MOTOROLA

MVME320/D1

MVME320
VMEbus Disk Controller Module
User's Manual

MICROSYSTEMS

QUALITY • PEOPLE • PERFORMANCE

MVME320/D1

FEBRUARY 1985

MVME320

VMEbus DISK CONTROLLER MODULE

USER'S MANUAL

WARNING

THIS EQUIPMENT GENERATES, USES, AND CAN RADIATE RADIO FREQUENCY ENERGY AND IF NOT INSTALLED AND USED IN ACCORDANCE WITH THE INSTRUCTIONS MANUAL, MAY CAUSE INTERFERENCE TO RADIO COMMUNICATIONS. IT HAS BEEN TESTED AND FOUND TO COMPLY WITH THE LIMITS FOR A CLASS A COMPUTING DEVICE PURSUANT TO SUBPART J OF PART 15 OF FCC RULES, WHICH ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST SUCH INTERFERENCE WHEN OPERATED IN A COMMERCIAL ENVIRONMENT. OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE INTERFERENCE IN WHICH CASE THE USER, AT HIS OWN EXPENSE, WILL BE REQUIRED TO TAKE WHATEVER MEASURES NECESSARY TO CORRECT THE INTERFERENCE.

First Edition

SAFETY SUMMARY
SAFETY DEPENDS ON YOU

The following general safety precautions must be observed during all phases of operation, service, and repair of this equipment. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment. Motorola Inc. assumes no liability for the customer's failure to comply with these requirements. The safety precautions listed below represent warnings of certain dangers of which we are aware. You, as the user of the product, should follow these warnings and all other safety precautions necessary for the safe operation of the equipment in your operating environment.

GROUND THE INSTRUMENT.

To minimize shock hazard, the equipment chassis and enclosure must be connected to an electrical ground. The equipment is supplied with a three-conductor ac power cable. The power cable must either be plugged into an approved three-contact electrical outlet or used with a three-contact to two-contact adapter, with the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet. The power jack and mating plug of the power cable meet International Electrotechnical Commission (IEC) safety standards.

DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE.

Do not operate the equipment in the presence of flammable gases or fumes. Operation of any electrical equipment in such an environment constitutes a definite safety hazard.

KEEP AWAY FROM LIVE CIRCUITS.

Operating personnel must not remove equipment covers. Only Factory Authorized Service Personnel or other qualified maintenance personnel may remove equipment covers for internal subassembly or component replacement or any internal adjustment. Do not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, always disconnect power and discharge circuits before touching them.

DO NOT SERVICE OR ADJUST ALONE.

Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

USE CAUTION WHEN EXPOSING OR HANDLING THE CRT.

Breakage of the Cathode-Ray Tube (CRT) causes a high-velocity scattering of glass fragments (implosion). To prevent CRT implosion, avoid rough handling or jarring of the equipment. Handling of the CRT should be done only by qualified maintenance personnel using approved safety mask and gloves.

DO NOT SUBSTITUTE PARTS OR MODIFY EQUIPMENT.

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification of the equipment. Contact Motorola Microsystems Warranty and Repair for service and repair to ensure that safety features are maintained.

DANGEROUS PROCEDURE WARNINGS.

Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed. You should also employ all other safety precautions which you deem necessary for the operation of the equipment in your operating environment.

WARNING

Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting.

PREFACE

Unless otherwise specified, all address references are in hexadecimal throughout this manual.

An asterisk (*) following the signal name for signals which are level significant denotes that the signal is true or valid when the signal is low.

An asterisk (*) following the signal name for signals which are edge significant denotes that the actions initiated by that signal occur on a high to low transition.

Signal names in parentheses denote internal module (onboard) signals.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1

CHAPTER 1

GENERAL INFORMATION

1.1 INTRODUCTION

This manual describes the Motorola MVME320 Disk Controller Module. The MVME320 interfaces to industry-standard Winchester disk drives (such as the Seagate ST506 and Shugart SA1000) and floppy disk drives (5 1/4-inch and 8-inch). The manual includes a general description, specifications, hardware preparation and installation instructions, and a functional description. A typical module is shown in Figure 1-1.

1.2 FEATURES

The features of the MVME320 include:

- Combined control for hard and floppy disk drives
- Winchester disk interface (SA1000 and ST506)
- Supports MFM and FM recording
- Soft sector
- Cylinder number: 8- or 16-bit
- Programmable hard-disk format
- Standard IBM formats
- Multiple-sector read/write
- Implied seek
- Error correction up to 11 bits
- 32-bit ECC code
- Serial data rates up to:
  5 megabits/sec (standard)
  8 megabits/sec (optional)
- Self diagnostics
- Automatic bad-sector handling
- Single-phase read and write clocks
- 8-bit register transfer, 16-bit DMA data transfer
- High-level commands
- 24-bit host address and 6-bit address modifier
- VMEbus-compatible

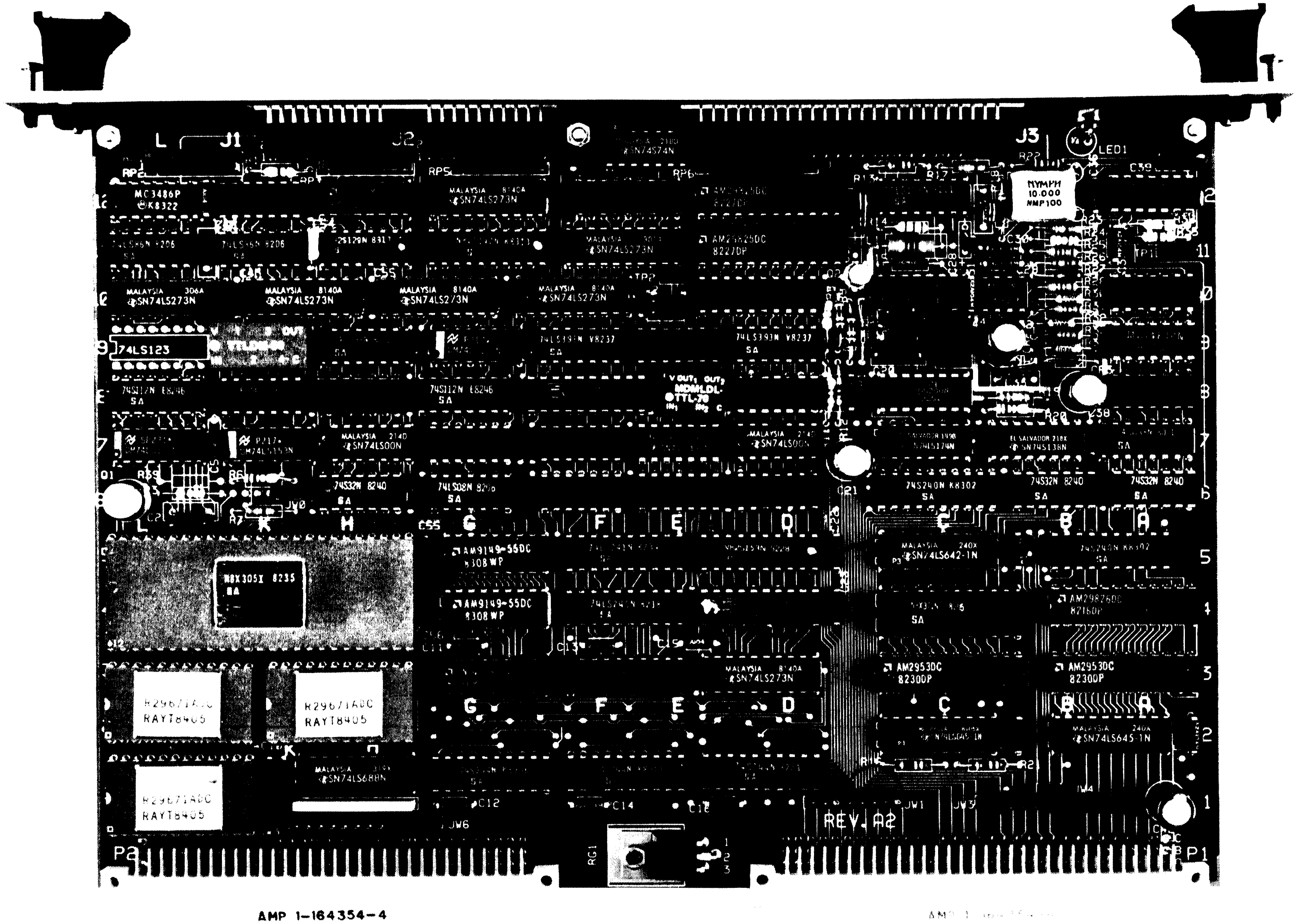


FIGURE 1-1. MVME320 Disk Controller Module

1.3 SPECIFICATIONS

The MVME320 is a VMEbus-compatible module. The specifications (which are subject to change without notice) are given in Table 1-1.

Table 1-1. MVME320 Specifications

CHARACTERISTIC	SPECIFICATIONS
Physical characteristics	
Height	5.4 inches (137 mm)
Width	10.0 inches (254 mm)
Thickness	0.5 inches (12.7 mm)
Power requirements	2.6A typical (3 A max.) at +5 Vdc 20 mA typical (30 mA max.) at +12 Vdc/-12 Vdc
Operating temperature	0° C to 50° C ambient max.
Storage temperature	-40° C to 85° C ambient max.
Relative humidity	0% to 90% (noncondensing)
Altitude	0 ft. to 10,000 ft.
Serial data rate	5 megabits/second max.
Configuration	
DTB master	Slots A24, D16
DTB slave	Slots A24, D8

1.4 GENERAL DESCRIPTION

The MVME320 is a dual-height module that provides the traditional and advanced features required to control Winchester-type hard disk drives and floppy disk drives. It can control up to four disk drives (up to two hard disk drives or up to four floppy disk drives) in many combinations.

The MVME320 supports single- and double-density (FM and MFM) recordings on floppy disk drives, and double-density (MFM) recording on hard disk drives.

- IBM 3740 single-density format (FM)
- IBM System 34 double-density format (MFM)
- Hard disk format

The controller is programmable using high-level commands over the VMEbus. The Direct Memory Access (DMA) data transfer on the host bus is 16-bit in parallel at one megabyte/second. The MVME320 is capable of handling serial data rates up to five megabits/second over the disk drive interface.

The disk interface consists of two sections:

. Input and output ports that sense and generate "slow changing" or static control signals.

. Serial data Input/Output (I/O) and high-speed disk control.

Because both sections are controlled by a fast internal microprocessor (8X305), the MVME320 is able to control a variety of different disk types and media formats.

The host system communicates with the MVME320 through Event Control Areas (ECA's) which reside in system memory. An ECA parameter block is set up for each disk drive (up to four) to be controlled. These areas contain information required by the MVME320 to execute disk commands. This information includes data about the requested command and the disk drive. DMA operations are defined by the ECA description of the command.

The MVME320 contains seven internal 8-bit registers. The host system requests an operation using these registers. Registers 1, 3, 5, and 7 are loaded with the 24-bit address of the pointer to the ECA parameter block to be acted upon. Command execution does not start until drive availability is determined. The MVME320 provides a global semaphore that can be used for this purpose. After a command is accepted for execution, no further interaction between the host processor and the MVME320 need occur until the command is completed. See Figure 4-1 for the register formats.

The MVME320 microprogram uses the data contained in the ECA block to generate disk interface signals and perform the requested drive I/O. Data transfers to and from floppy disks are executed in real-time (with 2-byte buffering). Data transfers to and from Winchester drives are buffered on a sector basis.

Prior to informing the host of a command completion, the MVME320 writes the return status information to the appropriate ECA block memory. Then the host is informed of completion of a command by an interrupt request signal from the MVME320. On interrupt service, the host reads the interrupt vector number from the internal vector number register on the data bus.

1.5 APPLICATIONS

Examples of MVME320-compatible drives:

- 8-inch hard disk drives:

  Shugart SA1000 series
  Quantum
  Data Peripherals

- 5 1/4-inch hard disk drives:

  Seagate ST506, ST512, ST406, ST412
  Shugart SA600
  RMS 500, 506, 512
  Tandon
  International Memories Inc

- 8-inch floppy disks:

  Shugart SA800, SA801, SA810, SA850

- 5 1/4-inch disks:

  Shugart SA400, SA410, SA450

1.6 REFERENCE MANUALS

The following manuals are applicable to MVME320.

MVMEBS	VMEbus Specification Manual
MVME702	MVME702 Disk Interface Module User's Manual

CHAPTER 2

HARDWARE PREPARATION AND INSTALLATION INSTRUCTIONS

2.1 INTRODUCTION

This chapter provides unpacking, hardware preparation, and installation instructions for the MVME320.

2.2 UNPACKING INSTRUCTIONS

NOTE

If shipping carton is damaged upon receipt, request carrier's agent be present during unpacking and inspection of module(s).

Unpack the MVME320 from its shipping carton. Refer to packing list and verify that all items are present. Save packing material for storing and reshipping of the module.

CAUTION

AVOID TOUCHING AREAS OF INTEGRATED CIRCUITRY; STATIC DISCHARGE CAN DAMAGE CIRCUITS.

2.3 HARDWARE PREPARATION

The MVME320 contains the following jumper-selectable configuration options. The as-shipped factory jumper header configurations are listed in Table 2-1. Jumper header locations are shown in Figure 2-1.

2.3.1 PROM Size (JW0)

Permanently installed at the factory, this jumper is used to select either 2K x 8 or 4K x 8 PROM's for control store. Typically, 4K x 8 PROM's are used.

Location K6

JW0	o	o	o	o	o—	—o	Default = 4K x 8 PROM's
	1	2	3	4	5	6	

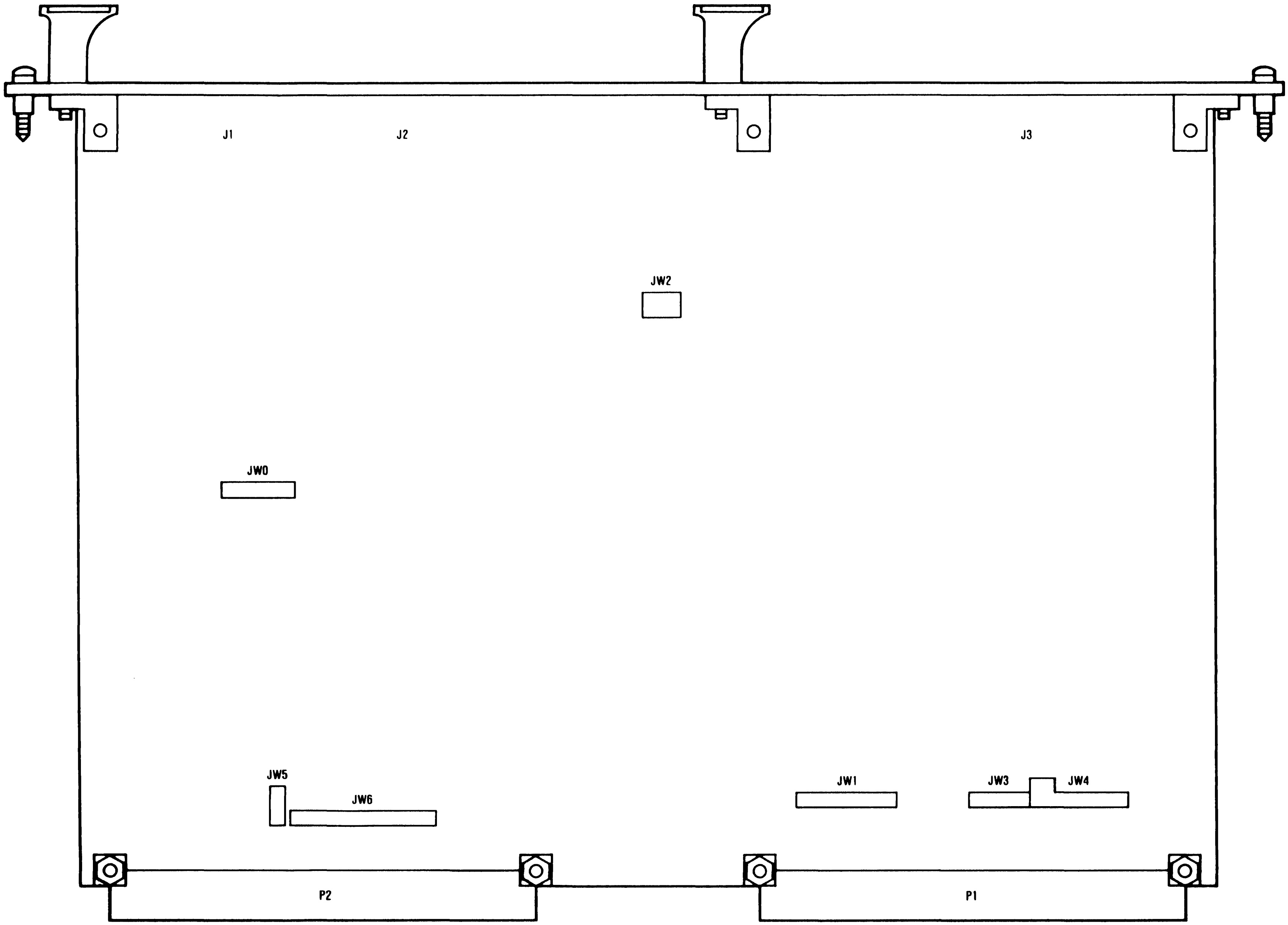


FIGURE 2-1. MVME320 Jumper Header Locations

TABLE 2-1. As-Shipped Factory Jumper Header Configurations

JUMPER HEADER	INITIAL FACTORY JUMPER PLACEMENT
JW0	5-6
JW1	3-8
JW2	2-5, 3-6
JW3	1-2
JW4	1-9, 2-10, 3-4, 5-6, 7-8
JW5	2-3
JW6	1-2, 3-4, 9-10

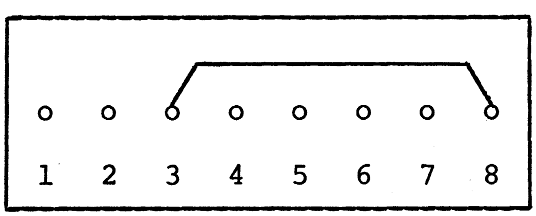
2.3.2 Interrupt Level (JW1 and JW2)

Interrupt level selection is accomplished at jumpers JW1 (IRQ) and JW2 (IACK).

Interrupt Level		JW1 (IRQ)	JW2 (IACK)	LEVEL	PINS 1-4	PINS 2-5	PINS 3-6
(highest)	6	8-6	(highest)	6	ON	ON	OFF
	5	8-5		5	ON	OFF	ON
	4	8-4		4	ON	OFF	OFF
(default)	3	8-3	(default)	3	OFF	ON	ON
	2	8-2		2	OFF	ON	OFF
(lowest)	1	8-1	(lowest)	1	OFF	OFF	ON

Location D1

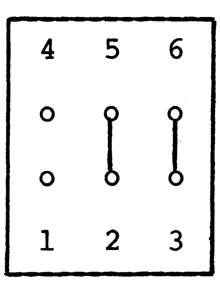
JW1



Default =
IRQ Level 3

Location E10

JW2



Default = IACK Level 3

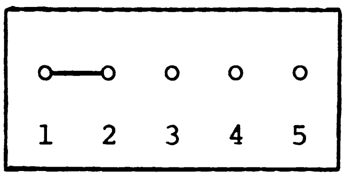
2.3.3 Bus Arbitration Level (JW3 and JW4)

Bus arbitration priority level is selected at jumpers JW3 (BUS REQUEST) and JW4 (BUS GRANT).

Bus Priority Level		JW3 (BUS REQ)	JW4 (BUS GRANT)		
(highest)	3	1-2	10-2	9-1	(DEFAULT)
	2	1-3	10-4	9-3	
	1	1-4	10-6	9-5	
(lowest)	0	1-5	10-8	9-7	

Location C1

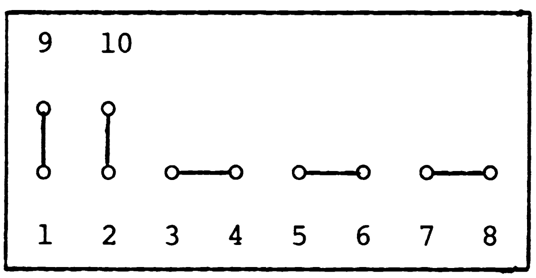
JW3



Default = BUS REQ Level 3

Location B1

JW4



Default =
BUS GRANT
Level 3

<u>NOTE</u>

Placing the module in a slot closer to slot 1 will result in a higher priority within a given priority level.

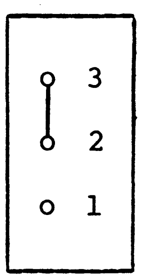
2.3.4 Address Modifier (JW5)

The address modifier is selected at jumper JW5.

