MXV50
DISK CONTROLLER
MANUAL

840003-102 REV B

JANUARY 18, 1986

Micro Technology, Inc.

The information in this manual is for information purposes and is subject to change without notice.

Micro Technology, Inc. assumes no responsibility for any errors that may appear in this manual.

Printed in the U.S.A.

REVISION HISTORY

Revision A	Production Release
Revision B	Incorporate RX50 drive option. Added Scatter Gather for MicroVax I. Additional information for logical unit numbers provided. Cleared up format description for double sided operation. Corrected miscellaneous typographical errors.

PRODUCT STATUS

All known problems resolved.

Contents

Figures

Tables

Section 1

General Information

1. INTRODUCTION

The MXV50 is a dual density controller compatible with the DEC* RX50 diskette subsystem. Configured with any Shugart compatible drive, it is a direct replacement for the RX50 subsystem. The controller provides 400 Kbytes of storage on a single sided diskette. Using double sided diskettes, the MXV50 provides 800 Kbytes per drive or a total of 3.2 Mbytes

All electronics are contained on one dual-vide board which plugs directly into any standard LSI-11 backplane and interfaces through a 34 conductor ribbon cable to any Shugart compatible drive. All controllers are 100% tested and ready for plug in and operation. The controller is configured for the standard device address and the interrupt level is factory set to level four. Features include:

- o 22 bit addressing
- o Double sided drive support.
- o Transparent firmware bootstrap.
- o Blank diskette formatting.
- o Alternate address selection.
- o Power fail protection for data integrity.
- o Internal diagnostics

*DEC is a registered trademark of Digital Equipment Corporation.

1.1. COMPONENTS

The MXV50 controller is provided with the following components:

MXV50 5 1/4" floppy disk controller
MXV50 disk controller manual

1.2. COMPATIBILITY

This section discusses the aspect of hardware, software and media compatibility with Digital Equipment's RX50 subsystem. The information will aid the user in data interchanging with foreign systems.

1.2.1. Hardware

The controller is compatible with the LSI-11, LSI-11/2, LSI-11/23, LSI-11/73, MicroVax I and MicroVax II processors. All circuitry is contained on one dual-wide board that plugs directly into any standard LSI-11 backplane. Alternate address selection and four-level device interrupt priority scheme provide the user added flexibility for expanded system configurations. Shugart compatible drive logic is interfaced through a 34-pin ribbon connector.

1.2.2. Software

The MXV50 is compatible with RQDX register definition and command protocol. All DEC-supplied software designed to operate with the RQDX series of controllers will operate with the controller without modification.

1.2.3. Drives

When coupled with a single sided 96 TPI 5 1/4" floppy drive the MXV50 provides 409 Kbytes of storage capacity. Using double sided drives the controller provides in excess of 819 Kbytes of storage per drive. The following is a partial list of recommended drives:

Shugart	SA 410/460	Single/double sided, full height drive
Tandon	TM65-4	Double sided, half height drive
Tandon	TM100-4	Double sided, full height drive
Mitsubishi	M4853	Double sided, half height drive

1.2.4. Media

Either RX50 preformatted or blank soft sectored diskettes may be used with the controller. The following list summarizes the suggested media:

VERBATIM	MD550-01-18188
3M	DC 051111-00234
3M	DC 051111-00300 (RX50 compatible)

1.3. MEDIA FORMAT

The diskette is divided into 160 concentric tracks. Eighty (80) of these tracks are on each surface and are numbered zero (0) through seventy nine (79). A track is formatted starting at the physical index. The first field is termed Gap 1 and consists of 48 bytes of "4E" data followed by 8 bytes of "00" data. The next field is the first ID record consisting of three address marks of "A1" data. These address marks are unique containing a missing clock pattern. Following these address marks are the Index Header byte "FE", one byte of track address, one byte representing the side, one byte representing the sector address, one byte indicating sector length, and finally two CRC bytes. The following field is Gap 2 or the ID gap. This field consists of 22 bytes of "4E" data followed by 12 bytes of "00" data. The next field is the first data record consisting three address marks of "A1" data with missing clock patterns, the data header byte "FB", 512 bytes of user data and finally two bytes of CRC data. Following the data field is Gap 1 leading to the next ID field. Following the tenth data field is Gap 3 or the pre-index gap consisting of approximately 70 bytes of "4E" data.

Each track is formatted in the above manner (refer to Figure 1-1). The sector header field of each sector contains information describing both the sector and track number. All fields are encoded in MFM.

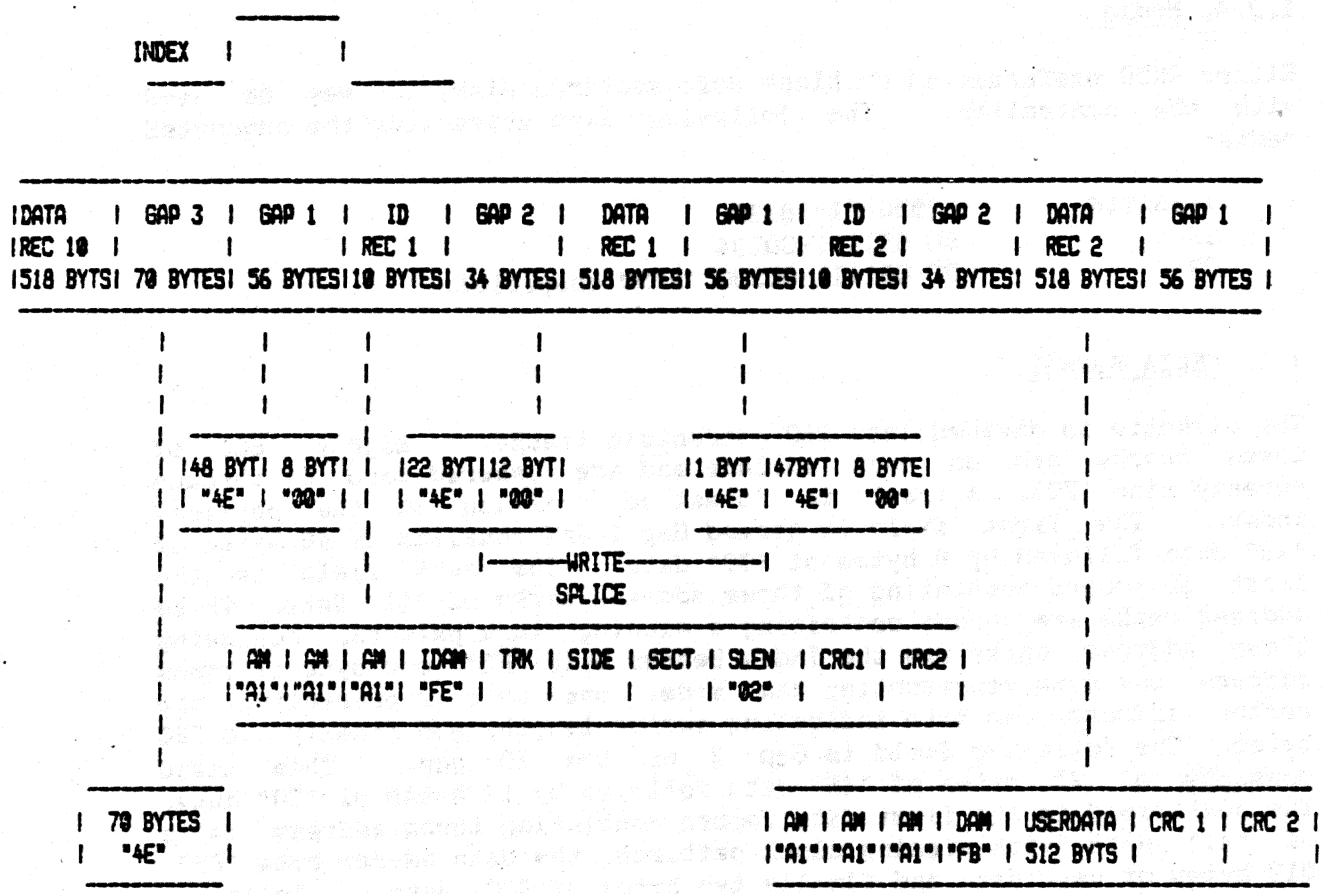


Figure 1-1: MXV50 5 1/4-Inch Logical Track Format

1.3.1. Sector Header Field

The header field consists of 10 bytes of information. Preceding the header is a field of 8 bytes of "zero" data for synchronization.

1. Byte 1,2,3 Address mark--a unique mark consisting of an "A1" data pattern with a missing clock transition using a "0A" clock pattern. This mark is decoded by the controller to indicate the start of either a sector or a data field.

2. Byte 4 ID Header--a single byte of "FE" data identifying the field as sector header.

3. Byte 5 Track Address--this byte indicates the absolute (0,117(8)) track address. Each sector contains this track information to locate its position on one of the eighty tracks.

4 Byte 6 Side--this byte indicates on which surface the track is located. Side zero is indicated by "00", side one by "01".

5 Byte 7 Sector address--This byte indicates the absolute (1-12(8)) sector number. Each sector contains this information to identify its position as one of the ten sectors on the track.

6 Byte 8 Sector Length--this byte indicates the length of the user data in the data field. 01 = 256 bytes 02 = 512 bytes, etc.

7. Byte 9, 10 CRC -- these two bytes represent the 16 bit cyclic redundancy check and are calculated for each sector header from the first 8 bytes of information using the standard CRC-CCITT polynomial.

1.3.2. Data Field

The data field consists of 518 bytes of information. Preceding the data field is a field of 12 bytes of "zero" data for synchronization.

1. Byte 1,2,3 Address Mark-- a unique mark consisting of an "A1" data pattern with a missing clock transition using a "0A" clock pattern. This mark is decoded by the controller to indicate the start of either a sector or a data field.

2. Byte 4 Data Field Header -- a single byte of "FB" data identifying the field as a data field.

3. Bytes 5-516 User Data-- this field contains 512 bytes of user data encoded in MFM.

4 Bytes 517,518 CRC -- these two bytes represent the 16 bit cyclic redundancy check and are calculated for each data field from the first 516 bytes of information using the standard CRC-CCITT polynomial.

1.3.3. Recording Scheme

The MXV50 MFM encoding is compatible with the DEC RX50 subsystem. MFM encoding consists of flux transitions for a logic "one" and no flux transitions for a logic "zero". A clock transition only occurs between two consecutive logic "zeros" as shown in Figure 1-2 below. The MFM bit cell time is 4us.

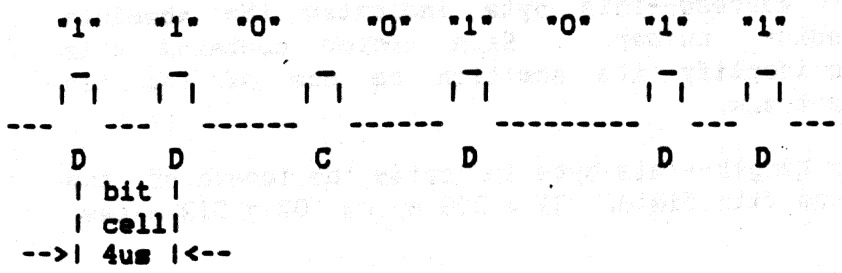


Figure 1-2: MFM Recording Characteristics

Table 1-1 summarizes the standard MFM encoding algorithm.

DATA		ENCODED DATA		
DN-1	DN	DN-1	CN	DN
0	0	0	1	0
1	0	1	0	0
0	1	0	0	1
1	1	1	0	1

Table 1-1 Standard MFM Encoding

1.3.4. Cyclic Redundancy Check

Each sector header field and data field has two byte CRC character appended. This 16 bit character is the remainder that results when dividing the data bits [represented as a polynomial M(x)] by a generator polynomial G(x). The polynomial used for RX50 is the standard CRC-CCITT G(x) = X^16 + X^12 + X^5 + 1. For the sector header the data bits include bytes 1 through 8. For the data field the data bits include bytes 1 through 516.

1.4. SPECIFICATIONS

1.4.1. General

Device Address

Standard	17772150	4th Alternate	17760414
1st Alternate	17760334	5th Alternate	17772144
2nd Alternate	17760354	6th Alternate	17772154
3rd Alternate	17760374	7th Alternate	Reserved

Vector

Standard (software configured)

Media Format	RX50 Compatible
Encoding Methods	MFM
Sectors/track	10
Tracks/surface	80
Surface/diskette	2
CRC	Standard CRC-CCITT

1.4.2. Mechanical

Size: Dual-height, standard length module.

Connectors: Two connectors used:

1. Standard Q-22 LSI-11 bus edge connector using the A and B rows
2. 34 pin right angle flat cable connector located at the handle end of the module for interface to industry standard SA 410/460 drives.

Jumpers: Jumpers are available to select address, logical unit numbers, interrupt priority, and various drive characteristics.

1.4.3. Electrical

Power requirements: +5VDC ± 5% at 2.5 Amps (max.)

LSI-11 bus loading:
1 DC bus load
2 AC bus loads

LSI-11 bus: Adheres to the Q-22 specification but does not generate or check parity. Only uses the A and B rows.

Receives the following LSI-11 bus signals:

BDAL00-BDAL17, BDIN, BDOUT, BSYNC, BRPLY, BWTBT, BBS7, BIRQ4, BIRQ5, BIRQ6, BIAKI BDMGI, BINIT, BDCOK, BPOK

Drives the following LSI-11 bus signals:

BDAL00 - BDAL21, BDIN, BDOUT, BSYNC, BRPLY, BWTBT, BBS7, BIRQ4, BIRQ5, BIRQ6, BIRQ7, BIAKO, BDMR, BDMGO, BSACK

Does not interface the following LSI-11 bus signals:

BHALT, BEVNT

Drive Interface:

Adheres to the Shugart SA410/460 floppy disk interface for minidiskette drives.

Signal assertion 0.0vdc to 0.4vdc @ 40 ma signal non assertion 2.5vdc to 5.25vdc @ 250 ma

Impedance 220/330 ohms

Receives the following signals:

INDEX, TRACK 0, WRITE PROTECT, READ DATA, DRIVE STATUS

Drives the following signals

DRVSEL1 - DRVSEL4, MOTOR ON, DIRECTION, STEP, WRITE DATA, WRITE GATE, HEAD LOAD, SIDE

Does not interface to the following signals:

DOOR LOCK, IN USE

1.4.4. Environmental

Temperature:

5 to 50 degrees C (41 to 122 degrees F)

Humidity:

10% to 90% (non-condensing)

1.4.5. Performance

Transfer rate:

Burst 31.25 Kbytes/sec
Average 15.5 Kbytes/sec

Capacity:

Single sided media	404,480 bytes
Double sided media	808,960 bytes
Maximum (4 drives)	3,235,840 bytes

1.4.6. Reliability

MTBF

36000 hrs.

MTTR

.5 hr.

