MVME319/D2

MVME319
Intelligent Disk/Tape Controller
User's Manual

MOTOROLA

MVME319

INTELLIGENT DISK/TAPE CONTROLLER

USER'S MANUAL

(MVME319/D2)

PREFACE

This manual provides general information, hardware preparation, installation instructions, functional description, operating instructions, and support information for the MVME319 Intelligent Disk/Tape Controller.

This manual is intended for anyone who wants to design OEM systems, supply additional capability to an existing compatible system, or in a lab environment for experimental purposes.

A basic knowledge of computers and digital logic is assumed.

To use this manual, you should be familiar with the publications listed in the *related documentation* paragraph in Chapter 1 of this manual.

Throughout this manual the paragraph headings conform to the following convention:

IDTC COMMANDS	(this is a main topic heading)
General Commands	(this is a subordinate topic heading under a main topic)
Configure IDTC	(this is a subordinate topic heading under the subordinate topic)

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 July 1988

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 Field Service Division 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.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

TABLE OF CONTENTS (cont'd)

Page

TABLE OF CONTENTS (cont'd)

TABLE OF CONTENTS (cont'd)

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1 - GENERAL INFORMATION

INTRODUCTION

This manual provides general information, hardware preparation, installation instructions, functional description, operating instructions, and support information for the MVME319 Intelligent Disk/Tape Controller.

FEATURES

The features of the MVME319 include:

- MC68121 Intelligent Peripheral Controller (IPC)
- VMEbus requester with four selectable priority levels
- VMEbus interrupter with seven selectable priority levels
- SYSFAIL driver and front panel indicator
- DMA controller
- 16K onboard firmware to support the IPC
- 32K onboard dual-ported RAM to support the IPC
- Eight-bit microprocessor built into the IPC
- Sixteen-bit timer built into the IPC

SPECIFICATION

The MVME319 specifications are identified in Table 1-1.

1

TABLE 1-1. MVME319 Specifications

CHARACTERISTICS	SPECIFICATION
Microprocessor	MC68121 IPC
Memory size capability	
RAM	32Kb
EPROM	16Kb
Indicators	FAIL LED on front panel
Operating temperature	0 degrees C to +55 degrees C at point of entry of forced air (approximately 150 LFM)
Storage temperature	-40 degrees C to +100 degrees C
Operating humidity	0% to 90% (non-condensing)
Power requirements	+5 Vdc +/- 5%, 3.7 A (typ), 4.6 A (max.) +12 Vdc +/- 5%, 30 mA (typ), 50 mA (max.) -12 Vdc +/- 5%, 20 mA (typ), 35 mA (max.)
Physical characteristics	Double height VME board with front panel
Board dimensions	
Height	233.35 mm (9.187 in.)
Depth	160 mm (6.299 in.)
Front panel	
Height	262 mm (10.48 in.)
Width	20 mm (0.8 in.)

Cooling Requirements

The Motorola MVME319 is specified, designed, and tested to operate reliably with an incoming air temperature range from 0 degrees C to 55 degrees C (32 degrees to 131 degrees F) with forced air cooling at a velocity typically achievable by using a 71 CFM axial fan. Temperature qualification is performed in a standard Motorola VMEsystem 1000 chassis. Twenty-five watt load boards are inserted in two card slots, one on each side, adjacent to the board under test, to simulate a high power density system configuration. An assembly of three axial fans, rated at 71 CFM per fan, is placed directly under the VME card cage. The incoming air temperature is measured between the fan assembly and the card cage, where the incoming airstream first encounters the module under test. Test software is executed as the module is subjected to ambient temperature variations. Case temperatures of critical, high power density integrated circuits are monitored to ensure component vendors specifications are not exceeded.

While the exact amount of airflow required for cooling depends on the ambient air temperature and the type, number, and location of boards and other heat sources, adequate cooling can usually be achieved with 150 LFM flowing over the module. Less airflow is required to cool the module in environments having lower maximum ambients. Under more favorable thermal conditions, it may be possible to operate the module reliably at higher than 55 degrees C with increased airflow. It is important to note that there are several factors, in addition to the rated CFM of the air mover, which determine the actual volume and speed of air flowing over a module.

GENERAL DESCRIPTION

The MVME319 is an intelligent disk/tape controller containing a complete microcomputer which can control up to eight hard disk controllers via the SASI/SCSI host adapter and either four floppy disk drives or one FloppyTape drive plus two floppy disk drives directly. The onboard processor, supported by firmware in ROM, an interrupt controller, a DMA controller, and 32K of dual-ported RAM, relieves the system processor of disk/tape controlling tasks.

Commands and status messages are transferred through a dual-ported RAM which is shared between the operating system and the MVME319 firmware. MVME319 drivers are available for VERSAdos and SYSTEM V/68.

RELATED DOCUMENTATION

The following publications may provide additional helpful information. If not shipped with this product, they may be purchased from Motorola Literature Distribution Center, 616 West 24th Street, Tempe, AZ 85282; telephone (602) 994-6561.

DOCUMENT TITLE	MOTOROLA PUBLICATION NUMBER
MC68120/MC68121 Intelligent Peripheral Controller User's Manual	M68120UM
M68000/IPC Command Channel Software Interface Reference Manual	M68KIPCS
The VMEbus Specification	HB212/D

1

DOCUMENT TITLE	MOTOROLA PUBLICATION NUMBER

NOTE: Although not shown in the above list, each Motorola MCD manual publication number is suffixed with characters which represent the revision level of the document, such as "/D2" (the second revision of a manual); supplement bears the same number as the manual but has a suffix such as "/A1" (the first supplement to the manual).

The following publications are available from the sources indicated.

ACB 4000 5-1/4" Winchester Disk Controller OEM Manual; ADAPTEC Inc., 580 Cottonwood Drive, Milpitas, CA 95035.

Series 525 FloppyTape Cartridge Tape Drive Product Description; Cipher Data Products Inc., 10225 Willow Creek Rd., San Diego, CA 92131.

SASI Bus Specification; Shugart Corp., 475-T Oakmead Pky, Sunnyvale, CA 94086.

XEBEC S1410/S1410A 5.25 Inch Winchester Disk Controller Manual; XEBEC Systems Inc., 432-T Lakeside Drive, Sunnyvale, CA 94086.

MANUAL TERMINOLOGY

Throughout this manual, a convention has been maintained whereby data and address parameters are preceded by a character which specifies the numeric format as follows:

$	dollar	specifies a hexadecimal number
%	percent	specifies a binary number
&	ampersand	specifies a decimal number

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 high to low transition.

In this manual, assertion and negation are used to specify forcing a signal to a particular state. In particular, assertion and assert refer to a signal that is active or true; negation and negate indicate a signal that is inactive or false. These terms are used independently of the voltage level (high or low) that they represent.

CHAPTER 2 - HARDWARE PREPARATION AND INSTALLATION INSTRUCTIONS

INTRODUCTION

This chapter provides unpacking, hardware preparation, and installation instructions for the MVME319.

UNPACKING INSTRUCTIONS

NOTE

If shipping carton is damaged upon receipt, request carrier's agent be present during unpacking/inspection of equipment.

Unpack equipment from shipping carton. Refer to packing list and verify that all items are present. Save packing material for storing or reshipping the equipment.

HARDWARE PREPARATION

To select the desired configuration and ensure proper operation of the MVME319, certain modifications may be made to the module. These modifications are made through jumper arrangements. Header locations are shown in Figure 2-1. The module is shipped with factory-installed jumper configurations which make the module ready for single disk operation in a VMEbus system.

The header functions and possible options are listed in Table 2-1.

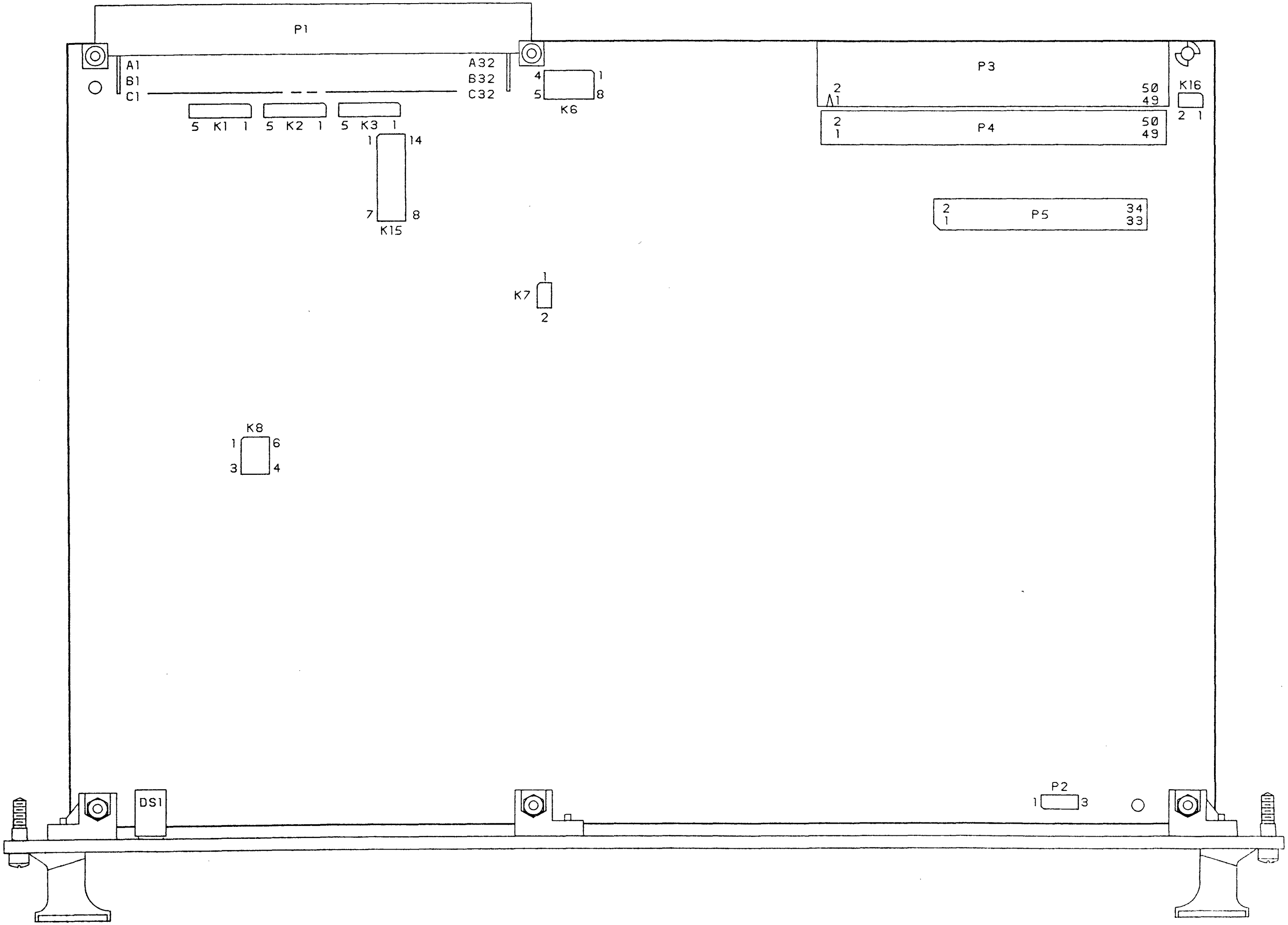


FIGURE 2-1. MVME319 Header Locations

2

TABLE 2-1. MVME319 Header Functions

HEADER	FUNCTION	OPTIONS
K1,K2,K3	VMEbus requester priority level	Select level 0, 1, 2, or 3
K6,K8	VMEbus interrupter	Select level 0 through 7
K7	System fail enable	Enable/disable SYSFAIL* output
K15	Command channel base address	Select any 512 byte boundary in the short I/O address space
K16	Drive select 3 enable	Select DRV3* or READY* on P5 pin 6
P2	Factory use only	

VMEbus Requester Priority Level Select Headers (K1,K2,K3)

Headers K1, K2, and K3 determine the VMEbus requester priority level. Four levels are available. Refer to the table below. On header K3 the bus request output of the VMEbus requester is connected to the appropriate bus request line, on K1 and K2 the VMEbus requester is placed in the corresponding bus grant daisy-chain. The as-shipped factory configuration is level 3.

