bus reset, to the controller, means 'wait for the MSCP initialisation sequence'. This bus reset aborts WOMBAT and enters the controllers MSCP routines - which is not very helpful if you really want to run WOMBAT!

With these PDP-11's you must enter the following small program:

```
10000/      12737       ;MOV
10002/      250         ;#250,
10004/      'CSR'       ;controllers CSR - e.g. 172150
10006/      0           ;HALT
10010/      137         ;JMP
10012/      2000        ;#2000
```

Start the program. It will halt to give the controller time to load the WOMBAT communication routine into memory. When you 'Continue' the processor will start WOMBAT running.

VAX CPU

The following details the procedure for invoking WOMBAT in VAX computers using ODT. You must 'reboot' to exit WOMBAT.

VAX 11/750

When the 750 halts at the end of its initialisation phase:-

>>>	**D/L F30808 80000002**	(use F32808 in SUB - set up the appropriate Unibus map entry)
>>>	**D/W 'CSR' AC**	(ask WOMBAT to load the communications program into memory)
>>>	**D/L 483 'CSR'**	(patch the correct CSR address into the comm-unication program)
>>>	**D/G F 400**	(set up the start address of the communication program)
>>>	**C**	(start the program without a bus reset)

VAX 11/780

When the 780 halts at the end of its initialisation phase:-

>>>	**D/L 20006808 80000002**	(set up the appropriate Unibus map entry)
>>>	**D/W 'CSR' AC**	(ask WOMBAT to load the communications program into memory)
>>>	**D/L 483 'CSR'**	(Patch the correct CSR address into the comm-unication program)
>>>	**S 419**	(Start the program)

3.2 WOMBAT Menu options

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

```
      WUSMD   WOMBAT Version: 2.3


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

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

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

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

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

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

3.3 Master Menu Options

Option 1 - Structure Disc

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

```
      ** Disc Structure Menu **

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

Option 2 - Test Disc

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

```
** Disc Test 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
6   Read Physical Block
7   Display Error Statistics
8   Zero Error Statistics
9   Test Cache, RAM & ROM
```

Option 3 - Manage Bad Blocks

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

```
** Bad Block Management Menu **

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

Option 4 - Display Error

Selecting this option causes WOMBAT to display a message which explains the most recent controller detected fatal error. Refer to Table 4-6. The occurence of an error is indicated by the controller hanging with its red LED flashing. If no error has occurred this option produces a meaningless error message.

Option 5 - Shadow Options

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

```
** Shadow Options Menu **

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

Setting Up a New Disc

The procedure for structuring a new disc is as follows:

1 Create disc structure
2 Format the disc
3 Write the disc structure
4 Pattern test the disc
5 Replace bad blocks

Once these 5 steps have been undertaken the host operating system may use the disc.

3.4 Disc Structure Menu Options

**** Disc Structure Menu ****

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

Option 1 - Create Disc Structure

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

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

The create disc structure dialogue is divided into three parts.

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

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

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

Drive Structure Specification

Cylinders: The number of cylinders on the drive.

Heads: The number of heads on the drive is displayed.

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

Track spiralling: Track spiralling improves disc performance on data transfers over more than one track. Sector 0 on each track is offset by a nominated factor to allow head select and positioning before sector 0 on the next track is reached. The recommended factor is four. Note that the Track Spiralling factor must not be changed after the disc is formatted. If it is changed, incorrect bad block replacement will occur. If you wish to try a different Track Spiralling factor to that with which the disc is currently formatted, then the disc MUST be reformatted with the new Track Spiralling factor.

Interface type: Enter the correct value for the SMD interface type being used. It is possible to have a different SMD interface for each drive connected to the controller. The values are :
0 - Standard
1 - Modified
2 - Enhanced
3 - Extended
4 - SMD-E

Optimisation: The seek optimisation strategy can be either:

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

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

2 (Elevator) - Processes requests as it moves in one direction along the 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.

3 (Forward) - This processes requests from the

lowest cylinder number to the highest in one direction only.

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

Fairness count: The fairness count determines the number of times an I/O request will be passed over by the controller's seek optimisation setting before it is executed. A reasonable count for normal use would be around 25. Every time a request is passed over its fairness count is decremented. When that count reaches zero that request will be selected, no matter what optimisation strategy is in effect.This count has no effect if no optimisation is selected.

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

Unit Structure Specification

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

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

Serial number: The MSCP protocol returns a 32-bit volume serial number as a part of its response when an

"on-line" command is issued. WOMBAT defaults this field to zero. To change this enter the desired serial number. This field is used, for example, by RSX-11M-PLUS, when a disc is initialised with the "INI DU:" command. It sets up the volume serial number.

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

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

Controller Wide Tuning Parameters

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

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

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

Front panel type: This option allows the correct selection of front panel type. See 2.4 Front Panel Connections.

0 - None
1 - Passive
2 - Active

Note with version 2.4 (or later) firmware the front panel type is set to a default value of 0 (none). Earlier firmware versions are default 1 (passive).

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

Option 2 - Format Disc

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

Option 3 - Write Disc Structure

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

Option 4 - Update Header Blocks

This is similar to Write disc structure except that the replacement block table is not written, thus preserving any blocks which may have been replaced.

Option 5 - Display Disc Structure

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

Option 6 - Change Unit Number

It is sometimes necessary to change a unit number in order to resolve a duplicate unit number or to satisfy operating system requirements. This method is a safe and simple way of doing so. WOMBAT prompts for a unit number on the current drive, and then for the new number for that unit.

Display Statistics

Statistics about disc and cache usage are maintained, and recorded on the disc once per 1000 disc accesses. They are displayed as :

Controller statistics report

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

# of writes xxxx

Drive statistics report

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

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

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

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

3.5 Disc Test Menu Options

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. The disc structure must be written to the disc before bad blocks can be replaced.

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

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

2. If WOMBAT is running from the Console terminal, type CTRL/C.

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

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

**** Disc Test 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
6 Read Physical Block
7 Display Error Statistics
8 Zero Error Statistics
9 Test Cache, RAM & ROM

Option 1 - Read All Disc

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

```
Pass: 1. Errors: 0.  Passes Since Last Error: 0
Pass: 2. Errors: 0.  Passes Since Last Error: 1
```

This function does not destroy any information.

Option 2 - 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    (Error message)
Pass:1. Errors: 1.  Passes Since Last Error: 0
Pass:2. Errors: 1.  Passes Since Last Error: 1
```

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

Option 3 - Pattern Test

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

Option 4 - Random Writes

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

Option 5 - ECC Validation

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

Option 6 - Read Physical Block

WOMBAT prompts for a block number anywhere on the disc. It then converts that block number into a physical address consisting of cylinder, head, and sector, and displays these values. Then it reads that sector and displays a message indicating the success or failure of the read.

Option 7 - 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.
```

Option 8 - Zero Error Statistics

Zeroes the error statistics table & redisplays Test Menu options.

Option 9 - Test Cache, RAM & ROM Option

CAUTION

> Execution of the Test Cache option causes drive table parameters stored in RAM to be destroyed. Drive error statistics will also be lost.

This test continuously writes test patterns throughout the entire cache and reads them back testing for veracity. A separate part of the test automatically checks that the parity logic is functioning correctly by forcing incorrect parity and checking that an error occurred. The cache pattern tests use special microcode instructions which iteratively read and write large blocks of cache memory at high speed. The Static RAM at L5 is also tested and the code PROM is Checksummed.

With version 2.4 (or later) firmware the Test Cache option also reports the size of onboard cache memory and the location of any failed memory.

3.6 Bad Block Management Menu Options

**** Bad Block Management Menu ****

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

Option 1 - Manually Replace Bad Blocks

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

Option 2 - Automatically Replace Bad Blocks

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

Option 3 - Display Replaced Bad Blocks

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

Option 4 - Enter Defect Map

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

Enter defect map for drive n

Enter all values in decimal

Bytes per sector: The number of bytes per sector as set up in the drive switches (hard sector drives) or calculated using the procedure in section 3.12.

Cylinder: Cylinder number of defect, RETURN or ^C if no more.

Head: Head number of defect.

Bytes past index: Location past index of defect.

Bit **length of defect:**Length of defect in BITS.

This data is then used to compute the address of a block on the disc. If it does not match a block on the specified disc track the error message "!! Beyond last sector" is produced. This may mean that the defect is located beyond the last data sector on the track.

3.7 Shadow Options Menu Option

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

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

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

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

Shadow options menu

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

Option 1 - Shadow Copy Options

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

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

1 - **On power up**	This causes the copying to be done whenever the drive(s) are first powered up, or WOMBAT has been invoked.
2 - **Not ready/ready**	The copying is to be done whenever the controller detects the shadow unit going from a not ready to ready condition - e.g. a removable drive being bought on-line.
3 - **Power up or Not ready/ready**	A combination of 1 and 2 above.

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

Option 2 - Set Shadow Units

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

Primary unit:	The unit to be shadowed.
Shadow unit:	The unit number of the shadow.

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

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

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

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

WARNING:- a potentially serious problem exists with the automatic copy of data from the primary to the shadow at power up (shadow copy types 1 or 3). If the primary has been corrupted, or the data is invalid, IT WILL DESTROY YOUR BACKUP. If you have trouble with your primary unit use WOMBAT to change it from a shadow primary. You may

even want to change its unit number via the 'Change Unit number' option, and then bring in the shadow unit as its replacement. Please realize that the controller MUST assume the validity of the primary unit at the first initialization after power up if you have selected one of the power up copy types.

Option 3 - Reset shadow units

This option allows disabling of an assigned shadow unit :

Primary unit: The primary unit.

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

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

Option 4 - Copy Unit To Unit

To copy a primary unit to its shadow :

Primary unit The number of the unit you want copied.

Shadow unit The unit you want the data copied to.

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

Option 5 - Compare Unit to Unit

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

Primary unit One of the units.

Shadow unit The other.