Section 2

Installation

2. GENERAL

The controller is shipped with the standard address 17772150(8) set. The device interrupt priority is set to level four. The firmware bootstrap is disabled.

Most options are factory foil-etched (on the solder side of the PCB) to the most often used configuration. The foil jumpers must first be cut before the alternate jumpers are inserted. Refer to Tables 2-1, 2-2, and 2-3 for alternate options and Figure 2-1 for jumper location. Several of the options are selectable by using AMP 530153-2 pin jumpers. If these pin jumpers are not available use #30 wire wrap.

2.1. CONFIGURATION

2.1.1. Address Vector Selection

The controller is shipped with the DEC standard device address assignment preset to 17772150(8). Any change in this assignment would necessitate a change in system software. However, six alternate address options are selectable and are defined as 17760334, 17760354, 17760374, 17760414, 17772144 and 17772154. To select the desired alternate address, jumper W14, 15, 16, 17, 61, 62, 63 and 64 as shown in Table 2-2.

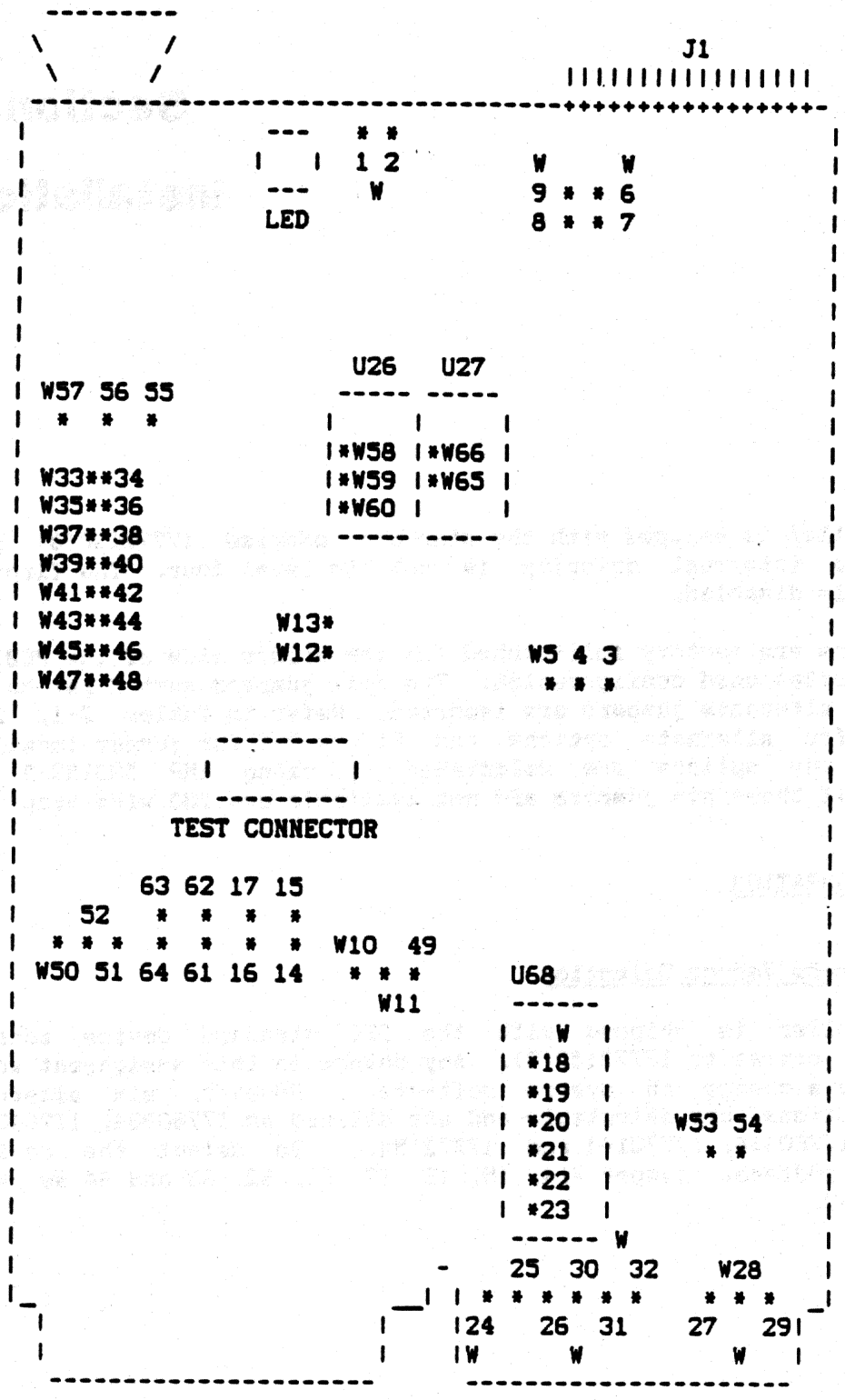


Figure 2-1: Configuration Jumper Locations

JUMPER	USAGE	STATUS
W1 - W2	Power Monitor Enable	IN
W3 - W4	Drive 1,2 Enable 2 sided	OUT
W4 - W5	Drive 3,4 Enable 2 sided	OUT
W6 - W7	5 1/4" Drive Status Connect	IN
W8 - W9	5 1/4" Head Load Connect	IN
W10-W11	BBS7 Enable	IN
W11-W49	BBS7 Disable	OUT
W12-W13	Multiple Drive	OUT
W14-W15	Pri/2nd/4th/6th Alt. Add.Select	IN
W16-W17	1st/3rd/5th/7th Alt. Add.Select	OUT
W18-W19	BIRQ 6 Monitor Disable	OUT
W19-W20	BIRQ 6 Monitor Enable	IN
W21-W22	BIRQ 5/7 Monitor Disable	OUT
W22-W23	BIRQ 5/7 Monitor Enable	IN
W24-W25	BIRQ 6 Assertion Disable	IN
W25-W26	BIRQ 6 Assertion Enable	OUT
W27-W28	BIRQ 7 Select	OUT
W28-W29	BIRQ 5 Select	IN
W30-W31	BIRQ 5/7 Assertion Disable	IN
W31-W32	BIRQ 5/7 Assertion Enable	OUT
W33-W34	LUN 1	OUT
W35-W36	LUN 2	OUT
W37-W38	LUN 4	OUT
W39-W40	LUN 8	OUT
W41-W42	Boot Enable	OUT
W43-W44	Step Rate 1,2 (Out = 3ms)	OUT
W45-W46	Step Rate 3,4 (Out = 3ms)	OUT
W47-W48	Diagnostic Enable	OUT
W50-W52	Alternate Boot Address	OUT
W51-W52	Primary Boot Address	IN
W53-W54	Wake-Up Circuit Enable	IN
W55-W56	Factory Test	IN
W56-W57	Factory Test	OUT
W58-W59	Precomp Enable	IN
W59-W60	Precomp Disable	OUT
W61-W62	1st Extended Alt. Addressing	OUT
W63-W64	2nd Extended Alt. Addressing	OUT
W65-W66	RX50 Drive Option	IN

Table 2-1 Factory Option Summary

OPTIONS	JUMPERS			
	14 - 15	16 - 17	61 - 62	63 - 64
Standard Address* 17772150[Oct](LSI-11) 20001468[Hex](MicroVax I & II)	IN	OUT	OUT	OUT
1st Alternate Address 17760334[Oct](LSI-11) 2000006C[Hex](MicroVax I & II)	OUT	IN	OUT	OUT
2nd Alternate Address 17760354[Oct](LSI-11) 200000EC[Hex](MicroVax I & II)	IN	OUT	IN	OUT
3rd Alternate Address 17760374[Oct](LSI-11) 200000FC[Hex](MicroVax I & II)	OUT	IN	IN	OUT
4th Alternate Address 17760414[Oct](LSI-11) 2000010C[Hex](MicroVax I & II)	IN	OUT	OUT	IN
5th Alternate Address 17772144[Oct](LSI-11) 20001464[Hex](MicroVax I & II)	OUT	IN	OUT	IN
6th Alternate Address 17772154[Oct](LSI-11) 2000146C[Hex](MicroVax I & II)	IN	OUT	IN	IN
7th Alternate Address RESERVED	OUT	IN	IN	IN
*Factory Preset				

Table 2-2 Address Option Configuration

2.1.2. Logical Unit Numbers

The MXV50 allows selection of the starting logical unit number (LUN) and provides selection from LUN 0 to LUN 15. The default LUN is 0 and is selected by removing all the jumpers W33-W34, W35-W36, W37-W38, W39-W40. Some operating systems use this option to organize logical units from different controllers. If a particular LUN is required insert the appropriate jumpers as indicated by their binary weights referring to Table 2-3.

LOGICAL UNIT NUMBER	JUMPERS			
	W39-W40	W37-W38	W35-W36	W33-W34
0*	OUT	OUT	OUT	OUT
1	OUT	OUT	OUT	IN
2	OUT	OUT	IN	OUT
3	OUT	OUT	IN	IN
4	OUT	IN	OUT	OUT
5	OUT	IN	OUT	IN
6	OUT	IN	IN	OUT
7	OUT	IN	IN	IN
8	IN	OUT	OUT	OUT
9	IN	OUT	OUT	IN
10	IN	OUT	IN	OUT
11	IN	OUT	IN	IN
12	IN	IN	OUT	OUT
13	IN	IN	OUT	IN
14	IN	IN	IN	OUT
15	IN	IN	IN	IN

*Factory Preset

Table 2-3: Logical Unit Number Option

2.1.3. Device Interrupt Priority

The MXV50 supports the four-level device interrupt priority scheme compatible with the LSI-11/23. The controller asserts interrupt requests and monitors higher level request lines during interrupt arbitration as described in Table 2-3. The level four request is always asserted by the controller, regardless of its priority, to maintain compatibility with the LSI-11 and LSI-11/2 processors. The interrupt priority level is configured to level four at the factory. If a different interrupt level is desired cut the foil-etch jumpers between W19-W20, W22-W23, W24-W25, W28-W29, and W30-W31. Refer to Table 2-4 for the proper jumpers to insert for the desired priority level.

PTY LEV	ASRT	MON	JUMPERS									
			18-19	19-20	21-22	22-23	24-25	25-26	27-28	28-29	30-31	31-32
4*	4	5,6	Out	In	Out	In	In	Out	Out	In	In	Out
5	4,5	6	Out	In	In	Out	In	Out	Out	In	Out	In
6	4,6	7	In	Out	Out	In	Out	In	In	Out	In	Out
7	4,6,7	None	In	Out	In	Out	Out	In	In	Out	Out	In

*Factory Preset

Table 2-4: Priority Level Configuration

2.1.4. Bootstrap

The controller board incorporates a transparent firmware bootstrap compatible with the LSI 11 series processors. The bootstrap is initiated whenever program execution is started at location 17773000(8) homing all drives to track 0. Next, block 0 of unit 0 is read and loaded into memory starting at location 0. Program execution is then transferred to location 0. Controllers are shipped with this feature disabled. To enable the bootstrap insert the jumper between W41 and W42 as shown in Table 2-5.

NOTE

Only one bootstrap should be enabled in a system for proper operation. If another bootstrap exists in the system, it must be disabled before enabling the controller bootstrap.

BOOTSTRAP	JUMPERS
	W41 - W42
ENABLED	IN
DISABLED *	OUT

*Factory Preset

Table 2-5: Bootstrap Option

An alternate bootstrap address selection is available for systems with multiple boot controllers. The alternate address is 17765000(8). Since some bootstrap controllers may use this location it is advisable to consult the documentation for the various bootstrap controllers to avoid multiple addressing problems. Controllers are shipped with the primary boot address enabled. To select the alternate boot address cut the jumper between W51 to W52 (on the solder side of the PCB) and insert a jumper between W50 and W52 as shown in Table 2-6.

BOOTSTRAP ADDRESS	JUMPERS	
	W50 - W52	W51 - W52
Primary 17773000(8)*	OUT	IN
Alternate 17765000(8)	IN	OUT

*Factory Preset

Table 2-6: Bootstrap Address

2.1.5. Internal Reset

The controller provides an internal wake-up circuit. This logic initializes the controller at power up, resetting all control outputs and forcing the Micro sequencer to begin program execution at location zero. Controllers are shipped with this feature enabled. Although it is not recommended to disable this feature, it can be disabled by cutting the etch jumper between W32 and W33

INTERNAL RESET	JUMPER
	W53 - W54
ENABLED *	IN
DISABLED	OUT

*Factory Preset

Table 2-7: Internal Reset Option

2.1.6. RX50 Drive Option

The controller is shipped with this option disabled. Enabling this option provides compatibility with the DEC RX50 drives as drives 0 and 1 and compensates for the shared actuator. To enable cut the jumper between W65 and W66 on the solder side of the card as indicated in Table 2-8. This jumper is located behind U27 as shown in figure 2-1.

RX50 DRIVE OPTION	JUMPER W65 - W66
ENABLED	OUT
DISABLED *	IN
*Factory Preset	

Table 2-8 RX50 Drive Option

2.1.7. Multiple Drive Mode

The controller is shipped with multiple drive mode disabled. Enabling this option provides control of up to four drives. To enable the multiple drive mode, jumper W12 and W13 as shown in Table 2-9.

MULTIPLE DRIVE MODE	JUMPER W12 - W13
ENABLED (4 Drives)	IN
DISABLED (2 Drives)*	OUT
*Factory Preset	

Table 2-9: Multiple Drive Mode

2.1.8. Step Rate Control

Up to four drives can be interfaced with the MXV50 controller. These drives are organized into two pairs. A jumper is provided to select either a 6MS or 3MS step rate for each pair of drives. The controller is shipped configured with a 3ms step rate for both pair of drives.

Alternate step rates can be selected for each pair of drives. Refer to Table 2-10 for desired step rate option and associated jumpers.

STEP RATE		JUMPERS	
DS1/DS2	DS3/DS4	W43 - W44	W45 - W46
3 *	3 *	OUT	OUT
6	3	IN	OUT
3	6	OUT	IN
6	6	IN	IN

* Factory Preset

Table 2-10: Step Rate

2.1.9. Double Sided Operation

The MXV50 controller offers control for double sided drives and when enabled provides twice the storage of standard RX50 media. The controller recognizes the presence of single sided or double sided media whenever an ONLINE or SET UNIT CHARACTERISTICS command is issued to the controller. Double sided media are created by formatting with this option enabled. All other media are recognized as single sided.

NOTE

Do not enable double sided on drives 1 and 2 when using the RX50 drive option

WARNING

Some software may be unable to use the double sided feature.

CAUTION

When double sided diskettes are interchanged to controllers not recognizing the extra capacity, the software may recognize a capacity difference between the size initialized on the media and that which the controller provides the system. This may result in errors.

Controllers are shipped with this feature disabled. To enable double sided operation on drives 1 and 2 jumper W3 to W4. In four drive systems to enable double sided operation on drives 3 and 4 jumper W4 and W5. Refer to Table 2-11.