In VME systems containing a single level bus arbiter, the VMEbus bus request must be configured for level 3. For use with a multi-level arbiter, any one of the four priority levels may be selected.

In multi-level arbitration systems, the bus grant daisy-chain signals which are not used by the MVME319 should be jumpered on the VMEbus backplane.

Headers K1, K2, and K3 are shown in the figure and table below.

```
   +---+
 5 | o |
   |   |
 4 | o |
   |   |
 3 | o |   K1
   |   |
 2 | o |
   | | |
 1 | o |
   +---+
 5 | o |
   |   |
 4 | o |
   |   |
 3 | o |   K2
   |   |
 2 | o |
   | | |
 1 | o |
   +---+
 5 | o |
   |   |
 4 | o |
   |   |
 3 | o |   K3
   |   |
 2 | o |
   | | |
 1 | o |
   +---+
```

K1,K2,K3 CONNECTIONS	BUS REQUEST PRIORITY LEVEL
K1: 1-5, K2: 1-5, K3: 1-5	level 0
K1: 1-4, K2: 1-4, K3: 1-4	level 1
K1: 1-3, K2: 1-3, K3: 1-3	level 2
K1: 1-2, K2: 1-2, K3: 1-2	level 3

VMEbus Interrupter Priority Level Select Headers (K6,K8)

Headers K6 and K8 determine the priority level of the VMEbus interrupter. On K6 the interrupt request output of the VMEbus interrupter is connected to the appropriate interrupt request line. K8 presents the binary code of the interrupt level to the level comparator. The as-shipped factory configuration is level 3.

```
          K6                              K8
       +-------+                        3  2  1
     5 | o   o | 4                   +-----------+
       |       |                     | o   o   o |
     6 | o +-o | 3                   |         | |
       |   |   |                     | o   o   o |
     7 | o | o | 2                   +-----------+
       |   |   |                       4   5   6
     8 | o-+ o | 1
       +-------+
```

K6, K8 CONNECTIONS	BUS INTERRUPT PRIORITY LEVEL
K6: 4-8, K8: 1-6, 2-5	level 1
K6: 5-8, K8: 1-6, 3-4	level 2
K6: 3-8, K8: 1-6	level 3
K6: 6-8, K8: 2-5, 3-4	level 4
K6: 2-8, K8: 2-5	level 5
K6: 7-8, K8: 3-4	level 6
K6: 1-8, K8: none	level 7

System Fail Enable Select Header (K7)

A self test failure or a severe firmware malfunction on the MVME319 lights the FAIL LED on the front panel. This local failure signal is optionally output to the VMEbus SYSFAIL* line when a jumper is installed on header K7. The as-shipped factory configuration is with SYSFAIL* output enabled.

```
      K7
  +-------+
  | o---o |
  +-------+
    1   2
```

K7 CONNECTION	VMEbus SYSTEM FAIL OUTPUT
1-2	SYSFAIL* output enabled
none	SYSFAIL* output disabled

Command Channel Base Address Select Header (K15)

The command channel consists of 256 consecutive words in the VMEbus short supervisory I/O address space. The MVME319 decodes the address modifier code $2D and the VMEbus address lines A09-A15.

Header K15 determines the base address of the MVME319 command channel relative to the base address of the VMEbus short supervisory I/O address space. Each of the VMEbus address lines A09-A15 is represented by one pin on K15 and may be chosen to be low (with jumper installed) or high (with jumper removed) for selecting the module. This allows the command channel to be placed on any 512 byte boundary within the 64Kb short supervisory I/O address space. The as-shipped factory configuration is command channel base address = $0000.

```
                K15

   7   6   5   4   3   2   1
 +---------------------------+
 | o   o   o   o   o   o   o |
 | |   |   |   |   |   |   | |
 | o   o   o   o   o   o   o |
 +---------------------------+
   8   9  10  11  12  13  14
  A09 A10 A11 A12 A13 A14 A15
```

K15 CONNECTION	BASE ADDRESS OFFSETS
1-14 removed	Base address offset = $8000
2-13 removed	Base address offset = $4000
3-12 removed	Base address offset = $2000
4-11 removed	Base address offset = $1000
5-10 removed	Base address offset = $0800
6-9 removed	Base address offset = $0400
7-8 removed	Base address offset = $0200

Drive 3 Enable Select Header (K16)

With respect to 5-1/4 inch floppy disk drives, there is an inconsistency of the READY* signal location at the drive connector. Some older drives use pin 6 for the READY* signal and support only three drive select lines, whereas most modern drives locate the READY* signal at pin 34 and use pin 6 for a fourth drive select line. To support disk drives of either type, the MVME319 carries the drive select signal DRV3* via header K16 to pin 6 of connector P5.

If 5-1/4 inch floppy disk drives with the READY* signal at pin 6 are used in the system, the jumper at header K16 must be removed, and only up to three 5-1/4 inch floppy disk drives may be installed.

CAUTION

INCORRECT CONFIGURATION OF HEADER K16 MAY CAUSE SEVERE SYSTEM MALFUNCTIONS BY SHORTING DRV3* AND READY*.

Whenever the READY* signal is not available, MVME319 must be initialized to use the INDEX* signal for detecting drive readiness (refer to Chapter 4 for details).

The as-shipped factory configuration is with DRV3* enabled.

```
 K16
+---+
| o |
| | |
| o |
+---+
```

K16 CONNECTION	CONNECTOR P5 CONFIGURATION
1-2	DRV3* enabled. Only valid if no 5-1/4 inch floppy disk drives with READY* at pin 6 are used.
none	DRV3* disabled. Jumper must be removed if any 5-1/4 inch floppy disk drive uses pin 6 for READY*.

Factory Use Only (P2)

Connector is used for serial data transfer during factory testing of the MVME319.

INSTALLATION INSTRUCTIONS

When the MVME319 has been configured as desired by the user, it can be installed in the system as follows:

a. Turn all equipment power OFF and disconnect power cable from ac power source.

<u>CAUTION</u>

CONNECTING MODULES WHILE POWER IS APPLIED MAY RESULT IN DAMAGE TO COMPONENTS ON THE MODULE.

<u>WARNING</u>

DANGEROUS VOLTAGES, CAPABLE OF CAUSING DEATH, ARE PRESENT IN THIS EQUIPMENT. USE EXTREME CAUTION WHEN HANDLING, TESTING, AND ADJUSTING.

b. Remove chassis cover as instructed in the equipment user's manual.

c. Remove the filler panel(s) from the appropriate card slot(s) at the front of the chassis. Do not install in card slot 1.

d. Insert the MVME319 into the selected card slot. Be sure module is seated properly into the connectors on the backplane. Fasten the module in the chassis with the screws provided.

e. Connect the appropriate controllers and disk drives as instructed in the following paragraphs and the controller or disk drive manual.

f. At the MVME319 slot on the backplane, remove the IACKIN*/IACKOUT* jumper and the BGIN*/BGOUT* jumper for the request level at which the MVME319 is configured.

g. Replace cover (if removed).

h. Turn equipment power ON.

2

Hard Disk Configuration

The MVME319 is capable of driving up to eight non-arbitrating hard disk controllers, each of them being connected with one or two hard disk drives. Supported hard disk controllers are XEBEC S1410/S1410A or ADAPTEC ACB 4000.

An example of a hard disk configuration with the XEBEC S1410 is shown in Figure 2-2.

A 50-conductor flat ribbon cable provides the SASI/SCSI bus between connector P3 on the MVME319 and the hard disk controllers. It is recommended to use a 3M 3425-3000 connector at P3. The connectors used at the hard disk controllers must be chosen according to the controller specifications. The total length of the connector/cable assembly should not exceed 10 feet.

The SASI/SCSI bus signals driven by the MVME319 must be terminated on the last hard disk controller in the line, using a 220/330 ohm resistor network.

For each hard disk controller, the position of the address jumper specifies the logical controller number used in the MVME319 commands as follows:

Jumper ADDR0 installed: Controller number = 0
Jumper ADDR1 installed: Controller number = 1
Jumper ADDR2 installed: Controller number = 2
Jumper ADDR3 installed: Controller number = 3
Jumper ADDR4 installed: Controller number = 4
Jumper ADDR5 installed: Controller number = 5
Jumper ADDR6 installed: Controller number = 6
Jumper ADDR7 installed: Controller number = 7

The address jumpers must be configured according to the hard disk controller installation instructions. For selecting the controller numbers, the user may choose any order along the line and if less than eight controllers are connected, any subset of controller numbers. However, duplicate selections must be avoided.

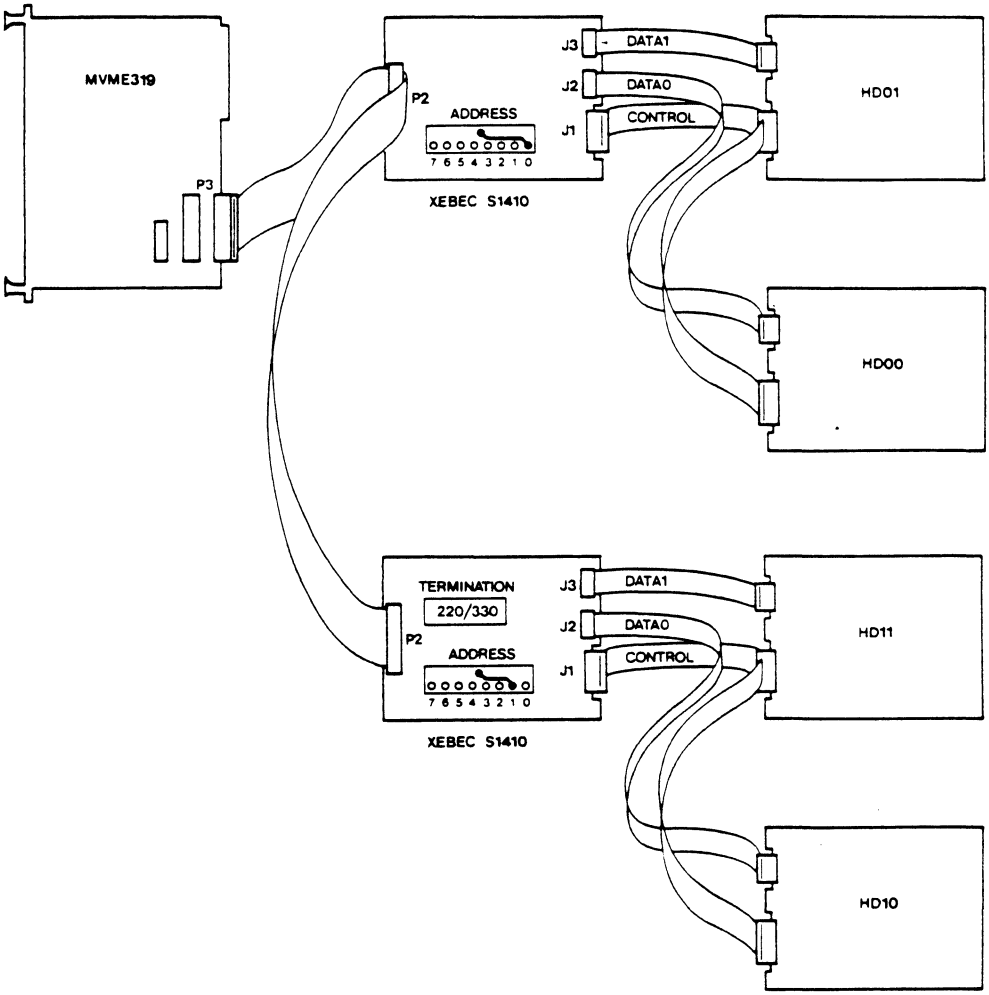


FIGURE 2-2. Hard Disk Configuration

Floppy Disk/Tape Configuration

The MVME319 is capable of controlling up to four 5-1/4 inch and/or 8 -inch floppy disk drives, or two floppy disk drives plus one Cipher CT 525 FloppyTape drive. The following paragraphs give installation instructions for 8-inch only, 5-1/4 inch only, mixed size floppy disk, and FloppyTape configurations with either 8-inch or 5-1/4 inch floppy disk drives.

