

**MXV42C**  
Disk Controller  
Manual

  
Micro Technology, Inc.

**MXV42C**  
Disk Controller  
Manual

MARCH 1982 PRELIMINARY  
MICRO TECHNOLOGY, INCORPORATED

## CONTENTS

### SECTION 1 - GENERAL INFORMATION

1.1	INTRODUCTION.....	1-1
1.2	COMPONENTS.....	1-2
1.3	COMPATIBILITY.....	1-2
1.3.1.1	Logical Track Format (5¼" Mini Floppy).....	1-4
1.3.1.2	Logical Track Format (8" Floppy).....	1-6
1.3.2	Sector Header Field.....	1-8
1.3.3	Data Field.....	1-9
1.3.4	Recording Scheme.....	1-10
1.3.5	Double Frequency (FM).....	1-10
1.3.6	DEC Modified MFM.....	1-10
1.3.7	Cyclic Redundancy Check.....	1-11
1.4	SPECIFICATION.....	1-12

### SECTION 2 - INSTALLATION

2.1	CONFIGURATION.....	2-1
2.1.1	Address/Vector Selection.....	2-1
2.1.2	Device Interrupt Priority.....	2-4
2.1.3	Bootstrap.....	2-4
2.1.4	Write Precompensation.....	2-5
2.1.5	Step Rate Control.....	2-6
2.2	DRIVE CONFIGURATION.....	2-6
2.2.1	Mini-Floppy Configuration.....	2-6
2.2.2	8" Floppy Configuration.....	2-7
2.3	CABLING.....	

## FIGURES

Figure 1-1:	5¼" Mini-Floppy Logical Track Format.....	1-5
Figure 1-2:	8" Floppy Logical Track Format.....	1-7
Figure 1-3:	FM Recording Characteristics.....	1-10
Figure 1-4:	MFM Recording Characteristics.....	1-10
Figure 2-1:	Configuration Jumper Location.....	2-2
Figure 2-2:	Printed Circuit Assembly.....	2-3
Figure 2-3:	Drive/Controller Cabling Method A.....	2-9
Figure 2-4:	Drive/Controller Cabling Method B.....	2-11
Figure 2-5:	Connector Pin Definition.....	2-12
Figure 4-1:	Format Subroutine.....	4-4
Figure 4-2:	Write Data Subroutines.....	4-5
Figure 4-3:	Read Data Subroutines.....	4-6

## TABLES

Table 1-1:	Address Marks.....	1-9
Table 1-2:	Standard MFM Encoding.....	1-11
Table 1-3:	Modifying Algorithm (Special Case).....	1-11
Table 2-1:	Address/Vector Selection.....	2-1
Table 2-2:	Priority Level Configuration.....	2-2
Table 2-3:	Bootstrap Option.....	2-5
Table 2-4:	Write Precompensation.....	2-5
Table 2-5:	Step Rate.....	2-6
Table 2-6:	Mini-Floppy Drive Configuration.....	2-6
Table 2-7:	8" Floppy Drive Configuration.....	2-7
Table 4-1:	5¼" Mini-Floppy Command Word Formats.....	4-3
Table 4-2:	8" Floppy Command Word Formats.....	4-3

### SECTION 3 - FUNCTIONAL DESCRIPTION

3.1	GENERAL.....	3-1
3.2	REGISTER DEFINITIONS.....	3-2
3.2.1	MXVCS - Command and Status Register (5¼"-177170 <sub>8</sub> ).....	3-2
3.2.2	MXVDB - Data Buffer (5¼"-177172 <sub>8</sub> ) (8"-177176 <sub>8</sub> or 177152).....	3-3
3.2.3	Extended Status Registers.....	3-7
3.3	COMMAND PROTOCOL.....	3-8
3.3.1	Fill Buffer (000).....	3-8
3.3.2	Empty Buffer.....	3-9
3.3.3	Write Sector (010).....	3-10
3.3.4	Read Sector (011).....	3-11
3.3.5	Set Media Density (100).....	3-12
3.3.6	Read Status (101).....	3-13
3.3.7	Write Deleted Data Sector (110).....	3-13
3.3.8	Read Error Code (111).....	3-13

### SECTION 4 - CONTROLLER OPERATIONS

4.1	GENERAL.....	4-1
4.2	BOOTSTRAPING THE CONTROLLER.....	4-1
4.2.1	Bootstrapping Operation.....	4-2
4.3	FORMAT OPERATIONS.....	4-3
4.3.1	5¼" Mini-Floppy.....	4-3
4.3.2	8" Floppy.....	4-3
4.4	FILL/WRITE OPERATIONS.....	4-5
4.5	READ/EMPTY OPERATIONS.....	4-6
4.6	WRITE CURRENT CONTROL.....	4-7
4.7	WRITE PRECOMPENSATION.....	4-7
4.8	POWER FAIL PROTECTION.....	4-9
4.9	OPERATION USING RT11.....	4-9
4.9.1	Creating a DY-Compatible System Disk.....	4-9
4.9.2	Creating a Mini-Floppy RT11 System Diskette.....	4-10
4.9.3	Dikette Copy Program.....	4-12

## 1.1 INTRODUCTION

The MXV42C emulates the operation of two DEC\* RX02 sub-systems. The controller contains two sets of register. One set controls 5¼" mini-floppy drives and the other controls 8" floppy drives. When configured with any Shugart compatible drive, each set acts as a direct replacement for the RXV21 controller. Either single or double density recording is available on each drive, providing up to 1M bytes of storage per 8" diskette and 512K bytes per 5¼" diskette. Single density 8" diskette recording is compatible with IBM 3740 equipment.

All electronics are contained on one dual wide board which plugs directly into any standard LSI-11 backplane and interfaces through a 50 pin ribbon cable to two Shugart compatible 8" and two Shugart compatible 5¼" drives. (See Section ). All controllers are 100% tested and ready for operation. The 5¼" controller is configured with a device address of 177170<sub>8</sub> and interrupt vector of 264<sub>8</sub>. The 8" floppy controller is configured with a device address of 177174<sub>8</sub> or 177150<sub>8</sub> and interrupt vector of 270<sub>8</sub>. The interrupt level is factory set to level four. Features include:

- Transparent firmware bootstrap automatically loads either single or double density 5¼" mini-floppy diskettes.
- Formatting capability permits writing sector headers, checking the written headers, and writing the data fields in the user selected density.
- Jumper selectable four-level drive interrupt priority compatible with the LS-11/23.
- Provides power fail protection for data integrity.
- Write current control signal for tracks greater than **forty-three**.
- Write precompensation for reduced error rates.

\*DEC, PDP-11, LSI-11, Unibus and Q-bus are registered trademarks of Digital Equipment Corporation.

## 1.2 COMPONENTS

The controller is provided with the following components:

Floppy disc controller  
Manual

## 1.3 COMPATIBILITY

This section discusses the aspects of hardware, software and media compatibility with Digital Equipment's RX02 system. The information will aid the user in data interchange with foreign systems.

### Hardware

The controller is compatible with the LSI-11, LSI-11/2 and LSI-11/23 processors. All circuitry is contained on one dual-wide board that plugs directly into any standard LSI-11 backplane. A four-level device interrupt priority scheme provides the user added flexibility.

### Software

The MXV42C is completely compatible with RXV21 register definition and command protocol. All DEC-supplied software designed to operate with the RX02 system will operate with the controller without modification.

### Drives

#### 5 $\frac{1}{4}$ " Mini-Floppy

A double sided 96 TPI Shugart compatible 5 $\frac{1}{4}$ " mini-floppy drive must be used to attain 512K bytes of storage capacity. Two drives which are available are the Shugart SA460 and the Tandon T100-4.

#### 8" Floppy

Any Shugart compatible 8" floppy drive may be used.

### Media

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

#### 5 $\frac{1}{4}$ " Mini-Floppy

VERBATIM (Data Life).      5 $\frac{1}{4}$ " 96 TPI double density, double sided

#### 8" Floppy

IBM                              Single or double density

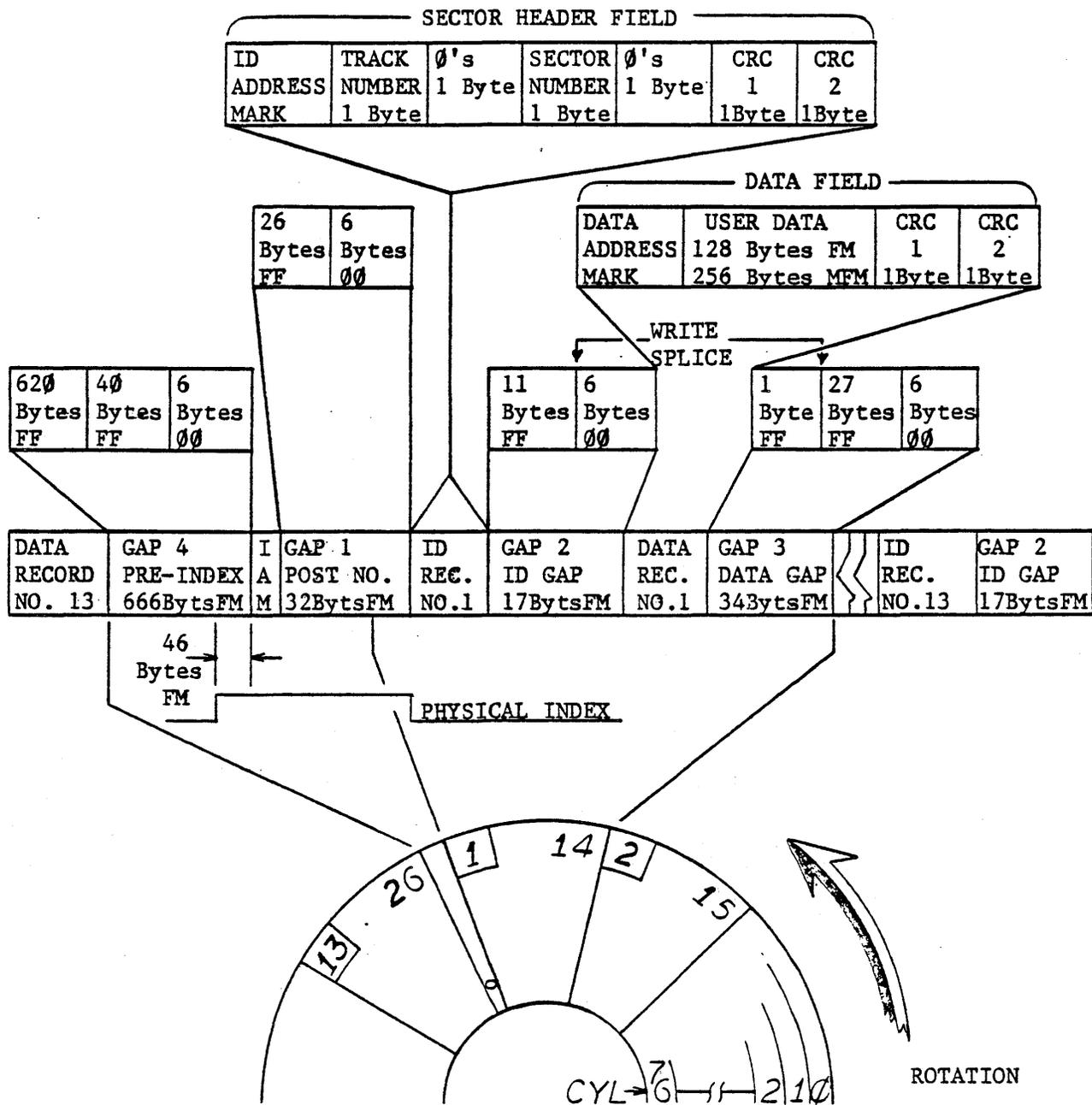
DEC                                RX01/RX02

This sheet purposely left blank.