To Select Address Modifier	Indicating	Jumper Pins	
(29 or 2D) hex	User or Supervisor	2-3	(DEFAULT)
only 2D hex	Supervisor only	1-2	

Location K1

JW5



Default = (29 or 2D) hex
(User or Supervisor access)

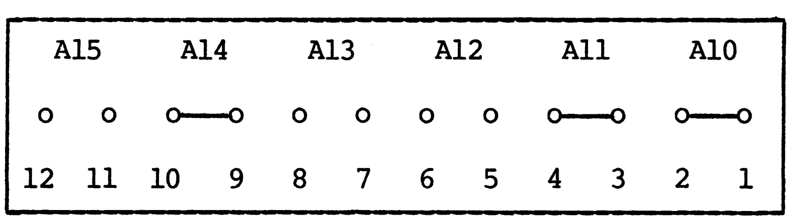
2.3.5 Memory Mapping (JW6)

Memory mapping in 1K-byte pages is done at jumper JW6, where address bits are set by: jumper inserted = 0, jumper deleted = 1.

Address Bit	Pins
10	1- 2
11	3- 4
12	5- 6
13	7- 8
14	9-10
15	11-12

Location H1

JW6



Default = 101100

2.4 INSTALLATION CONSIDERATIONS

This manual should be read and understood prior to installing the MVME320.

2

WARNING

WHEN HANDLING OR TRANSPORTING THE MVME320, CARE MUST BE TAKEN TO AVOID STATIC DISCHARGE DAMAGE TO THE INTEGRATED CIRCUITS ON THE MODULE. CONDUCTIVE MATERIALS SUCH AS ALUMINUM FOIL OR SPECIALLY TREATED FOAM IN CONTACT WITH THE SOLDER-SIDE SURFACE OF THE MODULE WILL BE ADEQUATE PROTECTION AGAINST SUCH POTENTIAL DAMAGE.

a. Unpack the module and inspect it for signs of damage.

b. Be sure that socketed IC's are fully seated in their sockets.

c. Make sure that adequate cooling is provided so that an ambient temperature of 50°C will not be exceeded.

d. Ensure that the specifications outlined in paragraph 1.3 of this manual can be handled by the user's system.

e. Provide adequate grounding. It is important that the chassis ground is interconnected (e.g., drives, enclosure), and wired to the signal ground in one place only, typically at the power supply.

f. Plug in the MVME320 making sure that it is fully seated in its connector.

CAUTION

THE MVME320 SHOULD NEVER BE REMOVED OR INSERTED WHILE POWER IS ON OR DAMAGE TO THIS UNIT MAY RESULT.

2.5 CABLING

System configurations, floppy disk cabling, and examples of termination on 5 1/4-inch hard disks is presented in the following sections.

2.5.1 System Configurations

A maximum of four drives can be attached to the MVME320 at one time; they may be a mixture of hard and floppy drives, with certain restrictions.

All drives connected to the MVME320 must be soft-sectored.

The MVME320 supports single- and double-sided (single- and double-density) disks.

Up to two hard disks may be attached to MVME320 and they must both be either 8-inch or 5 1/4-inch drives. Crystal Y1 must be 10 MHz (shipped as standard) for 5 1/4-inch hard disks, or 8.64 MHz for 8-inch hard disks.

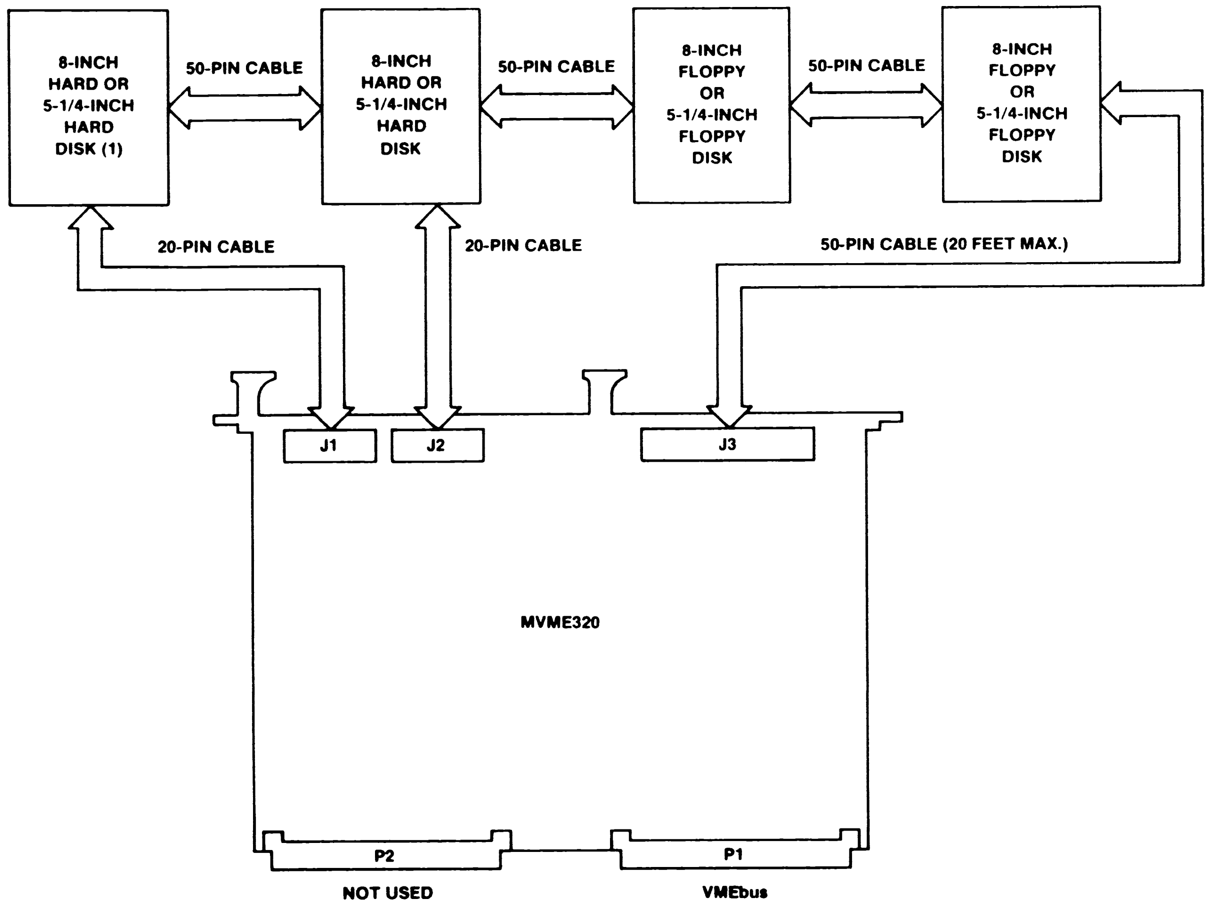
As shown below, connector J3 is the 50-pin header-type that connects all drives in a daisy-chain configuration. This connector carries control and data information for the floppy disk drives and control information only for the hard disk drives. Maximum cable length should not exceed 10 feet (3m). If using the MVME320 with the MVME702, refer to the MVME702 user's manual for cable interface.

If using double-sided floppy drives, isolate the "two-sided" status bit from the cable. Refer to the user's manual covering the drive.

When using the MVME320 without the MVME702, pins 4 and 8 on the 5 1/4-inch floppy disk interface need to be altered to move the signal from pin 4 to pin 8. This may be accomplished with a cable modification or by modifying the disk drive. Note that these modifications are unnecessary when using the MVME702.

Connectors J1 and J2 are 20-pin header-type connectors used to radially connect data lines from the hard disks to the controller. Maximum cable length should not exceed 20 feet (6 m). Cables for connectors J1, J2, and J3 should be kept as short as possible.

See Figure 2-1 to connect hard and floppy disks to the MVME320.



NOTES:

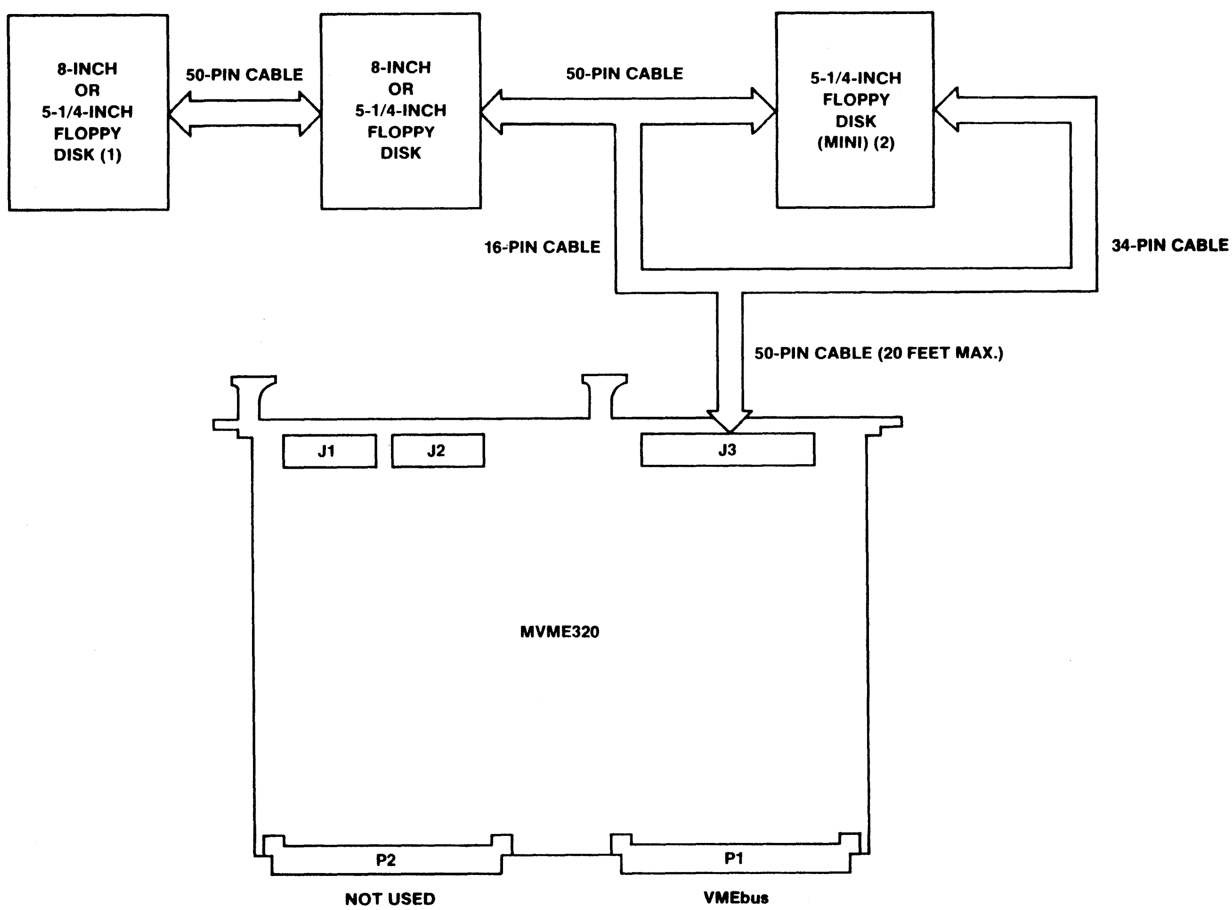
(1) Only the end of each signal line (typically on the last drive) requires termination.

2. In the above illustration, one or two floppies and one or two hard disks may be attached (daisy-chained) (e.g., two SA1000 and two SA850 disk drives). The 5 1/4-inch hard disk uses the first 34 lines (1-34) of the 50-pin cable, while the 5 1/4-inch floppy disk uses the last 34 lines (17-50).

3. The MVME702 is a disk interface module that performs signal switching and provides standard disk cable connections for different drives. When using the MVME702, refer to the MVME702 user's manual for cable configuration.

FIGURE 2-1. Hard and Floppy Disk Drive Connections to the MVME320

See Figure 2-2 to connect floppy disks only. (Refer to paragraph 2.5.2 for the actual cabling information.)



NOTES:

(1) Only the end of each signal line requires termination. For systems mixing 5 1/4-inch and 8-inch drives, this can be accomplished by using an 8-inch drive as the last drive installed and terminating all signal lines on that drive.

(2) For mini-floppy disk (pins 17-50), the cable will have a 34-pin connector spliced into the upper side of the 50-pin cable.

3. Pin 50 of the 50-pin cable connects to pin 34 at the 5 1/4-inch floppy edge connector. The 5 1/4-inch floppy uses only the upper 34 pins of the 50-pin cable (pins 17-50).

4. The MVME702 is a disk interface module that performs signal switching and provides standard disk cable connections for different drives. When using the MVME702, refer to the MVME702 user's manual for cable configuration.

FIGURE 2-2. Connections to Floppy Disks Only

2.5.2 Floppy Disk Cabling

<u>NOTE</u>

Disregard this section if using the MVME702 Disk Interface Module.

To avoid delays caused by head-settling times or motor-on delay periods, the MVME320 uses the signal line 50 to radially load heads on all 8-inch floppies and turn the motor-on for all 5 1/4-inch floppy disk drives every time an access is made on any of the floppy drives. This increases throughput substantially during copying between floppy drives.

Therefore, it is necessary to modify either the radial cable (J3) or the drives as shown in Figure 2-3.

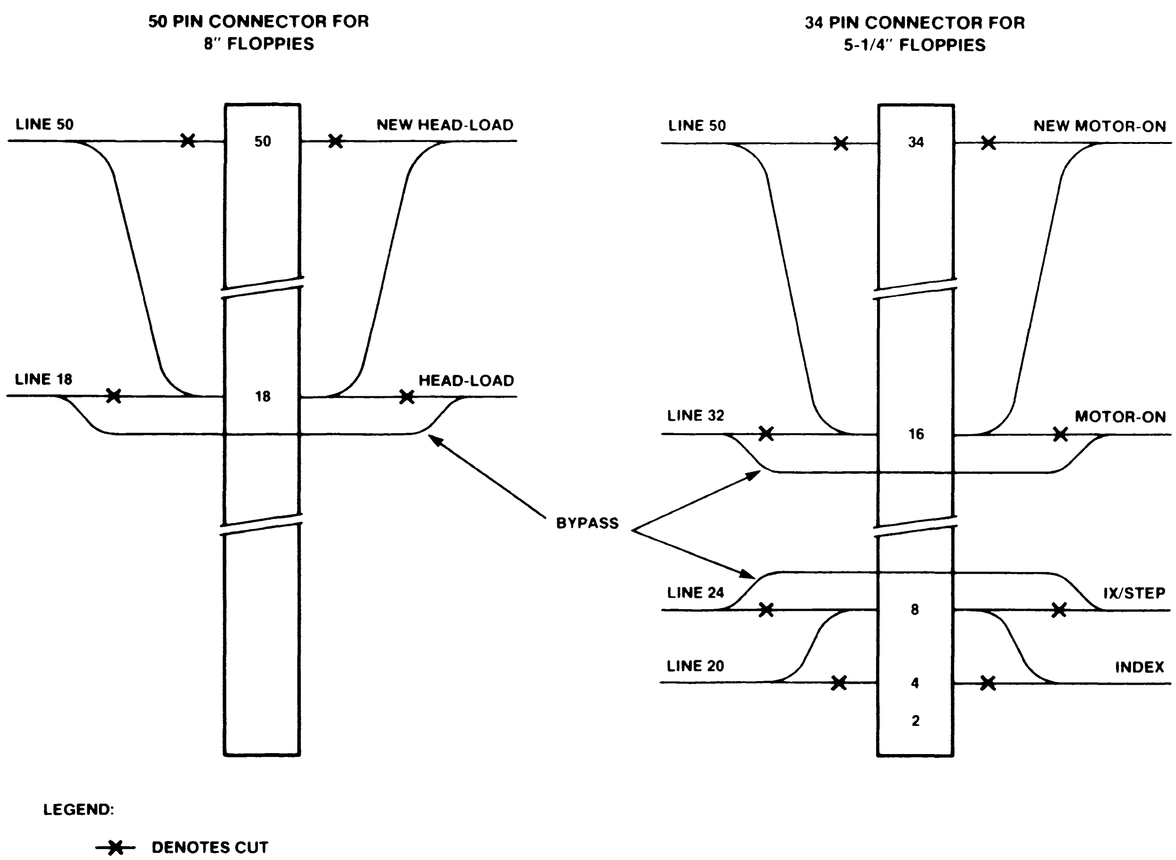


FIGURE 2-3. Floppy Disk Cabling

2.5.3 Examples of Termination on 5 1/4-inch Hard Disk

When using a 5 1/4-inch disk drive in configurations including 8-inch floppy disks on the hard disk, disconnect terminators on the following signals:

DIRECTION   HEAD 2^0 HEAD 2^1

by lifting pins on the terminator resistor network IC on the hard disk drive.

2

	Delete Pins
IMI 50xx....................	9, 10, 12
ST4xx........................	2, 3, 11
ST506........................	8, 9, 11
MINISCRIBE II (Mod#2006)....	1, 11, 12

Always consult the disk manufacturer's manual, and compare signal positions on connector J1. Each signal should be terminated only once at the physical end of the line.

If a good description or schematic is not available, use a signal-continuity meter to find the corresponding terminator pin for each connector J1 signal pin.

CHAPTER 3

FUNCTIONAL DESCRIPTION

3.1 INTRODUCTION

This chapter presents more detailed information about the MVME320 logic operation. Figure 3-1 shows a simplified functional block diagram of the MVME320.

The MVME320 logic is functionally divided as follows:

a. 8X305 MICROPROCESSOR -- executes the microprogram (in the 4K x 24 ROM) at 250-nsec instruction time. The MVME320 uses main SYSCLOCK 16 MHz divided by 2. The 8X305 processes commands and data from the host via the VMEbus, and controls the disk that will perform the user-selected operation. It also returns the result and status bytes to the host in the Event Control Area (ECA) table after the operation is complete.

b. MICROPROGRAM MEMORY -- The 4K x 24 ROM memory which stores the control microprogram that is executed by the 8X305 microprocessor.

c. DISK I/O PORTS -- Controls slow disk operations (e.g., stepping) as directed by the 8X305.

d. VMEbus I/O PORTS -- Interfaces with the host system using VMEbus I/O operation. This block has an I/O address comparator, control signal receivers, 8-bit data receivers/drivers and control logic.

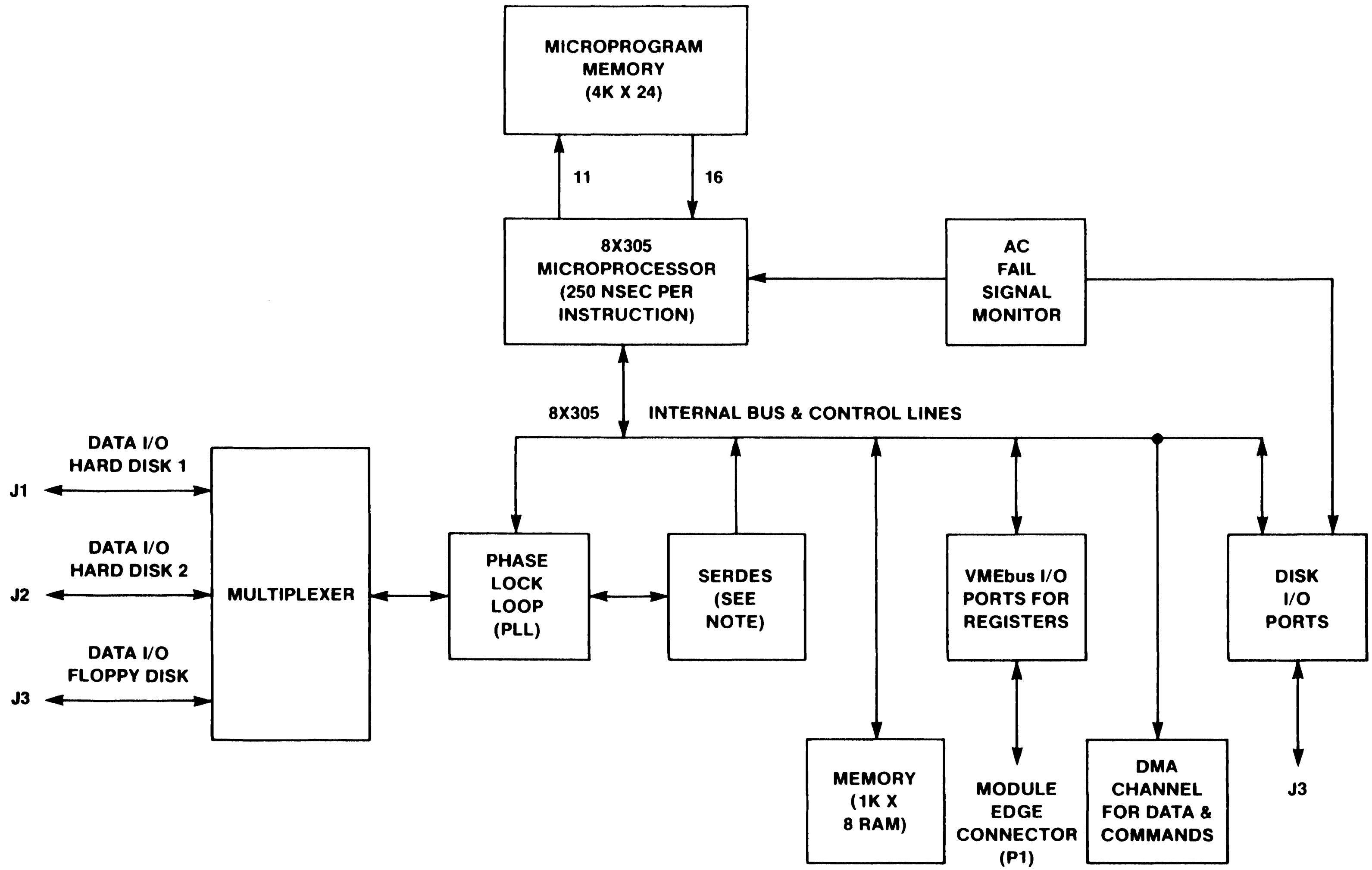
e. BUFFER MEMORY (1K x 8 RAM) -- Used both as a disk sector buffer and as scratchpad memory for MVME320 internal operations.

f. SERDES (i.e., SERIALIZER/DESERIALIZER) -- Disk data transfers are performed serially. The SERDES serializes data to be written to the disks and deserializes data received from the disks. The SERDES generates and checks CRC code. The SERDES also contains an error recovery or "watchdog" circuit that monitors for timely completion of various disk operations effected by the MVME320. If the specified disk operation was unsuccessful within the expected time, the MVME320 repeats the same operation up to the specified retry number times.

g. PHASE LOCK LOOP (PLL) -- The PLL circuit is a multifrequency data separator. The use of PLL assures reliable data recovery from the disk and accurate data writing to the disk.

h. MULTIPLEXER -- This circuit multiplexes I/O data from the disk units attached to the MVME320 to the PLL circuit.

i. AC FAIL SIGNAL MONITOR -- The microprogram monitors this signal using a sampling period <200 usec. Upon power-fail detection, all active disk drives are immediately deselected to protect drives, disk medium, and data.