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

3.8 WOMBAT Disc Structure

WOMBAT records the logical structure of the drive on track 0. All of track 0 is reserved for this and other testing purposes. The user area begins at the next block after track 0. This is the same as the number of sectors per track. The user area extends to the block before the beginning of the Replacement Control Table (RCT). The

size of the RCT is fixed and is always 2 tracks. The 2 tracks are accessed by different heads and contain identical data. All the blocks from the end of the second RCT track to the end of the media are reserved for replacement blocks.

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

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

3.9 Drives with Removable Media

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

3.10 Error Recovery Procedures

When a data error occurrs the transfer is retried 10 times. If any of the retries is successful, then the block is not replaced. Otherwise, the drive is recalibrated and a further 10 retries are executed. If any of these succeeds then the block is not replaced. If not, the block is assumed to have a hard error. When a hard error

is detected the error correction algorithm is invoked and dynamic replacement is scheduled.

When a hard error is detected, a quick check is made to verify that the problems were not caused by a drive hang. A seek to cylinder zero is performed followed by a read of block zero. Then a seek to the RCT cylinder is performed followed by a read of the first block of the RCT. If all of this succeeds then the drive is deemed to be 'OK' and the controller then proceeds with dynamic replacement.

When it has decided to replace a block, the controller first writes the data from the bad block to a special area of the rct together with a flag to indicate whether or not ECC was successful in recovering the data. This transfer is done as a write check operation. Next the bad block is tested by write checking data patterns to it a number of times. If the block appears good after this test, then it is assumed that the error was transient and the original data is written back to the block. The block is not replaced. If however the block is confirmed as being bad, then a new block is allocated to replace it, and the previously saved data is written to it.

When the saved data is written to the replacement block, the ECC success flag is tested. If it indicates that the data was irrecoverable then the block will be written with forced error status.

3.11 Drive Shadowing

The intent of this feature is to provide a continuous automatic backup of important data. You must first specify a unit to be the shadow unit, that is, the unit onto which the data will be backed up. The primary unit can then be specified - this is unit which the host will see. The normal case would be to have two identical drives with a single unit on each. The primary unit must be of identical size to the shadow. Once the primary/shadow pair has been defined, then the shadow unit will become invisible to the host. All writes directed to the primary will also be written to the shadow. If a read is directed to the primary, the controller will attempt to redirect that read to whichever drive of the pair has its heads positioned closest to the requested data.

This feature will only offer protection against the primary unit

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

If the primary does fail, then you will want to recover the data from the shadow. To do this WOMBAT must be run. First, disable the shadow unit with the "Reset shadow units" option of the Shadow Options Menu. Then (optionally) change the unit numbers of the two units with the "Change unit number" option of the Disc Structure Menu. Once the data has been recovered, the primary unit can be serviced and re-installed in the system.

3.12 Computing Sectors Per Track

To compute the number of sectors per track first determine the number of bytes per track from the drive manual. Divide this figure by the number of bytes per sector. The controller requires at least 596 bytes per sector. The drive switches should be set to provide the appropriate number of sectors per track. In some cases this will result in a short sector at the end of the track. If the drive has a short sector WOMBAT must be told when structuring the disc.

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

The unformatted capacity per track is 20,160 bytes. This is divided by 596 giving the number of sectors per track.

$$\frac{20{,}160}{596} = 33.823 \text{ sectors per track}$$

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

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

$$\frac{30{,}240}{596} = 50.738 \text{ sectors per track}$$

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

3.13 WOMBAT Error Messages

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

sector not found	Indicates that the sector asked does not exist or cannot be located.
drive fault	Indicates that the drive is faulty and that service is necessary.
drive timed out	Indicates that the drive has failed to complete an operation.
data field error	Indicates that bad data exists in a sector.
controller fault	Indicates controller failure. Service will be necessary.
block marked as bad	Indicates that the block has been flagged as bad. The controller will refer to the RCT for a replacement block.
data late	Data is lost due to internal overflow in controller memory before transmission over the bus to host.
forced error	A forced error occurs when a good block has bad data. The block is flagged forced error until good data is written to the block.
seek error	An error has occured on a seek operation.
rct full	The replacement control table is full. An error of this kind indicates that the disc has too many errors to be serviceable.
rct read error	Fatal error which indicates the controller cannot read the RCT.
rct write error	Fatal error which indicates the controller cannot write the location of replacement blocks.
illegal sector specified	A sector with an illegal number has been specified.
illegal block number	A block with an illegal number has been specified.
non existent drive	A drive has been specified which does not exist.

unit table full	The unit table is full. The maximum number of units is 16. To create a new unit an existing unit must be undefined.
no drive selected	A drive must be selected before WOMBAT can perform any operation. Select any valid drive.
non-existent unit	The unit selected has not been defined.
disc structure write error	WOMBAT is unable to write the structure to the disc. This is a fatal error indicating that the drive is not serviceable.

3.14 WOMBAT Self-Diagnostics

Initialization Procedures

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

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

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

Chapter 4
WUSMD MSCP Programming

4.1 Overview of MSCP

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

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

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

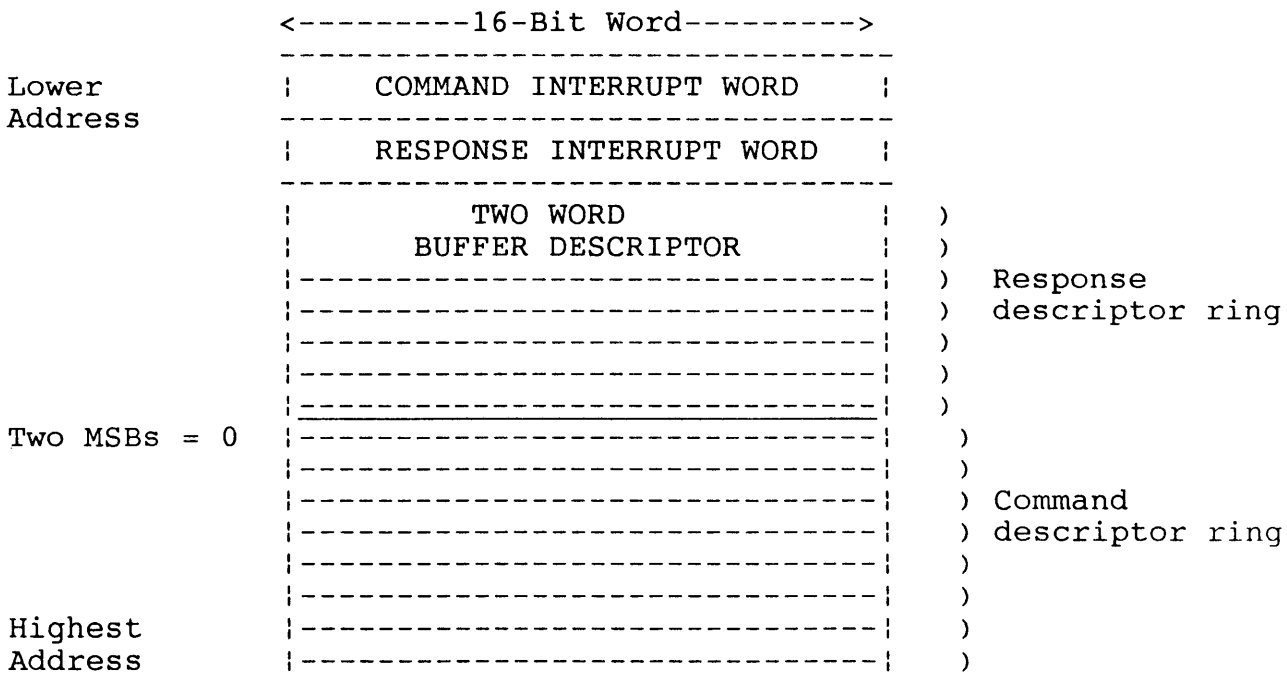
Disc drive parameters are transparent to both the host and controller resident layers of MSCP. The disc drive passes factors such as disc geometry, storage capacity or error retry counts to the disc controller on system start-up.

In addition to relieving the host of disc-specific data, the disc controller and disc provide the host with "clean" data. The disc drive handles some positioner errors entirely by itself but performs 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 a header area containing interrupt identification words and a variable-length section containing the response (receive) and command (send) lists, organised into ring buffers. Figure 4-1 shows the memory communications format.