DOUBLE SIDED	JUMPERS	
	W3 - W4	W4 - W5
DISABLED *	OUT	OUT
1 and 2 ENABLED	IN	OUT
3 and 4 ENABLED	OUT	IN
1,2,3, & 4 ENABLED	IN	IN

*Factory Preset

Table 2-11: Double Sided

2.1.10. Write Precompensation

The MXV50 controller provides hardware write precompensation to reduce the bit shift exhibited by all drives as the recorded flux density increases. The controller recognizes the patterns which produce bit shift and precompensates the written pattern. This unique feature allows the controller to perform reliably with any Shugart compatible drive.

Controllers are shipped with this feature enabled and it is recommended that for more reliable operation the feature not be disabled. However, if so desired, the feature can be defeated by cutting the foil-etched jumper between W58 and W59 and inserting a jumper between W59 and W60 as shown in Table 2-12.

WRITE PRECOMPENSATION	JUMPERS	
	W58 - W59	W59 - W60
ENABLED*	IN	OUT
DISABLED	OUT	IN

*Factory Preset

Table 2-12: Write Precompensation

2.1.11. 5 1/4 Inch Head Load Option

The controller provides a head load signal for drives requiring this option. Since several 5 1/4" drives use the pin reserved for head load operation for vendor unique functions, this option is normally disabled. If required jumper W8 to W9 to enable the head load signal to the drive as shown in Table 2-13.

HEAD LOAD OPTION	JUMPERS
	W8 - W9
ENABLED	IN
DISABLED*	OUT
*Factory Preset.	

Table 2-13: Head Load Option

2.1.12. Diagnostic Enable

The MXV50 controller incorporates a set of internal diagnostics to verify proper controller operation. With the diagnostic enable jumper removed the internal diagnostics only operate once during the power up cycle. Upon successful completion the green LED at the edge of the board is turned on. When the diagnostic enable jumper is installed, the controller continuously operates the self diagnostics, turning off the LED at initiation and turning on the LED upon successful completion of each pass of the diagnostics. The effect is to blink the LED once for each successful pass of the diagnostic. If an error occurs the diagnostics will loop on the failing test until successfully completed.

Self tests include:

a) Proper sequencer branch operation
b) Condition code testing
c) Register tests
d) ALU operation test
e) RAM verification
f) Instruction tests

The controller is shipped with this feature disabled. It is only necessary to enable the diagnostics to determine if the controller is not functioning properly. When the self diagnostics are enabled all other controller operations are suspended. To enable this option, install the jumper from W47 to W48. Refer to Table 2-14.

DIAGNOSTICS	JUMPERS
	47 - 48
DISABLED *	OUT
ENABLED	IN

*Factory Preset

Table 2-14: Diagnostic Enable

2.1.13. Miscellaneous Options

There are several options related to factory configuration of the controller. These options must be configured as shown for proper operation of the controller. Refer to Table 2-15 for these options. During DMA operations if the the bus address established extends into the peripheral address page the controller asserts bank select 7 (BS7) as required by normal bus protocol. If the application requires extended memory which overlaps the peripheral address page, this option can be disabled as indicated in Table 2-15.

OPTIONS	JUMPERS			
	W55 - W56	W56 - W57	W10 - W11	W11-W49
FACTORY TEST*	IN	OUT		
BS7 ENABLED*	-	-	IN	OUT
BS7 DISABLED	-	-	OUT	IN

* Factory Preset

Table 2-15: Miscellaneous Options

2.2. DRIVE CONFIGURATION

The controller provides an industry standard floppy interface compatible with most available drives. However, for proper operation, each drive must be configured with attention to several options. Table 2-16 summarizes these options.

OPTION	DESCRIPTION	DUAL DRIVE 0	DUAL DRIVE 1	SINGLE DRIVE 0
DS1	Drive select 0	IN	OUT	IN
DS2	Drive select 1	OUT	IN	OUT
DS3	Drive select 2	OUT	OUT	OUT
DS4	Drive select 3	OUT	OUT	OUT
HS		IN	IN	IN
HM		OUT	OUT	OUT
MUX		OUT	OUT	OUT

Table 2-16: Drive Configuration

2.2.1. Drive Selection Signals

The controller provides four radial drive select signals. Each drive must be configured to one of the appropriate drive select signals. Upon initiation of a function the controller selects the appropriate drive, performs the operation and after fifteen revolutions of inactivity deselects the drive.

2.2.2. Head Load Signal

A separate head load signal is provided to prolong media life. The read/write heads are only loaded on the media during read/write operations. All other operations are completed with the heads in the unloaded position.

2.2.3. Motor Control

The controller has been configured for ease of use with the never DC motor drives. A motor on signal is provided and is activated by the controller upon the initiation of any function. Before the drive is used for read/write operations, a motor delay timer insures the drive is up to speed. After fifteen revolutions of inactivity, the signal is deselected, prolonging the life of the drive.

The motor on delay timer has been configured to operate with drives jumpered with motor on as a function of drive select. If the selected device has not changed from the previous selection, the motor is on and the drive ready, the function is initiated. If the selected drive has changed or the motor has stopped, then the motor is started and a motor on delay is performed. A valid ready status initiates the function while a not ready status results in an error being reported and the operation aborted.

2.3. CABLING

A 34-conductor ribbon cable connects the controller to any Shugart compatible drive(s). If the optional cable is purchased with the controller, connect the socket connector to the 34-pin header located at

the edge of the controller board. Observe the alignment of pin 1 of the socket connector and header as indicated by the arrows shown in Figure 2-2. The two 34-pin connectors should be connected to the corresponding drives, again observing the location of pin 1.

If the optional cable is purchased from an independent source, the following list of materials (or equivalent) will help in the construction of the required cable.

QTY	DESCRIPTION	MFG	NUMBER
1 each	34 pin socket connector	3M	3414-6034
2 each	34 pin edge connector	3M	3463-0000
A/R	34 conductor ribbon cable	3M	3365-34

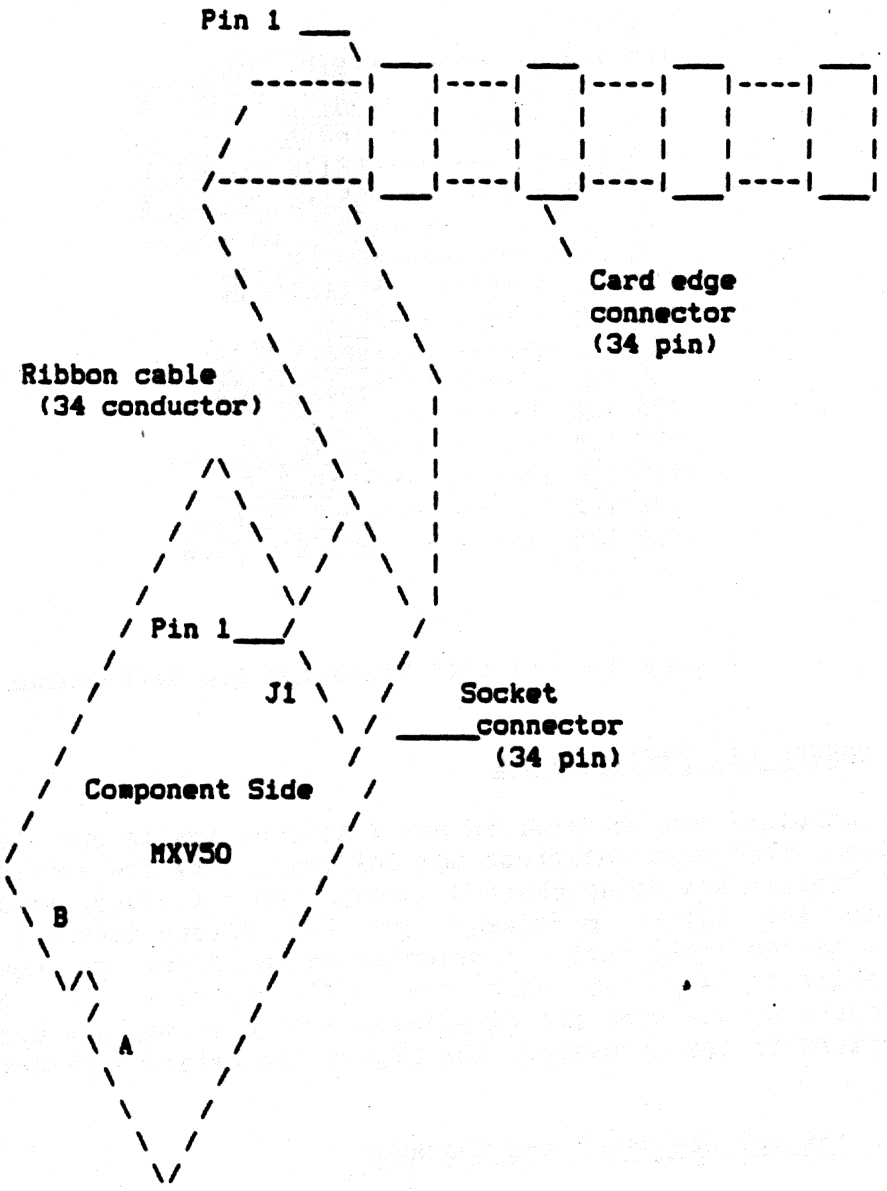


Figure 2-2: Drive/Controller Cabling

The connector pins illustrated in Figure 2-3 are compatible with both the Shugart SA 460 and Tandon 100-4. Any drive that has both a Shugart compatible interface and connector should function properly with the controller.

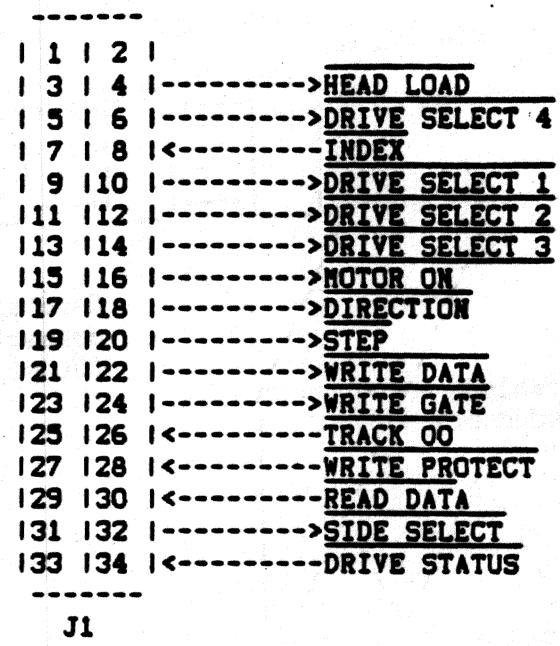


Figure 2-3: 5 1/4" Connector Pin Definitions

2.4. CONTROLLER INSTALLATION

The controller can be inserted and will function in any LSI-11 bus slot provided that both interrupt and DMA continuity are maintained. Since these signals are daisy chained through the bus slots, no unused slots between the LSI-11 processor and the floppy controller may exist. Determine the order that the priority chain flows by consulting the documentation supplied with the LSI-11 system. Note that when two interrupts of the same priority level are asserted, the closer a device is located to the processor, the higher its relative priority.

2.4.1. Initial Operation and Checkout

The MXV50 controller has been designed to operate with the entire series of Digital Equipment computers incorporating the Q-Bus architecture. These include the LSI 11/2, LSI 11/23, LSI 11/73, MicroVax I and MicroVax II computers. This section provides easy to follow procedures which can be used to verify proper controller operation. These procedures have been segmented in order to simplify initial checkout while additionally providing a comprehensive mechanism for the sophisticated user to isolate failure mode. In order to orient the procedures to the site specific requirements the procedures are further divided into three sections relating to the corresponding environment. These sections cover the LSI 11 series, the MicroVax I series and the MicroVax II series.

Before proceeding with the following verify that the controller and drive have been configured as described in sections 2.1 through 2.3.

NOTE

Before installation confirm that there is no address conflict with existing devices. The following procedures assume that the controller has been configured at the primary address 17772150(8) as unit 0.

NOTE

The onboard bootstrap must be disabled before attempting the following procedures.

1. Insure that the cables have been installed correctly and pover is applied to the drives.

2. Place the RUN/HALT switch on the processor to the HALT position and turn on the processor. The green diagnostic LED located at the edge of the controller should be illuminated after several seconds indicating the controller is ready for operation. If the LED remains off, first verify that the backplane voltage is equal to or greater than 4.75 volts. Secondly, check for proper seating of all socketed components. In the unlikely event the LED remains off after several attempts consult your local representative or contact the factory. This may indicate a faulty controller. Subsequent procedures can be used to confirm a failure.

3. Inspect the drives for activity. If the cable is correctly installed and both the controller and drive options have been properly configured there will be no drive activity. If any of the drive select indicators are illuminated and/or any of the drive motors are operating, reconfirm all drive options and insure that the cable is installed properly. A reversed cable asserts all drive signals.

NOTE

If this condition occurs any media which was in the drive have been partially erased and will need to be reformatted.

At this point the controller is installed correctly and is ready for use. If a more comprehensive checkout is required, refer to the corresponding site specific section which follovs for additional test procedures. The following site specific procedures can be used to isolate hardware or software failures in the event the controller does not function properly with a corresponding DEC operating system.

2.4.2. Operation and Checkout on LSI-11 Processors

The following procedures provide a comprehensive test of bus communication, interrupt and DMA operation by performing a controller initialization. In order to provide an environment for interrupt and DMA activity several memory and register locations must be defined. The processor will be forced to loop at location 0 in memory. Trap vectors are set up and used to capture interrupts in subsequent steps. Finally a pattern is set up in memory and will be used to verify DMA activity.

1. Place the RUN/HALT switch on the processor to the HALT position and either operate the RESTART switch or cycle the power off and on. An "@" character should be printed on the operators console indicating the console emulator is in effect.

2. To confirm proper register addressing, open the IP register by typing the address 17772150(8) followed by a slash. The line feed (LF) character will open the SA register which is the next address. An additional LF character will open the next address and should result with a question mark indicating an illegal address. Proper operation is indicated by the IP register containing the value 000000(8) and the SA register containing the valve 005500(8) as shown

```
@17772150/000000<LF>
17772152/005500<LF>
17772154/?
@
```

3. To confirm register and data path operations the controller's diagnostic wrap feature should be used. Open the SA register by typing 17772152 followed by a slash. Enter a 140000(8) followed by a carriage return to force the controller into diagnostic wrap mode. Subsequent entries into the SA register will be echoed for verification. Sequentially enter a pattern of zeros, all ones and two patterns of alternating ones and zeros as follows to confirm proper operation.

```
@17772152/005500 140000<CR>
@/140000 0<CR>
@/000000 177777<CR>
@/177777 125252<CR>
@/125252 052525<CR>
@/052525
```

4. The preceding operations confirm proper register response. In addition to bus interface and steering logic, the Supervisor data path is confirmed. If particular bits are dropped during this test it could indicate either failing bus transceiver chips or the supervisor data path chips. If bits are picked up during this test this could indicate corresponding bus transceivers, data path chips, or open etches on the controller.