#### 1.3.1.1 Logical Track Format (5¼" Mini Floppy)

The diskette surfaces are divided into 77 concentric cylinders numbered 0-76. Each cylinder consists of two tracks. The track on side 0 contains sectors 1-13 while the track on side 1 contains sectors 14-26. The track begins and ends at the index address mark. The track is formatted in such a way that this "soft" index is preceded by the leading edge of the physical index hole in the diskette. Following the physical index are 40 bytes of "FF" data, 6 bytes of "0" data, and the index address mark indicating the beginning of the track. Following the index address mark is the post index gap consisting of 26 bytes of "FF" data and 6 bytes of "0" data. The next field is the sector header for the first sector. On side 0 this is 1. On side 1 this sector is sector 14. Following the sector header is the ID gap consisting of 11 bytes of "FF" data and 6 bytes of "0" data. The next field is the data record for the sector. Following the data field is the data gap consisting of 28 bytes of "FF" data and 6 bytes of "0" data. This field leads to the next sector header. Following the 13th data record on side 0 and the 26th data record on side 1 is the pre-index gap consisting of approximately 620 bytes of "FF" 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 the track number. All the above fields are recorded in FM except as noted in the following sections.



NOTE 1: NUMBERS IN BLOCKS REFER TO SIDE Ø SECTOR NUMBERS.

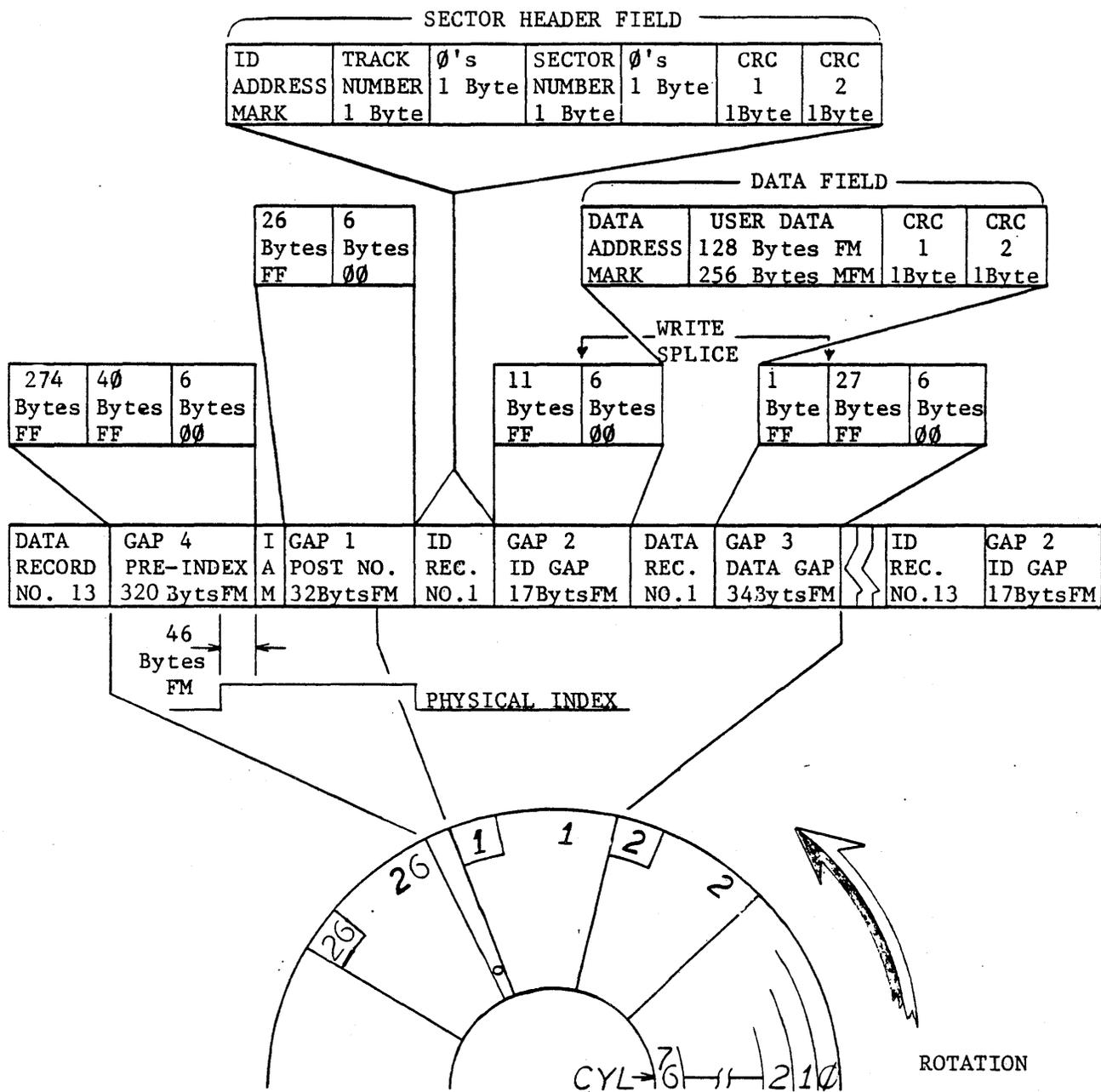
NOTE 2: SIDE Ø IS THE UNDERSIDE OF DISK.

Figure 1-1: 5¼" Mini-Floppy Logical Track Format

### 1.3.1.2 Logical Track Format(8" Floppy)

The diskette surface is divided into 77 concentric tracks numbered 0-76. Each track consists of 26 sectors numbered 1-26. A track begins and ends at the index address mark. Each track is formatted in such a way that this "soft" index is preceded by the leading edge of the physical index hole in the diskette. Following the physical index are 40 bytes of "FF" data, 6 bytes of "0" data, and the index address mark indicating the beginning of the track. Following the index address mark is the post index gap consisting of 26 bytes of "FF" data and 6 bytes of "0" data. The next field is the sector header for sector 1. Following the sector header is the ID gap consisting of 11 bytes of "FF" data and 6 bytes of "0" data. The next field is the data record for sector 1. Following the data field is the data gap consisting of 28 bytes of "FF" data and 6 bytes of "0" data. This field leads to the next sector header. Following the 26th data record is the pre-index gap consisting of approximately 274 bytes of "FF" data.

Each track is formatted in the above manner. Refer to Figure 1-2. The sector header field of each sector contains information describing both the sector and track number. All the above fields are recorded in FM except as noted in the following sections.



NOTE 1: NUMBERS IN BLOCKS REFER TO SIDE Ø SECTOR NUMBERS.

NOTE 2: SIDE Ø IS THE UNDERSIDE OF DISK.

Figure 1-2: 8" Floppy Logical Track Format

### 1.3.2 Sector Header Field

The header field consists of 7 bytes of information. Preceding the header is a field of bytes of "zero" data for synchronization. The header and this preamble are always recorded in FM.

1. Byte 1. ID Address Mark - A unique mark consisting of 1 byte of FE (hex) data with three missing clock-transitions using a C7 (hex) clock pattern. This mark is decoded by the controller and indicates the start of the sector header.
2. Byte 2. Track Address - This byte indicates the absolute  $(0-114_8)$  track address. Each sector contains this track information to locate its position on one of the 77 tracks.
3. Byte 3. "Zero"
4. Byte 4. Sector Address - This byte indicates the absolute  $(1-32_8)$  sector address. Each sector contains the information to identify its position on the track.
5. Byte 5. "Zero"
6. Byte 6, 7. CRC - This is the 16 bit cyclic redundancy character and is calculated for each header from the first 5 bytes of information using the IBM 3740 polynomial. (Refer to Cyclic Redundancy Check, Section 1.3.7)

### 1.3.3 Data Field

The data field consists of either 131<sub>10</sub> or 259<sub>10</sub> bytes of information depending upon the recording method. Preceding the data field is a field of 6 bytes of "zero" data for synchronization.

The preamble and data address mark are always written in FM. The user data and CRC character are either written in FM or MFM modified depending upon the formatted diskette density.

1. Byte 1. Data Address Mark - A unique mark consisting of a data byte (see Table 1-1) with three missing clock transitions using a C7 (hex) clock pattern. This byte is always written in FM and is decoded by the controller to indicate the start of the data field, its recording method (FM vs MFM) and if the field is a deleted data field.

ADDRESS MARK	INDICATED DENSITY	DATA	CLOCK
INDEX	NA	FC	D7
ID	NA	FE	C7
DATA	FM	FB	C7
	MFM Modified	FD	C7
DELETED	FM	F8	C7
	MFM Modified	F9	C7

Table 1-1: Address Marks

2. Bytes 2-129 (FM) or Bytes 2-257 (MFM Modified). User Data. This field is recorded in either FM or MFM modified. Depending upon the encoding scheme, either 128 or 256 bytes of information can be stored.
3. Bytes 130-131 or 258-259. CRC - This is the 16 bit cyclic redundancy character and is calculated for each data field from the first 129 or 257 bytes of information using the IBM 3740 polynomial. (Refer to Cyclic Redundancy Check, Section 1.3.7) These bytes are recorded with the same encoding scheme as the data field.

### 1.3.4 Recording Scheme

Two recording schemes are used: double frequency (FM) and DEC modified Miller code (MFM). FM is used for single density recording. DEC modified MFM is used for recording double density.

### 1.3.5 Double Frequency (FM)

FM recording is characterized by a flux transition at the beginning of each bit cell which is commonly termed a clock pulse or transition as shown in Figure 1-3. A logic "one" is represented by a flux transition within the bit cell; a logic "zero" is presented by the lack of any flux transition within a bit cell. In FM the bit cell time is 8us.

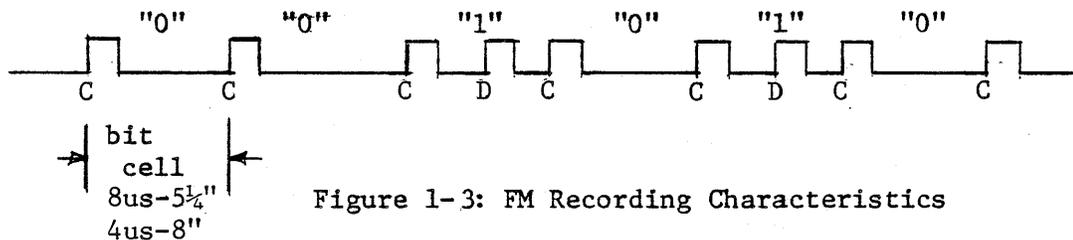


Figure 1-3: FM Recording Characteristics

### 1.3.6 DEC Modified MFM

MFM recording consists of flux transition for a logic "one" and no flux transitions for a logic "zero". A clock transition only occurs between two consecutive logic "zeroes" as shown in Figure 1-4 below. The MFM bit cell time is 4us.