FIGURE 4-1 MEMORY COMMUNICATIONS FORMAT



Command and Response Rings

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

FIGURE 4-2 DESCRIPTOR FORMAT

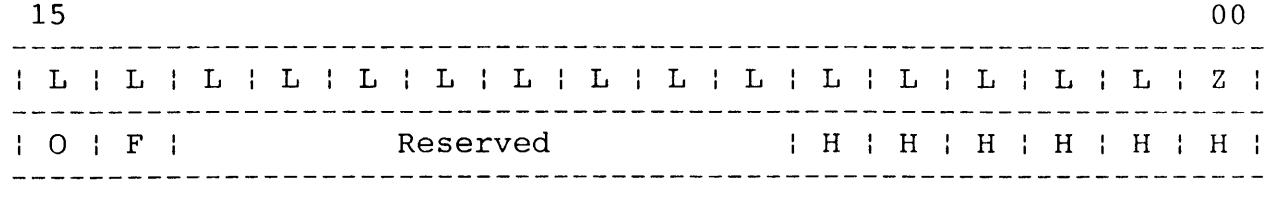


TABLE 4-1 COMMAND RING CODE DESCRIPTIONS

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

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

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

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

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

Message Packets

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

FIGURE 4-3 MESSAGE ENVELOPE FORMAT

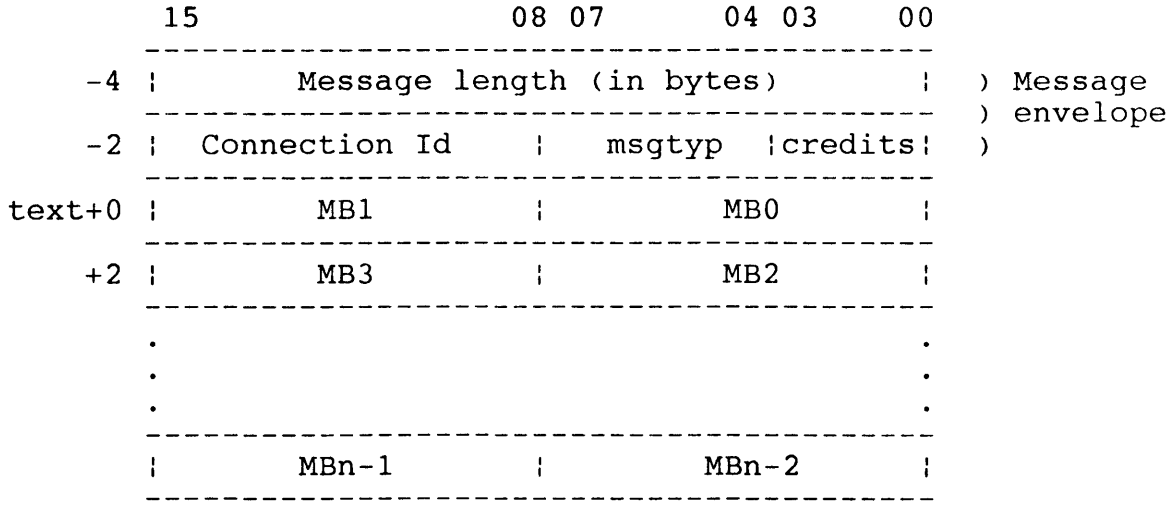


TABLE 4-2 WORD ENVELOPE CONTENTS

WORD	ENVELOPE CONTENTS
0	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 Unibus master for data transfer
   Failure to become Unibus master for interrupt
   Failure to access I/O page registers or communication area
   Unibus 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 Unibus addresses.

4.5 Initialization

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

Initialization Process

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

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

TABLE 4-3 INITIALISATION 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 Unibus 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 WUSMD are the Initialize and Poll register (IP) and the Status and Address register (SA).

Initialize and Poll Register (IP)

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

TABLE 4-4 WUSMD INITIALISATION WORDS

NUMBER octal	NUMBER hex	FUNCTION	PROCESSOR
000250	00A8	WOMBAT	PDP/11
000254	00AC	WOMBAT	VAX
000260	00B0	WOMBAT	On board port

Status and Address register (SA)

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

When read by the host during initialization, it communicates data and error information relating to the initialization process. When written by the host during initialization, it communicates certain host-specific parameters to the controller.

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

4.7 MSCP Commands

TABLE 4-5 WUSMD MSCP COMMANDS

COMMAND	FUNCTION
Access	Reads data from the specified unit.
Abort	Guarantees that referenced MSCP command will complete within the controller timeout period.
Available	If specified unit is on-line, returns it to the unit-available state. If specified unit is currently in the unit-available state, this command has no affect.
Compare Host Data	Reads data from the 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 48 bit ECC (error checking and correction) polynomial with an 11 bit correction span. ECC will first try to read or write a block up to 10 times before attempting to correct the error. If error correction fails a non-recoverable error is reported. Table 4-7 details the MSCP status code messages.

4.9 Fatal Controller Error

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

TABLE 4-6 FATAL CONTROLLER ERRORS

ERROR		DESCRIPTION
octal	hex	
125413	AB0B	RAM test failed
126013	AC0B	ROM test failed
126413	AD0B	Cache test failed
127013	AE0B	Cache unlock error
127413	AF0B	Qbus error - get host message
130013	B00B	Qbus error - clear host rings
130413	B10B	Buffers ; free buffer count error
131013	B20B	Queue empty - unfork
131413	B30B	Queue empty - wait
132013	B40B	Queue empty - pause
133013	B60B	Cache parity

TABLE 4-7 MSCP STATUS CODE MESSAGES

MESSAGE	MEANING
Command Aborted	The current command was aborted before it could be completed normally.
Compare Error	While performing a Compare command, a Discrepancy was found while comparing the disc data to the host data.
Controller Error	The WUSMD 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 WUSMD controller found some field in the command to be in error.
Success	The command was successfully completed.
Unit Available	The WUSMD controller is not on line, but it can accept an On Line command from the host.
Unit Offline	The WUSMD controller is not on line, and it cannot be brought on line.
Write Protected	A Write or Erase command was attempted to a unit that is logically write-protected.

Chapter 5
WUSMD Operating Systems

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

5.1 Operating Systems Overview

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

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

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

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

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

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

MSCP Devices

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

Unibus Addresses

The standard Unibus address of 772150 (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, Unibus 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 5-1 is a list of device names for five operating systems. Two controller and device names are given to indicate the numbering scheme.

TABLE 5-1 DEVICE NAMES IN DEC OPERATING SYSTEMS

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

5.2 RT-11 Operating System

Installation of a Single Controller

A single controller is installed at the Unibus address of 772150 (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 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 Unibus 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 CSR0     =   772150    (DEFAULT)
SET DU CSR1     =   760334
SET DU VECTOR   =   154       (DEFAULT)
SET DU VEC2     =   160
```

The second device can be at any unused address on the Unibus 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 Under RT11

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

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

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

2. The total number of logical 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
```

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

RSTS/E can support two MSCP type controllers. The first is located at the standard Unibus address (772150 octal) while the second can be located in floating address space. However, the recommended address for the second controller is 760334. A controller must be located at the standard Unibus 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.

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

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

The RSX-11M SYSGEN program is an interactive program that builds a complete and running RSX-11M system for a particular hardware configuration. RSX-11M SYSGEN supports autoconfigure. This program detects MSCP type controllers located at standard Unibus 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 Unibus address of 772150 (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 overridden. Answer **N** (no) to the following :-

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

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

4. Indicate the number of devices that are installed.

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

   Enter the value 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, Unibus 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:154,772150,,4,4]
   * **154,772150,3,4,4<return>**

Vector and Unibus 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 Unibus address of 772150 (octal) is used for the first controller, while the second can be 760334 (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 discettes. 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 WUSMD, 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.

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

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

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

Installing a Single Controller

A single controller is installed at the standard Unibus address of 772150 (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 WUSMD 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 772150 (octal) is used for the first controller, while the second can be 772154 (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.

Warning: Versions of RSX-11M Plus prior to 3.0C support discs of a maximum size of 1,044,480 blocks. Larger drives must be broken up into smaller virtual units.

5.6 MicroVAX/MicroVMS Operating Systems

The first WUSMD controller is located at the standard bus address of 772150 (Octal) and the second in floating address space. The MicroVMS SYSGEN utility can determine the Unibus 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 Unibus 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 Unibus 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 Unibus 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 Unibus by logical name. Devices with floating bus and vector addresses should be noted if it is intended to re-install them with the WUSMD 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 Unibus 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 Unibus 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 WUSMD by selecting the correct pin configuration on the PCB.

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

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

5.7 Autoconfigure

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

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

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

TABLE 5-2 SYSGEN DEVICE RANKING

Rank	Device	Number of Registers	Octal Modulus	Rank	Device	Number of Registers	Octal Modulus
1	DJ11	4	10	17	Reserved	4	10
2	DH11	8	20	18	RX11[2]	4	10
3	DQ11	4	10	18	RX211[2]	4	10
4	DU11, DUV11	4	10	18	RXV11[2]	4	10
5	DUP11	4	10	18	RXV21[2]	4	10
6	LK11A	4	10	19	DR11-W[3]	4	10
7	DMC11	4	10	20	DR11-B[3]	4	10
7	DMR11[1]	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[2]	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[2]	4	10	30	TU81	2	4
14	RL11[2]	4	10	31	KMV11	8	20
14	RLV11[2]	4	10	32	DHV11	8	20
15	LPA11-K[2]	8	20	33	DMZ32	16	40
16	KW11-C	4	10	34	CP132	16	40

[1] DZ11-E and DZ11-F treated as two DZ11s.
[2] The first device of this type has a fixed address while extra devices have floating addresses.
[3] The first two devices of this type have fixed addresses while extra devices have floating addresses.

An eight byte gap must also be reserved in floating address space for each device type not currently installed in the system. This gap must start on the proper boundary. See Table 5-6 for an example of gap placement.

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

TABLE 5-3 DEVICE REGISTERS AND WORD BOUNDARIES

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

Vector Addresses and Autoconfiguration

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

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

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

TABLE 5-4 FLOATING VECTOR ADDRESS DEVICE PRIORITY RANKING

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

continued on next page...

TABLE 5-4 FLOATING VECTOR ADDRESS PRIORITY RANKING

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

[1] KL11 or DL11 have fixed vectors when used as a console.
[2] The first device has a fixed vector all subsequent device of the same type have a floating vector.
[3] ML11 is a Mass Bus device which connects to the Qbus or Unibus via a bus adaptor.