NOTE: Serializer/Deserializer, with watchdog circuits.

FIGURE 3-1. MVME320 Block Diagram

CHAPTER 4

REGISTERS

4.1 INTRODUCTION

The MVME320 is a Direct Memory Access (DMA) transfer oriented device; only minimal control information needs to be transferred between the I/O and the seven internal registers. Figure 4-1 shows the format of the internal registers; each is 8-bits wide. The addresses for the internal registers are shown in Figure 4-2. The registers reside on the low-order data bus (D0-D7); therefore, their effective address is always an odd value ranging from 1 to D (hex). Each register is referred to by its address (e.g., register D). The contents of the internal registers control the command execution. The following paragraphs describe the contents and format of each register.

	bit 7							bit 0	
1	LEAST SIGNIFICANT BYTE OF THE ADDRESS OF ECA POINTERS								
3	SECOND LEAST SIGNIFICANT BYTE OF THE ADDRESS OF ECA POINTERS								
5	MIDDLE SIGNIFICANT BYTE OF THE ADDRESS OF ECA POINTERS								
7	MOST SIGNIFICANT BYTE OF THE ADDRESS OF ECA POINTERS UNUSED (0 DUE TO 24-BIT ADDRESS LIMIT)								
9	INTERRUPT VECTOR (SUPPLIED TO HOST ON INTERRUPT ACKNOWLEDGE)								
B	DR4 SRC	DR3 SRC	DR2 SRC	DR1 SRC	N/D	N/D	N/D	N/D	Interrupt Source Status
D	DR4 BUSY	DR3 BUSY	DR2 BUSY	DR1 BUSY	N/D	N/D	N/D	8/16 BIT (1)	Drive Status/Configuration
F									

NOTE: (1) Ignored.

Figure 4-1. Internal Registers

Register	A3	A2	A1	DS1*	DS0*
1	0	0	0	1	0
3	0	0	1	1	0
5	0	1	0	1	0
7	0	1	1	1	0
9	1	0	0	1	0
B	1	0	1	1	0
D	1	1	0	1	0

4

Figure 4-2. Internal Register Address

4.2 EVENT CONTROL AREA REGISTERS (REGISTERS 1-7)

These registers are read/write registers containing the 24-bit address of the Event Control Area (ECA) pointers. The least significant bit and the most significant byte of the address are not used by the MVME320. These registers are cleared by RESET to 00. The host subsequently initializes these registers by writing a pointer value to them before issuing a command. This indicates to the MVME320 the location of the ECA pointer table in the system memory. The ECA pointer table consists of four long-word addresses that indicate the location of the four ECA control blocks. The addresses are arranged in ascending order by drive number. Register 7 contains the most significant byte of the address and register 1 the least significant byte.

NOTE

Since the least significant address bit is not used, the effective address for the pointer table in registers 1-7 has to be the actual address divided by 2 (shifted 1 bit to the right).

Example: pointer table address: 00C1234
address in registers: 006091A

The host system must not modify ECA information after the command is issued and before the corresponding interrupt is received.

The ECA pointer table can be represented as:

31   0
ECA0
ECA1
ECA2
ECA3

4.3 INTERRUPT VECTOR REGISTER (REGISTER 9)

The contents of this register is presented to the host system during the interrupt-acknowledge cycle. The register is initialized to the value 0FH by a reset of the MVME320. This register contains the global interrupt vector assigned by the host system.

4

4.4 INTERRUPT SOURCE REGISTER (REGISTER B)

The four most significant bits of this register, one for each drive (Dr1-4), are used to indicate which drive is the source of an interrupt.

The reset signal initializes all four bits to 0, indicating that all four drives are inactive and none of them has initiated an interrupt.

When the MVME320 interrupts the host, it indicates which drive has caused the interrupt by setting the interrupt source bit in this register. When the host services the interrupt, it resets the interrupt bit by resetting the corresponding busy bit(s) in register D.

4.5 DRIVE STATUS AND CONFIGURATION REGISTER (REGISTER D)

The host can activate a drive by setting the corresponding "busy" bit in register D. The MVME320 responds by accessing the corresponding ECA and performing the requested action. At any time the host can abort the current drive operation by resetting the bit before the operation is completed. This initiates the normal MVME320 interrupt sequence. The MVME320 returns a status in the ECA to indicate that the command has been aborted.

The reset signal initializes all bits to 0, indicating that all four drives are inactive.

CHAPTER 5

EVENT CONTROL AREA

5.1 INTRODUCTION

The Event Control Area (ECA) is a dedicated block of data stored in the system memory. It consists of a fixed-length mailbox area for controlling the exchange of information between the MVME320 and the host system. The MVME320 requires one ECA for each drive connected. Figure 5-1 illustrates the ECA block format. Appendix C provides sample ECA block tables.

5.2 ECA FIELDS

ECA fields are listed and described in the following paragraphs.

5.2.1 Command Code (1 byte)

The command code indicates the command to be executed (refer to Chapter 6, Commands).

5

5.2.2 Main Status (1 byte)

This field contains general value-oriented status information about the command execution:

Value (decimal)	Indicates
0	Correct execution without error
1	Nonrecoverable error which cannot be completed (auto retries were attempted)
2	Drive not ready
3	Reserved
4	Sector address out of range
5	Throughput error (floppy data overrun)
6	Command rejected (illegal command)
7	Busy (controller busy)
8	Drive not available (head out of range)
9	DMA operation cannot be completed (VMEbus error)
10	Command abort (reset busy)
11-255	Not Used

15	0	
COMMAND CODE	MAIN STATUS	
EXTENDED STATUS		
MAXIMUM NUMBER OF RETRIES	ACTUAL NUMBER OF RETRIES	
DMA TYPE	COMMAND OPTIONS	
BUFFER ADDRESS MOST SIGNIFICANT WORD		
BUFFER ADDRESS LEAST SIGNIFICANT WORD		
BUFFER LENGTH REQUESTED		
ACTUAL NUMBER OF BYTES TRANSFERRED		
CYLINDER NUMBER		(1)
HEAD OR SURFACE NUMBER	SECTOR NUMBER	(1)
CURRENT CYLINDER POSITION		
RESERVED (5 WORDS)		
N0 -- PRE-INDEX GAP	N1 -- POST-INDEX GAP	(2)
N2 -- SYNC BYTE COUNT	N3 -- POST-ID GAP	(2)
N4 -- POST-DATA GAP	N5 -- ADDRESS MARK COUNT	(2)
SECTOR LENGTH CODE	FILL BYTE	(2)
RESERVED (3 WORDS)		
DRIVE TYPE	NUMBER OF SURFACES	
NUMBER OF SECTORS/TRACK	STEPPING RATE	
HEAD SETTLING TIME	HEAD LOAD TIME	

FIGURE 5-1. ECA Block Format

15 — 8	7 — 0	
SEEK TYPE	RESERVED FOR CONTROLLER	
LOW WRITE CURRENT BOUNDARY TRACK		
PRECOMPENSATION BOUNDARY TRACK		
ECC REMAINDER (3 WORDS)		(3)
APPEND ECC REMAINDER FROM DISK (3 WORDS)		(3)
RESERVED (2 WORDS)		
MVME320 WORKING AREA (6 WORDS)		
RESERVED FOR THE CONTROLLER		

NOTES:

(1) Physical starting sector number
(2) Track format fields (refer to Chapter 8)
(3) Last word (don't care)

FIGURE 5-1. ECA Block Format (cont'd)

5

5.2.3 Extended Status (2 bytes)

This field contains specific bit-oriented status information about the command execution. If errors result during execution, the corresponding bit will be ORed into extended status and kept there. This OR function is used by the host system for error logging. The extended status will automatically be reset to zero by the MVME320 at the beginning of command execution.

The error definitions are:

Bit 0	Write fault
Bit 1	CRC/ECC error on data or ID
Bit 2	Data overrun
Bit 3	No identifier found
Bit 4	Not ready
Bit 5	Deleted data address mark
Bit 6	Write on write-protected diskette
Bit 7	Positioning error
Bit 8	Data port timeout
Bit 9	Disk format error
Bit 10	Uncorrectable data error (ECC)
Bit 11	Command stop or drive not available
Bit 12	Drive type rejected
Bit 13	Positioning timeout
Bit 14	Wrong ID-DATA-ID sequence during read track command
Bit 15	Bus error fault

5.2.4 Maximum Number of Retries (1 byte)

This parameter specifies the maximum number of retries per command (not per sector) that the MVME320 should attempt after a disk operation error. The value of zero indicates that no retries will be attempted.

5.2.5 Actual Number of Retries (1 byte)

This byte is set by the MVME320 to the actual number of retries executed per command. All retries are accumulated by the MVME320 on a sector-by-sector basis.

5.2.6 DMA Type (1 byte)

This 1 byte field should be set to a zero value. This value specifies the Direct Memory Access (DMA) transfer mode of releasing the bus between word transfers. No other options are implemented at this time.

5.2.7 Command Options (1 byte)

This byte contains options that apply to the current command. The option selections consist of the following:

Bit	Option
0	Handling of deleted data address mark
1	Reserved (should always = 0)

The options for handling deleted data address marks are described in the table below. Sectors with the deleted data address mark (bits 0, 1 of the option field) will be handled in the following manner for read operations:

Bit	Option
X00	The sectors with deleted data address mark will be skipped as if it did not exist. A successful CRC/ECC check is not required.
X01	The data of the sector with deleted data address mark will be transferred to the host system regardless of the CRC/ECC check result, and the operation will be terminated.
2	Automatic Error Correction (after all retries failed).
1XX	Automatic error correction enabled. This option corrects data in the MVME320 internal memory. It uses the ECA remainder field of the ECA. Besides the correct data, the MVME320 will write the error correction vector and its relative position from the end of the data buffer into the ECA working area.
3-7	Not Used.

5.2.8 Buffer Address (4 bytes)

This is a 32-bit starting byte buffer address for the DMA transfers. The buffer can start on an odd address. The least significant byte of the address field defines the even or odd byte address of the 16-bit words when in 16-bit mode. The eight most significant bits should be set to zero.

5.2.9 Buffer Length Requested (2 bytes)

This field contains the requested number of bytes to be transferred. This field implicitly defines the number of multiple-sector transfers to be performed. A seek occurs only when a zero length is specified.

5.2.10 Actual Number of Bytes Transferred (2 bytes)

This field indicates how many bytes have actually been read or written during a command execution. The value can be used, for example, for a location of a problem sector with a nonrecoverable error. The value of this field can be changed by the MVME320 during execution of a command and used as an intermediate field.

5

5.2.11 Physical Starting Sector Number (4 bytes)

This field contains the physical disk location at which the command is to begin execution. The format of the field is:

Cylinder High	Cylinder Low
Head/Surface Number	Sector Number

5.2.12 Current Cylinder (2 bytes)

These two bytes will be updated by the MVME320 after each positioning. During the first positioning after reset, the MVME320 will execute an automatic recalibration. If the current cylinder = 0, prior to the execution of the next command, the MVME320 will check the track 0 signals. If neither of the two track signals is active, the MVME320 will execute a recalibration procedure and then start command execution.

5.2.13 Reserved (10 bytes)

This field is not used by the MVME320.

5.2.14 Disk Track Format Fields (8 bytes)

The disk track format fields are described in Chapter 8, Disk Track Format.

5.2.15 Reserved (6 bytes)

This field is not used by the MVME320.

5.2.16 Disk Control Fields (40 bytes)

These fields describe the drive interface to the MVME320.

a. Drive Type (1 byte) - The most significant bit of this byte designates hard or soft sectoring (a set bit indicates hard sectoring). The seven least significant bits of this byte specify the disk type.

Value	Indicates
0	IBM single-density format, 8-inch floppy disk
1	IBM double-density format, 8-inch floppy disk, or high-density, 5 1/4-inch Amlyn cartridge-floppy
2	Hard-disk format with ECC
3	Hard-disk format with CRC
4	Single-density, 5 1/4-inch floppy drive
5	Double-density, 5 1/4-inch floppy drive

b. Number of Surfaces (1 byte) - The number of heads (surfaces) is used by the MVME320 for calculating the increment sector address.

c. Number of Sectors Per Track (1 byte) - This value is used for calculating the sector address.

d. Stepping Rate (1 byte) - This field contains the head stepping rate in 500-usec units, if applicable.

e. Head Settling Time (1 byte) - This field contains the head settling time in 500-usec units. If a nonzero head settling time is specified, the MVME320 assumes that a seek-complete is not available from the drive.

f. Head Load Time (1 byte) - This field contains the head load time in 500-usec units. For drives with a seek-complete signal such as hard disks and the Amlyn cartridge-floppy, this value must always be set to zero by the host system.

g. Seek Type (1 byte) - This field defines the type of seek positioning to be performed.

Value	Indicates
0	Normal single-step seek
1	Accelerated seek (for ST506)
2	Buffered seek (for ST412)

h. Phase Count (1 byte) - This field contains the phase counter, which is a status of the command execution set by the MVME320. For all commands, this field must be initialized to zero by the host system prior to a new command to the MVME320.

i. Low Write Current Active Track (2 bytes) - This field defines to the MVME320 the starting track at which low current write current is to be used.

j. Precompensation Active Track (2 bytes) - This field defines to the MVME320 the starting track at which precompensation is to be used.

k. ECC Remainder (6 bytes) - This field contains the calculated ECC remainder generated by the MVME320 (should be 0). Since ECC is 32-bits long, the last two bytes are "don't care".

l. Appended ECC Remainder from Disk (6 bytes) - This field contains the actual ECC remainder attached to the data on the disk when written.

m. Reserved (4 bytes) - This field is not used by the MVME320.

n. MVME320 Work Area (12 bytes) - These six words are reserved for the MVME320 microprogram. The meaning of certain bytes of this area will be determined by the command being executed.

CHAPTER 6

COMMANDS

6.1 INTRODUCTION

Execution of all commands is automatic in the MVME320; it does not require any assistance from the system. The command set includes:

Command		Function
0	CALB	Recalibrate to track zero
1	WRDD	Write deleted data
2	VER	Verify (read without data transfer)
3	TSR	Transparent sector read
5	REMS	Read multiple sector with overlapped seek
6	WRMS	Write multiple sector with overlapped seek
7	FORM	Format a track

A new command is requested by the host by writing into the Drive Status and Configuration Register (register D). The user sets a bit corresponding to the drive to be serviced. This causes the MVME320 to start execution of the command on the requested drive.

After a command is started, it is executed without further communication with the host system. All drive/memory data transfers are handled automatically by the Direct Memory Access (DMA) controller of the MVME320. The MVME320 signifies through an interrupt signal that the command is completed.

The following is a generalized description of the sequence performed by the MVME320 when executing a command:

a. The MVME320 verifies that the drive is available. If it is not, the MVME320 generates an error status and a completion interrupt.

b. The MVME320 executes a track seek if necessary. For drives which have a seek-complete signal (as indicated by a zero value in the head settling time field of the Event Control Area (ECA)), a command termination can be caused by the timeout of the Seek Complete signal drive. For drives for which the head settling time is nonzero, the MVME320 waits the specified time after issuing the stepping pulses (i.e., there is no timeout on the seeks).

c. The head will be selected or, for a floppy disk, loaded. The MVME320 will wait for four byte times to ensure that head switching has occurred.

d. The MVME320 reads the sector identifier to locate the requested sector.

   If the MVME320 is unable to read the sector identifier, the command is terminated (step g) and the status "no identifier found" is returned.

If the matched sector identifier indicates a bad sector, (refer to Chapter 8, Disk Trace Format) the MVME320 uses the replacement information to again perform steps b, c, and d for the new sector. This bad-sector replacement is not counted as a retry.

e. After the requested sector has been located, the MVME320 performs the sector disk I/O (read or write) and DMA operations. If the operation cannot be completed, retry processing is attempted. A failure in this portion of the command is indicated by the return status "data part time out", which distinguishes it from the read identifier failure of step d. The MVME320 reads the data from the disk into an internal register first and then through the DMA to the VMEbus memory. The floppy disk read and DMA transfer are executed simultaneously.

f. If multiple sectors have been specified in the command, the MVME320 assumes contiguous physical sectors; steps b, c, d, and e are repeated for each sector. The MVME320 automatically performs any track seek required if the next physical sector is located on the next cylinder.

   In the case of bad-sector replacement, the MVME320 continues the multiple-sector operation (after replacing the bad sector) at the sector physically located after the bad sector. It performs all the track repositioning required to return the read/write head to the next physical sector.

g. Upon command completion, all relevant ECA fields are updated and the appropriate bits of the "Interrupt Source Status" and "Drive Status and Configuration" registers ($B and $D) are set, and an interrupt signal is generated by the MVME320. Upon interrupt acknowledge, the MVME320 presents the interrupt vector from the Interrupt Vector Register on the data bus.

6.2 COMMAND DESCRIPTION

In the command descriptions, the term "FIFO" refers to the MVME320 internal memory when it is used as a FIFO buffer. Command descriptions are as follows.

6.2.1 Recalibrate to Track Zero (CALB)

The function of this command is to retract the heads to track 0. The MVME320 issues one step "IN" and then steps "OUT" until signal track 0 becomes active or until the number of steps equals the maximum number of cylinders from the ECA. The check on seek completed timeout will be made for the hard disk.

6.2.2 Read Multiple Sector with Implied Seek (REMS)

After a successful seek to the desired sector, the MVME320 will start to read the data of the sector into the FIFO buffer and check the CRC or ECC code. Sector data is then transferred through the DMA interface to the host system memory. This procedure is repeated until the number of sectors, implied by the buffer length field of the ECA, have been read.

For sectors with deleted data address mark, the MVME320 will process them according to the command option field of the ECA.

6.2.3 Write Multiple Sector with Implied Seek (WRMS)

After the desired sector is located, the MVME320 will write the complete data part: preamble, address mark, flag, data, CRC or ECC, and postamble. The precise format is given by the drive type. The FIFO will be continuously filled with new data (or zeros if all buffer data from the buffer has been transferred). The number of data bytes written in one sector is determined by the MVME320 using information from the ECA fields. This operation is repeated until the number of sectors, implied by the buffer length field of the ECA, have been written.