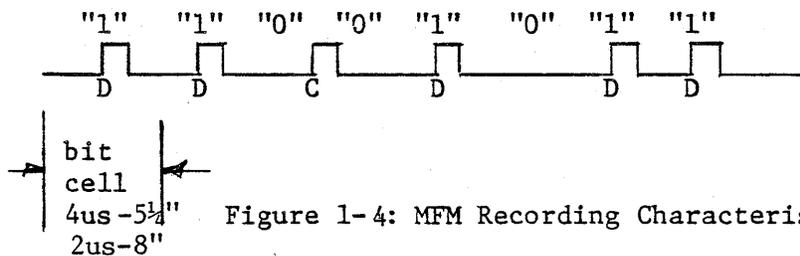


Figure 1-4: MFM Recording Characteristics

Table 1-2 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 1-2: Standard MFM Encoding

Because single density headers are used for both FM and MFM recording formats and since certain MFM patterns map into single density address marks, a modified algorithm is used. The mapping occurs when a data pattern of exactly four consecutive "ones" is encoded. Whenever this pattern is encoding a special algorithm is applied. Table 1-3 defines the encoding algorithm for this special case.

DATA											
DN-5		DN-4		DN-3		DN-2		DN-1		DN	
0		1		1		1		1		0	
X	0	1	0	0	0	1	0	0	0	1	0
CN-5	DN-5	CN-4	DN-4	CN-3	DN-3	CN-2	DN-2	CN-1	DN-1	CN	DN
ENCODED DATA											

Table 1-3 : Modifying Algorithm (Special Case)

When reading double density data fields the controller checks for a missing clock bit between two zero data cells and if found, substitutes one's for the two zero data bits (generated by the special encoding algorithm).

### 1.3.7 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 IBM 3740 is  $G(x) = X^{16} + X^{12} + X^5 + 1$ . For the sector header the data bits include byte 1 thru 5. For an FM data field the bits include byte 1 thru byte 129. For an MFM data field the bits include byte 1 thru 257.

1.4 SPECIFICATION

RECORDING TECHNIQUE:

5¼" Mini Floppy

Single Density

Modified IBM 3740 FM

Double Density

Modified DEC MFM

8" Floppy

Single Density

IBM 3740 FM

Double Density

DEC Modified MFM

POWER REQUIREMENTS:

Voltage

Single 5V supply  
(from LSI-11 backplane)

Current

2.5A Typical

ENVIRONMENTAL

Temperature

0 - 45°C

Humidity

10% - 95% non-condensing

## 2.1 CONFIGURATION

The controller is shipped with standard options configured. The device interrupt priority is set to level four. The firmware bootstrap is disabled. Write precompensation is enabled.

Most options are factory foil-etched to the most often used configuration. The foil jumpers must first be cut before the alternate jumpers are inserted. Refer to Tables 2-1 and 2-2 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.

### 2.1.1 Address/Vector Selection

The MXV42C is shipped with both the primary and alternate device address and vector options enabled. The primary address ( $177170_8$ ) is used as the mini-floppy ( $5\frac{1}{4}$ " ) controller address. The alternate address ( $177174_8$  or  $177150_8$ ) is used as the 8" floppy controller address. The vectors are respectively  $264_8$  and  $270_8$ . This is the only option available for the MXV42C and any changes to the factory jumpering may result in improper controller operation. These jumpers are summarized in Table 2-1.

OPTION	JUMPERS			
	15-16	16-17	7-8	8-9
Dual Address and Vectors	IN	OUT	IN	OUT

Table 2-1: Address/Vector Selection

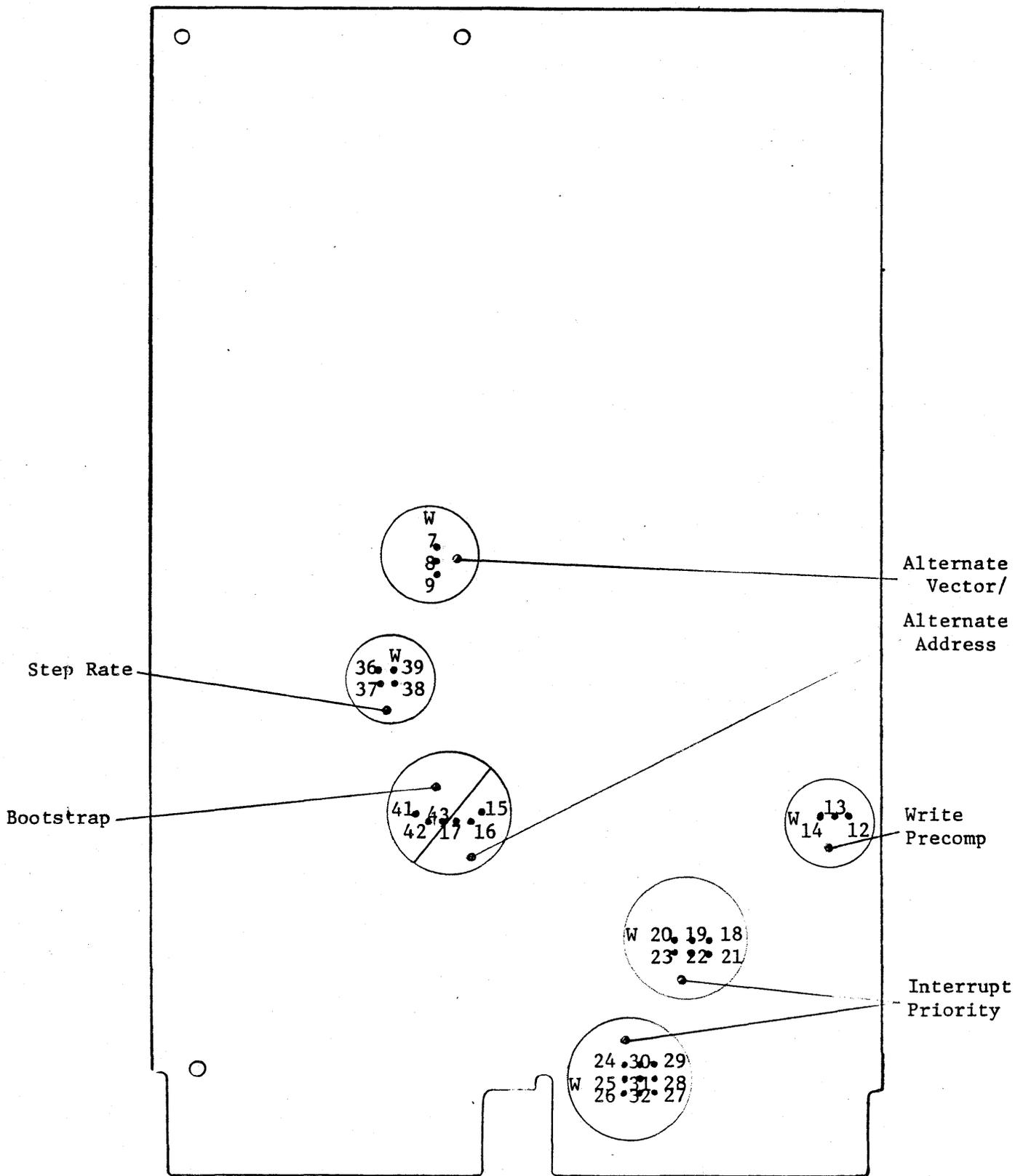


Figure 2-1: Configuration Jumper Location



### 2.1.2 Device Interrupt Priority

The MXV42C 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 2-1. 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 the following foil-etched jumpers must be cut. Refer to Table 2-2 for the proper jumpers to insert the desired priority level.

PRIORITY LEVEL	ASSERT	MONITOR	JUMPER									
			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 2-2: Priority Level Configuration

### 2.1.3 Bootstrap

#### 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.

The controller board incorporates a transparent firmware bootstrap. The bootstrap is initiated whenever program execution is started at location 173000<sub>g</sub>, homing both drives to track 0. Next, track 1, sector 1, of unit 0 is read and diskette density is determined. If the diskette is single density, sectors 1, 3, 5, and 7 are loaded. Program execution is then transferred to location 0.

Controllers are shipped with this feature disabled. To enable the bootstrap remove the jumper between W42 and W43 and insert the jumper between W42 and W41 as shown in Table 2-3.

OPTION	JUMPERS	
	41-42	42-43
Bootstrap Enabled	IN	OUT
Bootstrap Disabled*	OUT	IN
*Factory Set		

Table 2-3: Bootstrap Option

#### 2.1.4 Write Precompensation

The MXV42C 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 W12 and W13 and inserting a jumper between W13 and W14 as shown in Table 2-4.

OPTION	JUMPERS	
	12-13	13-14
Write Precomp Enabled*	IN	OUT
Write Precomp Disabled	OUT	IN
*Factory Set		

Table 2-4: Write Precompensation

### 2.1.5 Step Rate Control

The MXV42C is shipped with a 3ms step rate for the mini-floppy (5¼") and 6ms step rate for the standard floppy 8". Alternate step rates can be selected. Refer to Table 2-5 for option description and jumper numbers.

OPTION STEP RATE	JUMPERS	
	36-37	38-39
5¼" = 3ms* 8" = 6ms*	IN	IN
5¼" = 6ms 8" = 6ms	OUT	IN
5¼" = 3ms 8" = 3ms	IN	OUT
5¼" = 6ms 8" = 3ms	OUT	OUT
*Factory Set		

Table 2-5: Step Rate

## 2.2 DRIVE CONFIGURATION

For proper operation the floppy drives must be configured with attention to several options. The controller uses radial drive selection. Thus the drive(s) should be correspondingly configured.

### 2.2.1 Mini-Floppy Configuration

The mini-floppy (5¼") drives are configured as physical drive 3 and drive 4. However, the drives are referenced by the primary controller address register set and hence appear to the operating software as unit 0 and unit 1 respectively. For details concerning these and other options refer to table 2-6.

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

Table 2-6: Mini-Floppy Drive Configuration

### 2.2.2 8" Floppy Configuration

When two 8" drives are used, the first should be drive 1 and the second drive 2. A particular drive is selected and remains selected after a function is complete to allow the controller to poll drive status. A separate head load signal is provided by the controller for read and write functions on the diskette. The "in use" logic of the drive should be configured as a function of head loading. Since the drives are homed without loading the heads during an initialize command, the drive should be configured to provide stepper motor power independent of head loading. For details concerning these and other options refer to Table 2-7.