5. Operate the RESTART switch or deposit a zero into the IP register to start initialization and force "STEP 1". Enter the corresponding data into memory and the registers as follows.

```
@0/XXXXXX  777<LF>
2/XXXXXX <LF>
4/XXXXXX 6 <LF>
6/XXXXXX 0 <CR>

@100/XXXXXX 102 <LF>
102/XXXXXX 2 <CR>

@400/XXXXXX 402 <LF>
402/XXXXXX 0 <LF>
404/XXXXXX 2 <CR>

@772/XXXXXX 7 <LF>
774/XXXXXX 1 <LF>
776/XXXXXX 2 <LF>
1000/XXXXXX 3 <LF>
1002/XXXXXX 4 <LF>
1004/XXXXXX 5 <LF>
1006/XXXXXX 6 <LF>
1010/XXXXXX 7 <CR>

@R6/XXXXXX 700 <LF>
R7/XXXXXX 0 <CR>
@RS/XXXXXX 0 <CR>
```

Then place the RUN/HALT switch in the RUN position and proceed by typing a "P". The run light on the processor should be illuminated indicating the "Run" condition.

```
@ P
```

If the processor halts, indicated by the address being printed on the operator console followed by the "@" character, recheck all entries and attempt the procedure again. Continued failure indicates improper processor operation. Remove the controller and reattempt the procedure. Continued failure indicates a processor or backplane problem.

If the run indicator does not remain illuminated and a halt condition is not entered the bus is hung. This could indicate interrupt or DMA continuity problems. Confirm proper installation of all controllers and peripherals in the system.

Halt the processor by placing the RUN/HALT switch in the Halt position. The location 000000 followed by the "@" character should be printed on the operators console

```
000000
@
```

6. The following procedure confirms proper controller interrupt operation and will force initialization to "STEP 2". In addition, the drive(s) will be accessed for the first time verifying cable installation. Open the SA register by typing 17772152 followed by a slash. The register should contain a 005500(8). Deposit a 100300(8) followed by a carriage return. Enter the run condition by placing the RUN/HALT switch in the RUN position and typing a "P" on the operators console. The drive(s) will be selected and the motor will be operated indicating "STEP 1" initialization. The processor will then HALT at location 404(8) indicating an interrupt occurred caused by completion of "STEP 1".

```
@17772152/ 005500 100300 <CR>
@ P
000404
@
```

If the processor does not halt within ten seconds interrupts are not functioning properly. If the run indicator does not remain illuminated the bus is hung indicating an interrupt continuity problem. To clear this condition place the RUN/HALT switch on the Halt position and operate the RESTART switch.

NOTE

The LSI 11/2 processor incorporates a time out on the interrupt acknowledge cycle. If a time out occurs the processor will halt at location zero indicating an interrupt continuity problem.

If the processor can be halted by placing the RUN/HALT switch in the Halt position either interrupts are not being requested or are disabled by the processor. Double check the contents of the processor status word by typing an RS followed by a slash. The register should contain zeros.

7. This procedure will confirm DMA operation. The SA register should contain a 010200(8) indicating "STEP 2". This can be verified by typing a 17772152 followed by a slash on the operators console. Deposit a 1000(8) into the SA register in order to force "STEP 3" by typing 1000 followed by a carriage return on the operators console. Place the RUN/HALT switch in the RUN position and enter the "run condition" by typing a "P". The processor should halt at location 404 indicated by a 000404 printed on the operators console followed by the "@" character. The SA register should now contain a 020300 indicating "STEP 3". Deposit a zero into the SA register and type a "P" on the operators console to force "STEP 4". The processor should halt at 404(8) as indicated by a 000404 followed

by a "@" character printed on the operators console. At this point memory locations 774(8) thru 1006(8) should have been cleared by the controller under DMA control. Verify these locations by typing a 772 followed by a slash on the operators console. Use the line feed character to open successive locations. Memory locations 772(8) and 1010(8) should still contain the original contents.

```
@17772152/ 010200   1000 <CR>
@ P
000404
@17772152/ 020300   0  <CR>
@ P
000404
@772/  000007 <LF>

774/0000007 <LF>
776/000000 <LF>
1000/000000 <LF>
1002/000000  <LF>
1004/000000  <LF>
1006/000000  <LF>
1010/0000007 <CR>
@
```

If the indicated locations are not cleared this could indicate DMA addressing problems. However, if the register operation and supervisor data path has been confirmed in the previous tests this is doubtful. The operation should be reconfirmed starting back at procedure 5 and repeating all procedures carefully. If the failure persists and the memory processor and backplane have previously been tested and are in good working order then the controller is possibly faulty.

8. This operation will complete controller initialization and should not create any additional interrupts. The SA register should contain a 040XXX(8) which can be verified by typing a 17772152 followed by a slash on the operators console. To complete initialization deposition a "go" bit 000001(8) into the SA register and type a "P" on the operators console to enter the "Run Condition". The processor should not halt. To further verify completion of initialization, halt the processor by placing the RUN/HALT switch in the Halt position. The Halt condition will be indicated by 000000 followed by the"@" character printed on the operators console. Both the IP and SA registers should contain zeros.

```
@17772152/ 040XXX 1 <CR>
@ P

000000
@17772150/ 000000 <LF>
17772152/000000 <CR>
@
```

At this point the majority of the Host interface functions of the controller have been verified. The following steps can be used to confirm both read and write operation of the drive interface.

9. Driver interface logic will be tested by formatting a blank diskette. Since the format operation uses a two pass technique which first writes headers followed by reading the headers and writing the corresponding data fields, both the read and write electronics are tested. Since each cylinder contains unique headers which carry position information the control logic is additionally checked. Any failure causes the format operation to be aborted. To begin place the RUN/HALT switch into the Halt position and operate the RESTART switch. This will place the controller into "STEP 1". Open the SA register by typing 17772152 followed by a slash on the operators console. It should contain a 005500(8). Deposit the keyword 152000(8) into the SA register to invoke format mode. The controller responds by setting "STEP 2". Reopen the SA register, it should contain 010000(8). Formatting is initiated on drive 0 by entering a 152000 into the SA register at this point. While the formatting operation is in process the SA register will contain zeros.

```
@17772152/005500  152000 <CR>
@/ 010000  152000 <CR>
@/ 000000 <CR>
```

Formatting takes approximately 90 seconds for a single sided diskette. Upon successful completion of the formatting operation the value 010000(8) is loaded into the SA Register; otherwise, both the SA and IP registers contain zero's.

```
@17772150/000000        <LF>
17772152/000000        <CR>
@
```

If an error occurs this indicates either write or read difficulties. Reconfirm cable connections and both controller and driver options and retest. If the failure persists consult your local representative or the factory for service.

2.4.3. Operation and Checkout on the MicroVax I Processor

The following sequence of operations will assist an interested user in determining if the MXV50 is operating properly.

These operations provide a comprehensive test of bus communication, interrupt and DMA operation by performing a controller initialization. In order to provide an environment for interrupt and DMA activity several memory and register locations must be defined. Trap vectors are set up and used to capture interrupts in subsequent steps. Finally a pattern is set up in memory and will be used to verify DMA activity.

Before operating this procedure the MicroVax I processor must be configured to allow HALT ON BREAK DETECT and HALT when a Halt instruction is encountered. This is accompolished by switching the option switches 3, 4, and 5 on the Data Path Module to the ON position Refer to the MicroVax I CPU Technical Description Manual EK-KD32A-TD-CN1, section 3. These switches are located in the upper left hand corner of the module adjacent to the system identification register switches.

1. Enter console mode by depressing the front panel halt switch, or if "Halt on Break" is enabled, by depressing the "Break" key on the console terminal. If the front panel halt switch is used, it should be depressed a second time. This allows the CPU to operate when the continue command is given.

2. To confirm proper register addressing, examine the IP register by typing E/W/P 20001468 followed by a carriage return. An E + followed by a carriage return will open the next address which is the SA register. An additional E + <CR> will open the next location and should result with a question mark indicating an illegal address. Proper operation is indicated by the IP register containing the value 0000(16) and the SA register containing the value 0B40(16) as shown

```
>>> E/W/P 20001468<CR>
  P 20001468  00000000
>>> E +<CR>
  P 2000146A  00000B40
>>> E +<CR>
?20
>>>>
```

3. To confirm register and data path operations using the controller's diagnostic wrap feature deposit C000(16) into the SA register by typing D/W/P 2000146A C000 followed by a carriage return. This forces the controller into diagnostic wrap mode. Sequentially enter a pattern of zeros, all ones and two patterns of alternating ones and zeros as follows to confirm proper operation.

```
>>> D/W/P 2000146A C000<CR>
>>> E/W/P 2000146A<CR>
  P 2000146A  0000C000
>>> D/W/P 2000146A FFFF<CR>
>>> E/W/P 2000146A<CR>
  P 2000146A  0000FFFF
>>> D/W/P 2000146A AAAA<CR>
>>> E/W/P 2000146A<CR>
  P 2000146A  0000AAAA
>>> D/W/P 2000146A 5555<CR>
>>> E/W/P 2000146A<CR>
  P 2000146A  00005555
>>>
```

4. The preceding operations confirm proper register response. In addition to bus interface and steering logic, the Supervisor data path is confirmed. If particular bits are dropped during this test it could indicate either failing bus transceiver chips or the supervisor data path chips. If bits are picked up during this test it could indicate corresponding bus transceivers, data path chips, or open etches on the controller.

5. Deposit a zero into the IP register to start initialization and force "STEP 1". Enter the corresponding data into memory and registers as follows.

```
>>> I
>>> U
>>> D/L/P/ 0 FFFFFFFF<CR>
>>> D + FFFFFFFF<CR>
>>> D + FFFFFFFF<CR>
>>> D + FFFFFFFF<CR>
>>> D + FFFFFFFF<CR>
>>> D + FFFFFFFF<CR>
>>> D + FFFFFFFF<CR>
>>> D + FFFFFFFF<CR>
>>> D/L 1000 FFFD3101<CR>
>>> D/L 2000 200<CR>
>>> D/L 500 2000<CR>
>>> D/I 11 200<CR>
>>> D/G F 1000<CR>
>>> D/G E 200<CR>
>>> D P 04130000<CR>
```

Ensure the Halt switch is not engaged. Continue by typing a "C<CR>". The run light on the processor should be illuminated indicating the "Run" condition.

```
>>> C<CR>
```

If the processor halts recheck all entries and attempt the procedure again. Continued failure indicates improper system operation. Remove the controller and reattempt the procedure. Continued failure indicates a processor or backplane problem.

Halt the processor. The following message will be displayed on the console.

```
00001001 02
>>>
```

6. This operation will confirm proper controller interrupt operation and force initialization to "STEP 2". In addition, the drive(s) will be accessed for the first time verifying proper cable installation. Examine the SA register by typing E/W/P 2000146A followed by a <CR>. The register should contain a 0B40(16). Deposit a 80C0 (16) by typing D/W/P 2000146A 80C0 followed by a carriage return. Enter the run condition by typing a "C <CR>" on

the operator's console. The drive(s) will be selected and the motor(s) will be operated indicating "STEP 1" initialization. The processor will then HALT at location 2001(16) indicating an interrupt occurred caused by completion of "STEP 1".

```
>>> E/W/P 2000146A<CR>
  P  2000146A  00000B40
>>> D/W/P 200146A 80C0<CR>
>>> C<CR>
00002001 06
>>>
```

If the processor does not halt within ten seconds interrupts are not functioning properly.

If the processor can be halted interrupts are not being requested or are disabled by the processor. Double check the contents of the processor status word by typing an E P<CR>. The register should contain 04130000(16)

7. This operation will confirm DMA operation. The SA register should contain 1080(16) indicating "STEP 2". This can be verified by typing E/W/P 2000146A followed by a <CR> on the operator's console. Deposit a 10(16) into the SA register to force "STEP 3" by typing a D/W/P 2000146A followed by a 10(16) and ending with a carriage return on the console. Enter the "RUN" condition by typing a "C<CR>". The processor should halt at location 2001(16) indicated by 00002001 06 printed on the console followed by the ">>>" character. The SA register should nov contain a 20C0 indicating "STEP 3". Deposit a zero into the SA register and type a "C<CR>" on the operator's console to force "STEP 4". The processor should halt at 2001(16) as indicated by 00002001 06 followed by a ">>>" character printed on the operator's console. At this point memory locations C(16) thru 14(16) should have been cleared by the controller under DMA control. Verify these locations by typing E/L/P 8<CR> followed by E +<CR> four times to examine file successive double vord locations. Memory locations 8(16) and 18(16) should still contain the original contents (FFFFFFFF(16)).

```
>>> E/W/P 2000146A<CR>
  P 2000146A  00001080
>>> D/W/P 2000146A 10 <CR>
>>> C<CR>
00002001 06
>>> E/W/P 2000146A <CR>
P   2000146A  000020C0
>>> D/W/P 2000146A 0<CR>
>>> C<CR>
00002001 06
```

```
>>> E/L/P 8 <CR>
  P 00000008 FFFFFFFF
>>> E +<CR>
  P 0000000C 00000000
>>> E +<CR>
  P 00000010 00000000
>>> E +<CR>
  P 00000014 00000000
>>> E +<CR>
  P 00000016 FFFFFFFF
>>>
```

If the correct locations are not cleared this could indicate DMA addressing problems. However, if the register operation and supervisor data path have been confirmed in the previous tests this is doubtful. The operation should be reconfirmed starting back at procedure 5 and repeating all procedures carefully. If the failure persists and the memory processor and backplane have previously been tested and are in good working order then the controller may be faulty

8. This step will complete controller initialization and should not create any additional interrupts. The SA register should contain a 20C0(16) which can be verified by typing a E/W/P 200146A followed by a <CR> on the operator's console. To complete initialization deposit a "go" bit 0001(16) into the SA register and type a "C<CR>" on the operator's console to enter the "Run" condition. The processor should not halt. To further verify completion of initialization halt the processor and examine the IP and SA registers; both should contain zeros.

```
>>> E/W/P 200146A<CR>
  P 2000146A  00004079
>>> D/W/P 200146A 1<CR>
>>> C<CR>
00001001 02
>>> E/P 20001468<CR>
  P 20001468  00000000
>>> E +<CR>
  P 2000146A 00000000
```

At this point the majority of the Host interface functions of the controller have been verified. The following procedure can be used to confirm both read and write operation of the drive interface.

9. Drive interface logic will be tested by formatting a blank diskette. Since the format operation uses a two pass technique which first writes headers followed by reading the headers and writing the corresponding data fields, both the read and write electronics are tested. Since each cylinder contains unique headers which carry position information the control logic is additionally checked. Any failure causes the format operation to be aborted. Halt the processor and type a "U" followed by a <CR>.