System Configuration Example

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

TABLE 5-5 CSR AND VECTOR ADDRESS EXAMPLE

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

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

TABLE 5-6 FLOATING CSR ADDRESS ASSIGNMENT

Installed	Device		Octal Address
	DJ11	Gap	760010
	DH11	Gap	760020
	DQ11	Gap	760030
	DU11	Gap	760040
	DUP11	Gap	760050
	LK11A	Gap	760060
	DMC11	Gap	760070
----->	DZ11		760100
		Gap	760110
	KMC11	Gap	760120
	LPP11	Gap	760130
	VMV21	Gap	760140
	VMV31	Gap	760150
	DWR70	Gap	760170
	RL11	Gap	760200
	LPA11-K	Gap	760220
	KW11-C	Gap	760230
	Reserved	Gap	760240
	RX11	Gap	760250
	DR11-W	Gap	760260
	DR11-B	Gap	760270
	DMP11	Gap	760300
	DPV11	Gap	760310
	ISB11	Gap	760320
	DMV11	Gap	760330
	DEUNA	Gap	760340
----->	UDA50 (WUSMD)		772150[1]
----->	UDA50 (WUSMD)		760354
		Gap	760360
	DMF32	Gap	760400
		Gap	760440
	KMS11	Gap	760420
	VS100	Gap	761440
	TU81	Gap	761450
	KMV11	Gap	761460
----->	DHV11		761500
----->	DHV11		761520
		Gap	761530
	DMZ32	Gap	761540
	CP132	Gap	761600

[1] indicates a fixed address device

Chapter 6
WUSMD Cache Operation

6.1 WUSMD Disc Cache

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

6.2 Read Look-ahead

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

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

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

6.3 Cache Allocation

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

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

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

6.4 Cache Usage

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

6.5 Cache Assignment Algorithm

The WUSMD cache implements a contention based hashing algorithm to determine block replacement. A given disc block has a fixed cache address calculated as follows :-

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

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

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

6.6 Cache Operation

The following is a desciption of the cache operation algorithm :-

```
Read:
       Examine cache for data required.

           If all data in cache
           Transfer data from cache to Unibus
       else
           Assign cache (lock it and wait if locked already)
           Perform read
           Unlock cache and flag as valid
```

Write:
Assign cache (lock it and wait if locked already)
Transfer from Unibus to cache and flag as valid
Perform write
Unlock cache

6.7 Cache Disable

The WUSMD cache can be disabled for performance evaluation, engi neering and diagnostic application requirements by selecting that option in WOMBAT. The cache cannot be selectively disabled for a particular drive.

6.8 Early Write Notification

The WUSMD implements early write notification where data to be writ ten to disc is retained in the cache and the host is issued a write complete notification. The controller will then write the data to the disc at the most convenient time.

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

Chapter 7
WUSMD Unibus Interface

7.1 Unibus Interface

All data, address and control information transfers between the processor and disc controller are carried out over the Unibus. The WUSMD fully implements all current Unibus enhancements including 18-bit addressing and fully supports PDP-11 Unibus CPU designs, as well as VAX-11 and VAX 8600 32-bit CPUs.

The Unibus consists of 56 signal lines and a number of ground (logic reference) lines. The signals are partitioned into three groups :

* Three initialization/shutdown signals

Initialize	INIT
AC Low	AC LO
DC Low	DC LO

* Twelve arbitration signals

Bus Request	BR<07:04>
Bus Grant	BG<07:04>
Non-processor Request	NPR
Non-processor Grant	NPG
Selection Acknowlege	SACK
Bus Busy	BBSY

* Forty-one data transfer signals

Address	A<17:00>
Data	D<15:00>
Control	C0, C1
Master Sync	MSYN
Slave Sync	SSYN
Parity	PA, PB
Interrupt	INTR

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

7.2 Interrupts

Interrupt priority for the WUSMD is link selectable on the PCB. The recommended priority setting is five. Interrupts suspend program execution while the processor starts the device service routine at a vector address input from the requesting device.

Interrupts are serviced according to a two-dimensional priority arbitration system. There are five different priority levels of request and when more than one device generates a request at the same priority level, the processor gives preference to the device which is electrically closest. It is possible for the bus to carry out more than one transaction at a time viz. selecting a new master and transferring data.

The interrupt protocol has three phases :-

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

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

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

7.3 Direct Memory Access

The WUSMD supports Direct Memory Access (DMA). During a DMA transfer the processor passes mastership of the bus to the controller.

The controller transfers four 16-bit words per bus acquisition and inserts a four microsecond delay between each acquisition to allow service to other devices requiring access to the Unibus. The four microsecond delay and the number of words transferred per acquisition are default values and may be adjusted (using WOMBAT) to optimize system performance.

DMA protocol consists of three phases :-

1. Bus Mastership Acquisition Phase. The WUSMD requests control of the bus. The processor arbitrates the request then initiates the transfer of bus mastership.

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

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

For a detailed explanation of all Unibus functions the appropriate DEC manuals should be consulted.

Chapter 8
WUSMD SMD Interface

8.1 SMD Interface

The Storage Module Drive (SMD) interface is an industry standard for high capacity disc drives. The basic interface consists of a control cable, read/write cable, and a ground cable. The control and read/write cables are available in either flat-ribbon or twisted pair configurations. The WUSMD uses the flat-ribbon cable only.

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

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

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

The WUSMD simultaneously supports any combination of SMD interface drives and data rates on the one controller. Pin assignments for the various SMD interface types are set out in Tables 8-1, 8-2, 8-3, 8-4 and 8-5. The SMD documentation should be consulted for a detailed explanation of the specific function details of each SMD interface type.

8.2 Standard SMD Interface

TABLE 8-1 SMD STANDARD CONTROL CABLE SIGNALS

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

The standard SMD interface uses BUS0 through BUS9 with TAG1 (Set Cylinder & Seek Command) to address up to 1024 cylinders. Has only one disc status byte.

8.3 Extended SMD Interface

TABLE 8-2 SMD EXTENDED CONTROL CABLE SIGNALS

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

Uses BUS0 through BUS9 with TAG4 and TAG1 to provide a 20 bit cylinder address. Uses BUS0 through BUS9 to provide a 20 bit head address. Uses unit select lines with TAG4 to provide up to 16 Disc status bytes.

8.4 Modified SMD Interface

TABLE 8-3 SMD MODIFIED CONTROL CABLE SIGNALS

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

Uses BUS0 through BUS9, UNIT SELECT 0 & UNIT SELECT 1 with TAG1 to address up to 4096 cylinders. Has 4 disc status bytes selected by TAG4 and TAG5 (Modified Status Select).

8.5 Enhanced SMD Interface

TABLE 8-4 SMD ENHANCED CONTROL CABLE SIGNALS

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

Defines a spare SMD line as BUS10 to allow addressing of up to 2048 cylinders (Spare signal pin numbers are 59 and 60). Has only one disc status byte.

8.6 SMD-E Interface

TABLE 8-5 SMD-E CONTROL CABLE SIGNALS

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

Uses high order head select bits (BUS7 and BUS8) along with the standard cylinder address bits to address up to 4096 cylinders. Has 7 disc status bytes selected by TAG4, TAG5 and BUS0 and BUS1.

8.7 SMD Read/Write Cable Signals

The Read/Write cable is common to all of the SMD interface types. Pin assignments are shown below in Table 8-6.

TABLE 8-6 SMD READ/WRITE CABLE SIGNALS

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

Chapter 9
WUSMD Hardware Description

9.1 General Description

There are four major functional sections in the WUSMD :-

1. The microprogrammed sequencer combined with an 8-bit RALU, forms a high speed processor which performs the following functions :

   (a) Implement the MSCP register pair - the Initialization and Polling (IP) Register and the Status and Address (SA) Register. Respond to accesses of these registers by the host CPU.

   (b) When the Auto Boot option is enabled, respond to accesses of the selected bootstrap addresses and initiate the automatic bootstrap function.

   (c) Generate Unibus interrupt requests and control the vector transfer phase of the interrupt cycle.

   (d) Generate Unibus DMA requests and control the transfer of data between cache memory and Unibus main memory. During DMA transfers, detect memory parity and timeout errors.

   (e) During disc read/write operations, control the transfer of data between the disc FIFO buffer, (contained within the 8466 controller chip) and the cache memory.

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

2. The Unibus interface, which contains the necessary logic to support Data In (DIN), Data Out (DOUT), and Read Modify Write (DATIO) bus cycles from the host CPU. It also supports Interrupt Request cycles, DMA Request cycles and DMA transfer cycles with 18-bit addressing.

3. The SMD disc interface. Disc read, write and formatting functions are performed by the 8466 controller chip. Seek, head select, and drive select functions are performed by extra logic.

4. The cache memory. The WUSMD uses 36 256K rams to implement 1M byte of storage for use as a disc cache. This memory is parity protected at the byte level. Circuitry is provided to refresh the memory every 3.3 mS.

9.2 Detailed Description

Microprogrammed Sequencer

The sequencer consists mainly of a set of eight 1K registered PROMS which store a set of microcode routines which perform specific functions. Execution of a microcode routine is initiated by the P PROM at U12. The inputs of the P PROM are connected to various system states such that it can recognise 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, so that when the appropriate input conditions arise the P PROM will assert the starting address of the desired routine to be executed. 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 U17 has its inputs and outputs connected to the Next Address bus, allowing it to use the current address on this bus to determine 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 the address of each successive step of the routine being executed. At the end of the routine the TRAP signal is asserted, 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 22 system signals connected to the three 8-input multiplexers U21, U18, U23. The sequencer controls these multiplexers via the B PROM U22, 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. This mechanism allows the sequencer to make decisions based on various system states, and to take the appropriate action.

Seven other PROMS - the A,F,S,C,M,L, and E PROMS - are connected to the Next Address bus, each producing eight output signals which control the rest of the logic on the board. One of these signals, uA9, from the C PROM forms the most significant bit of the Next Address bus, and can be viewed as an extension of the N PROM. 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 2901 RALU, into the address latch made up of the five 74LS191 counters at U74, U79, U80, U84 and U85. The output of this latch addresses the U.V. erasable firmware PROM (EPROM) U69. The sequencer then asserts the uREAD

signal, followed by the FETCH signal, which places the output of the EPROM onto the Next Address bus via the octal latch U36. This starts up a microcode routine which performs the function specified by the value fetched from the EPROM; thus executing the instruction.

The 8-bit Register and Arithmetic Logic Unit (RALU) made up of the two 2901C bit slice chips at U38 and U46 performs all data manipulation functions and stores the "microprocessor" register set.

Unibus Interface

1. Address

Eighteen-bit address information is transferred between the WUSMD and the Unibus via the five 2908 bus transceivers : U14, U19, U24, U29 and U34. Incoming addresses appear on AL0 - AL17 and are decoded by the W PROM, U13. This PROM produces four outputs, one of which indicates that the address is within the I/O page. The other three outputs provide the three bootstrap address options,(773000, 774000, 771000). Address lines AL2 - AL12 are compared with the setting of the 10-way dip switch, U3, by the two 8131 comparators, U8 and U9. The output of this comparator (U9, pin 9) combined with the IOPAGE/ signal from the W PROM is used to indicate that the host CPU is addressing one of the two registers on the WUSMD. The IOPAGE/ signal, the address comparator output and the Unibus address strobe signal, RMSYN, are connected via the two 74S273 synchronising registers, U1, U7, into the P PROM. If these 3 signals are asserted, the P PROM starts a microcode routine to respond to the register access. The P PROM uses the state of the AL1 signal, also connected to its input, to determine whether the access is to the IP register or the SA register.