OPTION	DESCRIPTION	DUAL		SINGLE
		DRIVE 0	DRIVE 1	DRIVE 0
DS1	Drive select 1	In	Out	In
DS2	Drive select 2	Out	In	Out
DS3	Drive select 3	Out	Out	Out
DS4	Drive select 4	Out	Out	Out
A	Radial head loading option	In	In	In
B	Radial head loading option	In	In	In
C	Head load option	In	In	In
D	In use option	Out	Out	Out
X	Radial head loading option	Out	Out	Out
WP	Inhibit write when protect	In	In	In
NP	Allow write when protect	Out	Out	Out
DS	Stepper power from drive select	In	In	In
HL	Stepper from head load	Out	Out	Out
Z	In use from drive select	Out	Out	Out
Y	In use from head load	In	In	In
R*	Ready output	Out	Out	Out
I	Index output	In	In	In
DC	Disk change output	X	X	X
S	Sector output	X	X	X
800	Sector option Disable	In	In	In
801	Sector option Enable	Out	Out	Out
L	-5V DC Bias	In	In	In
T1	Termination HL	Out	In	In
T2	Termination Drive Select	In	In	In
T3	Termination Direction	Out	In	In
T4	Termination Step	Out	In	In
T5	Termination Write Data	Out	In	In
T6	Termination Write Gate	Out	In	In

\*If cable method B is used it is not necessary to remove this option

Table 2-7: 8" Floppy Drive Configuration

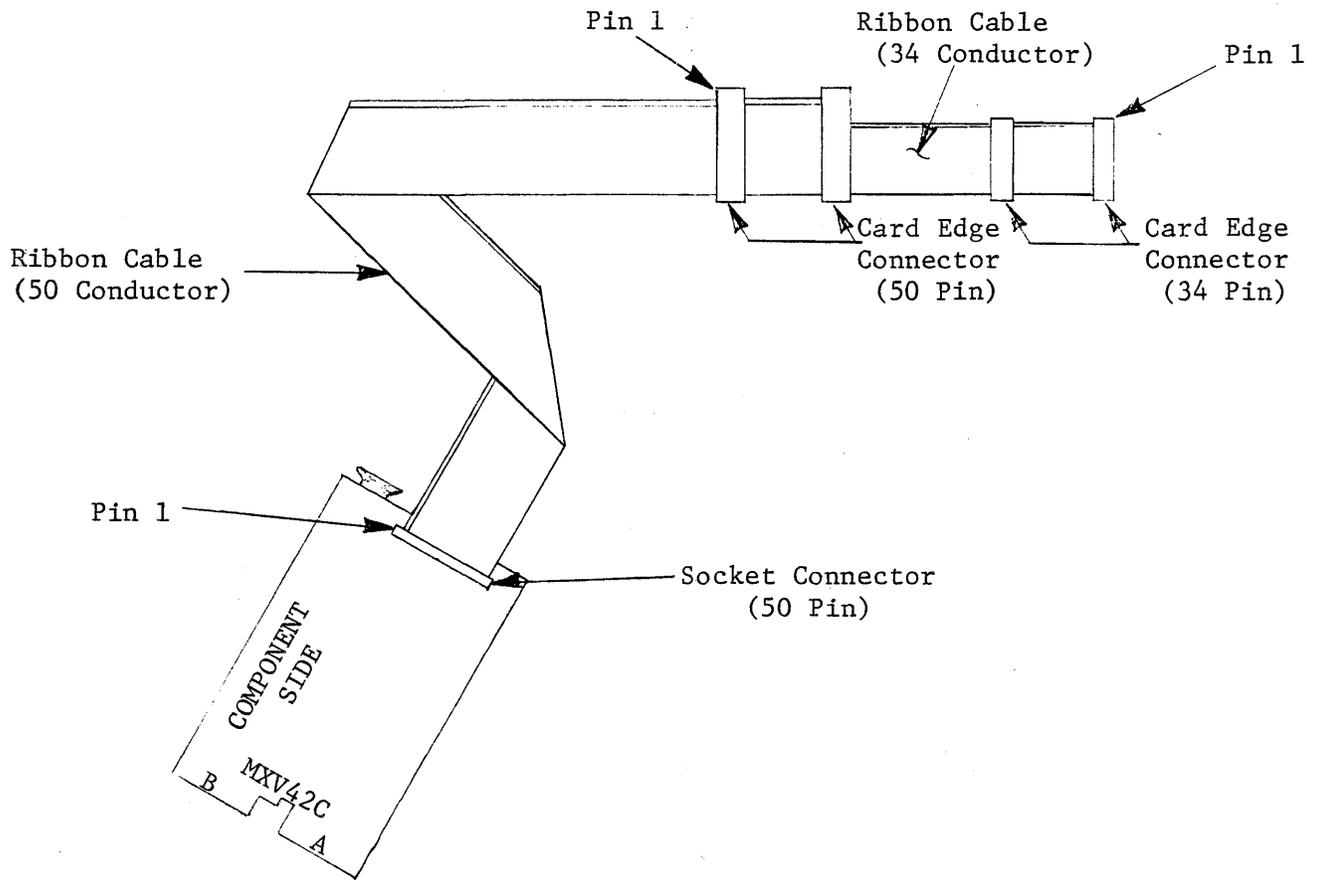
### 2.3 CABLING

The 50 pin header located at the top edge of the controller board provides all required interface signals for both the 8" and 5¼" Shugart compatible floppy disk drives. A 50 conductor ribbon cable is used to connect the controller to any 8" Shugart compatible drive. A 34 conduct ribbon cable, terminated to the upper 34 pins of the 50 pin socket connector, is used to terminate the controller to any 5¼" Shugart compatible drive. Two 8" and two 5¼" drives are simultaneously connected to the controller using one of the two following suggested methods.

Method A connects to all four drives using a single control cable (See Figure 2-3). At one end of the 50 conductor ribbon cable a 50 pin socket connector is terminated and connects the 50 pin socket connector to the controller card. At the required service length of the cable two 50 pin card edge connectors are terminated and connect to the 8" floppy disk drives. Beyond these connectors only the upper 34 conductors are required. The unused conductor should be trimmed. Again at the required service length two 34 pin card edge connectors are terminated and connect to the 5¼" mini-floppy drives. Using this cabling method requires the removal of the "READY" (R) option from both of the 8" floppy drives (Refer to Section 2.2).

#### NOTE

Only the last drive on the cable should be terminated. Remove the terminator resistor arrays from the other drives. If more than two termination networks remain, loading of the interface signals will result in intermittent operation and possible loss of data.



QTY	DESCRIPTION	MFG	NUMBER
1 ea	50 Pin Socket Connector	3M	3425-3000
2 ea	50 Pin Edge Connector	3M	3415-0001
2 ea	34 Pin Edge Connector	3M	3463-0001
AR	50 Conductor Ribbon Cable	3M	3365/50

Figure 2-3: Drive/Controller Cabling Method A

Method B connects the mini-floppy drive(s) to the controller using a cable configured with a 50 pin socket plug in place of the 50 pin edge connectors discussed in method A. See Figure 2-4. This socket plug will accept a standard 50 pin ribbon cable assembly. By removing pin 22 of this 50 pin plug connector the requirement to remove the "READY" (R) option from the 8" floppy drive(s) is eliminated. Standard configured 8" drives can then be directly plugged into this connector.

NOTE

Only the last drive on each cable assembly should be terminated. Remove the terminator resistor arrays from the other drives. If more than two termination networks remain loading of the interface signals will result in intermittent operation and possible loss of data.

In both methods observe the alignment of pin 1 of the socket connector and header as indicated by the arrows shown in Figure 2-3 and 2-4. The edge connectors should be connected to the corresponding drives, again observing the location of pin 1.

The connector illustrated in Figure 2-5 is compatible with both the Shugart compatible 8" and 5¼" floppy drives. Any drive that has both a Shugart compatible interface and connector should function properly with the controller.



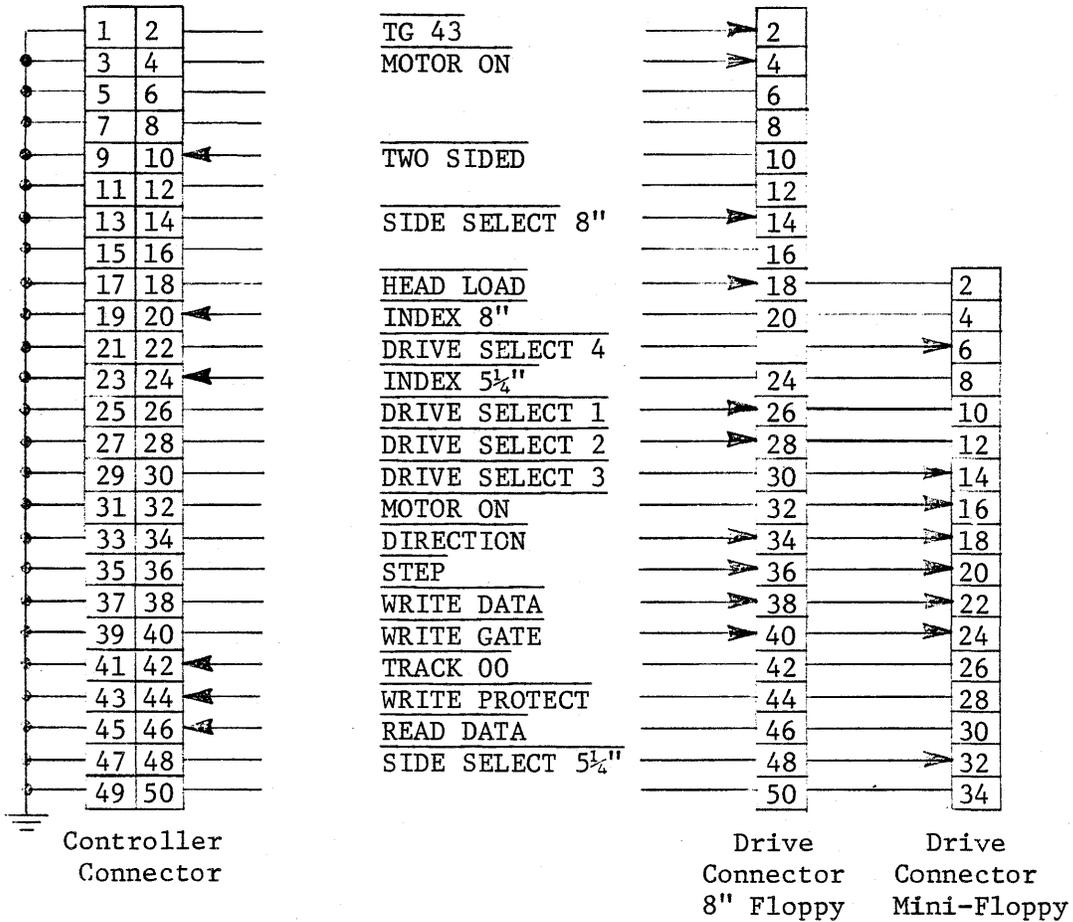


Figure 2-5: Connector Pin Definition

## 2.5 INITIAL OPERATION AND CHECKOUT

Before the following procedures are done, verify that the controller has been configured as described in Sections 2.1-2.4.

### NOTE

The bootstrap must be disabled for the following procedures.

1. Apply AC and DC power to the drive(s). The in-use indicators on all drives should be off.
2. Place the Run/Halt switch on the processor to the Halt position and turn on the processor. An "@" character should be printed on the terminal signifying that console ODT has been entered. All drives (first drive 1, then drive 0) of each set will step the heads inward 2-tracks, then step the heads outward until the home signal is detected. The heads will not load, and if the drive is configured as per Section 2.2, the in-use indicators will not light. If the above events do not occur, check the cabling and drive power supplies.
3. Open the CS register using ODT by typing 177170/ on the terminal. The processor will display the contents of the CS register. If the controller is operating properly a 004040<sub>8</sub> should be printed. Deposit a 40000<sub>8</sub> in the CS register by typing 40000 CR. This command will initialize the controller. Both drives of each set should calibrate for home position. First, drive 1 steps outward 2 tracks then inward one track at a time until the drive indicates track 0 has been reached. The procedure is repeated on drive 0. The heads will not load and if the drive is configured as per Section 2.2, the in-use indicators will not light.