This will place the controller into "STEP 1". Examine the SA register by typing E/W/P 2000146A followed by a <CR> on the operators console. It should contain a 0B40(16). Deposit the keyword D400(16) into the SA register to invoke format mode. The controller responds by setting "STEP 2". Reexamine the SA register; it should contain 1000(16). Formatting is initiated on drive 0 by depositing D400(16) into the SA register again. While the formatting operation is in process the SA register will contain zeros.

```
>>> E/W/P 2000146A<CR>
  P 2000146A  00000B40
>>> D/W/P 2000146A D400<CR>
>>> E/W/P 2000146A<CR>
  P 2000146A 00001000
>>> D/W/P 2000146A D400<CR>
>>>
```

Formatting takes approximately 90 seconds for a single sided diskette. Upon successful completion of the formatting operation the value 01000(16) is loaded into the SA Register otherwise both the SA and IP registers contain zero's.

```
>>> E/W/P 20001468<CR>
  P 20001468  00000000
>>> E +<CR>
  P 2000146A  00000000
>>>
```

If an error occurs this indicates either write or read difficulties. Reconfirm cable connections and both controller and drive options and retest. If the failure persists consult your local representative or the factory for service.

2.4.4. Operation and Checkout on the MicroVax II Processor

The following sequence of operations will assist an interested user in determining if the MXV50 is operating properly.

These operations provide a comprehensive test of bus communication, interrupt and DMA operation by performing a controller initialization. In order to provide an environment for interrupt and DMA activity several memory and register locations must be defined. Trap vectors are set up and used to capture interrupts in subsequent steps. Finally a pattern is set up in memory and will be used to verify DMA activity.

1. Enter console mode by depressing the front panel halt switch, or if "Halt on Break" is enabled, by depressing the "Break" key on the console terminal. If the front panel halt switch is used, it should be depressed a second time. This allows the CPU to operate when the continue command is given.

2. To confirm proper register addressing, examine the IP register by typing E/W/P 20001468 followed by a carriage return. An E followed by a carriage return will open the next address which is the SA register. An additional E <CR> will open the next location and should result with a question mark indicating an illegal address. Proper operation is indicated by the IP register containing the value 0000(16) and the SA register containing the value 0B40(16) as shown

```
>>> E/W/P 20001468<CR>
  P 20001468  0000
>>> E<CR>
  P 2000146A  0B40
>>> E<CR>
?1A ILL ADDR
>>>>
```

3. To confirm register and data path operations using the controller's diagnostic vrap feature deposit C000(16) into the SA register by typing D/W/P 2000146A C000 followed by a carriage return. This forces the controller into diagnostic vrap mode. Subsequent entries into the SA register will be echoed for verification. Sequentially enter a pattern of zeros, all ones and two patterns of alternating ones and zeros as follows to confirm proper operation.

```
>>> D/W/P 2000146A C000<CR>
>>> E/W/P 2000146A<CR>
  P 2000146A  C000
>>> D/W/P 2000146A FFFF<CR>
>>> E/W/P 2000146A<CR>
  P 2000146A  FFFF
>>> D/W/P 2000146A AAAA<CR>
>>> E/W/P 2000146A<CR>
  P 2000146A  AAAA
>>> D/W/P 2000146A 5555<CR>
>>> E/W/P 2000146A<CR>
  P 2000146A  5555
>>>
```

4. The preceding operations confirm proper register response. In addition to bus interface and steering logic, the Supervisor data path is confirmed. If particular bits are dropped during this test it could indicate either failing bus transceiver chips or the supervisor data path chips. If bits are picked up during this test it could indicate corresponding bus transceivers, data path chips, or open etches on the controller.

5. Deposit a zero into the IP register to start initialization and force "STEP 1". Enter the corresponding data into memory and registers as follows.

```
>>> I
>>> U
>>> D/L/P/N:1FF 20088000 80000000<CR>
>>> D/L/P/N:1FF 0 0<CR>
>>> D/L/P/N:0F 0 FFFFFFFF<CR>
>>> D/L 1000 FFFD3101<CR>
>>> D/L 2000 200<CR>
>>> D/L 500  2000<CR>
>>> D/W 20001F40  20<CR>
>>> D/I 11  200<CR>
>>> D R15 1000<CR>
>>> D R14 200<CR>
>>> D PSL 04130000<CR>
```

Ensure the Halt switch is not engaged. Continue by typing a "C<CR>". The run light on the processor should be illuminated indicating the "Run" condition.

```
>>> C<CR>
```

If the processor halts recheck all entries and attempt the procedure again. Continued failure indicates improper system operation. Remove the controller and reattempt the procedure. Continued failure indicates a processor or backplane problem.

Halt the processor. The following message will be displayed on the console.

```
?02  EXT HLT
PC = 00001001
>>>
```

6. This operation will confirm proper controller interrupt operation and force initialization to "STEP 2". In addition, the drive(s) will be accessed for the first time verifying cable installation. Examine the SA register by typing E/W/P 2000146A followed by a <CR>. The register should contain a 0B40(16). Deposit a 80C0 (16) by typing D/W/P 2000146A 80C0 followed by a carriage return. Enter the run condition by typing a "C <CR>" on the operator's console. The drive(s) will be selected and the motor(s) will be operated indicating "STEP 1" initialization. The processor will then HALT at location 2001(16) indicating an interrupt occurred caused by completion of "STEP 1".

```
>>>  E/W/P 2000146A<CR>
  P  2000146A  0B40
>>>  D/W/P 200146A 80C0<CR>
>>>  C<CR>
?06  HLT INST
PC = 00002001
>>>
```

If the processor does not halt within ten seconds interrupts are not functioning properly.

If the processor can be halted interrupts are not being requested or are disabled by the processor. Double check the contents of the processor status word by typing an E PSL<CR>. The register should contain 04130000(16)

7. This operation will confirm DMA operation. The SA register should contain 1080(16) indicating "STEP 2". This can be verified by typing E/W/P 2000146A followed by a <CR> on the operator's console. Deposit a 10(16) into the SA register to force "STEP 3" by typing a D/W/P 2000146A followed by a 10(16) and ending with a carriage return on the console. Enter the "RUN" condition by typing a "C<CR>". The processor should halt at location 2001(16) indicated by ?06 HLT INST, PC = 00002001 printed on the console followed by the ">>>" character. The SA register should now contain a 20C0 indicating "STEP 3". Deposit a zero into the SA register and type a "C<CR>" on the operator's console to force "STEP 4". The processor should halt at 2001(16) as indicated by ?06 HLT INST, PC = 00002001 followed by a ">>>" character printed on the operator's console. At this point memory locations C(16) thru 14(16) should have been cleared by the controller under DMA control. Verify these locations by typing E/L/P 8<CR> to examine successive locations. Memory locations 8(16) and 18(16) should still contain the original contents (FFFFFFFF(16)).

```
>>> E/W/P 2000146A<CR>
  P 2000146A  1080
>>> D/W/P 2000146A 10 <CR>
>>> C<CR>
?06 HLT INST
PC = 00002001
>>> E/W/P 2000146A <CR>
P   2000146A  20C0
>>> D/W/P 2000146A 0<CR>?
>>> C<CR>
?06 HLT INST
PC = 00002001


>>> E/L/P 8 <CR>
  P 00000008 FFFFFFFF
>>> E<CR>
  P 0000000C 00000000
>>> E<CR>
  P 00000010 00000000
>>> E<CR>
  P 00000014 00000000
>>> E<CR>
  P 00000016 FFFFFFFF
>>>
```

If the correct locations are not cleared this could indicate DMA addressing problems. However, if the register operation and supervisor data path have been confirmed in the previous tests this is doubtful. The operation should be reconfirmed starting back at procedure 5 and repeating all procedures carefully. If the failure

persists and the memory processor and backplane have previously been tested and are in good working order then the controller may be faulty.

8. This step will complete controller initialization and should not create any additional interrupts. The SA register should contain a 20C0(16) which can be verified by typing a E/W/P 200146A followed by a <CR> on the operator's console. To complete initialization deposit a "go" bit 0001(16) into the SA register and type a "C<CR>" on the operator's console to enter the "Run" condition. The processor should not halt. To further verify completion of initialization halt the processor and examine the IP and SA registers; both should contain zeros.

```
>>> E/W/P 200146A<CR>
  P 2000146A  4079
>>> D/W/P 200146A 1<CR>
>>> C<CR>
?02 EXT HLT
PC= 1001
>>> E/L/P 20001468<CR>
  P 20001468  00000000
```

At this point the majority of the Host interface functions of the controller have been verified. The following procedure can be used to confirm both read and write operation of the drive interface.

9. Drive interface logic will be tested by formatting a blank diskette. Since the format operation uses a two pass technique which first writes headers followed by reading the headers and writing the corresponding data fields, both the read and write electronics are tested. Since each cylinder contains unique headers which carry position information the control logic is additionally checked. Any failure causes the format operation to be aborted. Halt the processor and type a "U" followed by a <CR>. This will place the controller into "STEP 1". Examine the SA register by typing E/W/P 2000146A followed by a <CR> on the operators console. It should contain a 0B40(16). Deposit the keyword D400(16) into the SA register to invoke format mode. The controller responds by setting "STEP 2". Reexamine the SA register; it should contain 1000(16). Formatting is initiated on drive 0 by depositing D400(16) into the SA register again. While the formatting operation is in process the SA register will contain zeros.

```
>>> E/W/P 2000146A<CR>
  P 2000146A  0B40
>>> D/W/P 2000146A D400<CR>
  P 2000146A  1000
>>> D/W/P 2000146A D400<CR>
>>>
```

Formatting takes approximately 90 seconds for a single sided diskette. Upon successful completion of the formatting operation the value 01000(16) is loaded into the SA Register. Otherwise both

the SA and IP registers contain zero's.

```
>>> E/W/P 20001468<CR>
  P 20001468  0000
>>> E<CR>
  P 2000146A  0000
>>>
```

If an error occurs this indicates either write or read difficulties. Reconfirm cable connections and both controller and drive options and retest. If the failure persists consult your local representative or the factory for service.

Section 3

Hardware Description

3. GENERAL

This section reviews both hardware and operational features of the MXV50 controller. The controller is microprocessor based and as such, the majority of features are implemented in firmware. However, a general overview of the hardware will assist in the understanding of the controller.

3.1. HARDWARE OVERVIEW

The block diagram of the MXV50 is shown in Figure 3-1. Data and Address information are received and multiplexed onto an internal 8 bit data bus. An Address match circuit monitors the Data/Address lines for a register match condition representing either of the two controller registers, the IP and the SA Registers.

A match condition signals the sequencer to initiate the appropriate response. Reading the IP Register initiates a firmware polling routine while writing the IP causes the firmware to initialize. Writing the SA Register signals the CPU to accept the data available to the 8 bit Data bus. Likewise, reading the SA Register signals the CPU to place the contents of the SA Register on the data bus at which time it is made available to the bus transceivers thru the Latch.

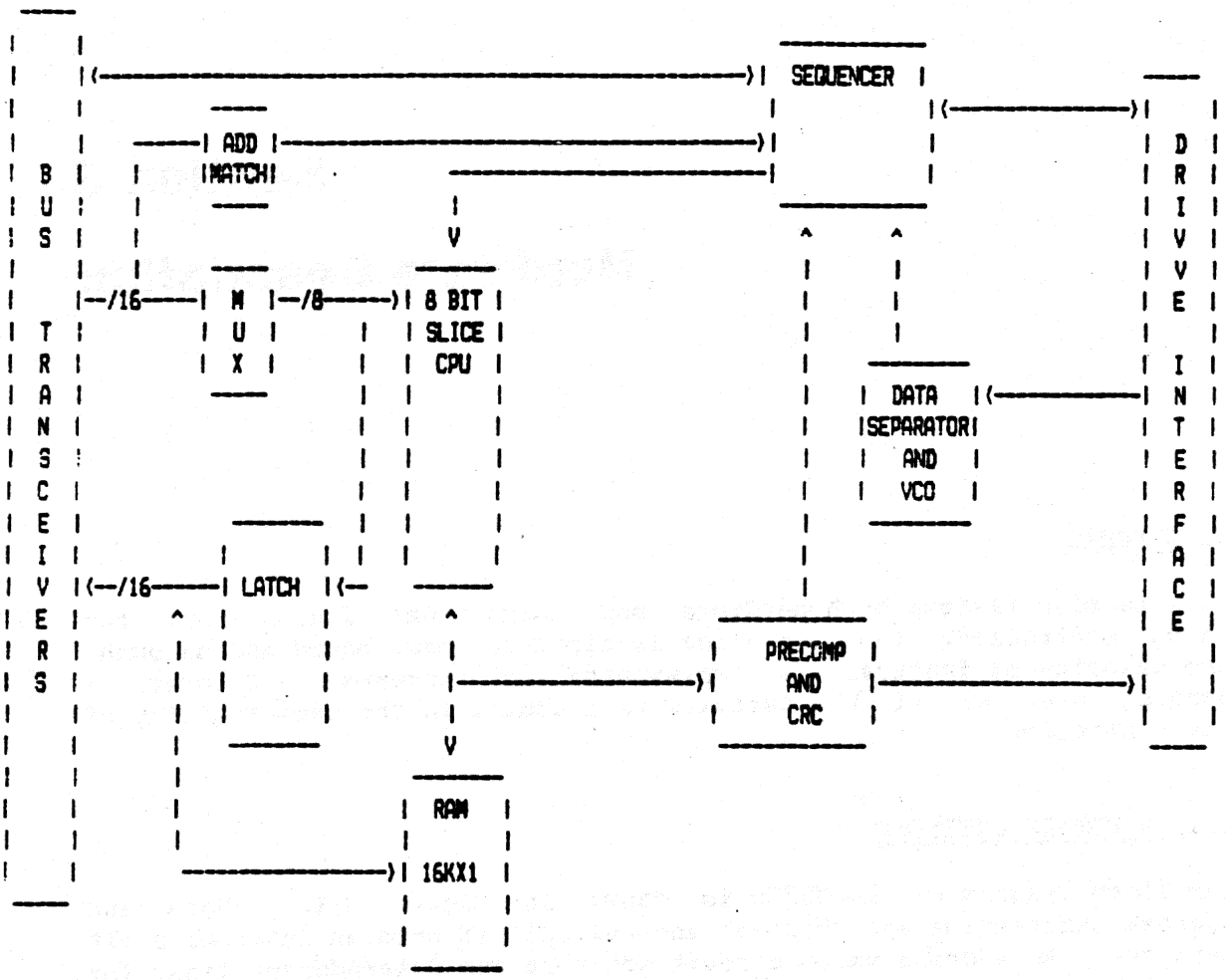


Figure 3-1 MXV50 BLOCK DIAGRAM

The MUX and LATCH logic is also used for all DMA transfers. Once the Q-BUS is acquired for a DMA cycle, address and data information are passed to the Q BUS through the LATCH logic. Data are read from the Q BUS through the MUX logic.