6.2.4 Format a Track with Implied Seek (FORM)

The MVME320 supports three different media formats (refer to Chapter 8, Disk Track Format, for detailed information). This allows the user to write the format information specified in the ECA track format fields to the recording media. The buffer address fields of the ECA table point to a memory space that contains sector information for each sector in the track to be formatted. The format for each section of the buffer information is as follows:

FOR FLOPPY DISKS:

BYTE 0	BYTE 1	BYTE 2	BYTE 3

where:

BYTE 0 = CYLINDER NUMBER
BYTE 1 = HEAD NUMBER
BYTE 2 = SECTOR NUMBER
BYTE 3 = SECTOR SIZE CODE, 00 = 128 BPS, 01 = 256 BPS

FOR HARD DISKS:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4

where:

BYTE 0 = CYLINDER NUMBER (MSB)
BYTE 1 = CYLINDER NUMBER (LSB)
BYTE 2 = HEAD NUMBER
BYTE 3 = SECTOR NUMBER
BYTE 4 = SECTOR SIZE CODE, 00 = 128 BPS, 01 = 256 BPS

Since each section of sector information contains a sector number field, the user can format any desired interleave factor on the disk. This allows the user to tailor the interleave factor to optimize system performance.

6.2.5 Write with Deleted Data (WRDD)

This command differs from the normal write only by writing the Deleted Data address mark (F8).

6.2.6 Verify (VER)

Verify with implied seek functionally is the same as the read command, but the data is not transferred to the host system.

6.2.7 Transparent Sector Read (TSR)

This command will read a single sector and transfer the data to the host system buffer regardless of a CRC/ECC check. Only the extended status will be updated with all the error conditions encountered.

This command can be used for diagnostic purposes and for recovery of damaged data with some manual intervention.

If an incorrect CRC/ECC remainder occurs, the MVME320 places the remainder into the ECA table, reserved for this purpose.

CHAPTER 7

ERROR CORRECTION

7.1 INTRODUCTION

This chapter describes how the MVME320 supports error correction.

7.2 GENERAL DESCRIPTION

The hard-disk format provides the space for a four-byte error correction code in the data part of the sector. The MVME320 has a bit-serial CRC/ECC generator.

The MVME320 supports error correction in the following ways:

- Generates the remainder during the sector-write operation.
- Checks the remainder during the sector-read operation.
- Delivers all four bytes of remainder for diagnostic purposes.
- Corrects the data either directly in the memory or provides the remainder in the ECC appended area of the ECA.

7

CHAPTER 8

DISK TRACK FORMAT

8.1 INTRODUCTION

The MVME320 supports the following track formats:

- IBM 3740 single-density floppy disk
- IBM System 34 double-density floppy disk
- Hard disk format

In each of the formats, data on the physical media is separated into sectors. The format serves several purposes in this arrangement -- it defines the structure of the sector, the location of the actual data, and provides overhead bytes that allow an interval for any switching required by the drive hardware. These intervals, in turn, allow the MVME320 to compensate for any variations in either the recording media and/or the drive hardware. Although each of these formats is well defined, variations in parameter values within a given format require that the MVME320 provide the user with the capability to adjust (program) them. The purpose of this chapter is to describe the formats that are supported and how to program them with the MVME320.

Because no single standard defines track format parameters, a description of parameters and other pertinent definitions are included later in this chapter. For the definition of any format parameter, refer to paragraph 8.4. Table 8-1 provides a summary of the formats and the programmable values.

The hard disk format is a MFM format which is nearly identical to the IBM System 34 double-density format. The differences are summarized below:

IBM FLOPPY DISK	HARD DISK
Uses index address marks	No index address marks
Sector ID part has one byte of cylinder data	Uses two bytes
Sector ID part has three address marks	Uses one address mark
Sector data part has three address marks	Uses one address mark
Data field check consists of 2-byte CRC only	Uses either 2-byte CRC or 32-bit ECC

The Format Track command allows the user to write formatting information to the recording media. Specification of format parameters is accomplished by changing appropriate Event Control Area (ECA) fields. This structure gives the user the flexibility to accommodate, through changes in the ECA, variations that occur within a given format.

TABLE 8-1. Summary of Format Parameters

DESCRIPTION	IBM-COMPATIBLE FLOPPY DRIVES					HARD DISKS	
	MODULE USES	HEX DATA VALUE	FM CNT	HEX DATA VALUE	MFM CNT	HEX DATA VALUE	PGM CNT
Pre-Index Gap	N0	FF	40	4E	80	-	-
Sync Field		00	6	00	12	-	-
Index Mark		FC/D7 (1)	1	C2 (2)	3	-	-
Index Flag		-	-	FC	1	-	-
Index Gap		FF	26	4E	50	4E	20
Sync Field		00	6	00	12	00	12
ID Address Mark		FE/C7 (1)	1	A1 (3)	3	A1 (3)	1
ID Address Flag		-	-	FE	1	FE	1
Cylinder	DMA		1		1		1or2
Side	DMA		1		1		1
Sector	DMA		1		1		1
Record Length	DMA	01 (4)	1	01 (4)	1		
CRC-CCITT			2		2		2
Post-ID Gap		FF	11	4E	22	00	3
Sync Field		00	6	00	12	00	12
Data Address Mark		FB/C7 (1)	1	A1 (3)	3	A1 (3)	1
Data Address Flag		-	-	FB	1	FB	1
Data (4)		Fill Byte	256	Fill Byte	256	Fill Byte	256
CRC-CCITT			2		2	(5)	2or4
Post-Data Gap	N4	FF	27	4E	54	00/4E	15
Inter-record Gap (4)		FF	170	4E	589	4E	346

NOTES:

(1) Shows data pattern and clock pattern (clock pattern normally is FF).
(2) Shows data pattern; clock pattern should suppress clock bit between data bit 3 & 4.
(3) Shows data pattern; clock pattern should suppress clock bit between data bit 4 & 5.
(4) This example is for a 256-byte sector; others will have different values.
(5) This can be either a CRC-CITT or a 32-bit ECC field.

The MVME320 writes the sector identifier and fills the data part with "fill byte" starting with the leading edge of the first index pulse and ending with the leading edge of the next pulse.

During the format-write operation, bad sectors can be replaced; a detailed description of the sector replacement algorithm and of the track format is given later in this chapter.

The "buffer length" field of the ECA designates the end of the formatting operation.

Sector interleaving can be done by filling the buffer.

8.2 BAD SECTOR HANDLING

Sometimes defective media sectors are encountered. The cost of perfect media is impractical. Since defective media areas cannot be used for recording data, the occurrence of bad sectors must be taken into account. It is suggested that a software imlementation similar to bad sector lockout (as used on Motorola's VERSAdos) or bad sector replacement (as used on Motorola's SYSTEM V/68) be implemented.

Sector lockout is accomplished by keeping a table of known bad sectors on the disk. The sectors on this list are then avoided. This table need only be checked when accessing sectors of unknown quality (e.g., writing a new sector).

A sector replacement algorithm involves keeping a list of known bad sectors and their replacement sector on the disk. This list is checked every time a disk access is made. A copy of this list should be kept in RAM for quicker disk accesses.

8.3 ECA TRACK FORMAT FIELDS

For the three formats supported by the MVME320, eight parameters are required to specify the format of the recording media to the MVME320. The format parameters are programmed by changing the values of the appropriate ECA fields. A detailed description of each of these parameters follows. The layout of the track format portion of the ECA is:

15 — 8	7 — 4 3 — 0
N0 -- Pre-Index Gap	N1 -- Post-Index Gap
N2 -- Sync Byte Count	N3 -- Post-ID Gap
N4 -- Post-Data Gap	N5 -- Address Mark Count
Sector Length Code	Fill Byte

where:

Sector Length Code = the exponent in the equation for sector length: 128×2^n

N0, N1, N2, N3, and N5 are fields reserved for future use; the MVME320 uses constants, which are listed in Table 8-1.

N4 must be initialized in the ECA table using the values listed in Table 8-1.

8

TABLE 8-2. Track Format Definitions and Abbreviations

NAME	SUBFIELD	FORMATS	DESCRIPTION
N0	Index	IBM only	Pre-Index Gap. This gap represents the number of bytes that appear prior to the index pulse.
N1	Index	All except hard sector	Post-Index Gap. This gap represents the number of bytes that appear after the index pulse and prior to the ID subfield.
N2	ID, Data, Index	All	Preamble Count or Sync. This is the number of sync bytes that precede the address mark.
N3	ID	All	Post-ID Gap. This count is the number of bytes that separate the ID subfield from the data subfield.
N4	Data	All	Post-Data Gap. This count is the number of bytes that separate the data subfield to the beginning of the ID subfield of the next sector.
N5	ID, Data, Index	IBM	Address Mark Count. This contains the number of address marks contained by the subfields. For single-density formats the count is one; for double-density formats the count is three, except hard disk where it is also one.

8

8.4 DESCRIPTION OF FORMAT PARAMETERS

Although exhibiting some differences, the parameters that constitute each of the three formats supported by the MVME320 are basically similar. In each format the data on the diskette or disk is separated into a logical data block or sector. The sector becomes the smallest block of information that can be addressed directly by the MVME320.

The hard disk and both IBM formats organize the physical disk into a circular path or track that is separated into several sectors. In this scheme, tracks on the media are referenced with respect to a physical index mark. An index mark pulse is generated by the media to indicate the beginning of a track. By specification of a track and sector, the location of a sector on the media is uniquely addressed (for at least one side of the media). The programmable hard-sectored format also uses this track and sector structure; however, it differs from the other formats in that in addition to the index pulse, each sector is preceded by a sector pulse.

In each of the formats a sector can be further separated into two parts or subfields -- an ID and a data subfield. The ID subfield contains a unique identifier (or address) and description of the sector. The data subfield contains the actual data of the sector. Both subfields contain three types of information:

a. address mark(s).
b. data (either actual or about the sector).
c. error report.

The following definitions apply to the information contained in these subfields:

Address Mark -- A unique character which precedes the data in the subfield. For MFM formats, (non-IBM single-density) it is followed by a single character, the address mark flag, which further describes the subsequent data.

Error Report -- A pattern generated by the MVME320 that is used to verify the data transmitted.

Both fields contain character sequences called "gaps" and "syncs". These sequences are used to differentiate the sector subfields and to provide an interval that allows any hardware switching to be performed. This in turn compensates for any timing problems caused by variations in either the recording media or the disk drive.

For the programmable hard-sectored and IBM formats (which use the index pulse), a third subfield is used -- the index subfield. This subfield appears prior to the first physical sector of a track on the recording media and has subfields which contain gap and sync sequences. Unlike the other two fields, it occurs only once per track and contains only the address mark information.

Table 8-2 provides definitions and abbreviations applicable to the track format specifications provided in this chapter.

CHAPTER 9

SYSTEM OPERATION

9.1 INTRODUCTION

This chapter provides information concerning MVME320 initialization, status on reset, host initialization, and ECA field types.

9.2 MVME320 INITIALIZATION

Before the MVME320 can process command requests from the host, it must go through an intialization sequence that places it into a known condition in preparation for commands. This intialization occurs in two distinct stages. The first is started by the activation of the Reset line and the second is performed by the host system. The MVME320 status to be considered consists of the following components:

- Content of the MVME320 control registers
- State of the control and signal lines
- Status of the MVME320 microprogram

9.3 STATUS ON RESET

On activation of the Reset line, the MVME320 performs the following:

- Clears the MVME320 memory
- Resets all control registers
- Three-states all bus interface and control lines
- Clears all internal registers and counters
- Places the MVME320 microprogram into an idle loop

9.4 HOST INITIALIZATION

The second stage of initialization occurs after the reset. The host must initialize the MVME320 before it can process any command requests. The initialization information must be passed from the host to the MVME320 control registers. The information consists of the following:

- Load an interrupt vector number into the Interrupt Vector Register.

- Load the Event Control Area (ECA) registers with the start location of the ECA pointer table in system memory.

The host prepares for MVME320 operation by loading the ECA pointer table in system memory with the address of each of the ECA blocks. The disk drive to ECA block assignment is determined by the relative position of the pointer in the table. The first table entry corresponds to drive one, the second entry to drive two, etc.

9.5 ECA FIELD TYPES

ECA formats, drive control parameters, command execution parameters, and command status are described in the following paragraphs.

9.5.1 ECA Formats

The ECA parameter blocks are the main means of communication between the host system and the MVME320. The information contained in its fields serves essentially two purposes. First, it provides the information required by the MVME320 to generate the disk drive interface signals. Second, the ECA is used by the MVME320 to pass to the host information required to execute system-level disk operations. To emphasize this second purpose, consider a system operation such as "open a read file". Such an operation implies that the host system issues several commands to the MVME320 to read the disk. Implementing such a system-level command requires that, at the completion of a command, the MVME320 pass to the host current information about the drive and the command (such as the current position of the read/write head). The host uses this information to prepare the next sequence of MVME320 commands.

As an MVME320 command proceeds through its execution, it references and alters various fields of the ECA. To gain a better understanding of the interaction between the MVME320 and the various ECA fields, the ECA data can be separated into three catagories:

- Drive control parameters
- Command execution parameters
- Command status

These categories differentiate the ECA fields with respect to content, access, and alterability. The following paragraphs describe the characteristics of each category.

9.5.2 Drive Control Parameters

These fields contain information that defines the disk interface to the MVME320. This information is provided by the host system and is not altered by the MVME320. The MVME320 microprogram that controls the disk drive I/O issued will continually reference this information during execution of a command.

9

To protect against positioning errors when switching from one ECA field to another, the field "current cylinder number" cannot be initialized by the host (since the MVME320 only writes into it and does not read from it).

9.5.3 Command Execution Parameters

These parameters consist of those ECA data fields required by the MVME320 to execute a command. These fields can be further separated into two groups -- static and dynamic. The static fields are valid for the duration of a command; the MVME320 does not alter any information in these fields while the command is being processed. The dynamic fields provide the "logical storage" needed to execute multiple-sector commands. The MVME320 uses the dynamic fields of the ECA to maintain the current execution status of the online disk drives.

Both the static and dynamic fields are loaded prior to initiation of a new command request by the host. The MVME320 provides only rudimentary checks of these fields to ensure integrity of the information. The host must provide the level of safeguards desired.

The static fields need to be verified only before a new command request is submitted to the MVME320. Because the dynamic fields are continually updated by the MVME320 during command execution, they reflect the current value of these parameters at termination. The dynamic information generated by the MVME320 is used by the host to continue execution of system-level disk drive operations by providing the host with information regarding where the drive "left off". Such dynamic information requires that the host provide validation checks of the dynamic parameter fields every time a new request is made to the MVME320.

9.5.4 Command Status

These fields report to the host the results of a command operation and are continually updated after each transfer. They are cleared by the MVME320 when a new request is accepted.

9.6 SUMMARY

Communications between the MVME320 and the host are established through the ECA. After a command has been accepted by the MVME320, the host must not alter the ECA.

The host can monitor progress by polling the main status byte. The MVME320 can modify other values for internal operations.

Implicit to the execution of any disk operation is the reading of the ECA data by the MVME320 to allow loading of drive control parameters. At the conclusion of a disk operation, the MVME320 performs a write operation to the corresponding ECA parameters to store the results of the command. Both reading and writing of the ECA fields by the MVME320 use the Host/Direct Memory Access (DMA) interface to arbitrate for the system bus and to perform required handshaking.

CHAPTER 10

SCHEDULER CONCEPTS

10.1 INTRODUCTION

The primary purpose of the MVME320 in a system is to reduce the overhead processing of a host involved with controlling several disk drives. Off-loading from the host performs those tasks directly associated with disk control and permits the host processor to concentrate on its main application(s). Although the MVME320 is capable of controlling four disk drives simultaneously, it perfoms these multiple control functions sequentially. Its operation can be viewed as a resource shared by all the drives online, with each drive competing for the services of the MVME320. In addition to arbitrating this competition, the scheduler must remain responsive to new command requests.

10.2 RESOURCES

The MVME320 resources available to the disk drives and the host processor consist of:

- Host/DMA interface
- Internal memory
- Control registers
- High-speed disk controls and serial I/O interface
- Low-speed disk controls

The following is the set of basic activities in which the MVME320 can interact with the drives and/or host:

a. Read control register
b. Write control register
c. Slow disk drive read
d. Slow disk drive write
e. Read FIFO (DMA)
f. Write FIFO (DMA)
g. Read ECA data field (DMA)
h. Write ECA data field (DMA)
i. Disk drive serial read (high-speed disk control)
j. Disk drive serial write (high-speed disk control)

Because performance of items a. to h. involves the common MVME320 data bus, (D00-D15) interface, the items are executed one at a time; that is, the MVME320 cannot perform them in parallel. The MVME320 is structured such that the host/DMA and the high-speed disk control interfaces operate asynchronouly with the MVME320 processor as well as with each other. This suggests that items i. and j. can be performed concurrently with any of the other 8 activities. This is true only on a limited basis. Because high-speed disk control implicitly uses the DMA interface, items i. and j. can be executed in parallel with any of the other listed activities until a DMA is required (items e. or f.). As a result, all of these activities can be considered mutually exclusive.

10

10.3 LOGICAL OPERATIONS

Theoretically, a new activity could be scheduled after any one of the basic operations has been performed. In practice, this is not an efficient use of the MVME320. Also, certain of these combinations are not logically useful. For example, during serial disk I/O it would not make sense to allow the ECA pointer register to be altered. For these reasons, the MVME320 does not execute a command in terms of these basic activities; instead, it joins them into the following logical groups:

- Load and initiate a new command request
- Perform disk positioning I/O
- Perform disk read/write I/O
- Disk write I/O for format commands
- Abort a command

These operations are the most fundamental level of MVME320 activity. The MVME320 commands are implemented essentially as combinations of these operations. For example, the MVME320 read multiple sector command consists of "disk positioning I/O" and several "disk read I/O" operations. For the remainder of this document this set of MVME320 operations will be referred to as "logical operations".

With several disk drives online, the scheduler should do more than sequence through the execution of a command; it must decide what drives are to be serviced and when certain operations are to be performed. The algorithm that the host incorporates must not only service the online drives, but it must do this in an equitable manner. No drive should be allowed to monopolize the MVME320 resources; for example, the unserviced drives in this situation would appear to "wait" while one drive completes a command. The host system should not be required to "idle", waiting to submit a new command request. This situation would degrade the performance of the entire system. Another responsibility of the host is to provide some safeguards against a command destroying information of another current command.

Although the host program will be responsible for making the decisions described above, the operational characteristics of the MVME320 dictate when new commands are allowed and when certain operations can be scheduled. The following describes each of the logical MVME320 operations and their interaction with an external host; this information constitutes the limitations and requirements the MVME320 imposes on this host scheduler.

10.4 COMMAND INITIATION

Although requesting the MVME320 to perform a command involves simply setting the start bit for the desired drive in the configuration and semaphore register, execution of the command is dictated by the status of the MVME320 when the request is received. Because the MVME320 is operating asynchronously with an external processor or processes, submitting a new command is not always as straight-forward as altering the content of a few MVME320 control registers. The following describes the MVME320 behavior when the scheduler makes a new command request.

For the remainder of this section the internal memory of the MVME320 will be referred to as the FIFO.

10.4.1 New Command Requests

The host can asynchronously signal the MVME320 that it has a new command by setting the bit assigned to the desired drive in the drive status and configuration register. This can be performed as long as the bus is available. The MVME320 accepts the request but does not necessarily begin to process it. If the disk controller is currently controlling another disk drive, the MVME320 accepts the request and treats it as a "pending" request for disk drive service. It does this to prevent forcing the host processor to wait for it to complete its current processing. If the command request is for the drive that is currently being serviced, the MVME320 responds by signifying that it is busy. This will be reported to the host through an interrupt and a status condition indicating the drive is busy.

Because four different drives can be online simultaneously, the drive status and configuration register may contain more than one pending request. When this happens, the MVME320 will process the drive with the lowest logical drive number. The logical drive number is assigned implicitly by the ECA pointer table.

10.4.2 Validity Checks for New Commands

The MVME320 can only perform limited checks on the Event Control Area (ECA) data. The host should perform these checks whenever a disk command request is made; this is especially important when the command request is a continuation of a "system-level" disk drive command. Three ECA fields are continually updated and are required for continuation of such commands. These fields are:

- Physical sector start
- Current cylinder
- Buffer start address

Validity check errors detected by the MVME320 will result in termination of the command request and initiation of the interrupt sequence. These checks are limited to the following:

- Valid command
- Valid options
- Valid number of retries requested

After a command has been accepted, the MVME320 will not interact with the host again until the command has been completed or terminated.

10.5 MVME320 DISK DRIVE CONTROLS

One of the most powerful features of the MVME320 is its ability to control a variety of different disk drive interfaces. This ability is achieved by providing it with facilities to easily "program" the desired disk drive interface. The MVME320 disk drive interface signals are generated by an onboard processor which is controlled by a "table-driven" microprogram. The host system defines the table parameters to the MVME320 microprogram through the device control parameter fields of the ECA. The processor uses this information to generate the appropriate signals for the interface.

Disk drive I/O can be separated into two groups -- positioning and read/write operations. The MVME320 drive interface, as described previously, consists of two sections:

- Slow disk control and status signals
- High-speed control and serial data read/write signals

Generally, the former controls the positioning, and the latter controls the read/write operation.