4. Reopen the CS register (location  $177170_8$ ) using ODT as described above. The contents of this location should be  $004040_8$ . Examining the next location  $177172_8$  should yield a  $04_8$ . For a detailed description of the register protocol and bit definition, refer to Section 3.
5. If the above procedures function as described, the controller is ready for use. Either diagnostics or an operating system can be booted. For details on bootstrapping refer to Section 4.2.
6. If the above procedures cannot be validated, consult the factory of your local representative for assistance.

NOTE

For an LSI-11/23 the CS and DB registers are addressed as  $777170_8$  and  $777172_8$ , respectively.

### 3.1 GENERAL

This section describes device registers and command protocol for the MXV42C.

The MXV42C controls two 5¼" mini-floppy drives and two 8" floppy drives. All software control of the MXV42C is performed by means of two device registers: the command and status (MXVCS) register and a multipurpose data buffer (MXVDB) register. There is a set of registers to control each type of drive. For the 5¼" drives the MXVCS is assigned the address 177170<sub>8</sub> and the MXVDB is assigned 177172<sub>8</sub>. The MXVCS and MXVDB addresses for the 8" drives are 177174<sub>8</sub> and 177176<sub>8</sub> or (177150 and 177152), respectively. The registers can be read or loaded, with a few exceptions, using any instructions referring to their addresses.

The MXVCS register passes control information from the CPU to the controller and reports status and error information from the controller to the CPU. The MXVDB is provided for additional control and status information between the CPU and the controller. The information that is present in the MXVDB at any given time is a function of the controller operation in progress.

#### NOTE

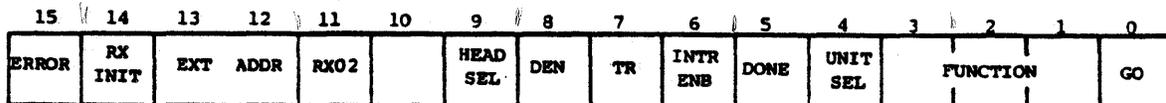
Loading one set of registers can affect the contents of the other set. This will not hinder operation of the controllers because once a command is initiated on a set of registers, only that set will act on data supplied to it.

The controller contains a sector buffer capable of storing a complete sector. For read/write operations the buffer is either "filled" before a write command or "emptied" after a read command under DMA control. During a write command the controller locates the desired sector and the buffer information is transferred to the diskette. During a read operation the desired sector is located and the sector data is transferred to the buffer.

### 3.2 REGISTER DEFINITIONS

#### 3.2.1 MXVCS - Command and Status Register (5¼" - 177170g)(8" - 177174g or 177150g)

The format of the MXVCS register is shown below. Functions are initiated by loading the command and status (CS) register, when not busy (bit 5=1), with bit 0=1. Command protocol is discussed in detail in Section 3.3.



8" Only

BIT	DESCRIPTION
-----	-------------

- |    |   |
|----|---|
| 15 | ERROR: This bit is set by the controller to indicate that an error has occurred during an attempt to execute a command. This bit is cleared by the initiation of a new command or by setting the initialize bit. When an error is detected the MXVES is read into the MXVDB. This bit is a read-only bit. |
| 14 | MXV22M INITIALIZE: This bit is set by the program to initialize the controller without initializing all the devices on the LSI-11 bus. This is a write-only bit.  |

**CAUTION**

Loading the lower byte of the MXVCS  
will also load the upper byte of the  
MXVCS.

When this bit is set, the controller will negate Done and move the head position mechanism of drive 1 (if two drives are available) to track 0. When completed, the controller will repeat the operation on drive 0.

The controller will then clear the error and status register, set Initialize Done, and set Drive Ready if drive 0 is ready. Finally, the controller will read sector 1, track 1, of drive 0.

- |       |   |
|-------|---|
| 12-13 | EXTENDED ADDRESS BITS: These bits are used to specify an extended bus address. Bit 12=MA16. Bit 13=MA17. These are write-only bits. |
| 11    | RX02: This bit is asserted by the controller to indicate that this is an RX02 type system. This is a read-only bit.                 |
| 10    | RESERVED: Must be written as zeroes.  |

- 09 5½" NOT USED
- 8" HEAD SELECT: This bit selects one of the two possible sides of the disk for execution of the desired function. When cleared, side 0 is selected; when set, side 1 is selected. This is a read/write bit.
- 08 DENSITY SELECT: This bit selects either single or double density operation. When cleared, single density is selected; when set, double density is selected. This is a read/write bit.
- 07 TRANSFER REQUEST: This bit signifies that the controller needs data or has data available. This is a read-only bit.
- 06 INTERRUPT ENABLE: This bit is set by the program to enable an interrupt when the controller has completed an operation and asserted the Done bit. The condition of this bit is cleared by initialize. This is a read/write bit.
- 05 DONE: This bit indicates the completion of a function. Done will generate an interrupt when asserted if interrupt enable (MXVCS bit 6) is set. This is a read-only bit.
- 04 UNIT SELECT: This bit selects one of the two possible disks for execution of the desired function. This is a read/write bit.
- 03-01 FUNCTION SELECT: These bits code one of the eight possible functions described in detail within this section. These are write-only bits.
- |     |                           |
|-----|---------------------------|
| 000 | Fill Buffer               |
| 001 | Empty Buffer              |
| 010 | Write Sector              |
| 011 | Read Sector               |
| 100 | Set Media Density/Format  |
| 101 | Read Status               |
| 110 | Write Deleted Data Sector |
| 111 | Read Error Code           |
- 00 GO: Initiates a command to the controller. This is a write-only bit.

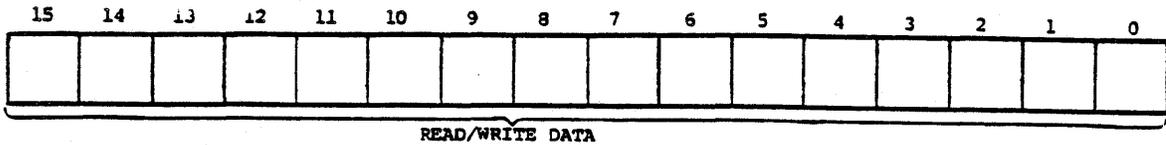
### 3.2.2 MXVDB - Data Buffer (5½" - 177172<sub>8</sub>) (8" - 177176<sub>8</sub> or 177152)

This register serves as a general purpose data path between the controller and the LSI-11. It will represent one of six registers according to the protocol of the function in process. These registers include the MXVDB, MXVTA, MXVSA, MXVWC, MXVBA, and MXVES.

This register is a read/write register if the controller is not in the process of executing a command (i.e., it may be manipulated without affecting the controller). When the controller is executing a command, this register can only be written to or read from when MXVCS bit 7 (TR) is set.

Data Buffer Register (MXVDB)

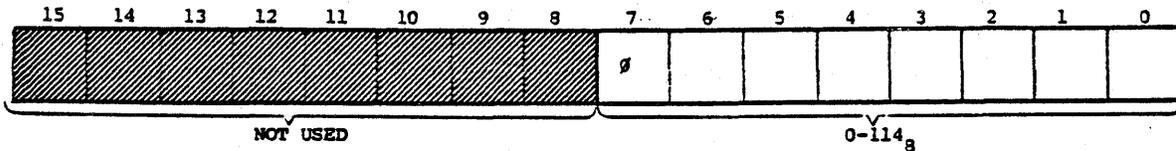
All information transferred to and from the floppy media passes through the MXVDB register and is addressable only under the protocol of the function in progress.



MXVDB FORMAT

Track Address Register (MXVTA)

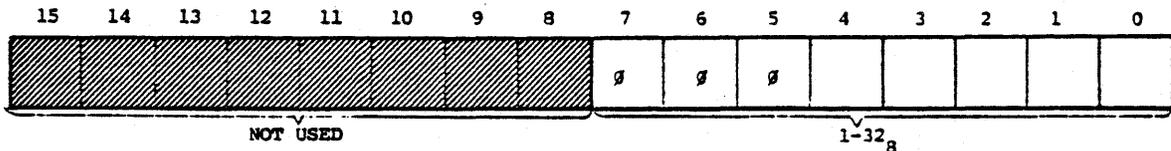
This register is loaded to indicate on which of the 115<sub>8</sub> (77 decimal) tracks a given function is to operate. It can be addressed only under the protocol of the function in progress. Bits 8 through 15 are not used and are ignored.



MXVTA FORMAT

Sector Address Register (MXVSA)

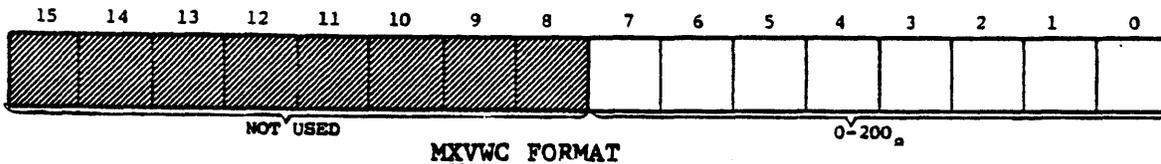
This register is loaded to indicate on which of the 32<sub>8</sub> (26 decimal) sectors a given function is to operate. It can be addressed only under the protocol of the function in progress. Bits 8 through 15 are not used and are ignored.



MXVSA FORMAT

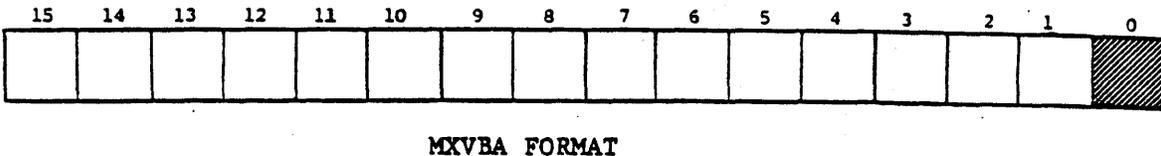
Word Count Register (MXVWC)

This 8-bit register is loaded with the number of words (maximum of 128 decimal) to be transferred. At the end of each transfer the word count register is decremented. When the contents of the register are decremented to zero, transfers are terminated; Done is set (MXVES bit 5); and, if enabled, an interrupt is requested. If the word count is greater than the limit for the density specified, the controller asserts a Word Count Overflow (bit 10 of the MXVCS). This register can be addressed only under the protocol of the function in progress. Bits 8 through 15 are not used and are ignored.



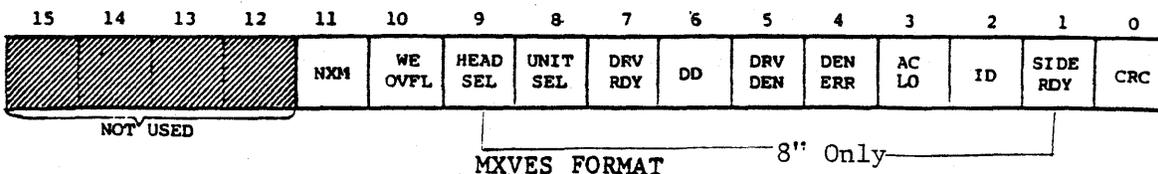
Bus Address Register (MXVBA)

This register is used to generate the bus address which specifies the location to and from which data are to be transferred. The register is incremented after each transfer. It will increment across 32K boundary lines via the extended address bits in the control and status register. Systems with only 16 address bits will "wraparound" to location zero when the extended address bits are incremented. This register can be addressed only under the protocol of the function in progress. Bit 0 is not used and is ignored.