Internally the controller is a byte oriented bit serial state machine. All data read or written are immediately converted to a continuous serial bit stream which is stored in the internal RAM buffer. As data are serially stored in the internal buffer it is passed to the CRC generator. Upon completion of a transfer the resultant CRC is transferred from the generator and appended to the data in internal RAM. This mechanism of calculating the CRC as data are transferred from Host memory places the entire controller into the error detection loop.

To encode, data are serially read from the RAM buffer, encoded into MFM data and stored in an intermediate 8 bit register in the 8 bit CPU slice. Subsequently, the encoded data are passed to the

precompensation circuitry which applies an early, late or nominal shift to each encoded pulse based on a predetermined algorithm.

Data are read from the drive in a similar manner. To synchronize, each encoded data pulse is sampled by the sequencer for the occurrence of eight consecutive bits of zero or one data. At this point the data separation and VCO circuitry is synchronized with the data and specific address mark patterns can be searched. Since each bit can be dynamically examined, it is a straight forvard task to locate sector headers and the corresponding data field. Once located each bit of data is sequentially stored into the RAM buffer and simultaneously passed to the CRC circuitry. Upon completion of the last bit, the CRC error flags are examined to confirm if a valid transfer has taken place. To transfer to the host a 16 bit vord is assembled from the serial bit stream and subsequently transferred under DMA control.

All miscellaneous control signals are handled by the sequencer. An integral control decode and latch can selectively activate or deactivate any one of 32 individual signals. Also incorporated into the sequencer is a conditional multiplexer which is used to determine the condition for the branch circuitry providing an easy mechanism to monitor various signals both from the interface connectors and internal to the controller.

3.1.1. Host Interface

The controller is interfaced to the BDAL<15:0> lines through a set of 8641 bus transceiver buffers. RDA<15:0> represents the data present on the BDAL lines at any moment in time. TDA<15:0> are gated to the BDAL lines with the BUSENB signal. The main bus control signals BDIN, BDOUT, BSYNC, BRPLY are similarly interfaced using an 8641 transceiver which is always enabled. The remaining signals are received with 8640 bus receivers or generated by 7438 bus drivers. Most signals are always enabled except the extended address signals which are gated by TSACK. This allovs multiple definition of these control signals.

Jumper options exist to select interrupt priority and to disable the firmvare generated TBS7 signal from being gated onto the Q BUS. This is necessary for special applications vhere memory is used in place of the I/O page.

3.1.2. Host Control

The RDA<12:1> signals are monitored by an address decoder and address latch creating four signals; MATCH, ALT, LA01, and BOOTL. MATCH is asserted for either the standard or alternate address. ALT is set when the alternate address is enabled and decoded.

The address decode PAL has the ability to recognize ten different addresses. Eight are used for the base register addresses and the remaining are used for the two bootstrap addresses. Table 3-1 lists the available addresses.

Description	Address	
	LSI-11	MicroVax I & II
Standard Address	17772150	20001468
1st Alt. Address	17760334	200000DC
2nd Alt. Address	17760354	200000EC
3rd Alt. Address	17760374	200000FC
4th Alt. Address	17760414	2000010C
5th Alt. Address	17772144	20001464
6th Alt. Address	17772154	2000146C
7th Alt. Address	Reserved	Reserved
Primary Boot Add.	17773000	NA
Alternate Boot Ad	17765000	NA

Table 3-1: Addressing Options

The Match signal is monitored by hardware and if enabled (logic RPLY ENB L = 0) is responded to by hardware relieving the firmware from this task when busy. In this case, only a BRPLY signal is generated resulting in zero data read during a BDIN cycle and the loss of data during a BDOUT cycle except for the interrupt enable signal which is latched in hardware.

Two edge sensitive circuits monitor respectively the DMG input signals and interrupt priority protocol, generating DM Grant status, interrupt grant status and interlocking the propogation of these signals down the bus.

3.1.3. Micro Processor

The controller is designed around an 8 bit ALU consisting of two 2901 processors. Coupled with a 16 line to 8 line MUX, controlled by the SEL BUS and the BR20 term, and an 8 to 16 line latch, loaded by the STBBAL signal, all interfaced to the RDA<15:0> and TDA<15:0> lines provides a direct communication path with the Host processor. The TDA<13:0> signals are also used as address inputs to the on board 16K x 1 RAM buffer. This provides a simple mechanism to convert parallel to serial data writing the most significant bit from the ALU into the RAM at a preselected location. The output of the addressed location can be interrogated by the condition multiplexer which is discussed later.

Control of the ALU is a function of the contents of the pipeline register and outputs from the instruction decoder.

3.1.4. Control ROM and Instruction Decode

Seven 2K x 4 Proms make up a 28 bit Micro Word which is addressed by the twelve memory address lines MA<11:0> from the main sequencer. Outputs of this control ROM are coupled either directly (non-pipelined) or through Registers (pipelined) to other components which make up the bit slice processor. The signals BR<11:0> are coupled to the main sequencer and based on instruction type become the next address field. BR<11:4> are used as the intrinsic input to the CPU while BR<9:4> are decoded by the control decode to generate various control signals. BR<21:17> are used by the condition multiplexer to select the desired condition input. BR<19:12> are decoded by the 2901 CPU's as specific A and B registers. In addition BR16 is used by the next address decoder to determine the polarity of the test condition multiplexer. BR<27:25>, BR22, and BR20 are used by the 2901 CPU to select specific instruction types. Also BR<27:23> are decoded by the instruction decoder to create the signals ID<3:0> used for additional instruction select inputs to the 2901 CPU's; CC1 and CC0 used by the control decoder to effect the type of parallel hardware actions to occur, BS1 and BS0 used to control the next address decoder which determines whether branch, return, or continue operations are to occur.

3.1.5. Branch MUX and Main Sequencer

The condition multiplexer is a 32 to 1 line MUX which feeds the condition latch. The selected signal is sampled each clock cycle. The latched signal is used with a microword dependent polarity status and next address type determined by BS1 and BS0 signals. These signals are decoded by the next address decoder to generate the signals required by the main sequencer to determine the appropriate instruction branch. The sequencer then either increments the memory address register for a continue instruction, branches to the location defined by R<11:0> for a branch instruction or returns to the location stored on an internal 4 word stack for a return instruction.

3.1.6. Control Decode and Latch

The control logic decodes the control actions under control of the microword instruction type. Decodes are available for Write Ram, Strobe Bus Address Register, Control Latch Strobe or no-operation.

When the control latch is selected BR<7:4> determine which of the 32 control signals is either asserted or negated. The entire latch is cleared by either a software reset or bus initialize.

An additional decoder using two of these control signals a select port signal derived directly from the microword; to either select the host bus, pipeline intrinsic or a jumper option register as a source for the 2901 CPU.

3.1.7. Data Separation and Master Timing

A 12 MHZ oscillator is divided down to create the 6 MHZ MCLK signal and further divided to provide the 1MHZ, 500KHZ and 250 KHZ reference signals for writing to the drive and as reference for the VCO circuitry. Three monostable one shot provide multiple fixed cell reference for the various data rates. The appropriate VCO division and cell reference are selected by the rate selector and directly applied to the input of the phase comparator. The phase difference creates either a pump-up or pump-down signal which is amplified by a current mirror, coupling a discrete amount of charge either to or from the loop filter. The corresponding voltage changes results in a change in the VCO frequency in a direction which corrects for the phase difference causing a "locked" condition.

Initially the VCO is locked to the crystal oscillator as a reference. When reading from the drive a preamble region is determined by searching for a uniform "high frequency" signal. The loop is then synchronously switched from the crystal to the data using several control signals to suspend the phase locked loop circuitry from changing. There is circuity which reinstates the loop "in phase", significantly reducing "lock time" and improving data reliability.

Once in lock the firmware can synchronize a hardware signal to the combined data/clock stream in order to separate these signals. Two latches store the respective data and clock values for a full cell width to simplify the interface to the bit size processor. Each bit is directly handled by the 2901 CPU, sequentially stored into RAM and simultaneously passed to the CRC logic.

3.1.8. Write Precompensation

Encoded data are generated by firmware and shifted bit-wise into the precompensation logic in order to write the drive interface. This logic maintains two bits of history and future in order to calculate an early, nominal or late correction factor which is applied to the encoded bit. This shift is necessary in order to cancel out the effective shift in read back position due to the interaction of the head and media. This is only applied to tracks above 43 where the effect becomes a problem.

3.1.9. CRC Logic

As data are sequentially written bit-wise into RAM either from host memory or when reading from the drive it is clocked simultaneously into the CRC logic. When the last bit of data is entered, the CRC code can be clocked from the CRC logic on the CRC generator and examined to determined if an error occured.

3.1.10. Drive Interface

The drive interface consists of several open collector drivers which meet the interface specifications. Input signals are terminated by a 220/330 termination and directly coupled to the condition multiplexer. A power fail protection and voltage monitor which interlocks the write gate signal through a discrete circuit, and protects the drive from inadvertently being written causing "bad blocks". An L.E.D. indicates the functionality of this circuitry. If this logic is active a write protect status is reported to the firmware alloving the processor to inform the host of this condition.

3.2. LSI-11 BUS PIN ASSIGNMENTS

Input/Output Pin Assignments

Pin	Mnemonic	Function
AA1	BIRQ5 L	Interrupt request priority level 5
AB1	BIRQ6 L	Interrupt request priority level 6
AC1	BDAL16 L	Address line 16/memory error line
AD1	BDAL17 L	Address line 17/memory error enable
AE1	SSPARE1	Special spare
AF1	SSPARE2	Special spare
AH1	SSPARE3	Special spare
AJ1	GND	Ground
AK1	MSPAREA	Maintenance spare
AL1	MSPAREB	Maintenance spare
AM1	GND	Ground
AN1	BDMRL	Direct Memory Access (DMA) request
AP1	BHALT	Processor halt
AR1	BREF L	Memory refresh
AS1	+5B or +12B	+12 or +5 Vdc battery backup power
AT1	GND	Ground
AU1	PSPARE1	Power spare 1
AV1	+5B	+5 Vdc battery power
BA1	BDCOK H	DC power OK
BB1	BPOK H	AC power OK
BC1	BDAL18 L	Address line 18
BD1	BDAL19 L	Address line 19
BE1	BDAL20 L	Address line 20
BF1	BDAL21 L	Address line 21

BH1	SSPARE	Special spare
BJ1	GND	Ground
BK1	MSPAREB	Maintenance spares
BL1	MSPAREB	Maintenance spares
BM1	GND	Ground
BN1	BSACK L	This signal is asserted by a DMA device in response to the processor's BDMGO L signal, indicating that the DMA device is bus master.
BP1	BIRQ7 L	Interrupt request priority level 7
BR1	BEVNT L	External event interrupt request
BS1	+12B	+12 Vdc battery backup power
BT1	GND	Ground
BU1	PSPARE2	Power spare 2
BV1	+5	+5 Vdc power
AA2	+5	+5 Vdc power
AB2	-12	-12 Vdc power (optional)
AC2	GND	Ground
AD2	+12	+12 V power
AE2	BDOUT L	Data output
AF2	BRPLY L	Reply is asserted in response to BDIN L or BDOUT L and during IAK transaction.
AH2	BDIN L	Data input is used for two type of bus operation: 1. When asserted during BSYNC L time, BDIN L implies an input transfer with respect to the current bus master. 2. When asserted without BSYNC L, it indicates that an interrupt operation is occurring.
AJ2	BSYNC L	Synchronize is asserted by the bus master device to indicate that it has placed an address on the bus.
AK2	BWTBT L	Write/byte is used in two ways to control a bus cycle: 1. It is asserted during the leading edge of BSYNC L to indicate that an output sequence is to follow. 2. It is asserted during BDOUT L, in a DATOB bus cycle, for byte addressing.
AL2	BIRQ4 L	Interrupt request priority level 4
AM2	BIAKI L	Interrupt acknowledge in
AN2	BIAKO L	Interrupt acknowledge out
AP2	BBS7 L	Bank 7 select--The bus master asserts this signal to reference the I/O page.
AR2	BDMGI L	Direct memory access grant--in
AS2	BDMGO L	Direct memory access grant--out
AT2	BINIT L	Initialize
AU2	BDAL0 L	Data/address line 00
AV2	BDAL1 L	Data/address line 01
BA2	+5	+5 Vdc power
BB2	-12	-12 Vdc power (optional)
BC2	GND	Power supply return
BD2	+12	+12 Vdc power

BE2	BDAL2 L	Data/address line 02
BF2	BDAL3 L	Data/address line 03
BH2	BDAL4 L	Data/address line 04
BJ2	BDAL5 L	Data/address line 05
BK2	BDAL6 L	Data/address line 06
BL2	BDAL7 L	Data/address line 07
BM2	BDAL8 L	Data/address line 08
BN2	BDAL9 L	Data/address line 09
BP2	BDAL10 L	Data/address line 10
BR2	BDAL11 L	Data/address line 11
BS2	BDAL12 L	Data/address line 12
BT2	BDAL13 L	Data/address line 13
BU2	BDAL14 L	Data/address line 14
BV2	BDAL15 L	Data/address line 15

3.3. DRIVE INTERFACE PIN ASSIGNMENTS

Pin#	Name	Description
2	Not used	
4	Not used	
6	DRVSEL4	A logic zero on this line selects the drive which has this unique drive select circuitry enabled.
8	INDEX	Pulsed by the drive once per revolution indicating the physical beginning of a track.
10	DRVSEL1	A logic zero on this line selects the drive which has this unique drive select circuitry enabled.
12	DRVSEL2	A logic zero on this line selects the drive which has this unique drive select circuitry enabled.
14	DRVSEL3	A logic zero on this line selects the drive which has this unique drive select circuitry enabled.
16	MOTOR ON	When activated this line turns on the motor alloving reading on writing on the drive.
18	DIRECTION	This signal defines the direction of motion the Read/Write heads will take when the step line is pulsed. A logic one is out and a logic zero is in.

20	STEP	Activation of this signal causes the read/write heads to move with the direction of motion as defined by the direction line.
22	WRITE DATA	This interface line provides the data to be written on the diskette
24	WRITE GATE	Activating this signal, logic zero, enables write data to be written on the diskette.
26	TRACK 0	The active state, logic 0, of this interface signal indicates that the read/write heads are positioned at track 0 (the outermost track), and the stepper is locked on a track.
28	WRITE PROTECT	This signal indicates when a write protected diskette is installed in the drive.
30	READ DATA	This interface signal provides the "Raw Data" (clock and data together) as detected by the drive electronics.
32	SIDE SELECT	This signal defines which side of a two sided diskette is to be written to or read from. Logic one selects side 0 and logic zero selects side 1.
34	DRIVE STATUS	This signal is asserted by the drive when a diskette is inserted the motor is on and at least two index pulses have occurred. A logic 0 indicates the drive is ready.