8-Inch Floppy Disk Configuration

The 8-inch floppy disk drives are connected to the MVME319 according to the example shown in Figure 2-3. A 50-conductor flat ribbon cable provides the bus between connector P4 on the MVME319 and the disk drives. The 50-pin dual row female connector used at P4 must not exceed a maximum height of 0. 475 in. (12 mm) to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M 3425-3000 connector at P4. The connectors used at the disk drives must be chosen according to the drive specifications. The total length of the connector/cable assembly should not exceed 10 feet.

The signals driven by the MVME319 must be terminated on the last disk drive in the line, using a 220/330 ohm resistor network. On all intermediate disk drives, the termination resistors must be removed.

For each disk drive, the position of the drive select jumper specifies the logical drive number used in the MVME319 commands as follows:

Jumper DSEL0 installed: Floppy disk drive number = 4
Jumper DSEL1 installed: Floppy disk drive number = 5
Jumper DSEL2 installed: Floppy disk drive number = 6
Jumper DSEL3 installed: Floppy disk drive number = 7

The drive select jumpers must be configured according to the floppy disk drive installation instructions. For selecting the drive numbers, the user may choose any order along the line and if less than four drives are connected, any subset of drive numbers. However, duplicate selections must be avoided.

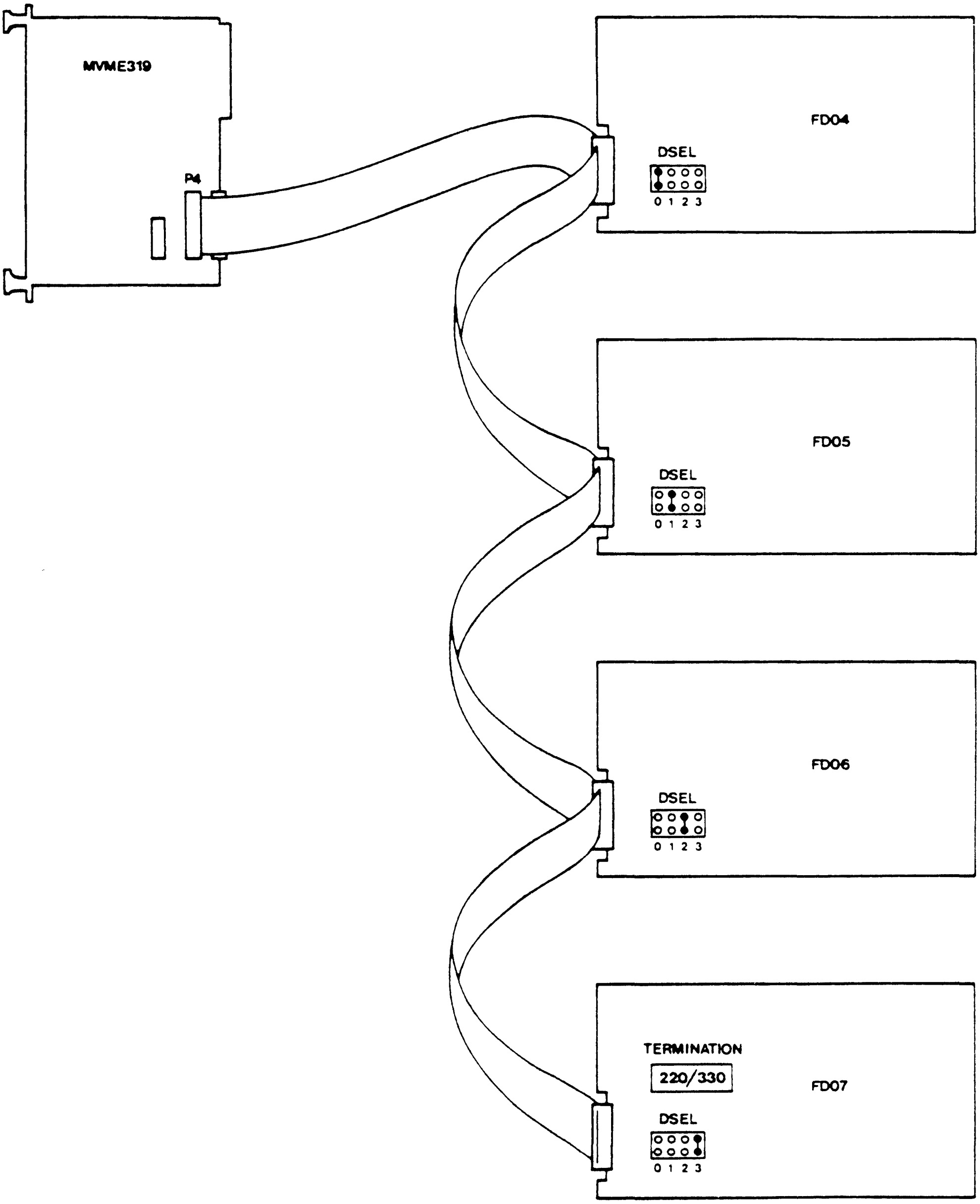


FIGURE 2-3. 8 -Inch Floppy Disk Configuration

2

5-1/4 Inch Floppy Disk Configuration

The 5-1/4 inch floppy disk drives are connected to the MVME319 according to the example shown in Figure 5-1. A 34-conductor flat ribbon cable provides the bus between connector P5 on the MVME319 and the disk drives. The 34-pin dual row female connector used at P5 must not exceed a maximum height of 0.475 in. (12 mm) to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M 3414-3000 connector at P5. The connectors used at the disk drives must be chosen according to the drive specifications. The total length of the connector/cable assembly should not exceed 10 feet.

The signals driven by the MVME319 must be terminated on the last disk drive in the line, using a 220/330 ohm resistor network. On all intermediate disk drives the termination resistors must be removed.

For each disk drive, the position of the drive select jumper specifies the logical drive number used in the MVME319 commands as follows:

Jumper DSEL0 installed: Floppy disk drive number = 4
Jumper DSEL1 installed: Floppy disk drive number = 5
Jumper DSEL2 installed: Floppy disk drive number = 6
Jumper DSEL3 installed: Floppy disk drive number = 7

The drive select jumpers must be configured according to the floppy disk drive installation instructions. For selecting the drive numbers, the user may choose any order along the line and if less than four drives are connected, any subset of drive numbers. However, duplicate selections must be avoided.

If 5-1/4 inch floppy disk drives shall be installed which use pin 6 of the connector for the READY* signal, refer to the *Drive 3 Enable Select Header (K16)* paragraph in this chapter for further configuration details.

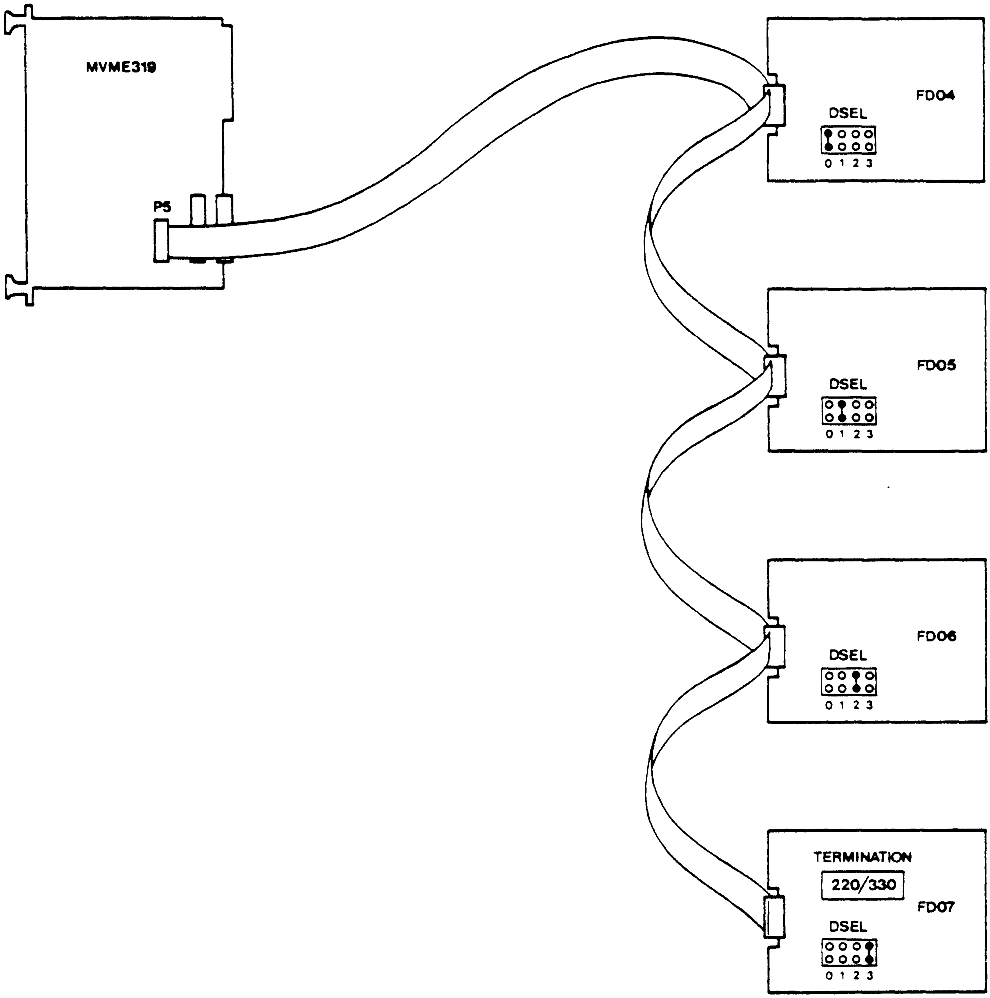


FIGURE 2-4. 5-1/4 Inch Floppy Disk Configuration

2

Mixed Size Floppy Disk Configuration

Mixed mode operation is not recommended, because some peripheral devices are unable to operate reliably under the conditions which it presents. Termination of signals driven by the MVME319 does not conform to accepted practice, which may cause unacceptably high noise levels. If mixed mode operation is necessary, careful testing should be performed to ensure that error rates do not exceed acceptable limits.