Error and Status Register (MXVES)

This register contains the current error and status conditions of the drive selected by bit 4 (Unit Select) of the MXVCS. This read-only register can be addressed only under the protocol of the function in progress. The MXVES is loaded in the MXVDB upon completion of a function.



- 15-12 Not Used
- 11 NONEXISTENT MEMORY ERROR: This bit is asserted by the controller when the memory address specified for a DMA operation is nonexistent.
- 10 WORD COUNT OVERFLOW: This bit indicates that the word count specified is greater than the limit for the density selected. Upon detecting this error the controller terminates the fill or empty buffer operation and asserts the Error and Done bits.
- 09 5¼" Not used
- 8" HEAD SELECT: This bit indicates the side currently selected. If cleared, it indicates side 0; if set, it indicates side 1.
- 08 UNIT SELECT: This bit indicates the drive currently selected. If cleared, it indicates drive 0; if set, it indicates drive 1.
- 07 DRIVE READY: This bit is asserted if the unit currently selected exists, is properly supplied with power, has a diskette installed correctly, has its door closed, and has a diskette up to speed. This bit is only valid when retrieved via a read status function or at the completion of initialize when it indicates the status of drive 0.
- 06 DELETED DATA: During data recovery, the identification mark preceding the data field was decoded as a deleted data mark.
- 05 DRIVE DENSITY: This bit indicates the density of the diskette in the selected drive. When zero, it indicates single density; when set to one, it indicates double density.
- 04 DENSITY ERROR: A density error was detected as the information was retrieved from the data field of the diskette (a density error occurs when the density selected differs from that of the data field). Upon detecting this error the controller loads the MXVES into the MXVDB and asserts the Error and Done bits.
- 03 ACLO - This bit is set by the controller to indicate a power failure.
- 02 INITIALIZE DONE: This bit is asserted to indicate completion of the initialize routine which can be caused by system power failure or programmable LSI-11 bus initialize.
- 01 5¼" Not Used
- 8" SIDE READY: This bit is asserted by the controller when a double sided drive is selected, is ready, and has double-sided media inserted. The assertion of this bit indicates that side 1 of the selected drive is available for read and write operations.
- 00 CRC ERROR: A cyclic redundancy check error was detected as information was retrieved from a data field of the diskette. The information stored in the buffer should be considered invalid. Upon detection of this error the controller loads the MXVES into the MXVDB and asserts the Error and Done bits.

### 3.2.3 Extended Status Registers

The controller has four internal status registers. These registers provide specific error information in the form of error codes as well as drive status information depending upon the general error type. The registers can be retrieved by a read error code function as described in Section 3.3.8.

#### Word 1 <7:0> -Definitive Error Code

##### Octal Code Error Code Meaning

040	Tried to access a track greater than 76.
050	Home was found before desired track was reached.
070	Desired sector could not be found after looking at 52 headers (2 revolutions).
120	A preamble could not be found.
150	The header track address of a good header does not compare with the desired track.
160	Too many tries for an IDAM (identifies header).
170	Data AM not found in allotted time.
200	CRC error on reading the sector from the disk.
240	Density Error.
250	Wrong Key word for Set Media Density Command.
260	Illegal Data AM.
270	Invalid POK during write sequence.
300	Drive not ready.
310	Drive write protected.

#### Word 1 <15:8> -Not Used

This register is always cleared by the controller.

#### Word 2 <7:0> -Current Track Address of Drive 0

This register is cleared during the initialize command in order to synchronize with actual track position. The register is updated with each seek on drive 0 and maintains current track position.

Word 2 <15:8> -Current Track Address of Drive 1

This register is cleared during the initialize command in order to synchronize with actual track position. The register is updated with each seek on drive 1 and maintains current track position.

Word 3 <7:0> Target Track of Current Disk Access

If legal, the track specified for the last read/write command is saved in this register.

Word 3 <15:8> Target Sector of Current Disk Access

The sector specified for the last read/write command is saved in this register.

Word 4 <15:8> Track Address of Selected Drive

This register contains the track address read from the sector header of the desired sector during the last read/write command.

**3.3 COMMAND PROTOCOL**

Data storage and recovery using the MXV22M controller is accomplished by careful manipulation of the MXVCS and MXVDB registers according to the strict protocol of the individual functions. The penalty for violation of protocol can be permanent loss of data. Each of the functions are encoded and written into the command and status register bits 1-3 as described in Section 3.2.1. The detailed protocol for each function is described below.

**3.3.1 Fill Buffer (000)**

This function is used to fill the controller buffer with data from the host processor. The number of words to transfer is specified by the host. The command density bit determines the buffer size (64 or 128 words). The controller zero-fills the remaining buffer space. If the word count is too large for the density selected the function is aborted, Error and Done are asserted and the Word Count Overflow bit is set in the MXVES.

The contents of the buffer may be written on the diskette with a subsequent write sector command or returned to the host processor using an empty buffer command.

When the command is loaded, MXVCS bit 5 (Done) is negated. MXVCS bit 8 (density) must be set to define the buffer size. MXVCS bits 12 and 13 (extended address bits) must also be asserted to define the extended memory segment used with the buffer address, yet to be specified, to form the absolute memory address of the data to be transferred. MXVCS bit 4 (unit select) is ignored since no drive operation is required. When MXVCS bit 7 (TR) is first asserted, the program must move the word count into the MXVDB which will negate TR.

When the controller again asserts TR, the program must move the buffer address into the MXVDB. The controller then negates TR, initiates a DMA cycle, and transfers the first word from the host processor to the controller buffer. At the end of the transfer the word count register is decremented and the buffer address is incremented by two. This cycle is repeated until the word count register becomes zero. The controller zero-fills the remaining buffer space, sets the Done bit, and if enabled, causes an interrupt request. After Done is asserted the MXVES is moved into the MXVDB.

During the Data Transaction, if any non-existent memory is addressed, the controller will time out and abort the function. The Error and Done bits will be asserted. MXVES bit 11 (NXM) will be set and the MXVES will be moved into the MXVDB; if enabled, an interrupt request will be generated.

### 3.3.2 Empty Buffer (001)

This function is used to transfer the contents of the controller to the host processor. The number of words to transfer is specified by the host. The command density bit determines the maximum legal word count. If the word count specified is too large for the density selected the function is aborted, Error and Done are asserted and the Word Count Overflow bit is set in the MXVES.

The contents of the buffer may be transferred to the host as many times as desired or may be written on the diskette with a subsequent write sector command. Unless a fill buffer or read sector command is issued, the controller buffer is not destroyed.

When the command is loaded, MXVCS bit (Done) is negated, MXVCS bit 8 (density) must be set to allow the proper word count limit. MXVCS bits 12 and 13 (extended address bits) must also be asserted to define the extended memory segment used with the buffer address, yet to be specified, to form the absolute memory destination address. MXVCS bit 4 (unit select) is ignored since no drive operation is required. When MXVCS bit 7 (TR) is first asserted the program must move the word count into the MXVDB which will negate TR. When the controller again asserts TR the program must move the buffer address into the MXVDB. The controller then negates TR, initiates a DMA, and transfers the first word of the buffer to the host processor. At the end of the transfer, the word count register is decremented and the buffer address register is incremented by two. This cycle is repeated until the word count register becomes zero. The controller then sets the Done bit and if enabled, causes an interrupt request. After Done is asserted the MXVES is moved into the MXVDB.

During the DMA transaction, if any non-existent memory is addressed, the controller will time out and abort the function. The Error and Done bits will be asserted. MXVES bit 11 (NXM) will be set and the MXVES will be moved into the MXVDB. If enabled, an interrupt request will be generated.

### 3.3.3 Write Sector (010)

This function is used to locate a desired track and sector and write the sector with the contents of the internal sector buffer. When the MXVCS is loaded with this command, the MXVES is cleared and both the TR and Done bits are negated. When TR is first asserted the program must load the desired sector address into the MXVDB which will negate TR. When TR is again asserted the program must load the desired track address into the MXVDB which will negate TR. The controller then seeks the desired track and attempts to locate the desired sector. The desired track is compared with the track field of the sector field. If they do not match the operation is aborted, the Error and Done bits are asserted, the MXVES is moved into the MXDB, and if enabled the controller will assert an interrupt.

If the densities agree but the controller is unable to locate the desired sector within two diskette revolutions, the controller will abort the operation, move the contents of MXVES into MXVDB, assert the Error and Done bits, and if enabled, assert an interrupt request.

If the desired track and sector are located and the densities agree, the controller will write the contents of the internal sector buffer followed by a CRC character, all in the function selected density. The controller completes the operation by moving the MXVES to the MXVDB, asserts Done, and if enabled, asserts an interrupt request.

**CAUTION**

The contents of the internal sector buffer are lost during a power failure. However, after power is brought back to normal, a write sector command will cause the random contents of the buffer to be written on the diskette with a valid CRC character.

**NOTE**

The contents of the sector buffer are not destroyed by a write sector operation.

#### 3.3.4 Read Sector (011)

This function is used to locate the desired track and sector and transfer the contents of the data field into the controller's internal sector buffer. When the MXVCS is loaded with this command, the MXVES is cleared and both the TR and Done bits are negated. When TR is first asserted the program must load the desired sector address into the MXVDB which will negate TR. When TR is again asserted the program must load the desired track address into the MXVDB which will negate TR.

Both the TR and Done bits remain negated while the controller attempts to locate the desired sector. If after two revolutions the controller is unable to locate the desired sector, the operation is aborted. The controller will move the MXVES into the MXVDB, assert the Error and Done bits, and if enabled, assert an interrupt request.

When the desired sector is located, the controller will then compare the desired track with the track field of the sector header. If they do not match, the operation is aborted. The Error and Done bits are asserted, the MXVES is moved into the MXVDB, and if enabled, the controller will assert an interrupt request.

**NOTE**

Since there is only enough memory to store one sector of data internal to the MXV42C controller, reading a sector using first one set of drives than the other will destroy the data from the first set. This does not present a problem since a read sector is normally followed by an empty buffer.

Note that user written drivers can transfer data between the 5 $\frac{1}{4}$ " and 8" diskettes without performing either a fill of empty buffer.

### 3.3.5 Set Media Density (100)

This function is dual purpose. The controller can set the media density by rewriting all the data address marks (single of double density) and writing zero data fields in the selected density. The controller can also "reformat" the entire diskette by rewriting both the sector headers and the data fields. The data fields are written in the selected density preceded by the corresponding data address mark. Both commands are initiated by the set media function but differ in the keyword required by the controller to execute the command.

When the MXVCS is loaded with the command, the MXVES is cleared and the Done bit is negated. When TR is set, the program must respond with a keyword. This keyword must be deposited in the MXVDB to complete the protocol. When the controller recognizes this character, it begins executing the command. If an illegal keyword is used, the operation is aborted. The MXVES is moved into the MXVDB, the Error and Done bits are set, and if enabled, the controller asserts an interrupt request.

If the keyword used is a 111g, the controller initiates a set media density operation. This operation starts at track 0, sector 1. Each sector header is located and a write operation is initiated. A data field is written with zero data in the density selected. If an error occurs reading any header, the operation is aborted. The MXVES is moved into the MXVDB, The Error and Done bits are set, and if enabled, the controller asserts an interrupt request. If the operation is successfully completed, Done is set and if enabled, the controller asserts an interrupt request.