The 4-way dip switch, U2, selects one (or none) of the 3 bootstrap options which is then connected to the P PROM via the synchronising registers. The P PROM uses this input combined with RMSYN and AL1 to start up the appropriate bootstrap service routine.

When the WUSMD is performing a DMA transfer it must specify the Unibus address to which the transfer is directed. It does this by first loading the five 2908 bus transceivers with the address information, one byte at a time, using the LDADL/, LDADM/, and LDADH/ signals. The LDADH/ signal also loads the Unibus control signal BC1L which specifies a read or write operation. This signal and all the Unibus address signals are then enabled onto the Unibus by the ABUSEN/ signal. The TMSYN signal is then asserted to strobe the Unibus address into the slave.

2. Data

Sixteen-bit data is transferred between the Unibus and WUSMD via the four 2908 bus transceivers, U66, U71, U76 and U81. Incoming data is enabled onto the internal 16-bit data bus, DB0 - DB15, by the DATEN/ signal. From here the data can be written directly into the cache or transferred to the 8-bit bus, D0 - D7, via the 74LS245 tranceivers, U70 and U75.

Outgoing data is loaded into the 2908 tranceivers by the LDBRL/ and LDBRH/ signals, either byte at time from the 8-bit data bus, or word at a time from the cache, and then enabled onto the Unibus by the DBUSEN/ signal.

3. Interrupts

To generate an interrupt request to the host CPU, the microcode asserts TBR5 which asserts BBR5L on the Unibus. The host grants the request by asserting BBG5IH which sets the INTG signal on U51 pin 9. This signal is connected to the P PROM via synchronising register U7, and causes a microcode routine to be started up which transfers the interrupt vector to the Unibus. The vector is loaded into the data tranceivers, U66, U71, U76, U81 and asserted onto the Unibus along with the BINTRL signal from U34 pin 6.

4. DMA

The microcode initiates DMA transfers by asserting the TNPR signal which asserts BNPRL on the Unibus. The host responds with BNPGIH which asserts the DMAG signal on pin 5 of U51. This signal is connected to the P PROM via U7 and initiates the DMA transfer routine in the microcode. This routine asserts TSACK to become Unibus master, then proceeds to transfer data over the Unibus.

SMD Interface

The 8466 disc controller chip performs the basic read, write and formatting functions required of the WUSMD. During a read, it accepts serial NRZ data from the disc, converts it to 16-bit words and stores them in an on-chip 16-word FIFO for retrieval by the external logic. Conversely, during a write, it removes 16-bit words from the FIFO and converts them to serial NRZ data to be written to the disc. The chip also performs sector header recognition, CRC generation and checking (for headers) and ECC generation and checking (for data). The chip contains 64 internal registers which are made accessable to the WUSMD firmware and allow programming of disc format parameters, ECC polynomials and disc commands.

The SMD interface consists of a 26-way data cable, over which serial disc data and clock information is transmitted, and a 60-way control cable which carries disc command and status information. Command

information to be transmitted to the drive is loaded into the two octal latches U26 and U31. Three TAG bits are then used to strobe this data into the drive. TAG1 strobes cylinder addresses, TAG2 strobes head addresses and TAG3 enables control information such as READ GATE and WRITE GATE. UNIT SELECT TAG selects the drive indicated by the UNIT SELECT bits.

Status information from the drive is received by U30 and U35 and is gated directly onto the 8-bit bus by the STAT2 signal. Some drives provide three more status bytes on these lines as selected by combinations of the TAG4 and TAG5 signals. This applies to MODIFIED SMD interfaces only. Drives with the EXTENDED SMD interface provide extra status bytes which are selected by UNIT SELECT bits and TAG4 together.

All signals on the SMD interface (except PICK and HOLD) are differential negative polarity signals which must be terminated by the 56 ohm and 82 ohm S.I.P. resistor packs. Incoming status signals are connected to the 3486 differential receiver chips which have tri-state TTL outputs. Incoming clock and NRZ data signals are received by the higher speed 26LS32 differential receivers.

Outgoing signals are driven by the 3453 differential drivers. These require a -5 volt supply to produce the negative polarity signals of the SMD interface. This is provided by a voltage inverter circuit which converts -15 volts from the Unibus backplane to a regulated -5 volt supply.

Cache Memory

An array of thirty six 256K dynamic RAM chips forms a one megabyte parity protected memory which is used as a disc cache. The array is arranged as two banks of 18 chips with 128K 16-bit words (and two parity bits) in each bank.

The array is addressed by the same address latch that addresses the fimware PROM mentioned above. 18 of these address bits are multiplexed by U89, U92 and U95 to form XA0 - XA8 which become the row and column address inputs for the array. Row addresses are strobed into the array by DRAS0/ or DRAS1/ depending on the state of the high order address bit A18. This provides a bank select function. Column addresses are strobed by DCASHI/ and DCASLO/. These two signals are asserted together to access a word, or individually to access the hi or lo byte only.

There is a parity bit assigned for each byte of data in the array. During a write cycle, parity information is generated by the two 74S280s U90, U93, and written into the parity chips U96, U97, (low bank) or U98, U99, (hi bank). Odd parity is used. During a read cycle, the same chips, U90, U93, check the parity bits and assert CPERR if an error is detected. If the WWP/ signal is asserted, wrong parity will be written into the array on all subsequent write cycles. This provides a diagnostic aid for testing the parity logic.

Dynamic RAMs require that 256 row addresses be refreshed every 4 mS. The 14-bit divider chip at U54 generates a refresh request every 12.8 uS by causing the REFR signal to be asserted at pin 6 of U63. This causes the sequencer to refresh one row address which is derived from the low order 8 bits of U54. By refreshing a row address every 12.8 uS, all 256 row addresses are refreshed in 3.3 mS.