10.5.1 Positioning Operations

During positioning operations, the read/write mechanism of the disk drive is moved to the selected track and the drive is prepared for the read/write operation. The positioning procedure performed by the MVME320 consists basically of the following seven steps:

a. Read the device control parameters from the ECA.

b. Read the slow drive status

c. Issue drive signals to the drive to move the heads to the desired track (cylinder). The mechanical motion associated with this movement is referred to as "seek".

d. Test the "seek-complete" flag for the drive until the operation is complete.

e. Wait for the "head settling time" as specified by the corresponding ECA field to ensure the heads have settled.

The MVME320 performs a seek to the desired track by issuing one seek pulse for each track the heads are to be moved. Because the actual "seeking" is a mechanical operation, it is a relatively slow activity; a minimum of 3 msec is required in drives without a buffered seek feature. The MVME320 will use this time to either attempt other drive control processing, return control to the scheduler, or continue generating and monitoring the seek signals until seek completion.

The "type of seek" field of the ECA defines which course of action is performed by the MVME320. Three types of seek operations are commonly available on most disk drives. The "normal" seek allows the MVME320 to issue a single seek pulse and after a sufficient stabilization period becomes available for other processing. The "buffered" and "accelerated" seeks require a dedicated processor to produce a continuous stream of pulses. Drives interfaced in this manner contain internal counters that store the "seek" count before initiating the actual seek operation. For such interfaces, the number of steps must be sent as a continuous stream of pulses to the disk drive. The MVME320 must issue all the required seek pulses before it can attempt other processing.

For buffered and accelerated seek devices, the maximum time for which the MVME320 is dedicated to positioning I/O generation can be easily derived. Each pulse requires a 6-usec period for hard disk drives and a 750-usec period for the Amlyn drive. Therefore, the time for the MVME320 to generate the required number of seeks is given by the product of the number of seeks and the pulse period per seek. For example, a 512-track seek would require 3 msec on a hard disk with a 6-usec pulse period.

10.5.2 Read/Write Operations

After the read/write head(s) have been positioned to the correct track, the MVME320 is ready to generate the high-speed disk control and serial data transfer signals needed for disk read/write. For the MVME320, this operation is closely integrated with the activities of the DMA controller. The disk drive and DMA interfaces operate in parallel and independent of each other and of the MVME320 control processor.

The MVME320 FIFO buffer is the primary data transmission facility between system memory and the disk drive. The DMA controller of the host/DMA interface is responsible for performing the actual transfer. For disk read, the serial data from the drive is placed into the FIFO after it is read from the drive. For disk write, data is loaded by the DMA into the FIFO from system memory before it is sent serially to the disk drive.

The "basic read and write" disk operations consist of the following sequence. The read/write operation for the format command is essentially executed as described in the following sequence. However, implementation of these commands involves additional processing, which is described in separate subsections.

The disk write operation consists of the following sequence:

a. Read the ECA for drive control parameters.

b. Transfer write data from system memory to MVME320 FIFO. Perform disk read operation to locate the data part of the requested sector (i.e., process the ID part of the sector). Refer to paragraph 10.5.2.1.

c. Transfer one operand from the FIFO to serial drive output.

d. Perform disk write I/O on the drive interface.

e. Continue steps c. and d. asynchronously until a sector has been written.

f. After a sector has been written, write the status and dynamic command execution information about the transfer into the ECA.

g. Repeat steps b. through f. until the requested number of sectors has been written.

The disk read operation consists of the following activities:

a. Read the ECA for drive control parameters.

b. Perform disk read operation to locate the data part of the requested sector (i.e., process the ID part of the sector). Refer to paragraph 10.5.2.1.

c. Perform disk read I/O on the drive interface. Move the character read into the FIFO buffer.

d. DMA transfer the operand from the FIFO to the system memory.

e. Continue steps c. and d. asynchronously until a sector has been read.

f. After a sector has been read, write the status and dynamic command execution information about the transfer into the ECA.

g. Repeat steps b. thorugh f. until the requested number of sectors has been read.

Notes on Basic Read/Write Operations.

The previous description for the MVME320 read/write I/O operations has been greatly simplified and some important points should not be overlooked. First, the actual DMA transfer involves three tasks:

a. Request mastership of the system bus.

b. After becoming bus master, generate the handshake signals required to send/receive data through the system bus.

c. Terminate the transmission and return the bus for arbitration.

Second, in step b. for both the read or write, the MVME320 processes the format information from the media. For a description of this activity, refer to Chapter 8, Disk Track Format.

10.5.3 New Command During Active Read/Write

The MVME320 DMA returns the bus to arbitration after a programmable number of operands have been transferred. Whenever the bus is available, it is possible for the host to request a new command by writing to the drive status and configuration register. The MVME320 cannot begin processing of new commands while it currently has an operation that involves a DMA transfer. There are two situations to be considered. First, if the request is for a drive which is not currently performing disk I/O, the MVME320 accepts the request and sets an internal flag to indicate a request is pending. The MVME320 completes the command and then processes the pending requests. This allows the host processor to continue processing and not be "idled", waiting for the MVME320 to become available.

Second, if the host makes the command request to the drive that is currently performing the disk drive I/O, the MVME320 responds by reporting to the host that the drive is busy through an interrupt.

10.5.4 DMA Transfers

The serial read/write I/O portion of the drive interface is a dedicated slave of the MVME320 disk drive interface; that is, a serial string is sent or received by the interface without interruption. The DMA interface, however, must surrender the bus for arbitration after the requested number of operands have been transferred. If the number of transfers does not provide the MVME320 with sufficient bus access time, the DMA could fail to keep track with the disk I/O operation.

For floppy disks, contiguous sectors can be transferred directly from the floppy disks via DMA. The floppy disk is slow enough to allow new arbitration for each word transfer without significant system restrictions on other components. The fastest floppy disk drive (double-density, 8-inch) allows 30 usec bus latency 62 usec for double-density, 5 1/4-inch; 125 usec for single-density, 5 1/4-inch).

For hard disks, however, the bus latency time needed for arbitration could cause throughput errors if data were transferred directly from hard disk to host memory. Therefore, the MVME320 reads the disk data into an internal one-sector buffer and then transfers from buffer to host memory. DMA buffering occurs only for hard disks.

10.6 DISK WRITE I/O FOR FORMAT COMMANDS

The format command write operation is performed by the MVME320 as integral operations on a per-track basis, as opposed to normal disk I/O which is on a sector basis. This technique was selected as a compromise to satisfy two conflicting requirements -- command efficiency versus equal access of online drives to the MVME320 resources. This command would be most efficient if allowed to monopolize the MVME320 resource. However, this situation would prevent any other drive from being serviced until the command had been completed. Normally this command is a background task with the other disk operations having a higher priority. For this reason, allowing the MVME320 to concentrate completely on either command is not a good system practice.

To allow other drives to be serviced without severely slowing down the execution of these commands, the MVME320 performs the I/O associated with them on a per-track basis.

The MVME320 becomes available for other procesing after a full track has been processed. For the worst-case situation for the format command, the time the MVME320 will not be able to process the other drives will not exceed two disk revolutions. This would occur if the index pulse were just "missed" and a complete revolution were required to find the pulse.

10.7 COMMAND TERMINATION

The stop command is initiated by placing a zero in the appropriate bit position in the configuration and semaphore register for the drive to be stopped. If the stop is issued to the drive that is currently processing a command, the command is immediately stopped; otherwise, the stop command is treated as a pending request to stop.

If there are both pending new command and stop command requests queued in the configuration and semaphore register, the MVME320 will first perform the stop requests.

CHAPTER 11

MAINTENANCE

11.1 INTRODUCTION

This chapter contains preventive and corrective maintenance instructions for the MVME320.

11.2 PREVENTIVE MAINTENANCE

Periodically, the user should perform the normal maintenance procedure for his system. Dirt should be cleaned from the MVME320 module and all filters. Verify the installation procedure for the MVME320 module (refer to installation instructions in Chapter 2).

<u>CAUTION</u>

WHEN USING THE MVME320 WITH A DISK DRIVE ENCLOSURE CONTAINING A SEPARATE POWER SUPPLY, TURN OFF POWER TO THE DRIVE ENCLOSURE BEFORE TURNING OFF POWER TO THE MVME320 TO PREVENT POSSIBLE DAMAGE TO THE SYSTEM. ALSO, WHEN POWERING UP THE SYSTEM, TURN ON THE POWER TO THE MVME320 BEFORE POWERING UP THE DISK DRIVE.

11.3 CORRECTIVE MAINTENANCE

In most cases, if the MVME320 fails, the module should be returned to the factory. However, the following items should be verified first as they could be causing the failure:

<u>CAUTION</u>

IF THE MVME320 MUST BE REMOVED WHEN PERFORMING ANY OF THE FOLLOWING CHECKS, TURN POWER OFF BEFORE REMOVING OR RE-INSERTING THE MODULE.

a. Perform the procedures in the installation section, verifying that the 50-pin connector and 20-pin connectors are properly plugged into both the MVME320 and the first disk drive.

b. Verify the connections between drives (e.g., the daisy-chain connections).

c. Verify that the last drive in the chain is terminated.

d. Be sure that all ICs on the MVME320 are fully seated in their sockets.

e. Be sure that the module edge connector is free of dirt and grease. If necessary, clean the edge connector. Be sure that the connector is fully inserted into the card cage in the VMEbus system.

CHAPTER 12

SUPPORT INFORMATION

12.1 INTRODUCTION

This chapter contains connector pin signal descriptions, schematic diagrams, and a parts list for the MVME320.

12.2 CONNECTOR PIN SIGNALS

Table 12-1 lists the VMEbus signals on connector P1 giving the signal mnemonic, connector and pin number, and signal characteristic. Connector P2 is not used.

12.3 PARTS LISTS

Table 12-2 lists the components of the MVME320. Figure 12-1 is the parts location diagram. The parts list reflects the latest issue of hardware at the time of printing.

12.4 SCHEMATIC DIAGRAMS

Figure 12-2 (6 sheets) contains a detailed schematic diagram for the MVME320. This schematic diagram represents the latest design. Occasionally, minor component changes are made at the factory. Therefore, when replacing a component, always use the same value as the defective component event though the schematic diagram may indicate a different value on type.

TABLE 12-1. Connector P1 Signals

SIGNAL MNEMONIC	CONNECTOR AND PIN NUMBER	SIGNAL NAME AND DESCRIPTION
ACFAIL*	1B: 3	AC FAILURE - Open-collector driven signal which indicates that the AC input to the power supply is no longer being provided or that the required input voltage levels are not being met.
IACKIN*	1A: 21	INTERRUPT ACKNOWLEDGE IN - Totem-pole driven signal. IACKIN* and IACKOUT* signals form a daisy-chained acknowledge. The IACKIN* signal indicates to the VME module that an acknowledge cycle is in progress.
IACKOUT*	1A: 22	INTERRUPT ACKNOWLEDGE OUT - Totem-pole driven signal. IACKIN* and IACKOUT* signals form a daisy-chained acknowledge. The IACKOUT* signal indicates to the next module that an acknowledge cycle is in progress.

12

TABLE 12-1. Connector P1 Signals (cont'd)

SIGNAL MNEMONIC	CONNECTOR AND PIN NUMBER	SIGNAL NAME AND DESCRIPTION
AM0-AM5	1A: 23 1B: 16,17, 18,19 1C: 14	ADDRESS MODIFIER (bits 0-5) - Three-state driven lines that provide additional information about the address bus, such as size, cycle type, and/or DTB master identification.
AS*	1A: 18	ADDRESS STROBE - Three-state driven signal that indicates a valid address is on the address bus.
A01-A23	1A: 24-30 1C: 15-30	ADDRESS bus (bits 1-23) - Three-state driven address lines that specify a memory address.
BBSY*	1B: 1	BUS BUSY - Open-collector driven signal generated by the current DTB master to indicate that it is using the bus.
BCLR*	1B: 2	BUS CLEAR - Totem-pole driven signal generated by the bus arbitrator to request release by the current DTB master in the event that a higher level is requesting the bus.
BERR*	1C: 11	BUS ERROR - Open-collector driven signal generated by a slave. This signal indicates that an unrecoverable error has occurred and the bus cycle must be aborted.
BG0IN*- BG3IN*	1B: 4,6, 8,10	BUS GRANT (0-3) IN - Totem-pole driven signals generated by the Arbiter or Requesters. Bus grant in and out signals form a daisy-chained bus grant. The bus grant in signal indicates to this module that it may become the next bus master.
BG0OUT*- BG3OUT*	1B: 5,7, 9,11	BUS GRANT (0-3) OUT - Totem-pole driven signals generated by Requesters. Bus grant in and out signals form a daisy-chained bus grant. The bus grant out signal indicates to the next board that it may become the next bus master.
BR0*-BR3*	1B: 12-15	BUS REQUEST (0-3) - Open-collector driven signals generated by Requesters. These signals indicate that a DTB master in the daisy-chain requires access to the bus.
DS0*	1A: 13	DATA STROBE 0 - Three-state driven signal that indicates during byte and word transfers that a data transfer will occur on data bus lines (D00-D07).
DS1*	1A: 12	DATA STROBE 1 - Three-state driven signal that indicates during byte and word transfers that a data transfer will occur on data bus lines (D08-D15).

12

TABLE 12-1. Connector P1 Signals

SIGNAL MNEMONIC	CONNECTOR AND PIN NUMBER	SIGNAL NAME AND DESCRIPTION
DTACK*	1A: 16	DATA TRANSFER ACKNOWLEDGE - Open-collector driven signal generated by a DTB slave. The falling edge of this signal indicates that valid data is available on the data bus during a read cycle, or that data has been accepted from the data bus during a write cycle.
D00-D15	1A: 1-8 1C: 1-8	DATA BUS (bits 0-15) - Three-state driven bidirectional data lines that provide a data path between the DTB master and slave.
GND	1A: 9,11, 15,17,19 1B: 20,23 1C: 9	GROUND
IRQ1*-IRQ7*	1B: 24-30	INTERRUPT REQUEST (1-7) - Open-collector driven signals, generated by an interrupter, which carry prioritized interrupt requests. Level seven is the highest priority.
LWORD*	1C: 13	LONGWORD - Three-state driven signal to indicate that the current transfer is a 32-bit transfer.
SYSCLK	1A: 10	SYSTEM CLOCK - A constant 16-MHz clock signal that is independent of processor speed or timing. This signal is used for general system timing use.
SYSFAIL*	1C: 10	SYSTEM FAIL - Open-collector driven signal that indicates that a failure has occurred in the system. This signal may be generated by any module on the VMEbus.
SYSRESET*	1C: 12	SYSTEM RESET - Open-collector driven signal which, when low, will cause the system to be reset.
WRITE*	1A: 14	WRITE - Three-state driven signal that specifies the data transfer cycle in progress to be either read or write. A high level indicates a read operation; a low level indicates a write operation.
+5V STDBY	1B: 31	+5 Vdc STANDBY - This line supplies +5 Vdc to devices requiring battery backup.
+5V	1A: 32 1B: 32 1C: 32	+5 Vdc Power - Used by system logic circuits.
+12V	1C: 31	+12 Vdc Power - Used by system logic circuits.
-12V	1A: 31	-12 Vdc Power - Used by system logic circuits.

TABLE 12-2. MVME320 Parts List

LOCATION/REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
--	84-W8285B01	PWB for MVME320
C1	21NW9629A16	Capacitor, mica, 50 pF at 500 Vdc, 5%
C2	21NW9629A22	Capacitor, mica, 82 pF at 500 Vdc, 5%
C10-C16,C22,C25, C29-C31,C34-C37, C41,C44-C47, C50-C53	21NW9632A03	Capacitor, ceramic, 0.1 uF at 50 Vdc, 20%
C17, C24	21NW9629A30	Capacitor, mica, 180 pF at 500 Vdc, 5%
C18	21NW9629A11	Capacitor, mica, 33 pF at 500 Vdc, 5%
C20	21NW9629A47	Capacitor, mica, 0.001 uF at 100 Vdc, 5%
C21,C32,C38,C42	23NW9618A79	Capacitor, 100 uF at 25 Vdc, 20%
C25,C26,C49	21NW9604A76	Capacitor, ceramic, 18K pF at 50 Vdc, 5%
C27,C33	21NW9604A07	Capacitor, ceramic, 2200 pF at 50 Vdc, 10%
C28	23NW9704B01	Capacitor, 4.7 uF at 10 Vdc, 20%
C39	21NW9629A02	Capacitor, mica, 10 pF at 500 Vdc, 5%
C48	21NW9629A06	Capacitor, mica, 82 pF at 500 Vdc, 5%
CR1, CR2, CR3	48NW9616A03	Diode, IN4148/IN914
J1,J2	28NW9802F98	Header, double row, 20-pin
J3	28NW9802F99	Header, double row, 50-pin
JW0,JW4,TP2	28NW9802D01	Header, double row, 2-pin
JW1	28NW9802D05	Header, single row, 8-pin
JW2	28NW9802B21	Header, double row, 6-pin
JW3,JW4	28NW9802F97	Header, single row, 13-pin
JW5, JW7, TP1	28NW9802D04	Header, single row, 3-pin
JW6	28NW9802F96	Header, single row, 12-pin
L1	24NW9708A03	Inductor, 100 mH
LED1	48NW9612A49	LED, Red
P1, P2	28NW9802E51	Connector, 96-pin

TABLE 12-2. MVME320 Parts List (cont'd)

LOCATION/REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
R1	06SW-124A39	Resistor, film, 390 ohms, 1/4 W, 5%
R2	06SW-960D81	Resistor, film, 68.1K ohms, 1/8 W, 1%
R3,R8,R11,R12,	06SW-124A65	Resistor, film, 4.7K ohms, 1/4 W, 5%
R6,R20	06SW-124A49	Resistor, film, 1.0K ohms, 1/4 W, 5%
R7,R17	06SW-124A45	Resistor, film, 680 ohms, 1/4 W, 5%
R9,R10,R13, R16,R21	06SW-124A73	Resistor, film, 10K ohms, 1/4 W, 5%
R14	06SW-960B93	Resistor, film, 909 ohms, 1/8 W, 1%
R15	06SW-960C50	Resistor, fixed, film, 3.24K ohms, 1/8W, 1%
R18	06SW-124A56	Resistor, film, 2.0K ohms, 1/4 W, 5%
R19	06SW-124A81	Resistor, film, 22K ohms, 1/4 W, 5%
R22	06SW-124A41	Resistor, film, 470 ohms, 1/4 W, 5%
R23	06SW-124A37	Resistor, film, 330 ohms, 1/4 W, 5%
R24	06SW-124A33	Resistor, film, 220 ohms, 1/4 W, 5%
R25,R28,R31,R33	06SW-960D47	Resistor, film, 30.1K ohms, 1/8 W, 1%
R26	06SW-960B18	Resistor, film, 150 ohms, 1/8 W, 1%
R27	06SW-960B09	Resistor, film, 121 ohms, 1/8 W, 1%
R30,R32	06SW-960B43	Resistor, film, 274 ohms, 1/8 W, 1%
R34	06SW-960D05	Resistor, film, 11K ohms, 1/8 W, 1%
R35,R36	06SW-124A43	Resistor, film, 560 ohms, 1/4 W, 5%
R37,R38	06SW-124A35	Resistor, film, 270 ohms, 1/4 W, 5%
RP1,RP3,RP5	51NW9626A53	Resistor network, 8-pin, 4 per package, 47 ohms
RP2	51NW9626B02	Resistor network, 8-pin, 7 per package, 47 ohms
RP4	51NW9626A41	Resistor network, 10-pin, 9 per package, 4.7K ohms

TABLE 12-2. MVME320 Parts List (cont'd)

LOCATION/REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
RP6	51NW9626A67	Resistor network, 10 pin, 8 per package, 220/330 ohms
Q1	48NW9610A34	Transistor, NPN, 2N5320
Q2	48NW9610A13	Transistor, NPN, 2N2222
--	29NW9805B17	Jumper, shorting, insulated
--	14NW9416A01	Pad, mounting (for Q1)
--	84-W8285B01	MVME320 PWB assembly
7D,7H	51NW9615E91	I.C., SN74LS00N
6G	51NW9615C22	I.C., SN74LS08N
6E,7G,12K	51NW9615E93	I.C., SN74LS14N
7F,8A	51NW9615C24	I.C., SN74LS32N
6F	51NW9615C25	I.C., SN74LS74AN
11K, 11L	51NW9615F01	I.C., SN74LS86N
7L	51NW9615C26	I.C., SN74LS123N
7E	51NW9615H56	I.C., SN74LS136N
7A,9H	51NW9615C69	I.C., SN74LS138N
7K,9G,12C	51NW9615E84	I.C., SN74LS153N
2D-2G, 2DE, 2FG	51NW9615P19	I.C., SN74LS163AD
7C	51NW9615C29	I.C., SN74LS174N
4EF	51NW9615E95	I.C., SN74LS240N
5EF	51NW9615F46	I.C., 74LS241N
10D	51NW9615F02	I.C., SN74LS244N
3D, 10F, 10G, 10 HK,10L	51NW9615F52	I.C., SN74LS273N
3G	51NW9615E98	I.C., SN74LS373N
3EF	51NW9615L69	I.C., SN74LS377N3