If the keyword used is a 222g, the controller initiates a format operation. This function starts at the physical index of track 0. Each track is written first with an index address mark, then 26 sector headers are written as described in Section 4.3. When each track (or cylinder) has been written, the controller initiates a set media density function as described above.

#### CAUTION

The set media density function takes about 30 seconds and the format function takes about 90 seconds for the 5¼" mini-floppy. For the 8" floppy, the times are approximately 25 seconds and 50 seconds, respectively. Neither operation should be interrupted. If an error occurs or either operation is interrupted, an illegal diskette may have been generated and the operation should be repeated.

### 3.3.6 Read Status (101)

This function is used to update the drive status information and is initiated by loading the command into the MXVCS. The Done bit is negated. MXVES bit 7 (Drive Ready) is updated by sampling the drive ready status line. Drive density is updated by loading the head of the selected drive and reading the first data address mark. The controller then moves the MXVES into the MXVDB, asserts Done, and if enabled, asserts an interrupt request. This operation requires about 250ms to complete.

### 3.3.7 Write Deleted Data Sector (110)

This operation is identical to Write Sector (010) with one exception. The data address mark preceding the data is not the standard data address mark. A single or double density deleted data address mark is written according to the density of the function.

### 3.3.8 Read Error Code (111)

This function is used to retrieve the extended status registers and is initiated by loading the MXVCS with the command. The Done bit is negated. When TR is asserted, the program must load the Bus Address into the MXVDB which negates TR. The controller assembles one word at a time and under DMA control, transfers them to memory starting at the address specified.

If non-existent memory is encountered during the transfer, the operation is aborted. The Error and Done bits will be asserted, MXVES bit 11 (NXM) will be set, and the MXVES will be moved into the MXVDB. If enabled, an interrupt request will be generated.

When all four words have been transferred the Done bit is set and if enabled, an interrupt request is generated.



#### 4.1 GENERAL

This section provides the user pertinent information concerning the description and use of the controller functions. The functions covered include: bootstrapping, formatting, fill/write operations, read/empty operations, write current control, write precompensation, and power fail protection. The section also reviews operation with a RT-11 operating system.

#### 4.2 BOOTSTRAPPING THE CONTROLLER

If the bootstrap is enabled as described in Section 2.1.2, the controller will respond to the standard bootstrap address  $173000_8$ . The controller is bootstrapped by typing  $173000G$  while in console ODT. This causes a bus INIT and transfers program execution to location  $173000_8$ . An alternate method is to strap the LSI-11 processor to power up in Mode 2. In this mode, when a power up occurs, the processor automatically starts execution at  $173000_8$ . Power-up strapping procedures for the LSI-11 processor can be found in the Microcomputer Processors Handbook.\*

To boot either a single or double density diskette use the following procedure:

1. Place the diskette in drive  $\emptyset$ .
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  $173000G$ .

#### NOTE

For an LSI-11/23 the boot address is  $773000_8$ , the controller address is  $777170_8$  and the alternate controller address is  $777174_8$  (or  $777150_8$ ).

\*Published by Digital Equipment Corporation. Maynard, Mass. 1979

#### 4.2.1 Bootstrapping Operation

The bootstrap is not a standard ROM program. It uses the controller's microprocessor to capture the bus; 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  $173000_8$  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  $173000_8$  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  $173000_8$ . 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  $173004_8$ . The controller passes the device address  $177170_8$  if the standard address is selected, or  $177174_8$  if the alternate address is selected. The processor moves the address into R1 and then attempts to fetch location  $173006_8$ . 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 initiates a Read Status function on drive 0 to determine diskette density. If the diskette is single density the controller reads sectors 1, 3, 5, and 7 of track 1 of drive 0 into location 2 through 176, 200 through 376, 400 through 576, and 600 through 776 respectively. If the diskette is double density the controller reads sectors 1 and 3 of track 1 of drive 0 into locations 2 through 376, and 400 through 776 respectively. 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.

### 4.3 FORMAT OPERATIONS

#### 4.3.1 5½" Mini-Floppy

The controller has the capability of formatting diskettes in a specified density. The formatting is accomplished in two passes. During pass 1, an index address mark is written on track 0, side 0, following the index hole. Thirteen sector headers, for sectors 1-13, are written with appropriate spacing following the index address mark. An index address mark is also written in track 0, side 1, followed by sector headers for sectors 14-26. Each of the remaining 76 cylinders are written in the same manner. When cylinder 76 is completed, pass 2 is initiated. The controller seeks track 0 and writes a data field consisting of all zero bytes in sector 1 using the selected density. The remaining sectors are written in the same manner. Table 4-1 lists the various command formats.

The format command selects diskette density and unit.

	Unit 0	Unit 1
Single Density	11 <sub>8</sub>	31 <sub>8</sub>
Double Density	411 <sub>8</sub>	431 <sub>8</sub>

Table 4-1: 5½" Mini-Floppy Command Word Formats

#### 4.3.2 8" Floppy

The controller has the capability of formatting diskettes in a specified density. The formatting is accomplished in two passes. During pass 1, an index address mark is written on track 0 following the index hole. Twenty-six sector headers, appropriately spaced, are written following the index address. Each of the remaining 76 tracks is written in the same manner. When track 76 is completed, pass 2 is initiated. The controller seeks track 0 and writes a zero data field in sector 1 using the selected density. The remaining sectors are written in the same manner.

The format command selects diskette density, unit and side (for dual headed drives). Table 4-2 lists the various command word formats.

	Unit 0	Unit 1
Single Density Side 0	11 <sub>8</sub>	31 <sub>8</sub>
Single Density Side 1	1011 <sub>8</sub>	1031 <sub>8</sub>
Double Density Side 0	411 <sub>8</sub>	431 <sub>8</sub>
Double Density Side 1	1411 <sub>8</sub>	1431 <sub>8</sub>

Table 4-2: 8" Floppy Command Word Formats

Figure 4-1 illustrates a format subroutine. The format command is loaded into MXVCS. When TR is set, the keyword 222<sub>g</sub> is loaded into MXVDB. When the diskette has been formatted a return is made.

FORMAT:

```

MOV #11, CMD           ;FORMAT
BIS DENS, CMD         ;DENSITY
BIS UNIT, CMD        ;UNIT
BIS SIDE, CMD        ;SIDE(For 8" Floppy Only)
MOV CMD, @#MXVCS     ;SELECT FUNCTION
JSR PC, TRWAIT       ;WAIT FOR TR
MOV #222, @#MXVDB    ;KEYWORD
JSR PC, DNWAIT       ;WAIT FOR DONE
TST @#MXVCS         ;ERROR
BMI FRMERR           ;BR IF SO
RTS PC

```

FRMERR:

Figur 4-1: Format Subroutine

Alternatively a diskette can be formatted using console ODT. Open the CS register and deposit the appropriate command. Then deposit the format key word 222<sub>g</sub> in the DB register. The following is an example of formatting unit 0 in double density:

<u>177170/</u>	004040	<u>411 &lt;LF&gt;</u>
177172/	000000	<u>222 &lt;CR&gt;</u>

#### 4.4 FILL/WRITE OPERATIONS

Figure 4-2 illustrates subroutines to write data on a diskette which is done by performing a fill buffer operation followed by a write sector.

The Fill Buffer command, specifying single or double density is loaded into the MXVCS. When TR is set, the word count is loaded into the MXVDB. When TR is again set, the bus address of the data is loaded into the MXVDB. A return is made when the controller's sector buffer is filled. The Write Sector command (specifying density, unit and side for 8" floppy only) is loaded into the MXVCS. When TR is set the sector address is loaded into the MXVDB. When TR is again set, the track address is loaded into the MXVDB. When the contents of the controller's sector buffer are written at the selected sector, a return is made.

##### FILLBF:

```
MOV    #1, CMD           ;FILL BUFFER
BIS    DENS, CMD         ;DENSITY
MOV    CMD, @#MXVCS     ;SELECT FUNCTION
JSR    PC, TRWAIT       ;WAIT FOR TR
MOV    COUNT, @#MXVDB   ;WORD COUNT
JSR    PC, DNWAIT
MOV    #BUFOUT, @#MXVDB ;BUS ADDRESS OF DATA
JSR    PC, DNWAIT       ;WAIT FOR DONE
TST    @#MXVCS          ;ERROR
BMI    ERFIL           ;BR IF SO
RTS    PC
```

##### ERFIL:

##### WSECT:

```
MOV    #5, CMD           ;WRITE, SECTOR
BIS    DENS, CMD         ;DENSITY
BIS    UNIT, CMD        ;UNIT
BIS    SIDE, CMD        ;SIDE(For 8" Floppy only)
MOV    CMD, @#MXVCS     ;SELECT FUNCTION
JSR    PC, TRWAIT       ;WAIT FOR TR
MOV    SECTOR, @#MXVDB  ;SECTOR
JSR    PC, TRWAIT
MOV    TRACK @#MXVDB    ;TRACK
JSR    PC, DNWAIT       ;WAIT FOR DONE
TST    @# MXVCS        ;ERROR
BMI    WSERR           ;BR IF SO
RTS    PC
```

##### WSERR:

Figure 4-2: Write Data Subroutines

#### 4.5 READ/EMPTY OPERATIONS

Figure 4-3 illustrates subroutines to read data from a diskette which is done by performing a Read Sector operation followed by an Empty Buffer operation.

The Read Sector command, specifying density, unit and side (for 8" floppy only) is loaded into the MXVCS. When TR is set the sector address is loaded into the MXVDB. When TR is again set, the track address is loaded into the MXVDB. When the contents of the selected sector are read into the controller's sector buffer, a return is made.

The Empty Buffer command, specifying density, is loaded into the MXVCS. When TR is set, the word count is loaded into the MXVDB. When TR is again set, the bus address of the storage buffer is loaded into the MXVDB. A return is made after the contents of the controller's buffer are transferred to the memory storage buffer.

RSECT:

```
MOV      #7,CMD          ;READ SECTOR
BIS      DENS, CMD       ;DENSITY
BIS      UNIT, CMD      ;UNIT
BIS      SIDE, CMD      ;SIDE(For 8" Floppy Only)
MOV      CMD, @#MXVCS   ;SELECT FUNCTION
JSR      PC, TRWAIT     ;WAIT FOR TR
MOV      SECTOR, @#MXVDB ;SECTOR
JSR      PC, DNWAIT
MOV      TRACK, @#MXVDB ;TRACK
JSR      PC, DNWAIT     ;WAIT FOR DONE
TST      @#MXVCS       ;ERROR
BMI      RSERR         ;BR IF SO
RTS      PC
```

RSERR:

EMPMF:

```
MOV      #3. CMD        ;EMPTY BUFFER
BIS      DENS, CMD      ;DENSITY
MOV      CMD, @#MXVCS   ;SELECT FUNCTION
JRS      PC, TRWAIT     ;WAIT FOR TR
MOV      COUNT, @#MXVDB ;WORD COUNT
JSR      PC, TRWAIT
MOV      #BUFFIN, @#MXVDB ;BUS ADDRESS FOR DATA
JSR      PC, DNWAIT     ;WAIT FOR DONE
TST      @#MXVCS       ;ERROR
BMI      ERMPEP        ;BR IF SO
RTS      PC
```

EREMP:

Figure 4-3: Read Data Subroutines

#### 4.6 WRITE CURRENT CONTROL

The controller provides a Write Current Control signal (TG43) which is asserted whenever a track address greater than 43 is accessed. This signal is required by some drives to reduce the effects of write saturation on the inner tracks. However, Shugart 850 series double sided drives require this signal (refer to the Shugart double sided diskette storage drive manual section 7.13). This signal is provided on pin 2 of the 50 pin ribbon and is disabled according to section 2.1.5.