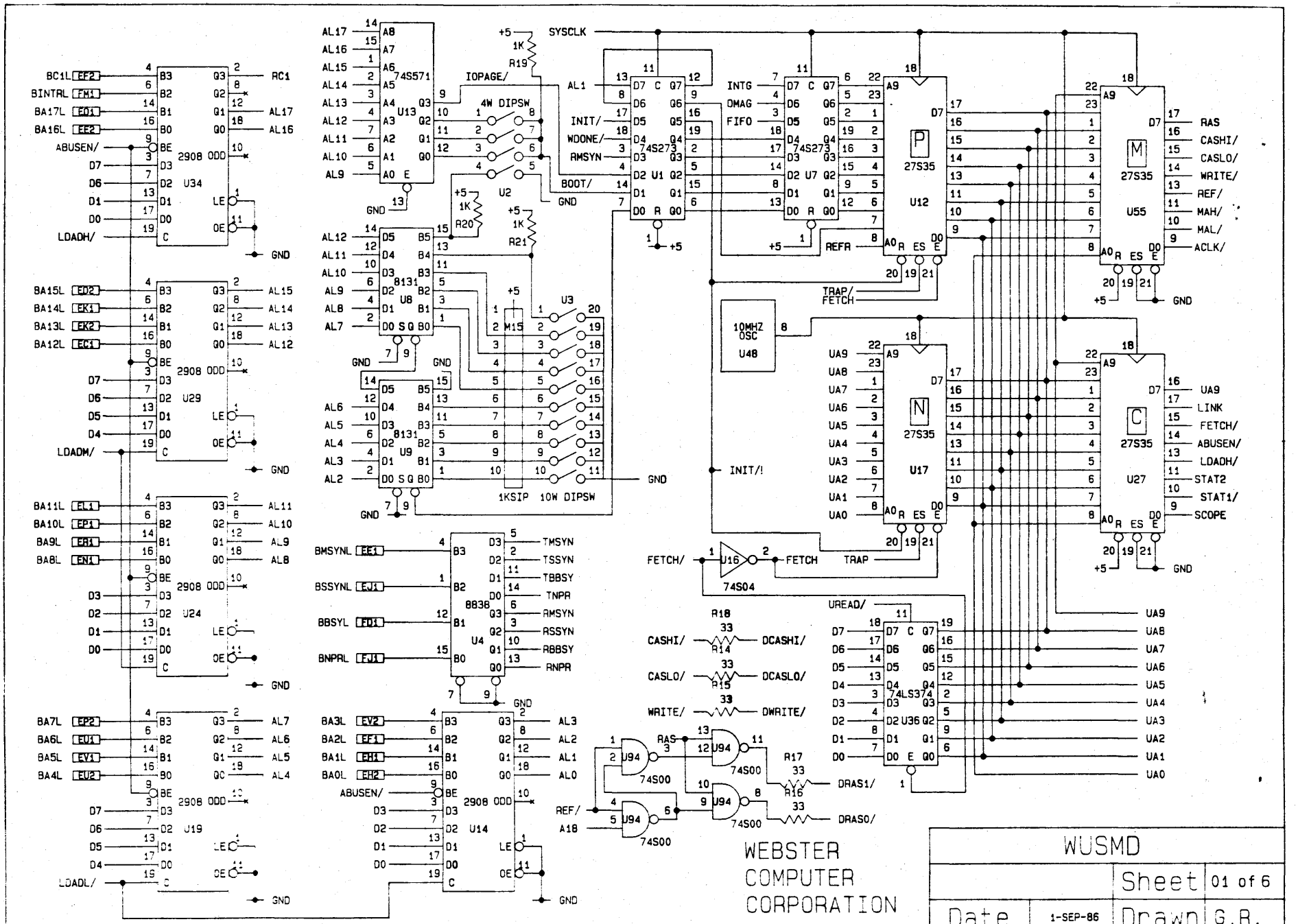
I/O Port

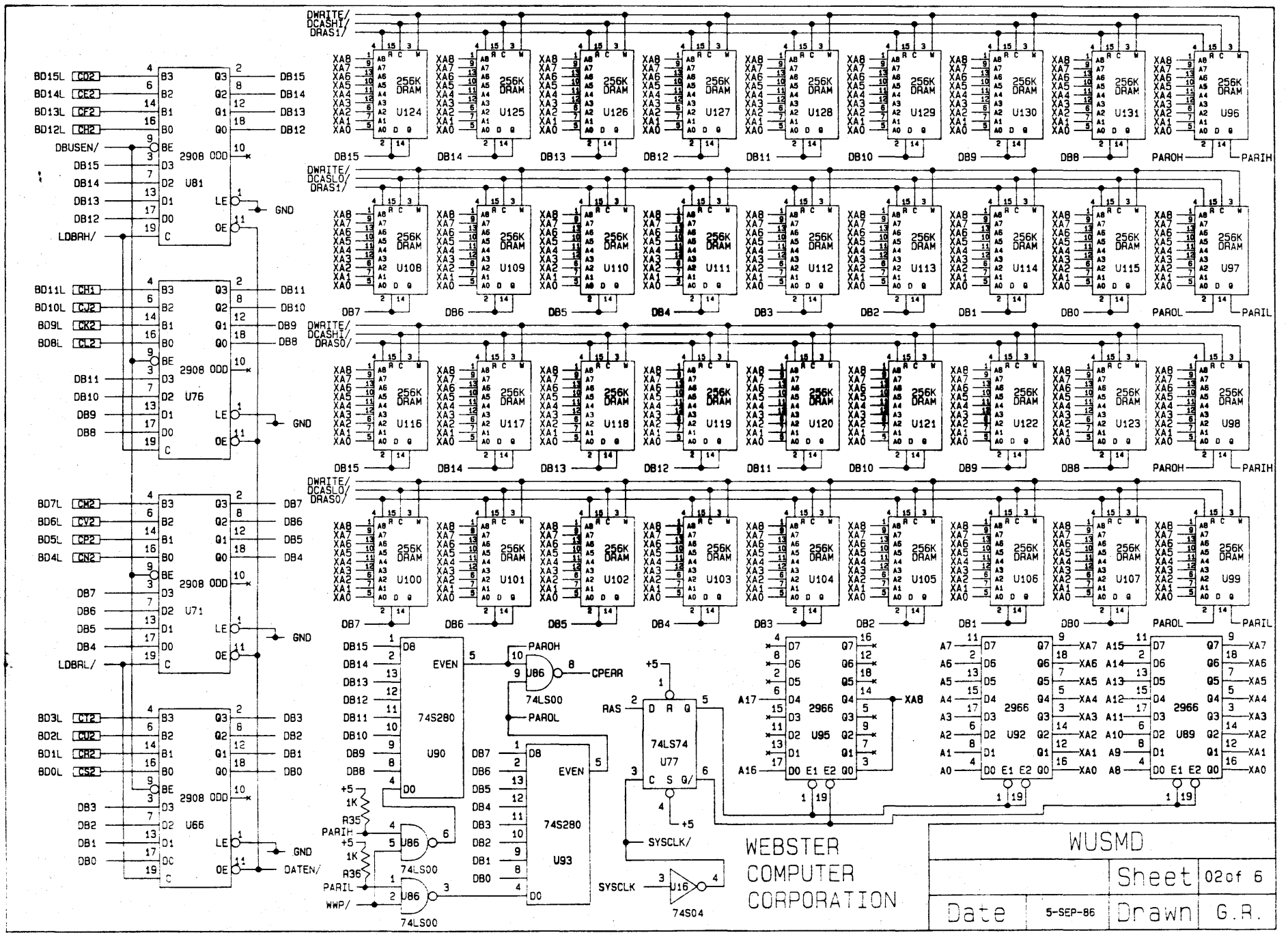
The I/O port, a 10-way connector at J1, serves two main functions. When running WOMBAT it can serve as an RS232 port to an ASCII terminal. RS232 input data is connected to J1 pin 8 where it is clipped by a zener diode and then received by the octal buffer U5. The SERIAL signal from U41 pin 12 drives the base of the 2N4356 transistor, Q1, which produces the necessary +-5V swing at its collector to drive the ACCESS/ RS2320 output on pin 3 of J1. The second function of the I/O port is to connect to a front panel, which, when the controller is on line (not running WOMBAT), can show disc activity and provide write protect capability on a per drive basis.

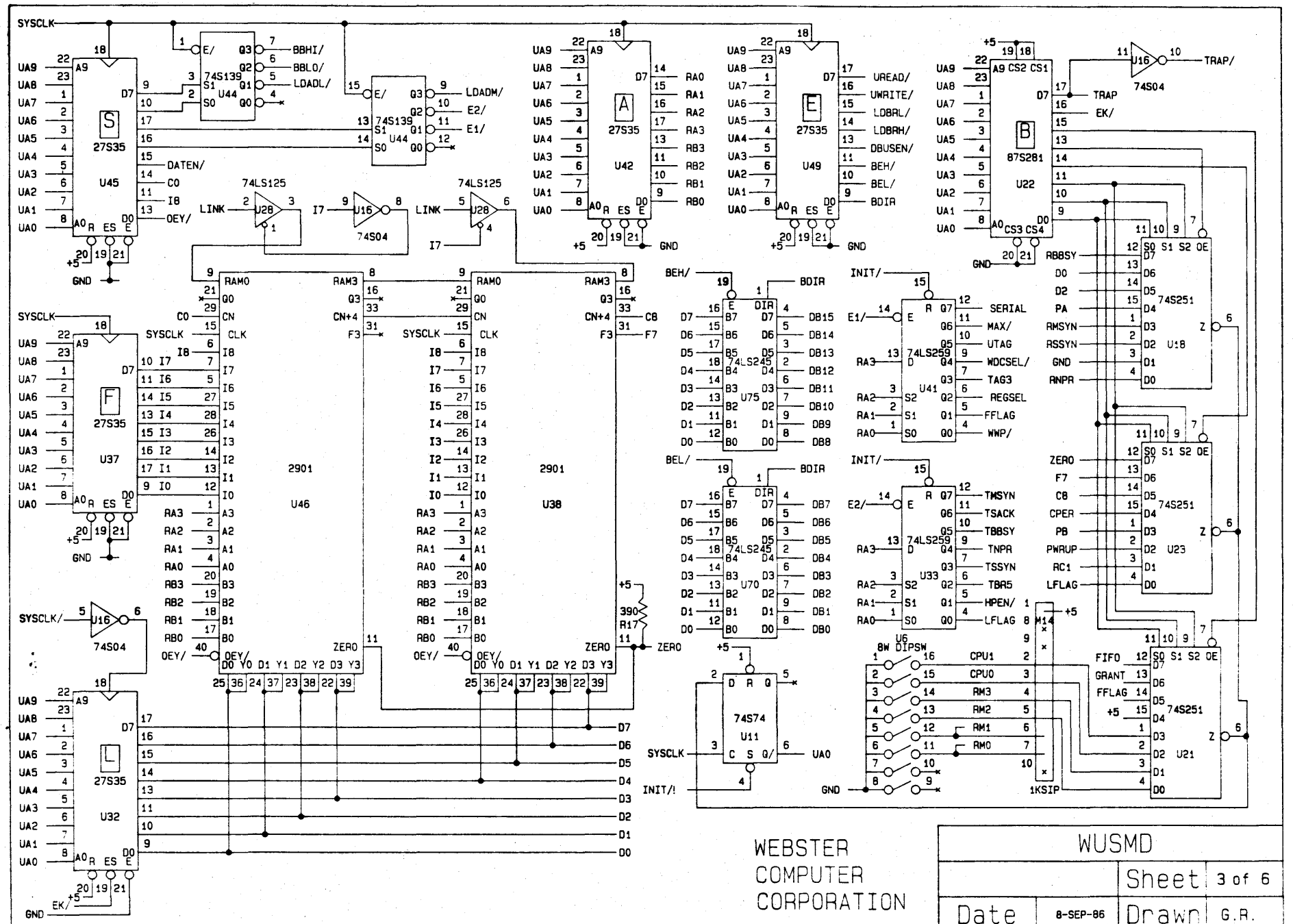
The two TTL signals ACCESS0/ and ACCESS1/, J1 pins 5 and 6, indicate disc activity on the drives connected to J3 and J4 respectively. These signals can be used to operate front panel indicators via appropriate driver circuits. The ACCESS/ RS232O signal is asserted when any of the four drives or the cache is accessed, and may be used to drive a front panel indicator. The four signals, WP0/ - WP3/, on J1 pins 8, 7, 10 and 1 can be externally grounded via a front panel push-button to write protect drives 0, 1, 2 and 3 respectively.

Chapter 10
Circuit Diagrams

The circuit diagrams for the WUSMD Disc Controller may be found in the following pages.