Multiple floppy disk drives including both 8-inch and 5-1/4 inch size are connected to the MVME319 according to the example shown in Figure 2-5. A 50-conductor flat ribbon cable provides the bus between connector P4 on the MVME319 and the 8-inch disk drives. A 34-conductor flat ribbon cable provides the bus between connector P5 on the MVME319 and the 5-1/4 inch disk drives. The dual row female connectors used at P4 and P5 must not exceed a maximum height of 0.475 in. (12 mm) to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M 3425-3000 connector at P4 and a 3M 3414-3000 connector at P5. The connectors used at the disk drives must be chosen according to the drive specifications. The total length of each connector/cable assembly should not exceed 10 feet.

On both buses, the signals driven by the MVME319 must be terminated on the last disk drive in each line, using a 220/330 ohm resistor network. On all intermediate disk drives the termination resistors must be removed.

For each disk drive, the position of the drive select jumper specifies the logical drive number used in the MVME319 commands as follows:

Jumper DSEL0 installed: Floppy disk drive number = 4
Jumper DSEL1 installed: Floppy disk drive number = 5
Jumper DSEL2 installed: Floppy disk drive number = 6
Jumper DSEL3 installed: Floppy disk drive number = 7

The drive select jumpers must be configured according to the floppy disk drive installation instructions. For selecting the drive numbers, the user may choose any order along the line and if less than four drives are connected, any subset of drive numbers. However, duplicate selections must be avoided.

If 5-1/4 inch floppy disk drives shall be installed which use pin 6 of the connector for the READY* signal, refer to the *Drive 3 Enable Select Header (K16)* paragraph in this chapter for further configuration details.

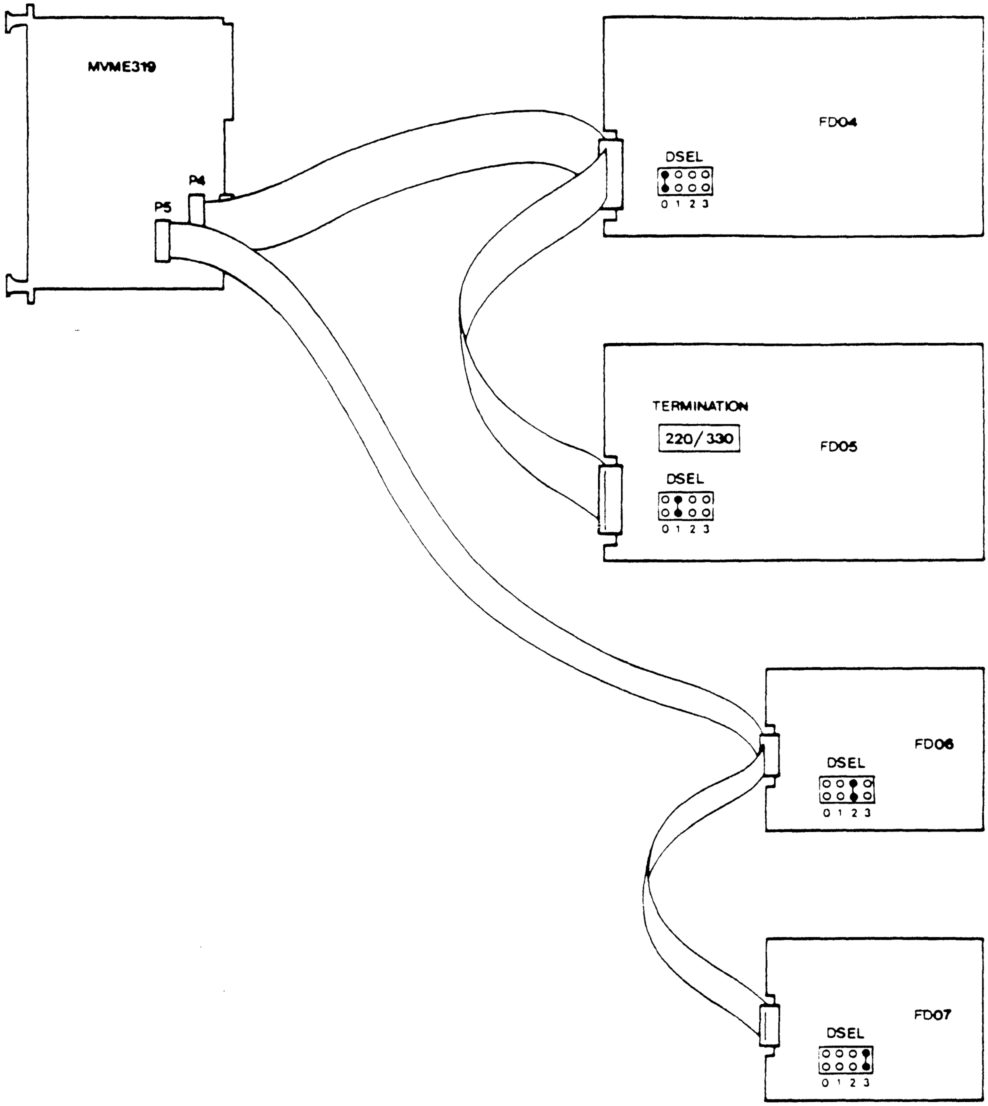


FIGURE 2-5. Mixed Size Floppy Disk Configuration

FloppyTape and 8-Inch Floppy Disk Configuration

A Cipher 525 FloppyTape and up to two 8-inch floppy disk drives are connected to the MVME319 in a daisy-chain configuration according to the example shown in Figure 2-6. A 50-conductor flat ribbon cable provides the bus between connector P4 on the MVME319 and the floppy disk and FloppyTape drives. The 50-pin dual row female connector used at P4 must not exceed a maximum height of 0.475 in. (12 mm) to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M 3425-3000 connector at P4. The connectors used at the disk and tape drives must be chosen according to the drive specifications. The total length of the connector/cable assembly should not exceed 10 feet.

The FloppyTape drive termination resistor network is difficult to remove from its socket. For this reason, it is recommended that the tape drive is the last drive in the chain. All termination resistors on floppy disk drives must be removed.

For each floppy disk drive, the position of the drive select jumper specifies the logical drive number used in the MVME319 commands as follows:

Jumper DSEL2 installed: Floppy disk drive number = 6
Jumper DSEL3 installed: Floppy disk drive number = 7

The FloppyTape drive is accessed as logical drive number 4. The internal jumpers W1 and W2 are installed and DRV0*, DRV1*, and SS1* are used to select the tape stream. This configuration may not be changed.

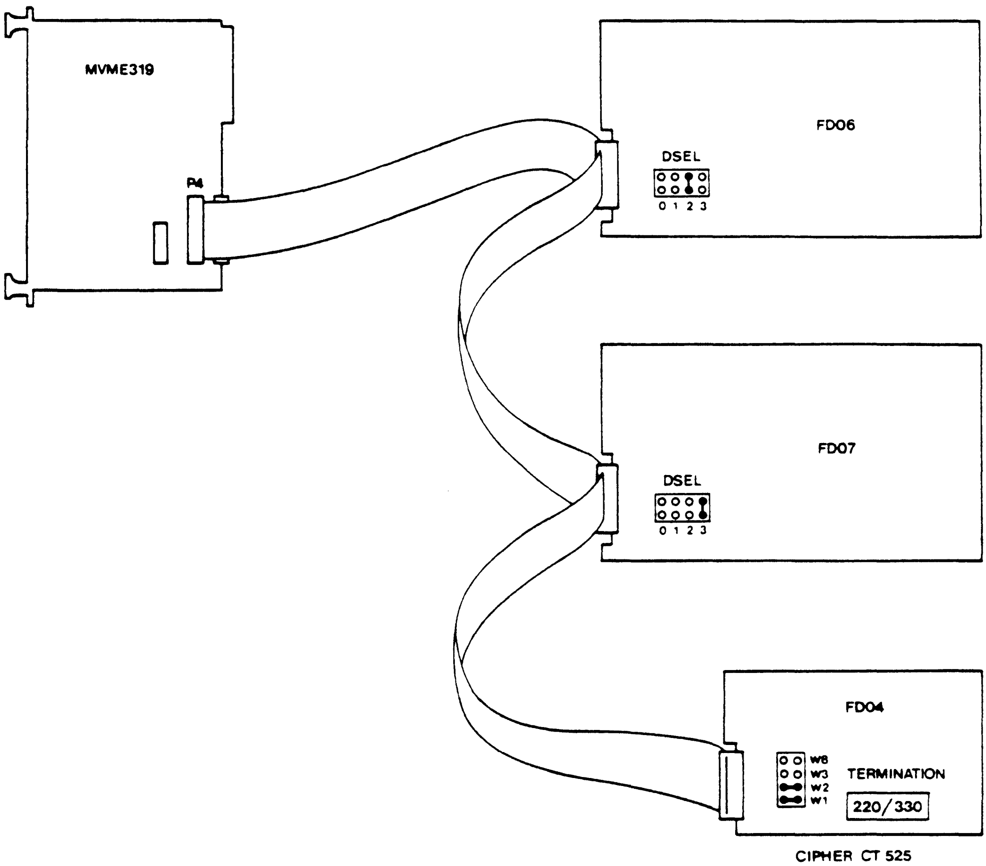


FIGURE 2-6. Tape Drive and 8-Inch Floppy Disk Configuration

FloppyTape and 5-1/4 Inch Floppy Disk Configuration

Mixed mode operation is not recommended, because some peripheral devices are unable to operate reliably under the conditions which it presents. Termination of signals driven by the MVME319 does not conform to accepted practice, which may cause unacceptably high noise levels. If mixed mode operation is necessary, careful testing should be performed to ensure that error rates do not exceed acceptable limits.

A Cipher 525 FloppyTape and up to two 5-1/4 inch floppy disk drives are connected to the MVME319 in a daisy-chain configuration according to the example shown in Figure 2-7. A 50-conductor flat ribbon cable provides the bus between connector P4 on the MVME319 and the FloppyTape drive and a 34-conductor flat ribbon cable between connector P5 on the MVME319 and the 5-1/4 inch floppy disk drives. The 50-pin and 34-pin dual row female connectors used at P4 and P5 must not exceed a maximum height of 0.475 in. (12 mm) to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M 3425-3000 connector at P4 and a 3M 3414-3000 connector at P5. The connectors used at the disk and tape drives must be chosen according to the drive specifications. The total length of the connector/cable assembly should not exceed 5 feet.

The last drive in each chain must be terminated, using a 220/330 ohm resistor network. On the intermediate disk drive the termination resistors must be removed.

For each floppy disk drive, the position of the drive select jumper specifies the logical drive number used in the MVME319 commands as follows:

Jumper DSEL2 installed: Floppy disk drive number = 6
Jumper DSEL3 installed: Floppy disk drive number = 7

The FloppyTape drive is accessed as logical drive number 4. The internal jumpers W1 and W2 are installed and DRV0*, DRV1*, and SS1* are used to select the tape stream. This configuration may not be changed.

If 5-1/4 inch floppy disk drives shall be installed which use pin 6 of the connector for the READY* signal, refer to the *Drive 3 Enable Select Header (K16)* paragraph in this chapter for further configuration details.