TABLE 12-2. MVME320 Parts List (cont'd)

LOCATION/REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
9D-9F	51NW9615F38	I.C., SN74LS393N
5C	51NW9615N81	I.C., SN74LS642-1N
2AB,2C	51NW9615H89	I.C., SN74LS645-1N
1H	51NW9615N77	I.C., SN74LS688N
8K	51NW9615D32	I.C., SN74S02N
8D,12A	51NW9615K71	I.C., 74F04PC
8H	51NW9615C56	I.C., SN74S08N
6D	51NW9615K68	I.C., 74F11PC
6A,6B,6H	51NW9615D27	I.C., SN74S32N
8G, 8L	51NW9615C97	I.C., SN74S112N
7B	51NW9615C34	I.C., SN74S138N
5AB, 6C	51NW9615F79	I.C., SN74S240N
1D,1EF,1G	51NW9615G07	I.C., SN74S244N
13EF	51NW9615C95	I.C., SN74S74N
9A	51NW9615K69	I.C., 74F10PC
10A	51NW9615J39	I.C., 74F74PC
12H	51NW9615L98	I.C., AM26LS31PC
11H	51AW1131X41	Programmed PROM
11G	51AW4800B01	Programmed PROM
5D	51AW4644B13	Programmed PROM
2L	51AW4803B01	Programmed PROM
2HK	51AW4803B02	Programmed PROM
1L	51AW4803B03	Programmed PROM
11D,12D	51NW9615N83	I.C., AM29825DC
4AB	51NW9615N84	I.C., AM29826DC
4C	51NW9615P24	I.C., N8X371N

TABLE 12-2. MVME320 Parts List (cont'd)

LOCATION/REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
3AB,3C	51NW9615P41	I.C., AM2953DC
4HKL	51NW9615N06	I.C., N8X305N
4D	51AW4722B02	Programmed PROM
8F	51AW4722B04	Programmed PROM
10BC	51NW9615P12	I.C., NE530N
10C	51NW9615A28	I.C., MC4024P
12L	51NW9615F60	I.C., MC3486P
4G,5G	51NW9615N51	I.C., UPD2149D
0F	51NW9615E11	I.C., MC7805CT
12F	51NW9615P20	I.C., DM74LS952N
8C	51NW9615N82	I.C., SD5000N
9K	01NW9804B83	Digital delay module, 50ns
8E	01NW9804C81	Digital delay module, 2-70ns
6A,6D,6F,6H,7G, 8D,8H,9A,10A, 10C,12A,13EF	21NW9634A01	Capacitor, ceramic, 14-pin
7B,8L,9G	21NW9634A02	Capacitor, ceramic, 16-pin
12EF	21NW9634A03	Capacitor, ceramic, 18-pin
6C,8F,10D,10F, 10K,11 EF,12G	21NW9634A04	Capacitor, ceramic, 20-pin
3AB,3C,4AB,11D, 12D	21NW9634A05	Capacitor, ceramic, 24-pin
2L	21NW9634A06	Capacitor, ceramic, 24-pin
8C	09NW9811A16	Socket, I.C., DIL, 16-pin
1L,2L,2HK	09NW9811A16	Socket, I.C., DIL, 24-pin
4HKL	09-W4659B25	Socket, 25-pin
4D	09-NW9811A78	Socket, I.C., DIL, 20-pin
12B	48NW9606A53	Crystal, 10 MHz

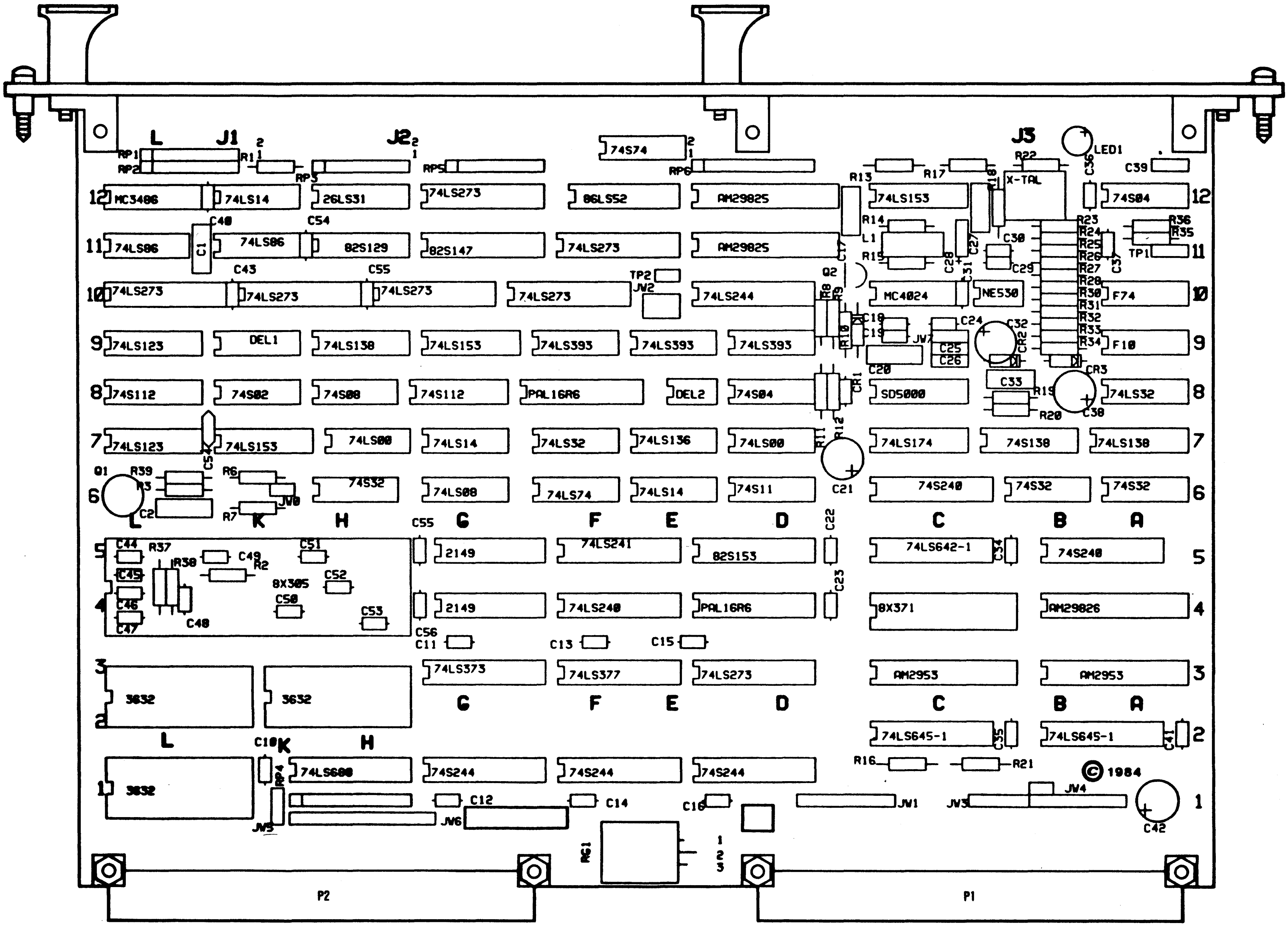


FIGURE 12-1. MVME320 Parts Location Diagram

NOTES:

1. FOR REFERENCE DRAWINGS REFER TO BILL(S) OF MATERIAL 01-W3285B01 .
2. UNLESS OTHERWISE SPECIFIED:
   ALL RESISTORS ARE IN OHMS, ± 5PCT, 1/4 WATT.
   ALL CAPACITORS ARE IN UF.
   ALL VOLTAGES ARE DC.
3. INTERRUPTED LINES CODED WITH THE SAME LETTER OR LETTER COMBINATIONS ARE ELECTRICALLY CONNECTED.
4. DEVICE TYPE NUMBER IS FOR REFERENCE ONLY. THE NUMBER VARIES WITH THE MANUFACTURER.
5. SPECIAL SYMBOL USAGE:
   * DENOTES - ACTIVE LOW SIGNAL.
6. INTERPRET DIAGRAM IN ACCORDANCE WITH AMERICAN NATIONAL STANDARDS INSTITUTE SPECIFICATIONS, CURRENT REVISION.

REVISIONS

ZONE	REV	DESCRIPTION	DATE	APPROVED
	A2	INITAL RELEASE BY SIGNETICS		
	B	REVISED AND REIDENTIFIED FROM SMVME4-300 FOR CONTROL AND PRODUCTION ASSEMBLY BY MICROSYSTEMS R.W. 3/26/84 ACD	4/5/84	
	C	PCCN 264: SH1: UPDATED REV STAT OF SHEETS. SH2: ADDED CONNECTION BETWEEN 8E-1 AND 6G-3. DELETED CONN BETW 8E-1 AND 6G-2. ADDED 8K, 9L, SIG1, SIG2 AND SYSCLK*. SH3: ADDED SIG1, SIG2 AND SYSCLK*. SH 2 THRU 6: REVISED AND UPDTD. SH TO SH REFS. M.H. 1-14-85 DcT	1/29/85	

REV STATUS	C	C	C	C	C	C
OF SHEETS	1	2	3	4	5	6

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS IN INCHES
METRIC DIMENSIONS IN (MILLIMETERS)
REMOVE ALL BURRS AND SHARP EDGES.
SURFACE TEXTURE IN MICROINCHES
TOLERANCE:
2 PLACE DEC. ±
3 PLACE DEC. ±
HOLES ± ANGLES ±

THIS DOCUMENT CONTAINS INFORMATION PROPRIETARY TO MOTOROLA INC. AND SHALL NOT BE USED FOR ENGINEERING DESIGN, PROCUREMENT OR MANUFACTURE IN WHOLE OR IN PART WITHOUT CONSENT OF MOTOROLA INC.

		DATE
DRAWN BY	R. WOODS	3/26/84
CHECKED BY	AC DEFOREST	4-5-84
PROJECT LEADER	P. WASIELEWSKI	4-5-84
DESIGN MANAGER	N.L. WAGNER	4/5/84
MFG ENGINEERING	G. CHAILLIE	4-5-84
DESIGN ENGINEER	P. WASIELEWSKI	4-5-84

DESCRIPTION | PART NO

MOTOROLA INC.
MOS Integrated Circuits Group
MICROSYSTEMS
2900 South Diablo Way, Tempe, Arizona 85282 U.S.A.

TITLE: SCHEMATIC DIAGRAM - MVME320 DISK CONTROLLER

D | FSCM NO. | DRAWING NO. 63DW3285B
SCALE: NONE | SHEET 1 OF 6

DWG NO 63DW3285B SH 1

9-82

FIGURE 12-2. MVME320 Schematic Diagram (Sheet 1 of 6)

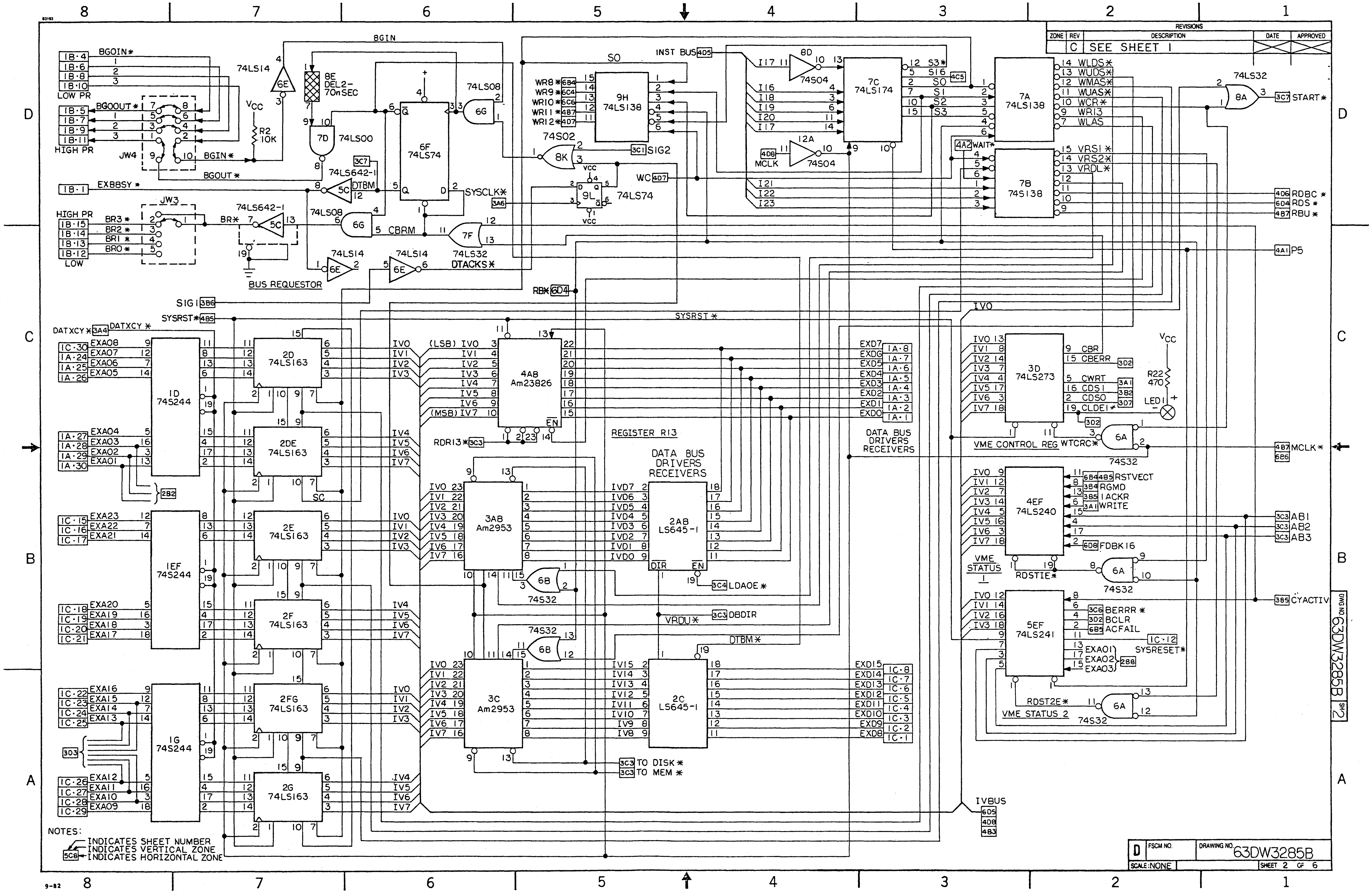


FIGURE 12-2. MVME320 Schematic Diagram (Sheet 2 of 6)

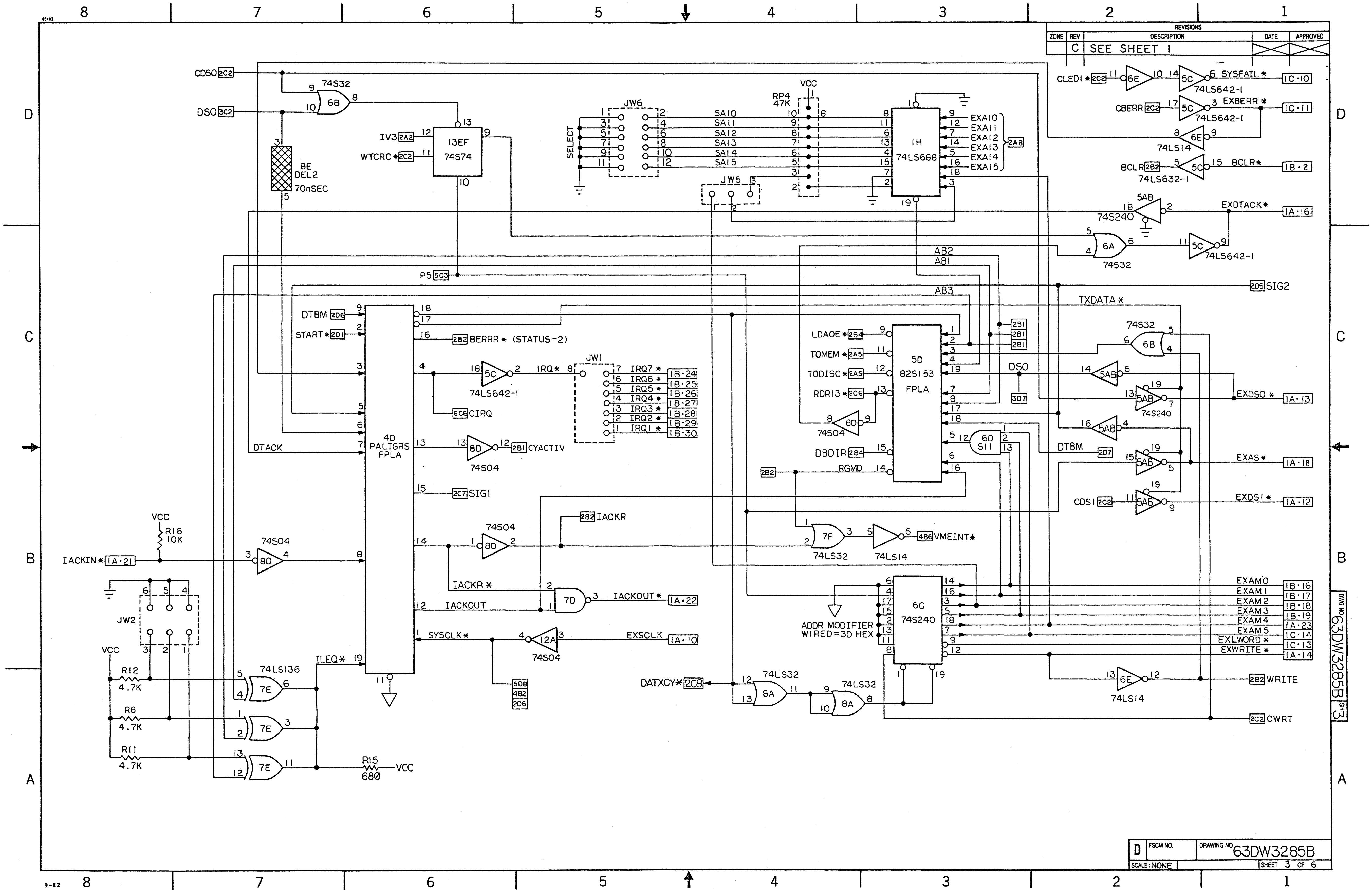


FIGURE 12-2. MVME320 Schematic Diagram (Sheet 3 of 6)

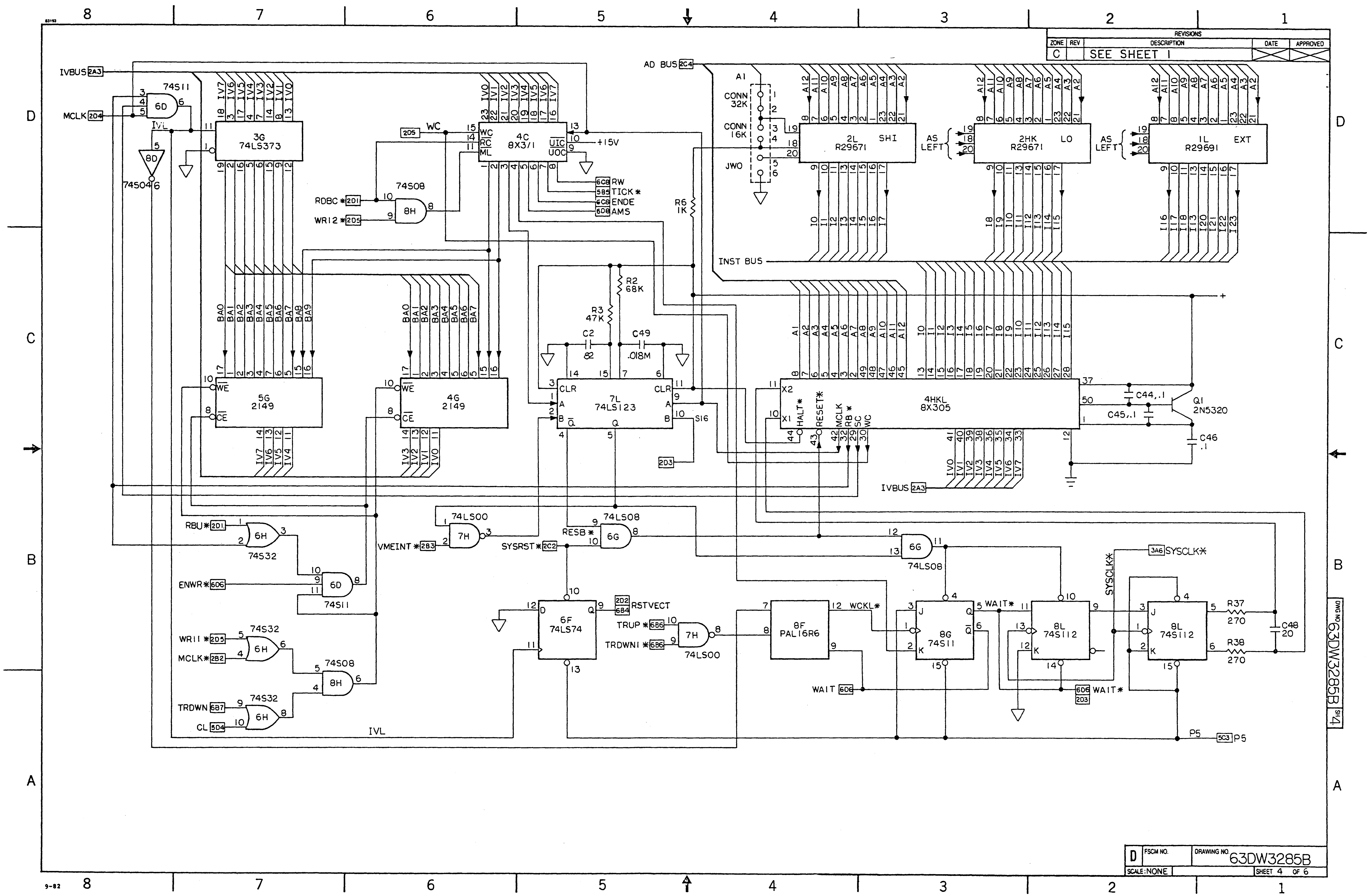


FIGURE 12-2. MVME320 Schematic Diagram (Sheet 4 of 6)