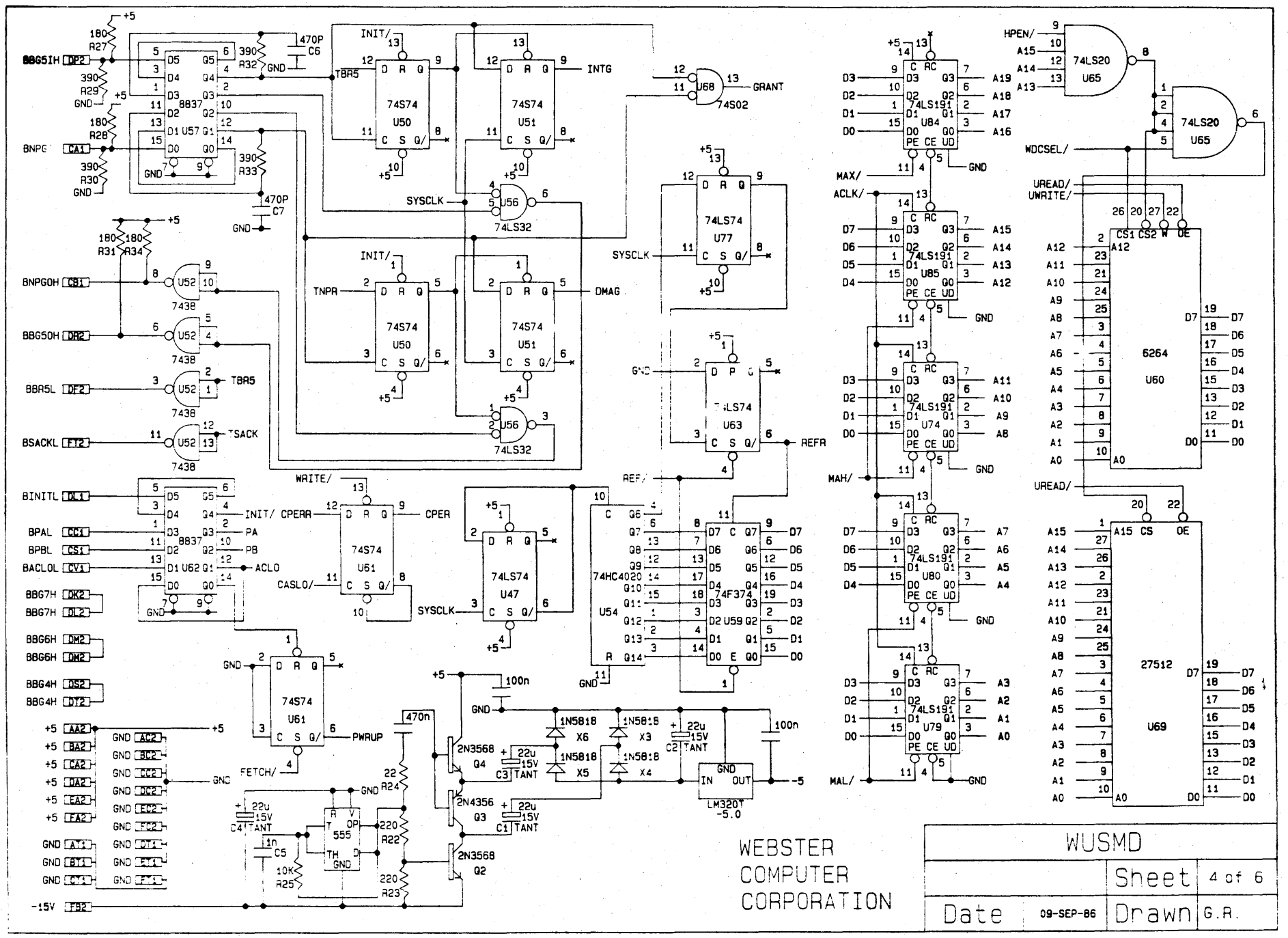

WEBSTER
COMPUTER
CORPORATION
WUSMD
Sheet 01 of 6
Date 1-SEP-86
Drawn G.R.
2908 ODD U34
2908 ODD U29
2908 ODD U24
2908 ODD U19
2908 ODD U14
74S571 U13
8131 U8
8131 U9
8838 U4
74S273 U2
74S273 U7
74LS374 U36
27S35 P U12
27S35 M U55
27S35 N U17
27S35 C U27
10MHZ OSC U48
74S04 U16
74S00
4W DIPSW
1KSIP 10W DIPSW
SYSCLK
IOPAGE/
INIT/
WDONE/
RMSYN
BOOT/
REFR
TRAP/
FETCH
INIT/!
UREAD/
CASHI/
CASLO/
WRITE/
REF/
RAS
DRAS1/
DRAS0/
DCASHI/
DCASLO/
DWRITE/
ABUSEN/
LDADH/
LDADM/
LDADL/
BMSYNL
BSSYNL
BBSYL
BNPRL
TMSYN
TSSYN
TBBSY
TNPR
RMSYN
RSSYN
RBBSY
RNPR
RAS
CASHI/
CASLO/
WRITE/
LINK
FETCH/
ABUSEN/
LDADH/
STAT2
STAT1/
SCOPE
ACLK/
MAH/
MAL/
REF/

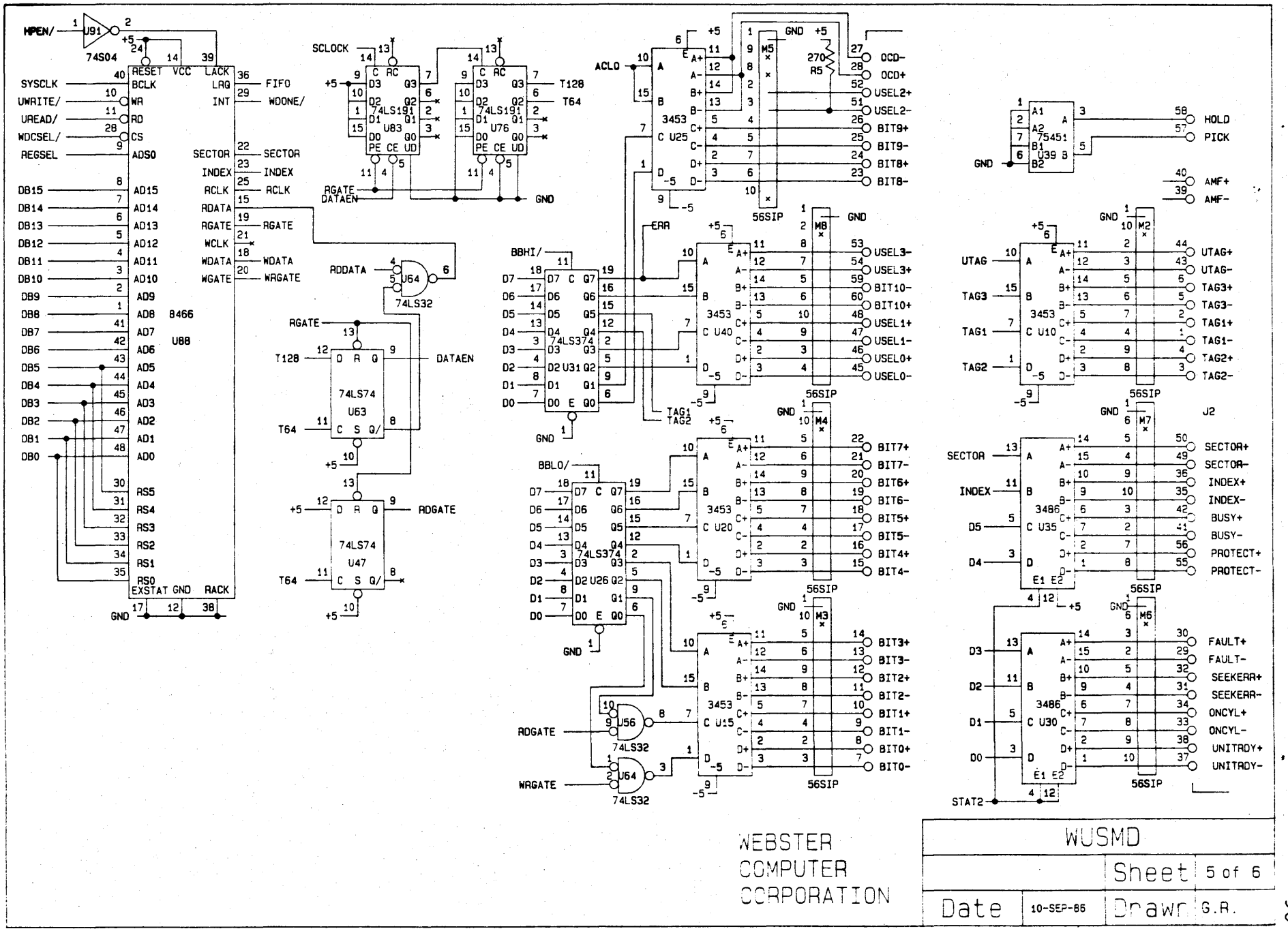

WEBSTER COMPUTER CORPORATION
WUSMD
Sheet 02of 6
Date 5-SEP-86
Drawn G.R.
2908 ODD U81
2908 ODD U76
2908 ODD U71
2908 ODD U66
74S280 U90
74S280 U93
74LS74 U77
74S04 U16
74LS00
2966 U95
2966 U92
2966 U89
256K DRAM
DWRITE/
DCASHI/
DCASLO/
DRAS1/
DRAS0/
DBUSEN/
LDBRH/
LDBRL/
GND
DATEN/
PAROH
PAROL
PARIH
PARIL
CPERR
RAS
SYSCLK/
SYSCLK
WWP/
PARIH
PARIL
XA8