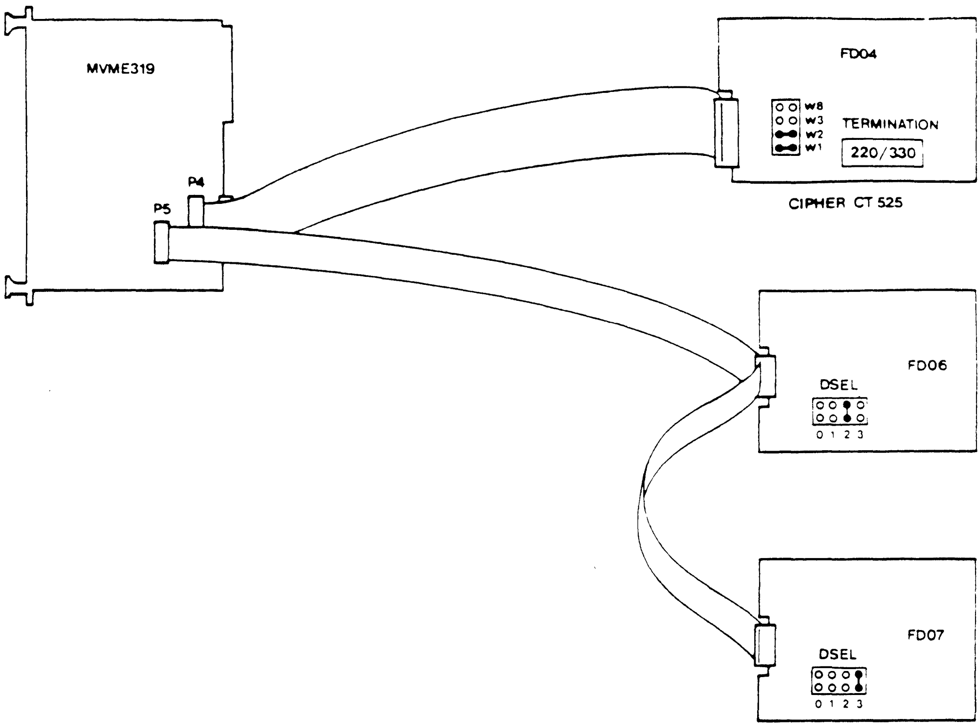


FIGURE 2-7. Tape Drive and 5-1/4 Inch Floppy Disk Configuration

CHAPTER 3 - OPERATING INSTRUCTIONS

INTRODUCTION

This chapter provides the details of the Intelligent Disk/Tape Controller (IDTC) command channel, commands for the hard disk, floppy disk, and FloppyTape drives and status messages generated by the IDTC.

MANUAL TERMINOLOGY

Throughout this manual, a convention has been maintained whereby data and address parameters are preceded by a character which specifies the numeric format as follows:

$	dollar	specifies a hexadecimal number
%	percent	specifies a binary number
&	ampersand	specifies a decimal number

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 high to low transition.

In this manual, assertion and negation are used to specify forcing a signal to a particular state. In particular, assertion and assert refer to a signal that is active or true; negation and negate indicate a signal that is inactive or false. These terms are used independently of the voltage level (high or low) that they represent.

CONTROLS AND INDICATORS

The MVME319 module has a FAIL LED on the front panel.

FAIL Indicator

If the self test program detects any hardware malfunctions, the IDTC lights the FAIL LED. The VMEbus SYSFAIL* signal is connected with the local failure signal via header K7.

A system failure is also indicated in the case of severe IDTC firmware malfunctions. All IDTC functions are supervised by a program time-out counter which is reset periodically by the IDTC firmware executive. The system failure signal is asserted if a program error occurs and the counter is not reset within 2.5 seconds.

When the IDTC has detected a system failure and turned on the FAIL LED, it immediately stops executing and does not accept further commands. The system must be reset for restarting the MVME319 module.

COMMAND CHANNEL

The MVME319 module contains a dual ported RAM which serves as a mailbox for command and status messages to be transferred between the operating system and the IDTC. This command channel is accessed from the VMEbus through an A16:D8 slave interface and is selected with the address modifier code $2D (short supervisory I/O access). In the VME system address map, the command channel appears as the lower (odd) data bytes in 256 consecutive words in the I/O address segment. The base address of the command channel interface within the 64K I/O address segment is selectable on header K15 and may be any 512 byte boundary. The command channel address map as seen from the VMEbus is shown in Table 3-1.

3

TABLE 3-1. IDTC Command Channel Address Map

ADDRESS	EVEN BYTES	ODD BYTES	ADDRESS
$1FE : :		status message buffers	$1FF : : $185
:		CMD ACK/NAK	$183
:		MSG SENT	$181
:		command message buffer	$17F : : $105
:		MSG ACK/NAK	$103
:	even bytes not used	CMD SENT	$101
: : $000		used by IDTC firmware. Not to be accessed from the VMEbus.	0FF : : $001

NOTE: The base address of this map is jumper selectable on 512 byte boundaries in the short I/O address space. Use address modifier code $2D when addressing the IDTC command channel.

In the following description, all command channel addresses are referenced as relative to the command channel base address.

The command channel contains two areas for message transfers and four locations for flags. The operating system may write into the command message buffer ($105-$17F) and into the CMD SENT ($101) and MSG ACK/NAK ($103) bytes. The IDTC may write into the status message buffer ($185-$1FF) and into the MSG SENT ($181) and CMD ACK/NAK ($183) bytes.

3

Although accessible from the VMEbus, locations $000 to $0FF must not be manipulated by the operating system. They are reserved for the IDTC firmware.

To prevent the information contained in the command channel being overwritten unintentionally, the information flow between the operating system and the IDTC has to follow a specified protocol. The IDTC firmware offers the choice between two such protocols: a simple protocol and an extended protocol. The operating system makes the command channel protocol selection when it configures the IDTC during the system initialization (refer to paragraph *Configure IDTC*).

In the simple command channel protocol, each command from the operating system is answered by an appropriate status message from the IDTC, even in the case of a command syntax violation. This protocol does not use the CMD ACK/NAK and MSG ACK/NAK bytes in the command channel.

In the extended command channel protocol, each command and status message may be acknowledged or not acknowledged by the receiving party. Thereby the transmitter of an invalid command or status message can be forced to correct and retransmit the packet immediately.

The Motorola MC68000 operating systems use different command channel protocols in their MVME319 drivers: SYSTEM V/68 uses the simple protocol, VERSAdos uses the extended protocol. Furthermore, the MVME319 disk/tape controller module may be used with any other operating system as long as the specification of one of the command channel protocols is obeyed.

The following paragraphs provide separate and detailed descriptions of both command channel protocols. The command and status message packets transferred between the operating system and the IDTC are described in paragraphs *IDTC Commands* and *IDTC Status Messages*.

Simple Command Channel Protocol

Prior to placing a command into the channel, the operating system tests the CMD SENT byte. A value of $00 indicates that any previous command has been transferred to the IDTC tasks and that the command message buffer is available. Then the operating system places a command packet into the command message buffer and writes $80 into the CMD SENT byte to indicate that a new command has been transmitted.

This write operation interrupts the IDTC processor. The IDTC executive tests the CMD SENT byte to verify the command transmission. Then the command is checked for proper format and syntax. In the case of command invalidity, the IDTC clears the CMD SENT byte and returns the "invalid command" message.

When the IDTC has found a correct command packet in the channel, it transfers the command to the appropriate task and clears the CMD SENT byte to allow new commands to be transferred.

The IDTC terminates any command execution with a status message transfer to the operating system. The protocol of this transfer is very similar to the command transfer described above.

Prior to placing a status message into the channel, the IDTC tests the MSG SENT byte. A value of $00 indicates that any previous status message has been acknowledged by the operating system and that the status message buffer is available. Then the IDTC places a status message packet into the status message buffer, writes $80 into the MSG SENT byte to indicate that a new status message is ready, and interrupts the operating system. Note that this interrupt is omitted for the configure IDTC command.

In the interrupt service routine, the operating system tests the MSG SENT byte to verify the status message transmission, fetches the status message from the command channel, and clears the MSG SENT byte to allow new status messages to be transferred.

A flow diagram of the simple IDTC command channel protocol is shown below.

```
Operating System                      MVME319 IDTC

test CMD SENT
      |
      V
CMD SENT clear:
place command in channel
set CMD SENT
      |
      +-------------------------------------+
                                            |
                                            V
                                      test CMD SENT
                                            |
                                            V
                                      CMD SENT set:
                                +---- test command ----+
                                |                      |
                                V                      V
                          command not ok:         command ok:
                                |                 fetch command
                                V
                          clear CMD SENT          clear CMD SENT
                                |                 execute command
                                |                       |
                                +-----------+-----------+
                                            |
                                            V
                                      test MSG SENT
                                            |
                                            V
                                      MSG SENT clear:
                                      place MSG in channel
                                      set MSG SENT
                                      assert IRQ
                                            |
      +-------------------------------------+
      |
      V
test MSG SENT
      |
      V
MSG SENT set:
fetch message
clear MSG SENT
```

Extended Command Channel Protocol

Prior to placing a command into the channel, the operating system tests the CMD SENT byte. A value of $00 indicates that any previous command has been transferred to the IDTC tasks and that the command message buffer is available. Then the operating system places a command packet into the command message buffer and writes $80 into the CMD SENT byte to indicate that a new command has been transmitted.

This write operation interrupts the IDTC processor. The IDTC executive tests the CMD SENT byte to verify the command transmission. Then the command is checked for proper format and syntax. Depending on the result of this test, the IDTC puts either a "not acknowledged" flag ($15) or an "acknowledged" flag ($06) into the CMD ACK/NAK byte and interrupts the operating system.

In the interrupt service routine, the operating system tests the CMS ACK/NAK byte. In case of a NAK flag, the operating system has to correct the command and retransmit the packet. In either case, the operating system clears the CMD ACK/NAK byte to indicate that the command check result has been read.

When the IDTC has found a correct command packet in the channel, it transfers the command to the appropriate task and clears the CMD SENT byte to allow new commands to be transferred.

The IDTC terminates any command execution with a status message transfer to the operating system. The protocol of this transfer is very similar to the command transfer described above.

Prior to placing a status message into the channel, the IDTC tests the MSG SENT byte. A value of $00 indicates that any previous status message has been acknowledged by the operating system and that the status message buffer is available. Then the IDTC places a status message packet into the status message buffer, writes $80 into the MSG SENT byte to indicate that a new status message is ready, and interrupts the operating system. Note that this interrupt is omitted for the configure IDTC command.

In the interrupt service routine, the operating system tests the MSG SENT byte to verify the status message transmission. Then the status message is checked for proper format and syntax. Depending on the result of this test, the operating system puts either a "not acknowledged" flag ($15) or an "acknowledged" flag ($06) into the MSG ACK/NAK byte.

This write operation interrupts the IDTC processor. The IDTC executive tests the MSG ACK/NAK byte. In case of a NAK flag, the IDTC corrects the status message and retransmits the packet. In either case, the IDTC clears the MSG ACK/NAK byte to indicate that the message check result has been read.