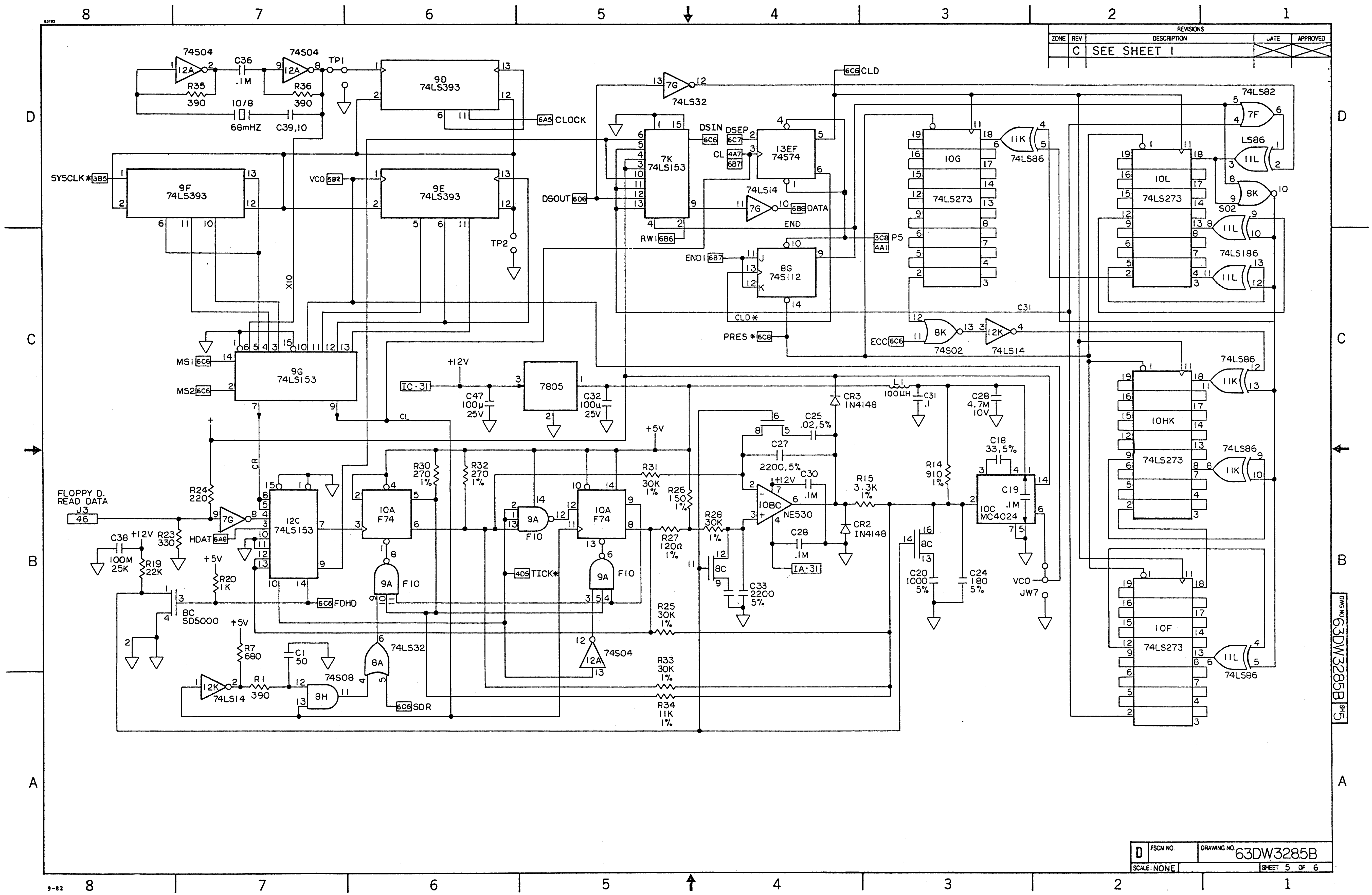


FIGURE 12-2. MVME320 Schematic Diagram (Sheet 5 of 6)

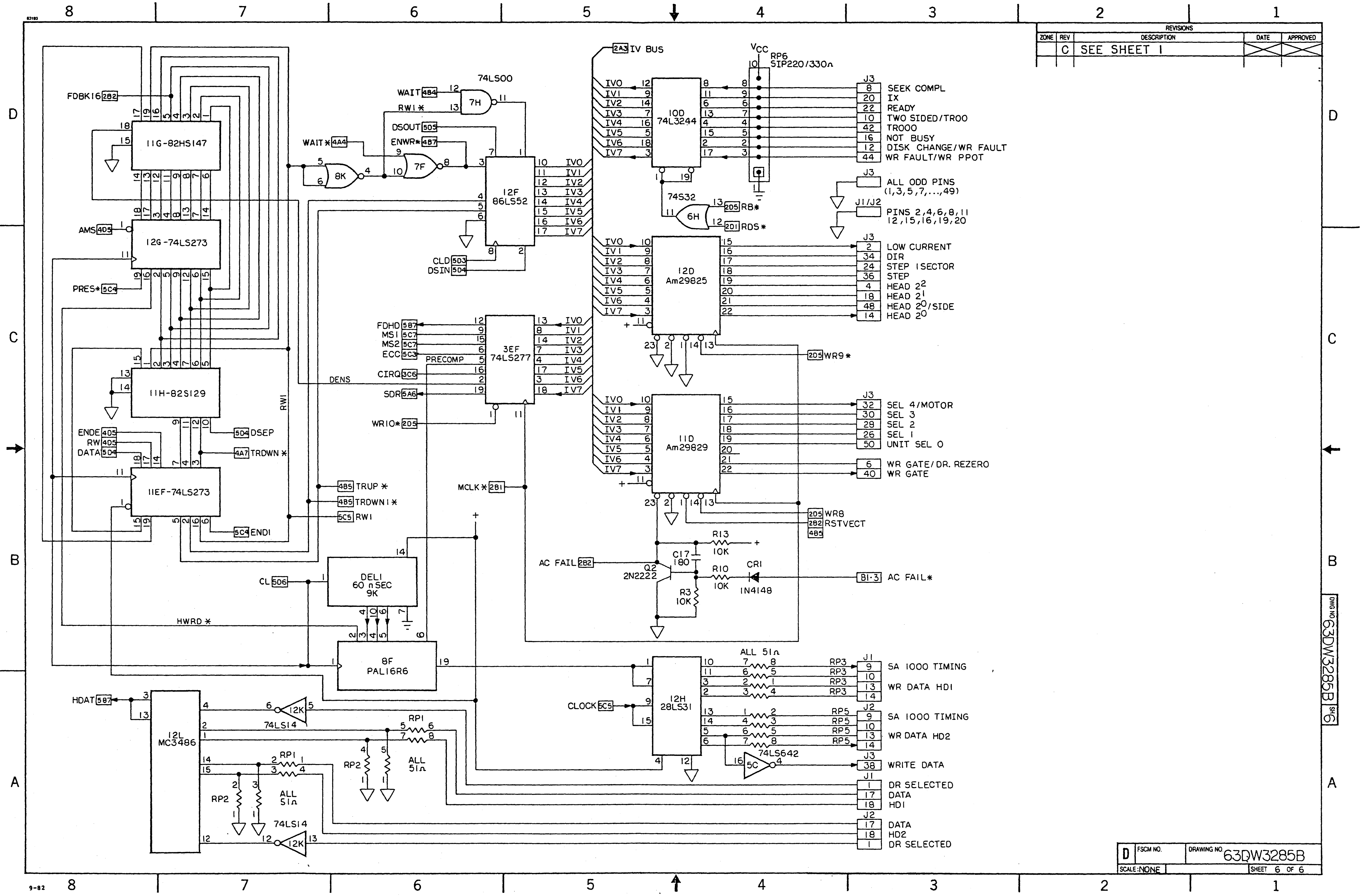


FIGURE 12-2. MVME320 Schematic Diagram (Sheet 6 of 6)

APPENDIX A

DISK FORMATS

A.1 INTRODUCTION

This appendix describes 8-inch and 5 1/4-inch hard disk and diskette formats.

A.2 8-INCH AND 5 1/4-INCH HARD DISK FORMATS

The industry standard format for 8-inch and 5 1/4-inch hard disks is 32 sectors/track, 256 bytes/sector; however, the MVME320 supports:

- Sector lengths of 128, 256, and 512 bytes per sector
- Up to 8 heads per drive, and
- Up to 64K cylinders per drive.

Each sector on the track consists of two fields separated by gaps to allow updating and recovery. The first field in the sector is the sector identifier (ID). The ID field contains four data type bytes for unique identification of the sector. These four bytes are:

- Cylinder number
- Head number
- Sector number
- Length code

The second field in the sector is the data field containing user data. Figure A-1 shows the track format used for 8-inch and 5 1/4-inch hard disk drives.

```
                                                             (Last Sector
         <-----------1 Sector----------->                      on Track)
 -----------------------------------------------------/-------------------
| GAP |SYNC| ID FIELD |GAP |SYNC | DATA  |  GAP  |   /  LAST DATA  | GAP |
|  1  |    |          | 2  |     | FIELD |   3   |  /     FIELD    |  4  |
 -----------------------------------------------------/-------------------
   |     |      |       |            |       |                        |
   |     |      |       |            |       |_15 bytes of 4E         |
   |     |      |       |            |                                |
   |     |      |       |_3 bytes    |              352 (NOM)         |
   |     |      |         of 4E      |              bytes of 4E_______|
   |     |      |                    |              (=ZTOL)
   |     |      |_Byte 1 = A1        |
   |     |              2 = FE       |_Byte 1   = A1
   |     |              3 = XX cyl hi         2   = FB
   |     |              4 = XX cyl lo         3-258   = user data
   |     |              5 = XX head           259-260 = XX CRC, ECC
   |     |              6 = XX sector         261-262 = XX ECC
   |     |              7 = 01 (N)
   |     |              8 = XX CRC1
   |     |              9 = XX CRC2
   |     |
   |     |_12 bytes of 00 (2 places/sector)
   |
   |_20 bytes of 4E
```

FIGURE A-1. Recommended Hard Disk Track format (256 Bytes/Sector)

The recording method used is MFM on all tracks. MFM is encoded using the following rules:

a. Write data bits at the center of the cell.

b. Write clock bits at the beginning of the cell if:

- No data was written in the previous cells
- No data will be written in the present cell.

A.3 8-INCH AND 5 1/4-INCH DISKETTE FORMATS

Table A-1 lists the track formats supported on the diskette drives.

TABLE A-1. Diskette Formats

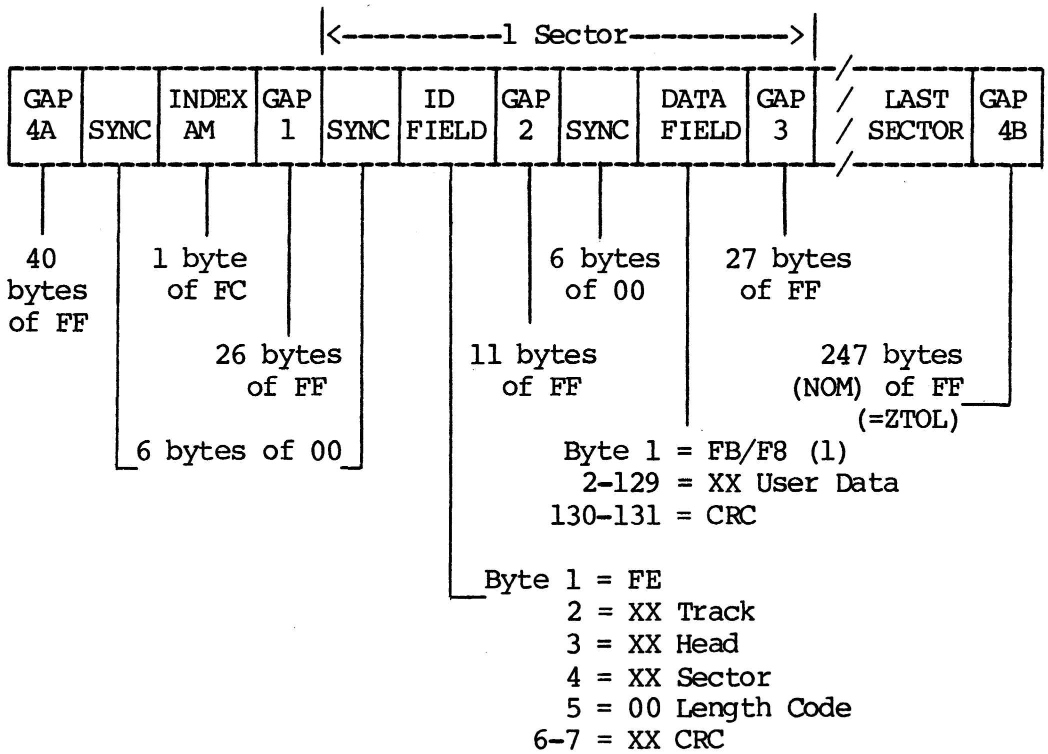
	8-INCH SINGLE-SIDED	8-INCH DOUBLE-SIDED	5 1/4-INCH SINGLE-SIDED	5 1/4-INCH DOUBLE-SIDED	5 1/4-INCH DOUBLE-SIDED 96TPI
	IBM	IBM	Formatted the same as 8-inch		
TRACKS/DISK	77	154	40	80	160
DATA RATE (BITS/SEC)	250K/500K		125/250K		
RECORDING METHOD	FM/MFM				
ERROR DETECT CORRECT METHOD	16-bit CRC				
BASIC FORMAT: (BYTES/SECTOR)					
BUFFERED	128/256/512				
UNBUFFERED	128 to 8192 for floppy only				
TYPICAL SECTORS/TRACK	26	26	16	16	
TRACKS/CYLINDER	1	2	1	2	
USER DATA BYTES/DISKETTE	246.272K (SS) 480K (DD)	985.088K (DS, DD)	250.25K (SS)	480K (DS)	

NOTE: SS = single-sided, DS = double-sided, DD = double-density

The recording formats for floppy disks are similar to those for hard disks. The MVME320 supports both FM and MFM formats for single-density and double-density floppies, respectively.

A.3.1 3740 IBM Format

This format uses FM single-density encoding on all tracks. Figure A-2 illustrates the track format.



NOTE: (1) Deleted data address mark byte:

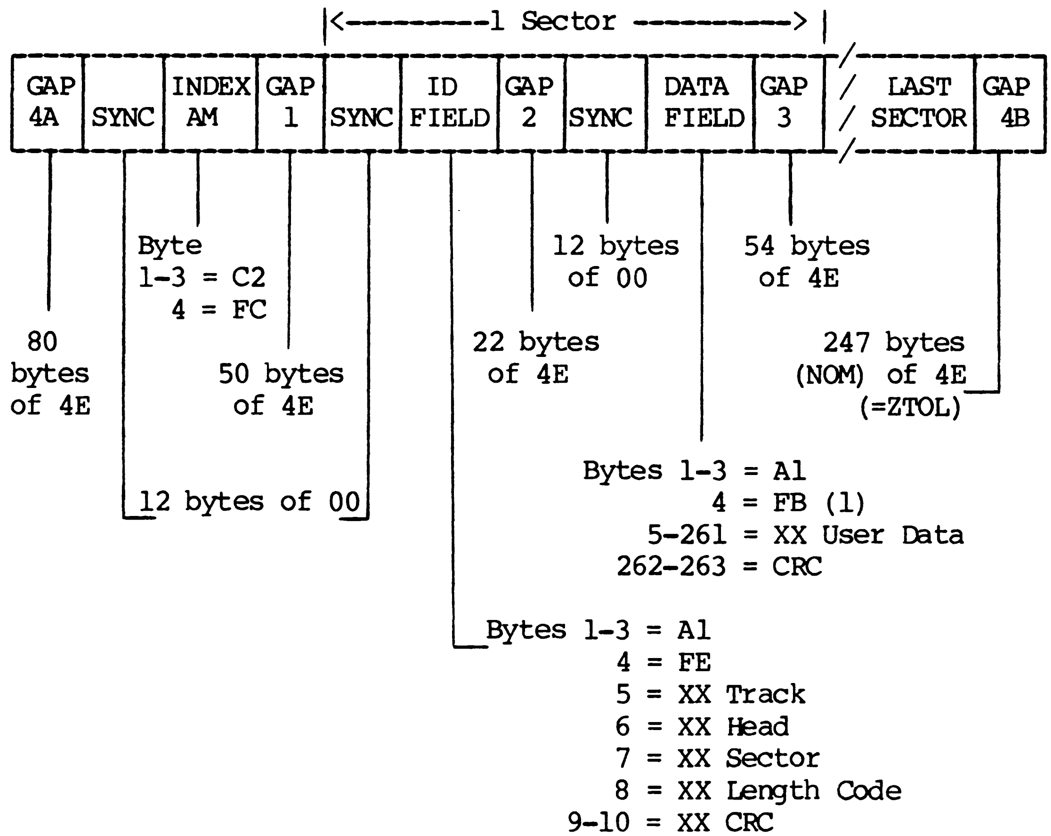
FB = data field contains normal data
F8 = data field contains deleted data

FIGURE A-2. IBM FM Track Format (128 Bytes/Sector)

A.3.2 System 34 IBM Format

This format uses MFM double-density encoding on all tracks except track 0, head 0 which uses the 3740 format described above. Figure A-3 illustrates the track format.

The MVME320 can support both FM and MFM on track 0.



NOTE: (1) Deleted data address mark:

FB = the data field contains normal data
F8 = the data field contains deleted data

2. TRACK 0, HEAD 0 uses the format shown in Figure A-2 above.

FIGURE A-3. IBM MFM Track Format (256 Bytes/Sector)

APPENDIX B

DISK INTERFACE CONNECTOR PIN ASSIGNMENTS

B

B.1 INTRODUCTION

This appendix defines the pin assignments for disk interface connectors J1, J2, and J3.