Section 4

Functional Description

4. OVERVIEW

This section describes communication between the host system and the MXV50 controller. Communication is accomplished using command and response message packets located in host memory.

Information about the message packets (e.g., location in memory, host/controller ownership, etc.) is contained in buffer descriptors which are organized into descriptor rings. A detailed description of the following is provided:

1. Register usage
2. Initialization

4.1. REGISTERS

There are two device registers, the IP (Initialization and Polling) and the SA (Status, Address and Purge). The IP register is always located on a double word boundary and the SA address at IP + 2. Only word transfers to/from the IP and SA are legal.

4.1.1. IP Register

The IP register performs two functions:

1. Any value written in the IP causes the initialization procedure to start.

2. During normal operation (an initialization is not being performed) reading the IP causes controller polling.

4.1.2. SA Register

The SA register performs four functions:

1. When the SA is read during initialization, data and error information is received.

2. When the SA is written during initialization, host specific parameters are communicated to the controller.

3. During normal operation reading the SA provides the host with status information concerning controller detected fatal errors.

4. For systems using purge interrupts, the host zeroes the SA to indicate the successful completion of a bus adaptor purge. This can occur during either initialization or normal operation.

4.1.3. Controller Detected Fatal Errors

If the controller detects a fatal error, is sets bit 15 of the SA register and places the error code in bits 0-10 of the SA register. The error codes are:

Code	Meaning
001	Envelope/Packet Read parity or timeout.
002	Envelope/Packet Write parity or timeout.
003	Controller ROM and RAM parity.
004	Controller RAM parity.
005	Controller ROM parity.
006	Ring Read parity or timeout.
007	Ring Write parity or timeout.
008	Interrupt Master.
009	Host Access timeout (higher-level protocol-dependent).
010	Credit Limit Exceeded.
011	Q-BUS Master Error
012	Diagnostic Controller Fatal Error.
013	Instruction Loop timeout.
014	Invalid Connection Identifier.
015	Interrupt Write
016	Maintenance Read/Write Invalid Region Identifier.
017	Maintenance Write Load to non-loadable controller.
018	Controller RAM error (non-parity)
019	INIT sequence error
020	High-level protocol incompatibility error
021	Purge/poll hardware failure

4.2. INITIALIZATION

4.2.1. General

The purpose of initialization is to identify the parameters of the communications area to the controller, provide a confidence check of controller functionality and bring the controller online to the host.

During initialization, the contents of the SA register depend on the initialization step and whether the SA is being read or written.

When being read, bits 11-15 of the SA have the same meaning for all steps:

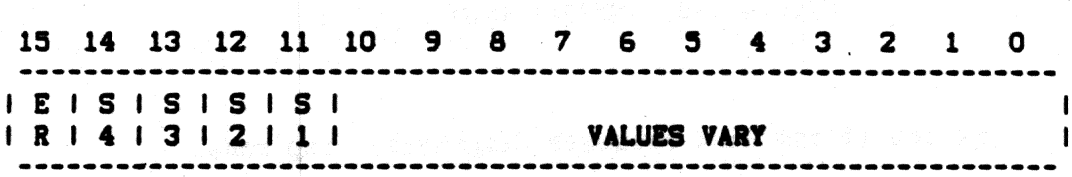


Figure 4-1: MXVER

Bits S1-S4 are set by the controller to indicate which step it is ready to perform. If more than one of these bits is set at any time, initialization should be restarted.

If the ER bit is set, either the controller has failed an internal diagnostic test or a fatal error has occurred.

The host begins initialization by issuing a bus INIT or by writing to the IP register.

4.2.2. Step 1

The host knows that Step 1 has begun when S1 is set. At that time the following pattern may be read from the SA.

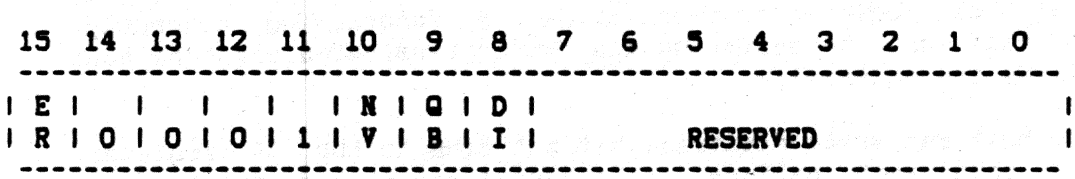


Figure 4-2: MXVSA1 (RO)

where:

NV — is cleared to indicate that the controller supports a host settable interrupt vector address.

QB — is set to indicate the controller is a Qbus device that supports 22-bit addressing.

DI — is set to indicate that the controller implements enhanced diagnostics, such as wraparound, purge and poll tests.

The host responds by writing the following into the SA,

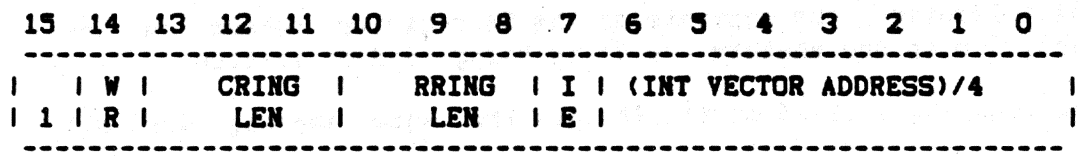


Figure 4-3: MXVSA1 (WO)

where:

WR — is set if the host requests diagnostic wrap mode.

CRING LEN — is the log (base 2) of the number of 32-bit slots in the command ring (e.g., 5 means 32 slots)

RRING LEN — is the log (base 2) of the number of 32-bit slots in the response ring

IE — is set if the host requests an interrupt at the completion of each of steps 1-3.

INT VECTOR — determines if interrupts are generated by the controller. If this is non-zero, interrupts can be generated during normal operation. A non-zero value is the interrupt vector address divided by 4.

If WR is not set by the host, the controller runs its internal diagnostics. Step 1 must complete within 10 seconds after the host writes the Step 1 data to the SA register. If WR is set, the controller enters diagnostic wrap mode (DWM). DWM allows the host to confirm that basic register communication is functioning correctly. Steps 2-4 will not be performed and DWM is terminated when the host re-initializes the controller.

In DWM, the host can write arbitrary bit patterns to the SA register, which will echo each pattern. If the SA register does not echo a pattern within 10 seconds, the controller has failed.

4.2.3. Step 2

When S2 is set, the host knows that Step 2 has started. At that time the following pattern may be read from the SA:

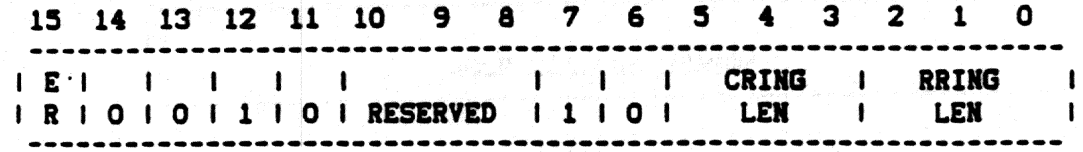


Figure 4-4: MXVSA2 (RO)

Bits 0-5 are the same values written to the SA during Step 1.

The host responds by writing into the SA the following:

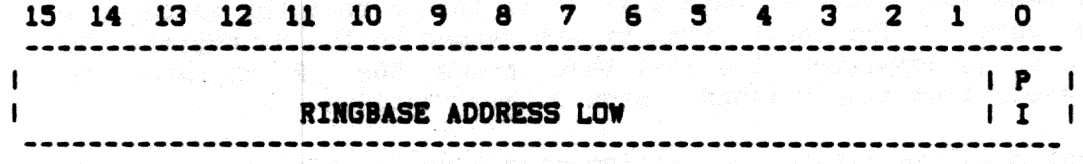


Figure 4-5: MXVSA2 (WO)

where:

RINGBASE ADDRESS LOW — is the low order address or word (ringbase + 0) of the communications area. This address is assumed to be on a word boundary (bit 0 = 0).

PI — is set if the host is requesting adapter purge interrupts.

Step 2 must complete within 10 seconds from the time the host writes the Step 2 data in the SA register.

4.2.4. Step 3

The host knows that Step 3 has begun when S3 is set. At that time, the following may be read from the SA:

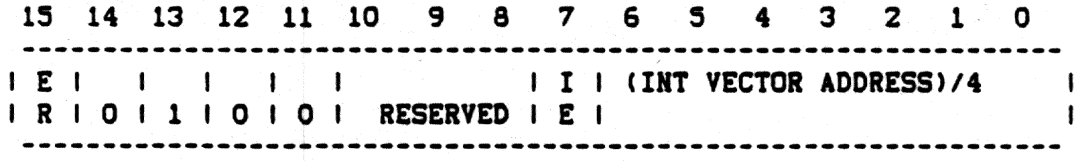


Figure 4-6: MXVSA3 (RO)

Bits 0-7 are the same values written in the SA during Step 1.

The host responds by writing into the SA the following,

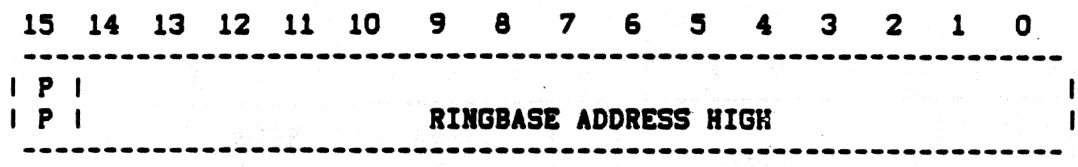


Figure 4-7: MXVSA3 (WO)

where:

PP is set if the host requests execution of purge and poll tests.

Ringbase Address High is the high order address of word [ringbase + 0].

If PP has been set, the host must wait until the SA register contains a value of zero. The host then writes zeroes to the SA register to simulate a purge complete. The host then reads the IP register to simulate a start polling command. Step 3 then continues.

Step 3 must complete within 100 milliseconds from the time the Step 3 data are written to the SA register if PP is not set and within 10 seconds if PP is set.

4.2.5. Step 4

The host knows that Step 4 has begun when S4 is set. At that time the following may be read from the SA:

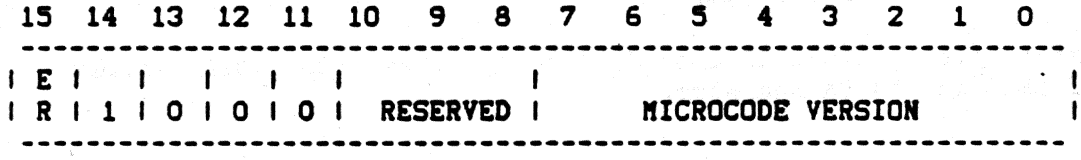


Figure 4-8: MXVSA4 (RO)

Bits 0-7 contain the version number of the controller microcode.

The host responds by writing into the SA the following,

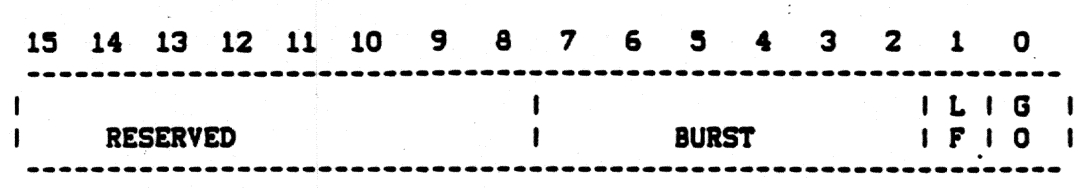


Figure 4-9: MXVSA4 (WO)

where:

BURST is one less than the maximum number of double words the host allows per DMA transfer. If this field is zero, then the controller is to use its default value.

LF is set when the host wants a last fail response packet when initialization is complete

GO is set when the controller should enter its functional microcode as soon as initialization is complete. If this field is not set, the controller reads the SA until it is set.

Section 5

Operational Description

5. GENERAL

This section provides pertinent information concerning controller operations. The features covered include on board bootstrap; utility features--formatting and diagnostics; Q-Bus communication logic; and diagnostic/power failure protection logic.

5.1. BOOTSTRAP FUNCTION (LSI-11 PROCESSORS ONLY)

If the bootstrap is enabled the controller will respond to the standard bootstrap address 17773000(8). The controller is bootstrapped by typing 17773000G while in console ODT. This causes a bus INIT and transfers program execution to location 17773000(8). An alternate method is to strap the LSI-11 processor to power up Mode 2. In this mode, when a power up occurs, the processor automatically starts execution at 17773000(8). Power-up strapping procedures for the LSI-11 processor can be found in the Microcomputer Processors Handbook.

To bootstrap a diskette, use the following procedure:

1. Place the diskette in drive 0.

2. If the processor is strapped for power-up Mode 2, operate the INIT (boot) switch or cycle DC power OFF and ON.

3. If the processor is not strapped for power up Mode 2 while in console ODT, type 17773000G.

5.1.1. Bootstrap Operation

The bootstrap is not a standard ROM program. It uses the controller's microprocessor to capture the bus; to read block 0 of the diskette into memory starting at location 0; and finally to transfer program execution to memory location 0.

Any attempt to read location 17773000(8) will result in a non-existent memory trap. The controller only responds to this address immediately after a bus INIT. For this reason the bootstrap is called

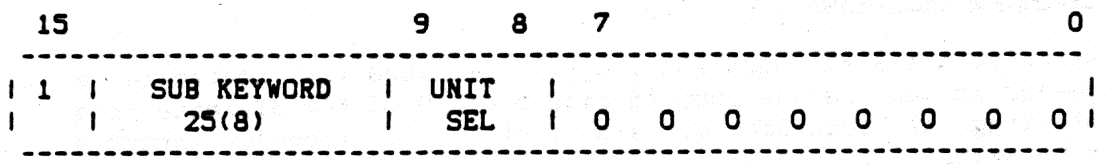
"transparent". When the processor attempts to fetch location 17773000(8) following a bus INIT, the controller responds by passing the processor a "CLEAR R0" instruction. The processor clears R0 and then attempts to fetch location 17773002(8). The controller passes the processor a "LOAD IMMEDIATE" instruction with R1 as the destination. The processor then attempts to fetch the source operand from location 17773004(8). The controller passes the zeroes, which results in R1 being cleared. The processor moves the zeroes into R1 and then attempts to fetch location 17773006(8). The controller first asserts a Direct memory Access Request (DMR) then passes the processor a "CLEAR PC" instruction. Before the processor executes the instruction it passes bus mastership to the controller. The controller moves a "BRANCH TO CURRENT LOCATION" instruction (777(8)) into memory location 0 under DMA control. When the controller releases bus mastership the processor executes the "CLEAR PC" instruction and, in so doing, transfers program execution to location 0. The processor is thus forced to loop at location 0. The controller reads block 0 of drive 0 into locations 2 through 776. Finally the controller DMA's location 0 with a NOP instruction (240(8)) allowing the processor to execute the system bootstrap. If there is no diskette in drive 0 nothing will be transferred to memory and the processor will continue to loop at location 0 until halted.