A flow diagram of the extended IDTC command channel protocol is shown below.

```
            Operating System                      MVME319 IDTC

            test CMD SENT
                  |
                  V
            CMD SENT clear:
            place command in channel
            set CMD SENT
                  |
+---------------> +-------------------------------------+
|                                                       |
|                                                       V
|                                                 test CMD SENT
|                                                       |
|                                                       V
|                                                 CMD SENT set:
|                                           +---- test command ----+
|                                           |                      |
|                                           V                      V
|                                     command not ok:         command ok:
|                                     set CMD NAK             set CMD ACK
|                                     assert IRQ              assert IRQ
|                                           |                    |  |
|                 +-------------------------+--------------------+  |
|                 |                                                 |
|                 V                                     +-----------+
|      +-- test CMD ACK/NAK --+                         |
|      |                      |                         V
|      V                      V                   fetch command
| CMD NAK set:           CMD ACK set:             clear CMD SENT
| clear CMD NAK          clear CMD ACK            execute command
| correct command                                       |
|      |                                                V
+------+                                          test MSG SENT
                                                        |
                                                        V
                                                  MSG SENT clear:
                                                  place message in channel
                                                  set MSG SENT
                                                  assert IRQ
                                                        |
                                                        |
                                                        V
```

```
                                                      V
                                                      |
               +--------------------------------------+ <---------------+
               |                                                        |
               V                                                        |
         test MSG SENT                                                  |
               |                                                        |
               V                                                        |
         MSG SENT set:                                                  |
     +-- test message --+                                               |
     |                  |                                               |
     V                  V                                               |
message ok:      message not ok:                                        |
set MSG ACK      set MSG NAK                                            |
     |   |              |                                               |
     |   +--------------+-----------------------+                       |
     |                                          |                       |
     V                                          V                       |
fetch message                       +-- test MSG ACK/NAK --+            |
clear MSG SENT                      |                      |            |
                                    V                      V            |
                               MSG ACK set:           MSG NAK set:      |
                               clear MSG ACK          clear MSG NAK     |
                                                      correct message   |
                                                      assert IRQ        |
                                                           |            |
                                                           +------------+
```

IDTC COMMANDS

IDTC operations are requested from the system MPU by writing commands into the command channel. IDTC commands are variable length packets containing all information required by the IDTC for executing the command autonomously. The general format of the command packets is as follows:

BYTE	CONTENTS	DESCRIPTION
0	$02	Byte 0 of the command packet is marked with the start-of-text character $02.
1	$xx	Byte 1 defines a message ID which is returned within the status message resulting from the command execution. This may be used by the operating system for identifying the status message.
2	$xx	Byte 2 specifies the command packet size in the total number of bytes.

BYTE	CONTENTS	DESCRIPTION
3	$yz	Byte 3 specifies the controller (y) and drive (z) number for the disk operation to be performed. For hard disk operations, valid controller numbers are 0-7, and valid drive numbers are 0 and 1. The IDTC translates the specified controller number into the corresponding address on the SASI/SCSI bus when selecting the controller. The drive number defines the Logical Unit Number (LUN) in the SASI/SCSI command descriptor block. For floppy disk operations, the controller number must be 0, and valid drive numbers are 4-7. Depending on the specified number, the IDTC asserts one of the drive select signals DRV0*-DRV3* on the floppy disk drive interface. For FloppyTape operations, the controller number must be 0 and the drive number must be 4. The FloppyTape drive uses the drive select signals DRV0* and DRV1*, together with SS1*, for the stream selection. The drive number 5 cannot be used with a FloppyTape. The drive numbers 6 and 7 are available for two additional floppy disk drives.
4	$xx	Byte 4 specifies the command type. The IDTC supports the command types "READ" ($10), "WRITE" ($20), "TEST" ($30), "CONTROL" ($40), and "HALT" ($50).
5	$xx	Byte 5 specifies the function code which selects a unique IDTC operation within the command type defined in byte 4.
6-(n-1)	$xx..xx	The following bytes are optional and are used for passing additional parameters to the IDTC. Data transfers from or to a disk or tape are specified by block size, Continuous Block Number (CBN) of the first block to be accessed, and the total number of blocks to be read or written. The block size may be specified as 256 ($100), 512 ($200), or 1024 ($400) bytes, but must at least be the size of the physical sectors on the selected disk or tape. The first block to be accessed is specified by its CBN, which is translated by the IDTC into the corresponding logical hard disk sector address, for floppy disks, into the corresponding side, cylinder, and physical sector numbers and for FloppyTape into the corresponding stream, segment and physical numbers. For further details, refer to Chapter 4.

BYTE	CONTENTS	DESCRIPTION
		Data transfers from or to system memory are specified by starting memory address and address modifier code. The starting address may be any even address within the 16Mb system address range ($000000-$FFFFFE). The address modifier code may be any in the range $01-$3F. The code $00 is a special case and is translated into code $3E (standard supervisory program access) when specified in the command. All further parameters are explained in the detailed command descriptions.
n	$03	The last byte of the command packet is marked with the end-of-text character ($03).

In the following paragraphs, all IDTC commands are described in detail in the order of general commands, hard disk commands, floppy disk commands, and FloppyTape commands.

To assist in tracing hard disk operations, the IDTC hard disk command descriptions contain the resulting SASI/SCSI controller commands. For details of the SASI/SCSI protocol, refer to Chapter 4.

General Commands

The group of general commands comprises commands for IDTC initialization and diagnostics. These commands are executed locally on the IDTC module and do not require any disk or tape drives being connected.

Configure IDTC

The configure IDTC command must be the first command sent to the IDTC after a system power up or reset. This command selects the command channel protocol (simple or extended) and initializes the interrupt status/ID byte of the VMEbus interrupter.

Note that the (simple and extended) protocols for the configure IDTC command itself differ from the following commands by the fact that the IDTC does not assert a VMEbus interrupt request when it returns the status message in the command channel (refer to paragraph 4.3).

For all other commands, the IDTC asserts a VMEbus interrupt request each time it has placed a CMD ACK/NAK byte or a status message in the command channel. The interrupt handler reads the specified status/ID byte during the interrupt acknowledge cycle for initiating the appropriate servicing routine.

3

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$1D	command packet size: 29 bytes
3	$00	filler
4	$50	command type: HALT
5	$00	function code
6	$xx	VMEbus interrupt status/ID byte ($00-$FF)
7	$xx	attribute byte (see below)
8-27	$00..00	reserved
28	$03	ETX mark

Attribute Byte:

BIT	DESCRIPTION
7	command channel protocol: %0 = extended protocol %1 = simple protocol
6-0	%000000 = reserved

Request IDTC Self Test

The request self test command executes the following tests on the IDTC module:

Floppy disk/tape controller:	Non-destructive test of the track, sector, and data registers.
SASI/SCSI bus interface:	Test of the control and data registers.
HARD disk controller:	If the previous test indicated that a hard disk controller is connected, the controller diagnostics command is transmitted. This causes the controller to perform a self test and to return a status message. If no hard disk controller is connected, this part of the IDTC self test is bypassed.

After completion of these tests the IDTC places a status message in the command channel which contains all results (refer to *Command Channel* paragraph).

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$00	filler
4	$30	command type: TEST
5	$FF	function code
6	$03	ETX mark

SASI/SCSI Controller Command: CONTROLLER DIAGNOSTICS

CLASS	OPCODE	DESCRIPTION
7	$04	Perform a controller hardware self test and return the results.

Hard Disk Commands

The group of hard disk commands comprises all commands required for performing hard disk operations. After having received such a command, the IDTC translates it into the corresponding disk controller command and supervises the hard disk controller operation.

3

Read Sectors

The read sectors command transfers data from contiguous blocks on the disk into system memory. The disk access is specified by the CBN of the first block, the number of blocks to be read, and the number of bytes per block. The system memory access is specified by address modifier code and starting address.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$10	command type: READ
5	$01	function code
6-7	$xxxx	number of blocks to be read ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$00xxxxxx	CBN of first block to be read ($000000-$1FFFFF)
20	$03	ETX mark

SASI/SCSI Controller Command: READ

CLASS	OPCODE	DESCRIPTION
0	$08	Read the specified number of sectors starting from the initial sector address given in the CDB.

Write Sectors

The write sectors command transfers data from system memory into contiguous blocks on the disk. The disk access is specified by the CBN of the first block, the number of blocks to be written, and the number of bytes per block. The system memory access is specified by address modifier code and starting address.

3

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$20	command type: WRITE
5	$02	function code
6-7	$xxxx	number of blocks to be written ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$00xxxxxx	CBN of first block to be written ($000000-$1FFFFF)
20	$03	ETX mark

SASI/SCSI Controller Command: WRITE

CLASS	OPCODE	DESCRIPTION
0	$0A	Write the specified number of sectors starting from the initial sector address given in the CDB.

Verify Sectors

The verify sectors command transfers data from contiguous blocks on the disk into local memory, thereby detecting uncorrectable data errors. The disk access is specified by the CBN of the first block, the number of blocks to be verified, and the number of bytes per block. The system memory is not accessed.

3

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 15 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$30	command type: TEST
5	$01	function code
6-7	$xxxx	number of blocks to be verified ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10-13	$00xxxxxx	CBN of first block to be verified ($000000-$1FFFFF)
14	$03	ETX mark

SASI/SCSI Controller Command: READ

CLASS	OPCODE	DESCRIPTION
0	$08	Read the specified number of sectors starting from the initial sector address given in the CDB.

Request Drive Status

The request drive status command first checks the ready signal of the specified drive. then a status message is returned indicating the device type and the total number of physical sectors on the disk (refer to *Drive Status* paragraph).

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$30	command type: TEST
5	$02	function code
6	$03	ETX mark

SASI/SCSI Controller Command: TEST DRIVE READY

CLASS	OPCODE	DESCRIPTION
0	$00	Select the drive specified by LUN and check READY signal. No further operands except the control field are required.

Format Track

The format track command formats one track on the disk. This command can only be used with a XEBEC S1410/S1410A disk controller. The track is specified by the CBN of the first block on the track. Unless otherwise defined by a previous initialize hard disk parameters command, the recording format defaults to the parameters described in the initialize hard disk parameters paragraph. If the XEBEC disk controller does not indicate the completion of the operation in one minute, the command execution is aborted and a controller error message is returned.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 15 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$40	command type: CONTROL
5	$01	function code
6-7	$xxxx	number of blocks per track
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10-13	$00xxxxxx	CBN of first block to be verified ($000000-$1FFFFF)
14	$03	ETX mark

SASI/SCSI Controller Command: FORMAT TRACK

CLASS	OPCODE	DESCRIPTION
0	$06	Format the specified track, clear "bad block" flag in all sectors, set ID fields according to the interleave factor in the CDB, and write $6C into all data fields.

Format Disk

The format disk command formats the complete disk. Unless otherwise defined by a previous initialize hard disk parameters command, the recording format defaults to the parameters described in the initialize hard disk parameters paragraph. This command is applicable for all supported hard disk controllers. When using the ADAPTEC ACB 4000 controller with automatic media detect handling, refer to *Format Disk With Media Detect Handling* paragraph.

Note that the format disk command is not supervised by any time-out function.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$40	command type: CONTROL
5	$02	function code
6	$03	ETX mark

SASI/SCSI Controller Command: FORMAT DRIVE

CLASS	OPCODE	DESCRIPTION
0	$04	Format the complete track, clear "bad block" flag in all sectors, set ID fields according to the interleave factor in the CDB, and write $6C into all data fields.

3

Format Disk With Media Defect Handling

The format disk with media defect handling command is only applicable for the ADAPTEC ACB 4000 controller and allows formatting of a hard disk with media defects by utilizing the defect handling capability of the ACB 4000. The media defect list must be created in the system memory by using the format listed below.

Note that the format disk with media defect handling command is not supervised by any time-out function.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 15 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$40	command type: CONTROL
5	$02	function code
6-7	$0000	filler
8-9	$xxxx	total length of media defect list in bytes
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting address of media defect list in system memory
14	$03	ETX mark

ADAPTEC ACB 4000 Media Defect List

BYTE	DESCRIPTION
0	0
1	0
2-3	Length of defect list in bytes (8*#defects)
4-6	Cylinder number of defect number 1
7	Head number of defect number 1
8-11	Bytes from index
.	.
.	.
.	.
8*N-4	Cylinder number of defect number N
to	Head number of defect number N
8*N+3	Bytes from index

The media defects supplied by the disk drive manufacturer must be inserted into the media defect list in the order of their related logical sector numbers, starting with the lowest sector number. Non-observance of this rule results in an IDTC hang up. Also, track 0 must be free of defects, because it is used for storing disk formatting information.