B.2 20-Pin Connectors (J1 & J2)

Signal	Pin	Pin	Signal
+DRIVE SELECTED	1	2	GROUND
	3	4	
	5	6	
	7	8	GROUND
+TIMING CLOCK	9	10	-TIMING CLOCK
GROUND	11	12	GROUND
+MFM WRITE DATA	13	14	-MFM WRITE DATA
GROUND	15	16	GROUND
+MFM READ DATA	17	18	-MFM READ DATA
GROUND	19	20	GROUND

c.3 50/34-Pin Connector (J3)

	5 1/4" FLOPPY SA450	8" HARD SA1000	8" FLOPPY SA800	8" FLOPPY (DS) SA850	5 1/4" HARD ST506
1 2		RWC		RWC	RWC
3 4					HEAD SEL 2^2
5 6					WR. GATE
7 8		SK. COMPLETE			SK. COMPLETE
9 10				(1) TWO SIDED	TRACK 00
11 12					WR. FAULT
13 14		HEAD SEL 20		SIDE SEL	HEAD SEL 20
15 16					reserved
17 18	---	HEAD SEL 21	HEAD LOAD	HEAD LOAD	HEAD SEL 21
19 20		INDEX	INDEX	INDEX	INDEX
21 22	---	READY	READY	READY	READY
23 24	IX (INDEX)				STEP
25 26	DR. SEL 1	DR. SEL 1	DR. SEL 1	DR. SEL 1	DR. SEL 1
27 28	DR. SEL 2	DR. SEL 2	DR. SEL 2	DR. SEL 2	DR. SEL 2
29 30	DR. SEL 3	DR. SEL 3	DR. SEL 3	DR. SEL 3	DR. SEL 3
31 32	MOTOR ON	DR. SEL 4	DR. SEL 4	DR. SEL 4	DR. SEL 4
33 34	DIRECTION	DIRECTION	DIRECTION	DIRECTION	DIRECTION
35 36	STEP	STEP	STEP	STEP	
37 38	WRITE DATA		WR. DATA	WR. DATA	
39 40	WRITE GATE	WRITE GATE	WR. GATE	WR. GATE	
41 42	TRACK 00	TRACK 00	TRACK 00	TRACK 00	> N/A
43 44	WR. PROTECT	WR. FAULT	WR. PROTECT	WR. PROTECT	
45 46	READ DATA		READ DATA	READ DATA	
47 48	SIDE SELECT				
49 50	"Special signal" Motor-on/Head-load for floppies				

NOTE: (1) Always disable two-sided status from drive.

APPENDIX C

ECA BLOCK TABLES

C.1 INTRODUCTION

Figures C-1 through C-6 are sample ECA block formats for various hard and floppy disks.

15	0	
COMMAND CODE	MAIN STATUS	
EXTENDED STATUS 0000		
MAXIMUM # OF RETRIES 0A	00 ACTUAL # OF RETRIES	
DMA TYPE 00	00 COMMAND OPTIONS	
BUFFER ADDRESS MOST SIGNIFICANT WORD 000C		
BUFFER ADDRESS LEAST SIGNIFICANT WORD 2000		
BUFFER LENGTH REQUESTED		
ACTUAL NUMBER OF BYTES TRANSFERRED 0000		
CYLINDER NUMBER		(1) (4)
HEAD OR SURFACE NUMBER 00	01 SECTOR NUMBER	(1)
CURRENT CYLINDER POSITION 0000		
RESERVED (5 WORDS) 0000		
N0 -- PRE-INDEX GAP 00	00 N1 -- POST-INDEX GAP	(2)
N2 -- SYNC BYTE COUNT 00	00 N3 -- POST-ID GAP	(2)
N4 -- POST-DATA GAP 1B	01 N5 -- ADDR MARK CNT	(2)
SECTOR LENGTH CODE 00	E5 FILL BYTE	(2)
RESERVED (3 WORDS) 0000		
DRIVE TYPE 04	02 NUMBER OF SURFACES	
# OF SECTORS/TRACK 10	0C STEPPING RATE	
HEAD SETTLING TIME 46	46 HEAD LOAD TIME	

FIGURE C-1. Sample ECA Block Format -- Shugart 450
5 1/4-inch Single-Density Floppy Disk
IBM 3740 Format, 128 Bytes/Sector

15 8	7 0	
SEEK TYPE 00	00 RESERVED FOR CONTROLLER	
LOW WRITE CURRENT BOUNDARY TRACK 0028		
PRECOMPENSATION BOUNDARY TRACK 0028		
ECC REMAINDER (3 WORDS) 0000		(3)
APPEND ECC REMAINDER FROM DISK (3 WORDS) 0000		(3)
RESERVED (2 WORDS) 0000		
MVME320 WORKING AREA (6 WORDS) 0000		
RESERVED FOR THE CONTROLLER		

NOTES:

(1) Physical starting sector number

(2) Track format fields (refer to Chapter 8)

(3) Last word (don't care)

(4) Cylinder number is used as the maximum track number per surface during format command.

5. This is an example, assuming the system memory address has been selected as 000C0000...000FFFFF for a 256K memory module.

6. All numbers are hexadecimal initial ECA values.

FIGURE C-1. Sample ECA Block Format -- Shugart 450
5 1/4-inch Single-Density Floppy Disk
IBM 3740 Format, 128 Bytes/Sector (cont'd)

15	0	
COMMAND CODE	MAIN STATUS	
EXTENDED STATUS 0000		
MAXIMUM # OF RETRIES 0A	00 ACTUAL # OF RETRIES	
DMA TYPE 00	00 COMMAND OPTIONS	
BUFFER ADDRESS MOST SIGNIFICANT WORD 000C		
BUFFER ADDRESS LEAST SIGNIFICANT WORD 2000		
BUFFER LENGTH REQUESTED		
ACTUAL NUMBER OF BYTES TRANSFERRED 0000		
CYLINDER NUMBER		(1) (4)
HEAD OR SURFACE NUMBER 00	01 SECTOR NUMBER	(1)
CURRENT CYLINDER POSITION 0000		
RESERVED (5 WORDS) 0000		
N0 -- PRE-INDEX GAP 00	00 N1 -- POST-INDEX GAP	(2)
N2 -- SYNC BYTE COUNT 00	00 N3 -- POST-ID GAP	(2)
N4 -- POST-DATA GAP 36	03 N5 -- ADDR MARK CNT	(2)
SECTOR LENGTH CODE 01	E5 FILL BYTE	(2)
RESERVED (3 WORDS) 0000		
DRIVE TYPE 05	02 NUMBER OF SURFACES	
# OF SECTORS/TRACK 10	0C STEPPING RATE	
HEAD SETTLING TIME 46	46 HEAD LOAD TIME	

FIGURE C-2. Sample ECA Block Format -- Shugart 450
5 1/4-inch Double-Density Floppy Disk
IBM 34 Format, 256 Bytes/Sector

15 8	7 0		
SEEK TYPE 00	00 RESERVED FOR CONTROLLER		
LOW WRITE CURRENT BOUNDARY TRACK		0028	
PRECOMPENSATION BOUNDARY TRACK		0028	
ECC REMAINDER (3 WORDS)		0000	(3)
APPEND ECC REMAINDER FROM DISK (3 WORDS)		0000	(3)
RESERVED (2 WORDS)		0000	
MVME320 WORKING AREA (6 WORDS)		0000	
RESERVED FOR THE CONTROLLER			

NOTES:

(1) Physical starting sector number

(2) Track format fields (refer to Chapter 8)

(3) Last word (don't care)

(4) Cylinder number is used as the maximum track number per surface during format command.

5. This is an example, assuming the system memory address has been selected as 000C0000...000FFFFF for a 256K memory module.

6. All numbers are hexadecimal initial ECA values.

FIGURE C-2. Sample ECA Block Format -- Shugart 450
5 1/4-inch Double-Density Floppy Disk
IBM 34 Format, 256 Bytes/Sector (cont'd)

15	0	
COMMAND CODE	MAIN STATUS	
EXTENDED STATUS 0000		
MAXIMUM # OF RETRIES 0A	00 ACTUAL # OF RETRIES	
DMA TYPE 00	00 COMMAND OPTIONS	
BUFFER ADDRESS MOST SIGNIFICANT WORD 000C		
BUFFER ADDRESS LEAST SIGNIFICANT WORD 2000		
BUFFER LENGTH REQUESTED		
ACTUAL NUMBER OF BYTES TRANSFERRED 0000		
CYLINDER NUMBER		(1) (4)
HEAD OR SURFACE NUMBER 00	01 SECTOR NUMBER	(1)
CURRENT CYLINDER POSITION 0000		
RESERVED (5 WORDS) 0000		
N0 -- PRE-INDEX GAP 00	00 N1 -- POST-INDEX GAP	(2)
N2 -- SYNC BYTE COUNT 00	00 N3 -- POST-ID GAP	(2)
N4 -- POST-DATA GAP 1B	01 N5 -- ADDR MARK CNT	(2)
SECTOR LENGTH CODE 00	E5 FILL BYTE	(2)
RESERVED (3 WORDS) 0000		
DRIVE TYPE 00	02 NUMBER OF SURFACES	
# OF SECTORS/TRACK 1A	06 STEPPING RATE	
HEAD SETTLING TIME 46	46 HEAD LOAD TIME	

FIGURE C-3. Sample ECA Block Format -- Shugart 860
8-inch Single-Density Floppy Disk
IBM 3740 Format, 128 Bytes/Sector

15	8	7	0	
SEEK TYPE	00	00	RESERVED FOR CONTROLLER	
LOW WRITE CURRENT BOUNDARY TRACK			0028	
PRECOMPENSATION BOUNDARY TRACK			0028	
ECC REMAINDER (3 WORDS)			0000	(3)
APPEND ECC REMAINDER FROM DISK (3 WORDS)			0000	(3)
RESERVED (2 WORDS)			0000	
MVME320 WORKING AREA (6 WORDS)			0000	
RESERVED FOR THE CONTROLLER				

NOTES:

(1) Physical starting sector number

(2) Track format fields (refer to Chapter 8)

(3) Last word (don't care)

(4) Cylinder number is used as the maximum track number per surface during format command.

5. This is an example, assuming the system memory address has been selected as 000C0000...000FFFFF for a 256K memory module.

6. All numbers are hexadecimal initial ECA values.

FIGURE C-3. Sample ECA Block Format -- Shugart 860
8-inch Single-Density Floppy Disk
IBM 3740 Format, 128 Bytes/Sector (cont'd)

15			0	
COMMAND CODE			MAIN STATUS	
EXTENDED STATUS			0000	
MAXIMUM # OF RETRIES	0A	00	ACTUAL # OF RETRIES	
DMA TYPE	00	00	COMMAND OPTIONS	
BUFFER ADDRESS MOST SIGNIFICANT WORD			000C	
BUFFER ADDRESS LEAST SIGNIFICANT WORD			2000	
BUFFER LENGTH REQUESTED				
ACTUAL NUMBER OF BYTES TRANSFERRED			0000	
CYLINDER NUMBER				(1) (4)
HEAD OR SURFACE NUMBER	00	01	SECTOR NUMBER	(1)
CURRENT CYLINDER POSITION			0000	
RESERVED (5 WORDS)			0000	
N0 -- PRE-INDEX GAP	00	00	N1 -- POST-INDEX GAP	(2)
N2 -- SYNC BYTE COUNT	00	00	N3 -- POST-ID GAP	(2)
N4 -- POST-DATA GAP	36	03	N5 -- ADDR MARK CNT	(2)
SECTOR LENGTH CODE	01	E5	FILL BYTE	(2)
RESERVED (3 WORDS)			0000	
DRIVE TYPE	01	02	NUMBER OF SURFACES	
# OF SECTORS/TRACK	1A	06	STEPPING RATE	
HEAD SETTLING TIME	46	46	HEAD LOAD TIME	

FIGURE C-4. Sample ECA Block Format -- Shugart 860
8-inch Double-Density Floppy Disk
IBM 34 Format, 256 Bytes/Sector

C

15 8	7 0	
SEEK TYPE 00	00 RESERVED FOR CONTROLLER	
LOW WRITE CURRENT BOUNDARY TRACK 0028		
PRECOMPENSATION BOUNDARY TRACK 0028		
ECC REMAINDER (3 WORDS) 0000		(3)
APPEND ECC REMAINDER FROM DISK (3 WORDS) 0000		(3)
RESERVED (2 WORDS) 0000		
MVME320 WORKING AREA (6 WORDS) 0000		
RESERVED FOR THE CONTROLLER		

NOTES:

(1) Physical starting sector number

(2) Track format fields (refer to Chapter 8)

(3) Last word (don't care)

(4) Cylinder number is used as the maximum track number per surface during format command.

5. This is an example, assuming the system memory address has been selected as 000C0000...000FFFFF for a 256K memory module.

6. All numbers are hexadecimal initial ECA values.

FIGURE C-4. Sample ECA Block Format -- Shugart 860
8-inch Double-Density Floppy Disk
IBM 34 Format, 256 Bytes/Sector (cont'd)

15	0	
COMMAND CODE	MAIN STATUS	
EXTENDED STATUS 0000		
MAXIMUM # OF RETRIES 0A	00 ACTUAL # OF RETRIES	
DMA TYPE 00	00 COMMAND OPTIONS	
BUFFER ADDRESS MOST SIGNIFICANT WORD 000C		
BUFFER ADDRESS LEAST SIGNIFICANT WORD 2000		
BUFFER LENGTH REQUESTED		
ACTUAL NUMBER OF BYTES TRANSFERRED 0000		
CYLINDER NUMBER		(1) (4)
HEAD OR SURFACE NUMBER 00	01 SECTOR NUMBER	(1)
CURRENT CYLINDER POSITION 0000		
RESERVED (5 WORDS) 0000		
N0 -- PRE-INDEX GAP 00	00 N1 -- POST-INDEX GAP	(2)
N2 -- SYNC BYTE COUNT 00	00 N3 -- POST-ID GAP	(2)
N4 -- POST-DATA GAP 0F	01 N5 -- ADDR MARK CNT	(2)
SECTOR LENGTH CODE 01	E5 FILL BYTE	(2)
RESERVED (3 WORDS) 0000		
DRIVE TYPE 03	06 NUMBER OF SURFACES	
# OF SECTORS/TRACK 20	00 STEPPING RATE	
HEAD SETTLING TIME 00	00 HEAD LOAD TIME	

FIGURE C-5. Sample ECA Block Format - ST412
5 1/4-inch Hard Disk (10 MHz, 15 Mbyte)

15	8 7	0	
SEEK TYPE 02	00 RESERVED FOR CONTROLLER		
LOW WRITE CURRENT BOUNDARY TRACK		0050	
PRECOMPENSATION BOUNDARY TRACK		0050	
ECC REMAINDER (3 WORDS)		0000	(3)
APPEND ECC REMAINDER FROM DISK (3 WORDS)		0000	(3)
RESERVED (2 WORDS)		0000	
MVME320 WORKING AREA (6 WORDS)		0000	
RESERVED FOR THE CONTROLLER			

NOTES:

(1) Physical starting sector number

(2) Track format fields (refer to Chapter 8)

(3) Last word (don't care)

(4) Cylinder number is used as the maximum track number per surface during format command.

5. This is an example, assuming the system memory address has been selected as 000C0000...000FFFFF for a 256K memory module.

6. All numbers are hexadecimal initial ECA values.

FIGURE C-5. Sample ECA Block Format - ST412
5 1/4-inch Hard Disk (10 MHz, 15 Mbyte) (cont'd)

15	0	
COMMAND CODE	MAIN STATUS	
EXTENDED STATUS 0000		
MAXIMUM # OF RETRIES 0A	00 ACTUAL # OF RETRIES	
DMA TYPE 00	00 COMMAND OPTIONS	
BUFFER ADDRESS MOST SIGNIFICANT WORD 000C		
BUFFER ADDRESS LEAST SIGNIFICANT WORD 2000		
BUFFER LENGTH REQUESTED		
ACTUAL NUMBER OF BYTES TRANSFERRED 0000		
CYLINDER NUMBER		(1)
HEAD OR SURFACE NUMBER 00	01 SECTOR NUMBER	(1)
CURRENT CYLINDER POSITION 0000		
RESERVED (5 WORDS) 0000		
N0 -- PRE-INDEX GAP 00	00 N1 -- POST-INDEX GAP	(2)
N2 -- SYNC BYTE COUNT 00	00 N3 -- POST-ID GAP	(2)
N4 -- POST-DATA GAP 0F	01 N5 -- ADDR MARK CNT	(2)
SECTOR LENGTH CODE 01	E5 FILL BYTE	(2)
RESERVED (3 WORDS) 0000		
DRIVE TYPE 03	04 NUMBER OF SURFACES	
# OF SECTORS/TRACK 20	00 STEPPING RATE	
HEAD SETTLING TIME 00	00 HEAD LOAD TIME	

FIGURE C-6. Sample ECA Block Format - Shugart 1000
8-inch Hard disk (8.64 MHz)

15 8	7 0	
SEEK TYPE 02	00 RESERVED FOR CONTROLLER	
LOW WRITE CURRENT BOUNDARY TRACK 0080		
PRECOMPENSATION BOUNDARY TRACK 0080		
ECC REMAINDER (3 WORDS) 0000		(3)
APPEND ECC REMAINDER FROM DISK (3 WORDS) 0000		(3)
RESERVED (2 WORDS) 0000		
MVME320 WORKING AREA (6 WORDS) 0000		
RESERVED FOR THE CONTROLLER		

NOTES:

(1) Physical starting sector number

(2) Track format fields (see section 8)

(3) Last word (don't care)

4. All numbers are hexadecimal initial ECA values.

5. This is an example, assuming the system memory address has been selected as 000C0000...000FFFFF for a 256K memory module.

FIGURE C-6. Sample ECA Block Format - Shugart 1000 8-inch Hard Disk (8.64 MHz) (cont'd)

APPENDIX D

PROGRAMMING CONCEPTS AND SEQUENCE

D.1 INTRODUCTION

Appendix D provides programming concepts and sequence and a command execution procedure for the MVME320.

D.2 PROGRAMMING CONCEPTS

In the VMEbus I/O space, the MVME320 is assigned a base address of the form:

ZZZR

where:

ZZZ = address bits A4 - A9 (don't care)
and A10 - A15 (from JW6)

R = any register number (odd numbers from 1 - D)

For example:

3C01 = register 1 at address 3C01

```
 15   12  11    8  7    4  3    0
+---------+----------+---------+---------+
| 0 0 1 1 | 1 1 X X  | X X X X | R R R R |
+---------+----------+---------+---------+
 \             /
  \__from JW6_/
```

where:

X = don't care
R = register number (binary expression of R above)
JW6 = jumper network JW6 (at location H1)

D.3 PROGRAMMING SEQUENCE

Any time after system power-up or system reset, and before executing a command for the first time, the host must:

- Initialize the ECA
- Set the ECA pointer (Registers 1, 3, 5, 7)
- Set the interrupt acknowledge status/ID byte (Reg 9)
- Set the interrupt acknowledge address in the trap area (if appropriate to the host, as described below)

a. Initialize the ECA - To create the ECA in main memory, the host must initialize the following portions of the ECA (see Figure 5-1).

- Command code
- Maximum number of retries
- Command options
- DMA type
- Buffer address most significant word
- Buffer length requested
- Cylinder number
- Sector number
- Head or surface number
- Post-data gap
- Fill byte
- Sector length
- Number of surfaces
- Drive type
- Stepping rate
- Number of sectors per track
- Head load time
- Head settling time
- Seek type
- Low write current boundary track
- Precompensation boundary track

The following portions of the ECA may be changed by the MVME320.

- Main status (The MVME320 sets main status to busy in its own memory. the host doesn't need to initialize main status. However, if the host wants main status to be meaningful during command execution, the host must set main status to busy before starting to execute a command.)
- Extended status
- Actual number of bytes transferred
- Current cylinder position
- ECC remainder
- Append ECC remainder from disk
- All reserved areas

D

b. Set the ECA Pointer (Registers 1, 3, 5, and 7) - This tells the MVME320 where the ECA was created in main memory.

c. Set the Interrupt Acknowledge Status/ID Bit (Register 9) - When the MVME320 wants to interrupt, it activates the interrupt line. The host acknowledges the interrupt request, and the MVME320 returns the status/ID byte from register 9.

d. Set the Interrupt Acknowledge Address in the Trap Area - If the host system is an MC68000-based system and if the system is interpreting the interrupt acknowledge status/ID bits as an interrupt vector number, the interrupt acknowledge address in the trap area must be initialized.

D.4 COMMAND EXECUTION

D

The following sequence is required each time a command is executed.

a. Check the busy bit in register D. Wait for 0.

b. Set main status in ECA to nusy.

c. Fill in and set up ECA.

d. Set busy bit in register D.

e. Wait for main status in ECA to change (polling mode)

or

Wait for interrupt and check register B for interrupting unit (interrupt mode).

f. If main status $\neq$ 0, then handle errors.

g. Clear busy bit in register D.

SUGGESTION/PROBLEM REPORT

MICROSYSTEMS
QUALITY • PEOPLE • PERFORMANCE

Motorola welcomes your comments on its products and publications. Please use this form.

To: Motorola Inc.
Microsystems
2900 S. Diablo Way
Tempe, Arizona 85282
Attention: Publications Manager
Maildrop DW164

Product: ______________________ Manual: ______________________

COMMENTS: ______________________

Please Print

Name ______________________ Title ______________________

Company ______________________ Division ______________________

Street ______________________ Mail Drop ____________ Phone ____________

City ______________________ State ____________ Zip ____________

For Additional Motorola Publications
Literature Distribution Center
616 West 24th Street
Tempe, AZ 85282
(602) 994-6561

Four Phase/Motorola Customer Support, Tempe Operations
(800) 528-1908
(602) 438-3100

MOTOROLA

MOTOROLA Semiconductor Products Inc.

P.O. BOX 20912 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.

17331 PRINTED IN USA (2/85) MPS 2M