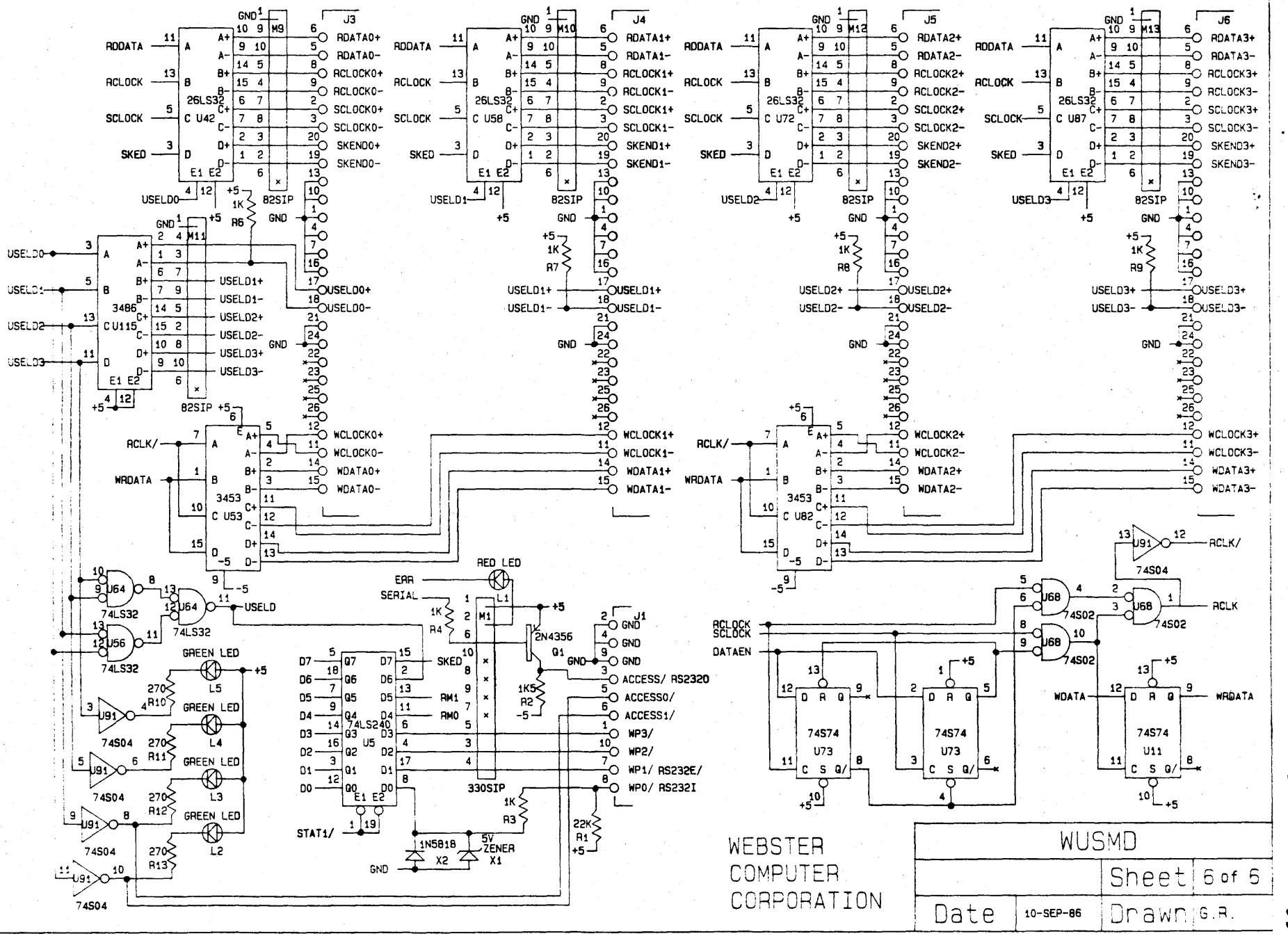


WEBSTER COMPUTER CORPORATION

WUSMD			
		Sheet	3 of 6
Date	8-SEP-86	Drawn	G.R.


WEBSTER
COMPUTER
CORPORATION
WUSMD
Sheet 4 of 6
Date 09-SEP-86
Drawn G.R.


HPEN/
U91
74S04
SYSCLK
UWRITE/
UREAD/
WDCSEL/
REGSEL
DB15
DB14
DB13
DB12
DB11
DB10
DB9
DB8
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
RESET
VCC
LACK
BCLK
LRQ
WR
INT
RD
CS
ADS0
AD15
AD14
AD13
AD12
AD11
AD10
AD9
AD8
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
8466
U88
RS5
RS4
RS3
RS2
RS1
RS0
EXSTAT
GND
RACK
FIFO
WDONE/
SECTOR
INDEX
RCLK
RDATA
RGATE
WCLK
WDATA
WRGATE
SCLOCK
74LS191
U83
U76
T128
T64
RGATE
DATAEN
GND
RDDATA
U64
74LS32
74LS74
U63
U47
DATAEN
RDGATE
ACLO
3453
C U25
OCD-
OCD+
USEL2+
USEL2-
BIT9+
BIT9-
BIT8+
BIT8-
M5
270
R5
56SIP
ERR
BBHI/
74LS374
U31
C U40
USEL3-
USEL3+
BIT10-
BIT10+
USEL1+
USEL1-
USEL0+
USEL0-
M8
TAG1
TAG2
BBLO/
U26
C U20
BIT7+
BIT7-
BIT6+
BIT6-
BIT5+
BIT5-
BIT4+
BIT4-
M4
C U15
BIT3+
BIT3-
BIT2+
BIT2-
BIT1+
BIT1-
BIT0+
BIT0-
M3
U56
WRGATE
75451
U39 B
HOLD
PICK
AMF+
AMF-
M2
UTAG
TAG3
TAG1
TAG2
C U10
UTAG+
UTAG-
TAG3+
TAG3-
TAG1+
TAG1-
TAG2+
TAG2-
J2
M7
SECTOR
INDEX
D5
D4
3486
C U35
SECTOR+
SECTOR-
INDEX+
INDEX-
BUSY+
BUSY-
PROTECT+
PROTECT-
E1 E2
M6
D3
D2
D1
D0
C U30
FAULT+
FAULT-
SEEKERR+
SEEKERR-
ONCYL+
ONCYL-
UNITRDY+
UNITRDY-
STAT2
WEBSTER
COMPUTER
CORPORATION
WUSMD
Sheet 5 of 6
Date 10-SEP-86
Drawn G.R.


WEBSTER
COMPUTER
CORPORATION
WUSMD
Sheet 6 of 6
Date 10-SEP-86
Drawn G.R.

Appendix A
Sample Switch Settings

Fujitsu Drive Switch Settings for the Webster WQSMD Controller.

1. M2322 8 inch 160 Megabyte drive.

Switch settings on logic PCB:

SW1	1	Off	)
	2	Off	)Drive 0
	3	Off	)
	4	On	Drive type M2322
	5	Off	Disable Tag 4/5
	6	Off	Hard sector mode
	7	Off	Write enabled
	8	Off	Horizontal or vertical mount

Set up bytes per sector as 596+1 = 597, plus one short sector of 182 bytes.

			Number of Bytes
SW2	1	Off	-
	2	Off	-
	3	On	4
	4	Off	-
	5	On	16
	6	Off	-
	7	On	64
SW3	1	Off	-
	2	Off	-
	3	On	512
	4	Off	-
	5	Off	-
	6	Off	-
	7	Off	-

M2322 Wombat Parameters

These are used in the 'Create Disc Structure' option.

823 cylinders
10 heads
34 sectors per track
1 short sector
Standard SMD interface
Spiralling factor = 4

2. M2333 8 inch 337 Megabyte drive.

Switch settings on logic PCB:

SW1	1	Off	)
	2	Off	)Drive 0
	3	Off	)
	4	Off	
	5	On	Drive type M2333
	6	On	
	7	On	
	8	Off	Tag4/5 disabled
	9	Off	Write enabled
	10	Off	Horizontal mount

Set up bytes per sector as 594 + 2 = 596, plus one short sector of 423 bytes.

			Number of Bytes
SW2	1	Off	-
	2	On	4
	3	Off	-
	4	On	16
	5	Off	-
	6	On	64
	7	Off	-
SW3	1	Off	-
	2	On	512
	3	Off	-
	4	Off	-
	5	Off	-
	6	Off	-
	7	Off	-

M2333 Wombat Parameters

These are used in the 'Create Disc Structure' option.

823 cylinders
10 heads
68 sectors per track
1 short sector
Standard SMD interface
Spiralling factor = 10

3. M2351A Eagle 490 Megabyte drive.

Jumper plug settings on logic PCB:

LOCATION AE7	3 - 4	Disable Tag4/5
	6 - 7	Enable SEEKEND after offset command reset
	10 - 11	Disable ready on drive fault

Set up bytes per sector as 595 + 1 = 596, plus one short sector of 148 bytes.

Jumper the following:		Number of Bytes
LOCATION BC7	3 - 4	-
	6 - 7	-
	9 - 10	4
	13 - 14	-
LOCATION BD7	2 - 3	16
	6 - 7	-
	9 - 10	64
	13 - 14	-
LOCATION BE7	3 - 4	-
	5 - 6	512
	10 - 11	-
	13 - 14	-
LOCATION BF7	3 - 4	-
	6 - 7	-
	10 - 11	-
	13 - 14	-

Switch settings on interface PCB:

SW1	1	Off	)
	2	Off	)Drive 0
	3	Off	)
	4	Off	Not used

M2351 Wombat Parameters

These are used in the 'Create Disc Structure' option.

842 cylinders
20 heads
47 sectors per track
1 short sector
Standard SMD interface
Spiralling factor = 6

4. M2361A Super Eagle 689 Megabyte drive.

BE SURE TO SET INPUT VOLTAGE CONNECTOR TO THE CORRECT VOLTAGE.

Display Panel Settings:

REMOTE/LOCAL	LOCAL (Spin up on power on)
ENABLE/DISABLE A	ENABLE (Enable Channel A)
ENABLE/DISABLE B	DISABLE (single port) ENABLE (dual port)
RLTM/ABSL	ONLY RELEVANT IF DUAL PORT OPTION FITTED
RTZ	MAINTENANCE AID (Manual Return To Zero)
STATE INDICATOR	ST2 (Not critical...Allows user to select state output on display)

Set up bytes per sector as 595 + 1 = 596, plus one short sector of 500 bytes.

			Number Of Bytes
Switch B:	1	Off	-
	2	Off	-
	3	Off	-
	4	Off	-
	5	Off	-
	6	Off	-
	7	On	512
	8	Off	-
Switch A:	1	Off	-
	2	On	64
	3	Off	-
	4	On	16
	5	Off	-
	6	On	4
	7	Off	-
	8	Off	-

Drive Address:

1	Off	Not used
2	Off	)
3	Off	) Drive 0
4	Off	)

M2361 Wombat Parameters

These are used in the 'Create Disc Structure' option.

842 cylinders
20 heads
68 sectors per track
1 short sector
Standard SMD interface
Spiralling factor = 10