Restore Head

The restore head command performs a recalibration of the read/write head in the specified drive. If the hard disk controller does not indicate the completion of the operation within 5 seconds, the command execution is aborted and a controller error message is returned.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$40	command type: CONTROL
5	$04	function code
6	$03	ETX mark

SASI/SCSI Controller Command: RECALIBRATE

CLASS	OPCODE	DESCRIPTION
0	$01	Position read/write head on track 0, and clear error status flag in the drive.

3

Seek Track

The seek track command positions the read/write head on the track which contains the specified sector. If the hard disk controller does not indicate the completion of the operation within 5 seconds, the command execution is aborted and a controller error message is returned.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 15 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$40	command type: CONTROL
5	$05	function code
6-7	$0000	filler
8-9	$0100	sector size in bytes
10-13	$00xxxxxx	CBN of sector on track to be sought ($000000-$1FFFFF)
14	$03	ETX mark

SASI/SCSI Controller Command: RECALIBRATE

CLASS	OPCODE	DESCRIPTION
0	$0B	Position read/write head on the track containing the sector specified in the CDB.

Initialize Hard Disk Parameters

The initialize hard disk parameters command initializes the hard disk controller with the parameters of the disk drive used. Each controller connected to the IDTC may be initialized with independent disk parameters.

Note that a parameter set specified for a XEBEC hard disk controller is valid for all disk drives connected to this controller. Each disk drive connected to an ADAPTEC controller may be initialized separately with a unique disk parameter set.

The IDTC firmware generates a disk parameter table for each of the 8 possible hard disk controllers. After power up or reset the following default parameters are effective:

Controller type	= XEBEC S1410
Number of heads per drive	= 4
Number of cylinders per disk	= 153
First cylinder with write precompensation . . .	= 64
First cylinder with reduced write current . . .	= 128
Number of sectors per track	= 32
Sector size in bytes	= 256
ECC data burst length in bits	= 11
Interleave factor	= 13
Stepping rate in milliseconds	= 3

Before accessing a hard disk for the first time after any reset, the corresponding disk parameter table should be updated with the initialize hard disk parameters command, specifying the capacities and characteristics of the actual disk drive.

The following paragraphs give separate descriptions for the XEBEC and ADAPTEC controller initialization commands.

IDTC Command Message for XEBEC S1410/S1410A

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$1D	command packet size: 29 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$40	command type: CONTROL
5	$08	function code
6-7	$0000	filler
8	$xx	number of sectors per track: $20 for sector size = 256 bytes $11 for sector size = 512 bytes
9	$xx	number of heads per drive ($01-$0F)
10-11	$xxxx	physical sector size ($0100, $0200) The physical sector size must also be selected by a jumper on the XEBEC S1410/S1410A controller.

BYTE	CONTENTS	DESCRIPTION
12-13	$xxxx	number of cylinders per disk ($0001-$FFFF)
14-15	$xxxx	first cylinder with write precompensation ($0000-$FFFF) For the XEBEC S1410A the most significant bit also selects the value of the write precompensation: bit 7 of byte 14 = 0 : 5 ns bit 7 of byte 14 = 1 : 10 ns
16-17	$xxxx	first cylinder with reduced write current ($0000-$FFFF)
18	$xx	ECC data burst length in bits ($01-$0F)
19	$00	filler
20	$xx	interleave factor ($01-$[sectors/track-1])
21-23	$000000	filler
24	$xx	attribute byte (see below)
25	$00	filler
26	$00	controller type (XEBEC S1410/S1410A)
27	$00	filler
28	$03	ETX mark

Attribute Byte:

BIT	DESCRIPTION
7-5	%000 = reserved
4	imbedded servo: %0 = drive without imbedded servo %1 = drive with imbedded servo (S1410A only)
3	%0 = reserved
2-0	stepping rate: %000 = 3 ms not buffered %100 = 200 us buffered %101 = 70 us buffered %110 = 30 us buffered %111 = 15 us buffered

Drives with imbedded servo are supported only by the XEBEC S14120A controller.

IDTC Command Message for ADAPTEC ACB 4000

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$1D	command packet size: 29 bytes
3	$yz	controller and drive number (y=0-7; z=0,1)
4	$40	command type: CONTROL
5	$08	function code
6-7	$0000	filler

BYTE	CONTENTS	DESCRIPTION
8	$xx	number of sectors per track: $09 for sector size = 1024 bytes $11 for sector size = 512 bytes and interleave = 1 $12 for sector size = 512 bytes and interleave > 1 $20 for sector size = 256 bytes and interleave = 1 $21 for sector size = 256 bytes and interleave > 1
9	$xx	number of heads per drive ($01-$10)
10-11	$xxxx	physical sector size ($0100, $0200, $0400)
12-13	$xxxx	number of cylinders per disk ($0001-$0800)
14-15	$xxxx	first cylinder with write precompensation ($0000-$07FF) This parameter is ignored by the reduced write current version of the ADAPTEC ACB 4000.
16-17	$xxxx	first cylinder with reduced write current ($0000-$07FF) This parameter is also taken as the write precompensation border by the reduced write current version of the ADAPTEC ACB 4000.
18-19	$0000	filler
20	$xx	interleave factor ($01-$[sectors/track-1]) A value of $00 for this parameter defaults to an interleave factor of 2. Refer to ADAPTEC ACB 4000 manual for details.
21-23	$000000	filler
24	$xx	stepping rate ($00 = 3 ms, $01 = 28 us, $02 = 12 us)
25	$00	filler
26	$02	controller type (ADAPTEC ACB 4000)
27	$00	filler
28	$03	ETX mark

SASI/SCSI Controller Command: INITIALIZE DRIVE CHARACTERISTICS

CLASS	OPCODE	DESCRIPTION
0	$0C	Initialize controller (for XEBEC S1410/S1410A).
0	$15	Mode select (for ADAPTEC ACB 4000).

Floppy Disk Commands

The group of floppy disk commands comprises all commands required for performing floppy disk operations.

Read Sectors With Retry

The read sectors with retry command transfers data from contiguous blocks on the disk into system memory. The disk access is specified by the CBN of the first block, the number of blocks to be read, and the number of bytes per block. The system memory access is specified by address modifier code and starting address. In the case of a seek error or a CRC error, the read operation is repeated up to seven times.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$xx	physical drive number ($04-$07)
4	$10	command type: READ
5	$01	function code
6-7	$xxxx	number of blocks to be read ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

Read Sectors

The read sectors command transfers data from contiguous blocks on the disk into system memory. The disk access is specified by the CBN of the first block, the number of blocks to be read, and the number of bytes per block. The system memory access is specified by address modifier code and starting address. In the case of a seek error or a CRC error, the command execution is aborted without any retry.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$xx	physical drive number ($04-$07)
4	$10	command type: READ
5	$02	function code
6-7	$xxxx	number of blocks to be read ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

Read Deleted Data

The read deleted data command transfers data from contiguous blocks on the disk into system memory without regarding the deleted data mark. The disk access is specified by the CBN of the first block, the number of blocks to be read, and the number of bytes per block. The system memory access is specified by address modifier code and starting address. In the case of a seek error or a CRC error, the read operation is repeated up to seven times.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$xx	physical drive number ($04-$07)
4	$10	command type: READ
5	$0D	function code
6-7	$xxxx	number of blocks to be read ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

Write Sectors With CRC Verify

The write sectors with CRC verify command transfers data from system memory into contiguous blocks on the disk. The disk access is specified by the CBN of the first block, the number of blocks to be written, and the number of bytes per block. The system memory access is specified by address modifier code and starting address. After writing, all sectors are read back for CRC verification. In the case of a seek error or a CRC error during verification, the write operation is repeated once.

3

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$xx	physical drive number ($04-$07)
4	$20	command type: WRITE
5	$01	function code
6-7	$xxxx	number of blocks to be written ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

Write Sectors

The write sectors command transfers data from system memory into contiguous blocks on the disk. The disk access is specified by the CBN of the first block, the number of blocks to be written, and the number of bytes per block. The system memory access is specified by address modifier code and starting address.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$xx	physical drive number ($04-$07)
4	$20	command type: WRITE
5	$02	function code
6-7	$xxxx	number of blocks to be written ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

Write Deleted Data

The write deleted data command transfers data from system memory into contiguous blocks on the disk and sets the deleted data address mark in the data field of each sector. The disk access is specified by the CBN of the first block, the number of blocks to be written, and the number of bytes per block. The system memory access is specified by address modifier code and starting address.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$xx	physical drive number ($04-$07)
4	$20	command type: WRITE
5	$0F	function code
6-7	$xxxx	number of blocks to be written ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be written ($0000-$FFFF)
20	$03	ETX mark

Verify Sectors

The verify sectors command transfers data from contiguous blocks on the disk into local memory, thereby detecting CRC errors in sector headers and data fields. The disk access is specified by the CBN of the first block, the number of blocks to be verified, and the number of bytes per block. The system memory is not accessed. In the case of a CRC error or a seek error, the read operation is repeated once.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 15 bytes
3	$xx	physical drive number ($04-$07)
4	$30	command type: TEST
5	$01	function code
6-7	$xxxx	number of blocks to be verified ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10-13	$0000xxxx	CBN of first block to be verified ($0000-$FFFF)
14	$03	ETX mark

Request Drive Status

The request drive status command first checks the ready signal of the specified drive. Then a status message is returned containing the device type, device status, and disk parameters (refer to *Drive Status* paragraph).

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$xx	physical drive number ($04-$07)
4	$30	command type: TEST
5	$02	function code
6	$03	ETX mark

Read Head Position

The read head position command first checks the ready signal of the selected drive. Then a status message is returned specifying the current position of the read/write head by the LSA of the first sector on the current track.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$xx	physical drive number ($04-$07)
4	$30	command type: TEST
5	$03	function code
6	$03	ETX mark

Format Track

The format track command formats one track on the disk. The track is specified by the CBN of the first block on the track. Unless otherwise defined by a previous initialize floppy disk parameters command, the recording format defaults to the parameters described in the initialize floppy disk parameters paragraph. After formatting, all sectors are read back for CRC verification. In the case of a CRC error or a seek error, the format operation is repeated.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 15 bytes
3	$xx	physical drive number ($04-$07)
4	$40	command type: CONTROL
5	$01	function code
6-7	$xxxx	number of blocks per track ($0001-$001A)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10-13	$0000xxxx	CBN of first block on the track ($0000-$FFFF)
14	$03	ETX mark

Format Disk

The format disk command formats the complete disk. Unless otherwise defined by a previous initialize floppy disk parameters command, the recording format defaults to the parameters described in the initialize floppy disk parameters paragraph. After formatting, all sectors are read back for CRC verification. In the case of a CRC error or a seek error, the format operation is aborted immediately.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$xx	physical drive number ($04-$07)
4	$40	command type: CONTROL
5	$02	function code
6	$03	ETX mark

3

Restore

The restore command checks the parameters of the specified drive. If the parameter table is locked by a previous initialize floppy disk parameters command, the restore command only returns eventual errors. If the parameter table is unlocked and the default parameters are in use, the restore command updates the parameter table of the specified disk drive. In detail, the following steps are performed:

a. The read/write head is positioned on track 0 and the first sector ID encountered is checked. If the verification is successful, no further actions are taken and the currently effective parameters remain unchanged.

b. The disk size is determined by measuring the time interval between two index pulses. An interval of 166 ms indicates an 8-inch drive, an interval of 200 ms indicates a 5-1/4 inch drive.

c. The data density is determined by stepping to track 1 and trying to verify a sector ID in either MFM or FM mode. A successful verification in MFM mode indicates double density, a successful verification in FM mode indicates single density. If both checks fail, an unformatted diskette is assumed, no further actions are taken and, depending on the disk size, the default parameters for either a double sided double density 5-1/4 inch disk or a double sided single density 8-inch disk are selected, according to the table below.

d. The number of disk sides is determined by trying to verify a sector ID on side 1. A successful verification indicates a double sided diskette, an error indicates a single sided diskette.

e. The remaining parameters are selected according to the default values given in the following table:

Disk size	5-1/4 in.	5-1/4 in.	8 in.	8 in.
Number of sides	1 or 2	1 or 2	1 or 2	1 or 2
Data density	single	double	single	double
Physical sector size in bytes	128	256	128	256
Number of sectors per track	16	16	26	26
Number of cylinders	80	80	77	77
Write precompensation	no	no	yes	yes
First cylinder with compensation	--	--	44	44
Interleave factor	1	1	1	1
Spiral offset	0	0	0	0
Drive with READY signal	yes	yes	yes	yes
Format	IBM3740	IBM34	Motorola	Motorola

Restore command does not change configuration between tape and disk.