5.2. UTILITY FEATURES

The MXV50 controller has two built-in utilities which enhance its capabilities. A formatting utility allows the controller to format blank diskettes to the RX50 media standard. This feature is supported entirely by the controller requiring no additional software support and is invoked by a simple command protocol. An internal set of diagnostics are available to diagnose proper controller operation. These diagnostics operate once during each application of DC power to the board or can be activated by option jumper.

5.2.1. Formatting

To invoke the formatting utility the MXVIP register is written to start initialization. The MXVSA will initially be cleared then will respond with the Step 1 indicator set (5500[Oct] 0B40[Hex]). The response to the Step 1 indicator is the formatting keyword 152000[Oct] (D400[Hex]). When the controller sets the Step 2 indicator 10000[Oct] 1000[Hex]), following must be input:



The following summarizes the formatting responses.

DESCRIPTION	MXVA RESPONSE
FORMAT DRIVE 0	152000[Oct] (D400[Hex])
FORMAT DRIVE 1	152400[Oct] (D500[Hex])
FORMAT DRIVE 2	153000[Oct] (D600[Hex])
FORMAT DRIVE 3	153400[Oct] (D700[Hex])

When the appropriate MXVSA response is input, the selected drive will be formatted. The MXVSA displays all zeroes while the formatting is taking place. When the formatting is complete, the Step 2 indicator is again set and another diskette can be formatted. Refer to Section 2.4 (Initial Operation and Checkout) for examples of formatting commands.

Formatting can be aborted by writing to the MXVIP register.

5.2.1.1. Format Operation

The MXV50 controller has the capability of formatting diskettes in an RX50 compatible format. Formatting is accomplished in two passes. Initiating the format utility homes the drive being formatted to track 0. During pass 1 starting at the physical index hole ten sector headers appropriately spaced, are written on the track followed by a "4E" pattern until the index hole is sensed. Each header is written with a unique address containing three address marks, an ID address mark track, side, sector and sector length information followed by two bytes of CRC data. If double sided operation is selected the second side of the cylinder will similarly be written except the side indicator will be set to side one. Each of the remaining 79 tracks are written in the same manner. When track 79 is complete pass 2 is initiated. The controller seeks track 0 locates sector 0 and writes a data field of three address marks, a data address mark, 512 bytes of "E5" data followed by two bytes of CRC. Each successive sector is located and the corresponding data field written. When double sided operation is selected as each track of side zero is completed each correspondingly sector will be located and its associated data field written. If an error occurs during format the operation will be aborted. Successful completion will be indicated by the Step 2 indicator being set in the SA register.

5.2.2. Diagnostics

The controller implements firmware diagnostics used to verify proper controller operation. These diagnostics include:

Instruction branch tests
Condition code verification
Register tests
ALU operation tests
RAM tests
Instruction tests

When the diagnostics enable jumper is removed the internal diagnostics only operate once during the power up cycle with boot disabled. When the diagnostic enable jumper is installed, after initialization the controller continuously operates the internal diagnostics blinking the LED once for each successful pass. If an error occurs the diagnostics will loop on the failing test until successfully completed.

5.3. SCATTER GATHER

The MicroVax I processor does not implement hardware mapping registers and requires any controller using DMA to implement a scatter gather algorithm. This algorithm is a mechanism for the controller to map virtual addresses into physical addresses for local memory.

The MXV50 implements the scatter gather algorithm as follows. When required the Host passes a mode bit indicating the requirement for scatter gather along with both the virtual address and base address of the location of the software registers. The controller adds bits nine through twenty-one of the virtual address to the register base address to form the physical address of the selected register. Subsequently the lower thirteen bits of the selected mapping register are combined with the lower nine bits of the virtual address to form the twenty-two bit physical address in local memory. Since the lower nine bits are carried forward from the virtual address it is only necessary to recalculate the physical address at the beginning of a transfer and whenever the lower nine bits overflow. Refer to figure 5-1.

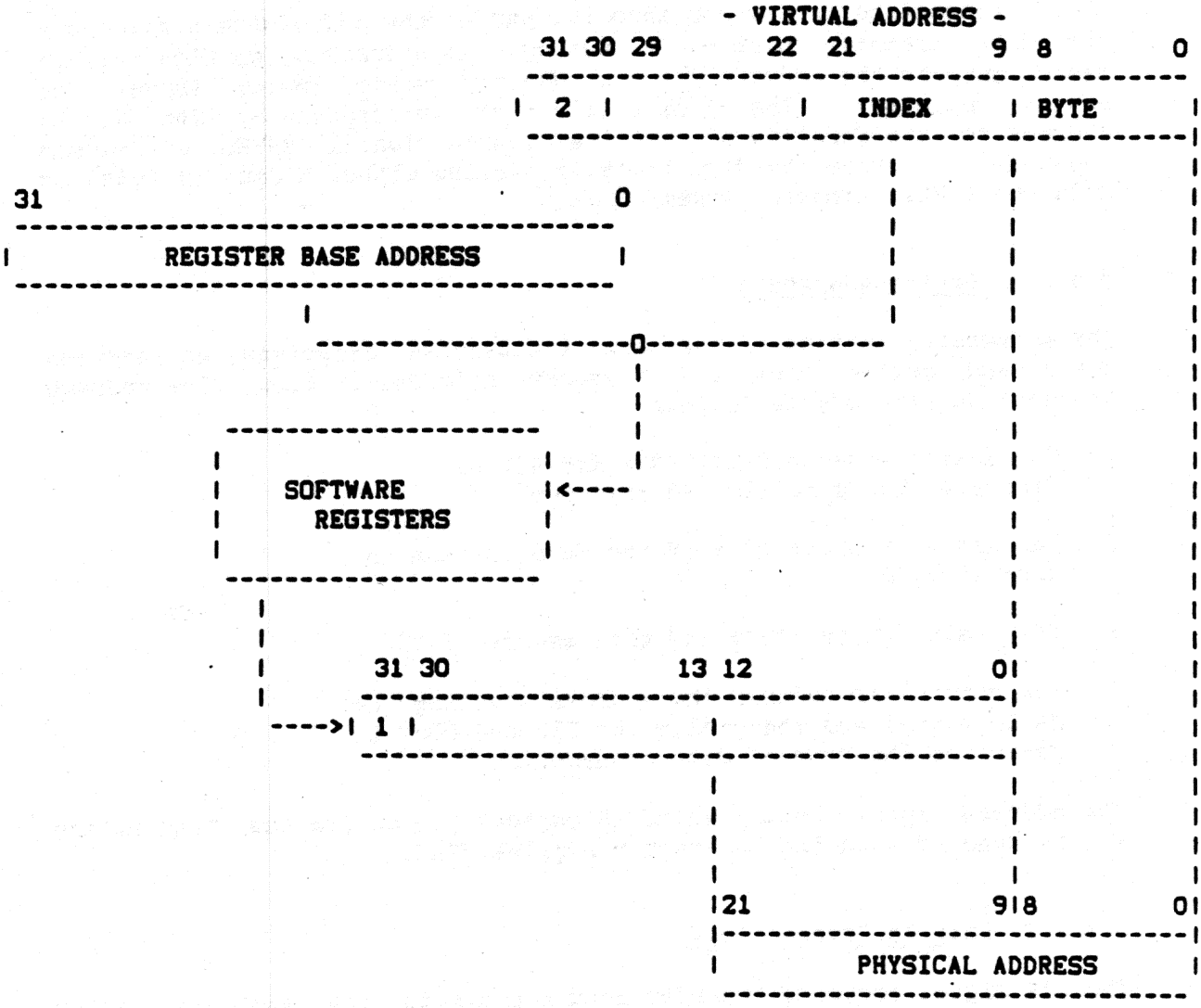


Figure 5-1 Scatter Gather Mapping Algorithm

5.4. Q-BUS COMMUNICATION LOGIC

The MXV50 interfaces to the Q-Bus plugging into any Q22 slot. Since the controller has interrupt capabilities and supports DMA data transfer, it is necessary to insure grant continuity to the slot in which the MXV50 is located. The controller supports standard data transfer, interrupt and DMA protocol. This section describes in detail these bus transactions as related to the controller.

5.4.1. Data Transfer

Data transfer takes place with one device as bus master transferring data to or from the other device as a slave. The type of transfer is indicated by a DIN or DOUT operation. The direction of data transfer is always specified with reference to the master device.

A bus transaction is divided into two parts; the addressing portion and the data transfer portion. During the addressing portion the bus master outputs the address for the desired slave device (memory or device register). The selected slave device responds by latching the address for the duration of the transaction (until BSYNC L becomes negated). During the data transfer portion either a data in (DIN) or data out (DOUT) transfer takes place.

5.4.1.1. Device Addressing

The addressing portion of the bus transaction comprises an address setup and deskev time and an address hold/deskev time. The address transaction proceeds as follows:

1. The master asserts DAL<17:00> and BS7 to address the specific I/O register

2. The master asserts WTBT if the data portion is a data cycle.

3. The master waits 150ns and then asserts SYNC.

4. The controller decodes the address sampling the MATCH signal and then polls the DIN and DOUT to determine the type of data transaction.

The address signal remains valid throughout the entire bus transaction and is negated when the bus master negates SYNC.

5.4.1.2. Data In Transactions

The DIN signal requests transfer from a slave to the master. After sensing a match followed by a DIN the controller determines which register was accessed by examining the LA01 signal, places the corresponding emulated register contents in the bus address register, enables the data to the QBUS with BUSENB and asserts REPLY. The controller then loops waiting for the bus master to negate DIN. When negated BUSENB and REPLY are cleared. The bus master responds by negating SYNC, ending the bus transaction.

5.4.1.3. Data Out Transaction

The DOUT signal requests transfer from the master to the slave device. After sensing a match followed by a DOUT the controller receives the data from the RDA lines saving it in the emulated register as defined by the LA01 signal. The controller asserts REPLY to indicate that the data were accepted. The controller loops waiting for SYNC to be negated. When negated, REPLY is cleared. The bus master responds negating SYNC, ending the bus transaction.

5.4.2. Interrupt Processing

The supervisor requests interrupt service from the host by asserting the interrupt request signal (IRQ). Priority is established at the next RDIN bus cycle. If the controller has priority the Interrupt Grant line is asserted blocking the interrupt Acknowledge from being passed down the bus. Upon receipt of the acknowledge signal (IAKI) the supervisor strobes the vector into the bus address register and asserts BUSENB followed by REPLY. When the host removes IAKI the supervisor negates IRQ, REPLY and finally BUSENB.

5.4.3. DMA Protocol

Direct memory access is initiated by the supervisor by assertion of the DMR signal. When the request is serviced by the host the DMGRANT line is asserted in response to the host DMGI signal; blocking DMGI from being passed down the bus. Upon receipt of DMGRANT the supervisor waits until Reply and RSYNC are both negated and then asserts TSACK acknowledging the GRANT. At this point the supervisor becomes bus master and can issue DATI and DATO bus cycles. At the completion of all bus cycles the supervisor negates DMR and then TSACK to release bus mastership.

5.5. DIAGNOSTIC/POWER FAIL PROTECTION LOGIC

The MXV50 incorporates a basic set of diagnostics to insure proper controller operation. These tests are initiated with application of power and upon successful completion the indicator located at the rear edge of the controller is illuminated. This indicator is also a function of the on-card power protection logic. If the DC voltage applied to the controller meets or exceeds 4.6 volts the indicator is enabled. After reaching this limit the voltage must drop below 4.2 volts before the protection logic will activate and turn off the indicator. When this logic is activated all writes to the drive interface are blocked and a write protected status is sensed by the supervisor. The host is subsequently alerted of this status in the appropriate end message.

The POK and DCOK signals on the bus are also sensed. If the POK signal is negated all write operations are aborted because of the impending power failure. If DCOK is negated the power protect logic is activated, blocking writes to the drive. However under this condition the indicator remains illuminated and the write protect status is not asserted to the supervisor.

Section 6

Software Considerations

6. GENERAL

The MXV50 is a dual height RX50 compatible floppy only controller incorporating features not available with other products. These features include a formatting utility and double sided drive support. This section revievs both diagnostic considerations for the controller as well as software considerations relating to the incorporated features.

6.1. DIAGNOSTIC SUPPORT

The MXV50 is compatible with both DEC's Performance Exerciser and Formatting Utility. However, several versions are available and exhibit different responses. The diagnostics supported include:

ZRQABO - Performance Exercisier
ZRQAEO

ZRQBAO - Formatting Utility
ZRQBC1

All of these diagnostics are available from DEC in XXDP+ format and operate similarly to other existing DEC diagnostics.

ZRQA has been designed to test the specific features of the RQDX series of controllers. Even though it is not organized to exercise the specific features of the MXV50 it can be used to verify proper controller operation. For a more comprehensive test of the controller the MTI MXV50D diagnostic should be used. This diagnostic is available from MTI.

ZRQABO will function with the controller as supplied from DEC. ZRQAE has a few bugs with its trap processing routines relative to a non-existent line clock address. Before use of this revision of the Exerciser the following patches should be applied.

Loc.	Old Value	New Value
31304	5037	240
31306	177546	240
31404	5037	240
31406	177546	240
42620	12737	240
42622	100	240
42624	177546	240

Figure 7-1 ZRQAE0 Patches

The ZRQB formatting utility is compatible with the MXV50 but because DEC does not support formatting on the RX50 media operating the diagnostic will cause the diagnostic to abort the first pass. The following message will be printed

```
$FTLERR INVALID CONNNECTION IDENTIFIER
PASS ABORTED THIS UNIT
ZRQB EOP 1
0 TOTAL ERRORS
```

6.2. DOUBLE SIDE SUPPORT

The MXV50 controller can easily be interfaced to double sided drives. When the double sided option is enabled and during either ONLINE or SET UNIT CHARACTERISTICS commands, the controller interrogates the second side of the media for a valid format and if found identifies the media as double sided. The end message packets are adjusted accordingly. All other media are recognized as single sided. Double sided media is created by formatting with the appropriate double sided jumper enabled.

Caution must be exercised when using this feature. Since other products may not support this feature, incompatibilities may arise. Particularly since software typically initializes a directory with embedded volume size, if double sided media are exchanged to a controller drive combination which does not support double sided a conflict will arise. The alternate controller will establish one volume size while the initialized directory differs resulting in various error messages. Also loss of data could occur if the alternate controller were forced to use the media, not being aware of the capacity difference.

However, the controller has been tested in the double sided configuration with RT11, RSX11M, RSX11M+, RSTS and VMS without any compatibility problems.

6.3. FORMATTING UTILITY

The MXV50 implements an internal formatting utility which can be envoked by forcing the controller into a special mode during initialization as described in Section 5. Simplified console ODT commands can be used to envoke this feature. At this time there are no online formatting utilities available.