#### 4.7 WRITE PROCOMPENSATION

Bit shift occurs on both single and double density diskettes. This shift is more noticeable with double density due to the smaller bit cell size and corresponding data and clock windows. Some aspects of bit shift are predictable and are dealt with by the precompensation scheme implemented in this controller; unpredictable effects are reduced by using PLL techniques.

Predictable bit-shift effects, result from normal read/write operation. Data are recorded by flux changes in the gap of the read/write head. These flux changes produce changes in magnetization in the oxide coating on the media. When reading the media the changing magnetization induces current in the read /write head. Since this change in current is not instantaneous, it takes a finite time to build up to a peak and return to zero. When the magnetic flux changes are close together the previous current transition may not reach zero before a second transition occurs. The summation of current pulses produces shifted peaks. Because the flux changes are closer together on the inner tracks (43 through 76) the bit shift is greater in this area.

Other causes of bit shift are variations in disk drive rotational speed. The specified  $\pm 2\%$  variation will produce bit shifts of  $\pm 40\text{ns}$ . Incomplete erasure of previously recorded data can produce bit shifts up to  $50\text{ns}$ .

Other miscellaneous components of bit shift include instantaneous speed variation, electrical noise, radial track alignment and nonsymmetry of the read/write head and associated electronics.

To improve margin the controller incorporates a scheme to recognize the data patterns which produce excessive bit shift and introduces a compensating bit shift. For tracks greater than forty-three the recorded bits are shifted 330ns for 5¼" mini-floppy and 165ns for the 8" floppy, either early or late, as determined by the two previously recorded bits and the subsequent two bits to be recorded. The controller also incorporates a phase-locked data recovery scheme which dynamically adjusts the recovery clock frequency to the data, reducing bit shifts due to rotational speed errors.

#### 4.8 POWER FAIL PROTECTION

The controller continuously monitors both the BPOK and BDCOK bus signals. Refer to the *Microcomputer Processors Handbook* for detailed descriptions of these signals. When asserted, BPOK signals an impending DC power failure and guarantees 4ms of operation before BDCOK is asserted and DC power fails. Assertion of BDCOK indicates invalid DC power. This signal is hardwired in the controller as an interlock on the Write Gate signal. When BDCOK is asserted the Write Gate signal is blocked and write operations are prevented.

Before initiating a write sequence, the controller interrogates the BPOK line. If an impending DC failure is indicated the operation is aborted.

#### 4.9 OPERATION USING RT11

The MXV42C (and RXV21) controller requires a different handler than the RX01 single density controllers. This new handler is configured to utilize the DMA transfer scheme of the controller. In addition, diskette density is determined by the handler without system intervention, allowing the use of either single or double density diskettes interchangeably.

This handler, designated "DY", is available in RT11-V03B and later revisions.

##### 4.9.1 Creating a DY-Compatible System Disk

The MXV42C controller requires the DY-based RT-11 monitor rather than the DX-based RT-11 monitor. The following procedures explain how to create a DY-based system.

Using an RX01 or equivalent system, or system which has an RX01 or equivalent peripheral device, the monitor file and other associated system files should be copied onto a single density diskette. These files can be obtained from the binary distribution media or by performing a SYSGEN and specifying DY as the system device (refer to the RT-11 System Generation Manual). The following commands will initialize the diskette and copy the necessary files to Drive 1:

##### RT11 V03B

```
INIT/NOQUERY DX1:
COPY/SYS DEV:SWAP.SYS DX1:
COPY/SYS DEV:DYMNXX.SYS DX1:
COPY/SYS DEV:TT.SYS DX1:
COPY DEV:DIR.SAV DX1
COPY DEV:PIP.SAV DX1:
COPY DEV:DUP.SAV DX1:
```

##### RT11 V04

```
INIT/NOQUERY DX1:
COPY/SYS DEV:SWAP.SYS DX1:
COPY/SYS DEV:RT11XX.SYS DX1:
COPY/SYS DEV:TT.SYS DX1:
COPY DEV:DIR.SAV DX1:
COPY DEV:PIP.SAV DX1:
COPY DEV:DUP.SAV DX1:
```

\*Published by Digital Equipment Corporation, Maynard, Mass. 1979

The bootstrap must then be copied from the monitor file to block 0 of the diskette. The following command will accomplish this on the diskette in drive 1.

```
RT11 V03B
COPY/BOOT DX1:DYMNXX DX1:
```

```
RT11 V04
COPY/BOOT DY1:RT11XX DX1:
```

This diskette can be used with the MXV22M controller but it is single density. To build a double density diskette the user must first format a diskette to double density as explained in Section 4.2. Boot the single density system diskette in drive 0. Use the following commands to initialize the formatted diskette in drive 1 and copy the system software from drive 0 to drive 1.

```
INIT/NOQUERY DY1:
COPY/SYS DY:*. * DY1:
```

Finally, copy the bootstrap to block 0 of the diskette in drive 1.

```
RT11 V03B
COPY/BOOT DY1:DYMNXX DY1:
```

```
RT11 V04
COPY/BOOT DY1:RT11XX DY1:
```

The diskette in drive 1 can now be booted as a double density system diskette.

#### 4.9.2 Creating a Mini-Floppy RT11 System Diskette

The mini-floppy will operate with standard DEC<sup>®</sup> software. To create a system diskette it is only necessary to copy to the 5¼" media RT11 organized to operate with the DY (RX02) device as the system device. Before transferring files a 5¼" diskette must be formatted (refer to section 4.3).

#### NOTE

The RT11 format program only performs a set media density function on the DY device. This function only sets the density of data fields and does NOT write sector headers.

The following commands will initialize the mini diskette, copy the necessary files to drive 0, and copy the bootstrap from the monitor file to block 0 of the diskette from any system device other than an RX02.

<sup>R</sup>Registered trademark of Digital Equipment Corporation, Maynard, Mass.

## RT11 V03B

INIT/NOQUERY DYØ:  
 COPY/SYS DEV:SWP.SYS DYØ:  
 COPY/SYS DEV:DYMNXX.SYS DYØ:  
 COPY/SYS DEV:TT.SYS DYØ:  
 COPY DEV:DIR.SAV DYØ:  
 COPY DEV:PIP.SAV DYØ:  
 COPY DEV:DUP.SAV DYØ:  
 COPY/BOOT DYØ:DYMNXX DYØ:  
 NOTE DY: = DYØ:

To use the dual control emulation of the MXV42C the DY driver (DY.MAC) must be reassembled for use with two controllers. Refer to the following procedure.

1. Using the 8" RXØ2 device, assembly the DY driver (DY.MAC) with the direct assignment statement:

DYT\$Ø = 1

This allows the driver to communicate with two controllers. Use the following equate statement to establish the alternate address compatible with the MXV22M:

DY\$CS2 = 177174 or 177150

2. Link the resulting object file (DY.OBJ) as DY.SYS. The old DY.SYS should be renamed prior to this step if it is to be retained.

## RT11 V04

INIT/NOQUERY DYØ:  
 COPY/SYS DEV:SWAP.SYS DYØ:  
 COPY/SYS DEV:RT11XX.SYS DYØ:  
 COPY/SYS DEV:TT.SYS DYØ:  
 COPY DEV:DIR.SAV DYØ:  
 COPY DEV:PIP.SAV DYØ:  
 COPY DEV:DUP.SAV DYØ:  
 COPY/BOOT DYØ:RT11XX.SYS DYØ:

### 4.9.3 DISKETTE COPY PROGRAM

To create a 5¼" system diskette from an 8" system diskette, the following program can be entered by using console ODT. It will copy the files from an 8" double density, single sided floppy diskette in drive 3 to a 5¼" double density, double sided mini-floppy diskette in drive 1. The program can be modified to change units or to select single density as described below.

```
@ 1 0 0 0 / XXXXXX 1 2 7 0 6 LF  
1002/XXXXXX 1 0 0 0 LF  
1004/XXXXXX 1 2 7 0 0 LF  
1006/XXXXXX 1 7 7 1 7 0 LF  
1010/XXXXXX 1 2 7 0 1 LF  
1012/XXXXXX 1 7 7 1 7 4 LF  
1014/XXXXXX 5 0 0 2 LF  
1016/XXXXXX 1 2 7 0 3 LF  
1020/XXXXXX 1 LF  
1022/XXXXXX 1 3 7 1 1 LF  
1024/XXXXXX 1 1 2 4 LF  
1026/XXXXXX 1 0 1 0 4 LF  
1030/XXXXXX 4 7 3 7 LF  
1032/XXXXXX 1 0 7 0 LF  
1034/XXXXXX 1 3 7 1 0 LF  
1036/XXXXXX 1 1 2 6 LF  
1040/XXXXXX 1 0 0 0 4 LF  
1042/XXXXXX 4 7 3 7 LF  
1044/XXXXXX 1 0 7 0 LF  
1046/XXXXXX 5 2 0 3 LF  
1050/XXXXXX 2 0 3 2 7 LF  
1052/XXXXXX 3 3 LF  
1054/XXXXXX 2 7 6 2 LF
```

1056/XXXXXX 5 2 0 2 LF  
 1060/XXXXXX 2 0 2 2 7 LF  
 1062/XXXXXX 1 1 5 LF  
 1064/XXXXXX 2 7 5 4 LF  
 1066/XXXXXX 0 LF  
 1070/XXXXXX 1 0 5 7 1 4 LF  
 1072/XXXXXX 1 0 0 3 7 6 LF  
 1074/XXXXXX 1 0 3 6 0 LF     10364  
 1076/XXXXXX 2 LF  
 1100/XXXXXX 1 0 5 7 1 4 LF  
 1102/XXXXXX 1 0 0 3 7 6 LF  
 1104/XXXXXX 1 0 2 6 4 LF  
 1106/XXXXXX 2 LF  
 1110/XXXXXX 3 2 7 1 4 LF  
 1112/XXXXXX 1 0 0 0 4 0 LF  
 1114/XXXXXX 1 7 7 5 LF  
 1116/XXXXXX 1 0 0 4 0 1 LF  
 1120/XXXXXX 2 0 7 LF  
 1122/XXXXXX 0 LF  
 1124/XXXXXX 4 0 7 LF  
 1126/XXXXXX 4 0 5 CR  
 @ 1 0 0 0 G     105714

The program will halt at location 1066 upon completion or at 1122 if an error is encountered. The copy operation will take about 30 minutes.

The read command for the 8" diskette is at location 1124 and the write command for the 5½" diskette is at 1126. The unit bit (bit 4) and the density bit (bit 8) can be changed although the two diskettes must have the same density and storage capacity.

A bootable version of the program on a 5½" diskette is available as an option.





**Micro Technology, Inc.**

1100-G W. Katella Avenue, Orange, California 92667, (714) 771-1042