3

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$xx	physical drive number ($04-$07)
4	$40	command type: CONTROL
5	$04	function code
6	$03	ETX mark

Seek Track

The seek track command positions the read/write head on the track which contains the specified block and verifies the first encountered sector ID. In the case of an error, a restore is executed and seek track command is repeated.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 15 bytes
3	$xx	physical drive number ($04-$07)
4	$40	command type: CONTROL
5	$05	function code
6-7	$0000	filler
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10-13	$0000xxxx	CBN of block on the track to be sought ($0000-$FFFF)
14	$03	ETX mark

3

Initialize Floppy Disk Parameters

The IDTC firmware generates a unique disk parameter table for each floppy disk drive. After system reset, device number 4 is initialized for a FloppyTape. The device numbers 6 and 7 are initialized for floppy disks with the following default parameters:

Disk size in inches	= 8
Number of sides	= 2
Data density	= FM
Physical sector size in bytes	= 128
Number of sectors per track	= 26
Number of cylinders	= 77
Write precompensation	= yes
First cylinder with write precompensation . .	= 44
Interleave factor	= 1
Spiral offset	= 0
Drive with READY signal	= yes
Format	= Motorola

Before accessing a floppy disk for the first time after any reset, the corresponding disk parameter table should be updated with the initialize floppy disk parameters command, specifying the capacities and characteristics of the actual disk drive.

If no FloppyTape is connected and floppy disk drives are connected to physical device numbers 4 and 5, device number 4 must be configured for a floppy disk before device number 5 is configured, even if no drive is connected to device number 4.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$1D	command packet size: 29 bytes
3	$xx	physical device number ($04-$07)
4	$40	command type: CONTROL
5	$08	function code
6	$xx	attribute byte 0 (see below)
7	$xx	attribute byte 1 (see below)
8	$xx	number of sectors per track ($01-$1A)
9	$xx	number of sides ($01-$02)
10-11	$xxxx	physical sector size in bytes ($0080, $0100, $0200, $0400)

BYTE	CONTENTS	DESCRIPTION
12-13	$xxxx	number of cylinders per disk ($0000-$00FF)
14-15	$xxxx	first cylinder with write precompensation ($0000-$00FF)
16-19	$00..00	reserved
20	$xx	interleave factor ($01-$[sectors/track-1])
21-27	$00..00	reserved
28	$03	ETX mark

Attribute Byte 0:

BIT	DESCRIPTION
7	%0 = IBM format (starting sector number on both sides = 1) %1 = Motorola format (starting sector number on side two = number of sectors/track+1)
6	%0 = no spiral offset %1 = spiral offset
5	%0 = single density (FM) %1 = double density (MFM)
4	%0 = single sided media %1 = double sided media
3	%0 = 5-1/4 inch drive with READY signal %1 = 5-1/4 inch drive without READY signal (drive ready detection with INDEX signal)
2	%0 = 5-1/4 inch disk size %1 = 8-inch disk size
1	%0 = precompensation or postcompensation %1 = no compensation
0	%0 = write precompensation (if command byte number 15 is not $00) %1 = read postcompensation

Attribute Byte 1:

BIT	DESCRIPTION
7-3	%00000 = reserved
2-0	stepping rate: %001 = 3 ms (8-inch), 6 ms (5-1/4 inch) %010 = 6 ms (8-inch), 12 ms (5-1/4 inch) %001 = 10 ms (8-inch), 20 ms (5-1/4 inch) %010 = 15 ms (8-inch), 30 ms (5-1/4 inch)

FloppyTape Commands

The group of FloppyTape commands comprises all commands required for performing FloppyTape operations.

Read Sectors with Retry

The read sectors with retry command transfers data from contiguous blocks on the tape into system memory. The tape access is specified by the CBN of the first block, the number of blocks to be read, and the number of bytes per block. The system memory access is specified by address modifier code and starting address. In the case of a seek error or a CRC error, the read operation is repeated up to seven times.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$04	physical device number
4	$10	command type: READ
5	$01	function code
6-7	$xxxx	number of blocks to be read ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

Read Sectors

The read sectors command transfers data from contiguous blocks on the tape into system memory. The tape access is specified by the CBN of the first block, the number of blocks to be read, and the number of bytes per block. The system memory access is specified by address modifier code and starting address. In the case of a seek error or a CRC error, the command execution is aborted without any retry.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$04	physical device number
4	$10	command type: READ
5	$02	function code
6-7	$xxxx	number of blocks to be read ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

3

Read Deleted Data

The read deleted data command transfers data from contiguous blocks on the tape into system memory without regarding the deleted data mark. The tape access is specified by the CBN of the first block, the number of blocks to be read, and the number of bytes per block. The system memory access is specified by address modifier code and starting address. In the case of a seek error or a CRC error, the read operation is repeated up to seven times.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$04	physical device number
4	$10	command type: READ
5	$0D	function code
6-7	$xxxx	number of blocks to be read ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

Write Sectors with CRC Verify

The write sectors with CRC verify command transfers data from system memory into contiguous blocks on the tape. The tape access is specified by the CBN of the first block, the number of blocks to be written, and the number of bytes per block. The system memory access is specified by address modifier code and starting address. After writing, all sectors are read back for CRC verification. In the case of a seek error or a CRC error during verification, the write operation is repeated once.

Note that the verification is carried out after writing a segment and before stepping to a new segment, this causes a reposition cycle to be initiated each time.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$04	physical device number
4	$20	command type: WRITE
5	$01	function code
6-7	$xxxx	number of blocks to be written ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

Do not write to sector number 3 (bad segment table).

Write Sectors

The write sectors command transfers data from system memory into contiguous blocks on the tape. The tape access is specified by the CBN of the first block, the number of blocks to be written, and the number of bytes per block. The system memory access is specified by address modifier code and starting address.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$04	physical device number
4	$20	command type: WRITE
5	$02	function code
6-7	$xxxx	number of blocks to be written ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be read ($0000-$FFFF)
20	$03	ETX mark

Do not write to sector number 3 (bad segment table).

Write Deleted Data

The write deleted data command transfers data from system memory into contiguous blocks on the tape and sets the deleted data address mark in the data field of each sector. The tape access is specified by the CBN of the first block, the number of blocks to be written, and the number of bytes per block. The system memory access is specified by address modifier code and starting address.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$04	physical device number
4	$20	command type: WRITE
5	$0F	function code
6-7	$xxxx	number of blocks to be written ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10	$xx	memory address modifier ($00-$3F)
11-13	$xxxxxx	starting memory address ($000000-$FFFFFF)
14-15	$0000	filler
16-19	$0000xxxx	CBN of first block to be written ($0000-$FFFF)
20	$03	ETX mark

Do not write to sector number 3 (bad segment table).

3

Verify Sectors

The verify sectors command transfers data from contiguous blocks on the tape into local memory, thereby detecting CRC errors in sector headers and data fields. The tape access is specified by the CBN of the first block, the number of blocks to be verified, and the number of bytes per block. The system memory is not accessed. In the case of a CRC error or a seek error, the read operation is repeated once.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 15 bytes
3	$04	physical device number
4	$30	command type: TEST
5	$01	function code
6-7	$xxxx	number of blocks to be verified ($0001-$FFFF)
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10-13	$0000xxxx	CBN of first block to be verified ($0000-$FFFF)
14	$03	ETX mark

Request Drive Status

The request drive status command first checks the ready signal of the specified drive. Then a status message is returned containing the device type, device status, and tape parameters (refer to *Drive Status* paragraph).

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$04	physical device number
4	$30	command type: TEST
5	$02	function code
6	$03	ETX mark

3

Format Tape

The format tape command formats all six streams of the tape. Unless otherwise defined by a previous initialize tape drive parameters command, the recording format defaults to the parameters described in the initialize FloppyTape parameters paragraph. After formatting, a verification pass is carried out and all segments with defective media are logged out. These segments are inserted into a bad segment list and the list is then written onto the 4th sector (sector number 3) of the first tape segment of stream 0, at the end of the command (refer to Chapter 4).

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$04	physical device number
4	$40	command type: CONTROL
5	$02	function code
6	$03	ETX mark

Rewind

The rewind command positions the tape head on stream 0 and rewinds the tape until the first segment on the stream is detected. No sector ID is verified.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$07	command packet size: 7 bytes
3	$04	physical device number
4	$40	command type: CONTROL
5	$04	function code
6	$03	ETX mark

Seek Segment

The seek segment command positions the read/write head on the segment which contains the specified block and verifies the first encountered sector ID. In the case of an error, the tape is rewound and seek segment command is repeated.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 15 bytes
3	$04	physical device number
4	$40	command type: CONTROL
5	$05	function code
6-7	$0000	filler
8-9	$xxxx	block size in bytes ($0100, $0200, $0400)
10-13	$0000xxxx	CBN of block on the segment to be sought ($0000-$FFFF)
14	$03	ETX mark

Initialize FloppyTape Parameters

The IDTC firmware generates a tape parameter table for FloppyTape drive. After system reset, device number 4 is initialized for tape and the following default parameters are effective:

Physical sector size in bytes . . . . = 256
Number of sectors per segment . . . . = 44
Skip factor . . . . = 1

If a different tape configuration is to be used, the actual parameters must be passed to the IDTC with the initialize FloppyTape parameters command before accessing the tape for the first time after a reset.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$0F	command packet size: 29 bytes
3	$04	physical device number
4	$40	command type: CONTROL
5	$08	function code
6	$00	filler
7	$80	device type: Tape
8	$xx	number of sectors per segment ($01-$2C)
9	$00	filler
10-11	$xxxx	physical sector size ($0100, $0200, $0400)
12-19	$00..00	reserved
20	$xx	segment skip factor (1-4)
21-27	$00..00	reserved
28	$03	ETX mark

IDTC STATUS COMMANDS

After execution of each command, the IDTC puts a status message into the command channel and asserts a VMEbus interrupt request. The status messages are variable length packets containing information about the completed command and, in the case of malfunctions, about detected errors. The general format of the status messages is as follows:

BYTE	CONTENTS	DESCRIPTION
0	$02	Byte 0 of the status packet is marked with the start-of-text character $02.
1	$xx	Byte 1 returns the message ID from the originating command as specified in byte 1 of the command packet. This may be used by the operating system for identifying the status message.
2	$xx	Byte 2 specifies the status packet size in the total number of bytes.
3	$yz	Byte 3 specifies the controller (y) and drive (z) number from the originating command as specified in byte 3 of the command packet.
4	$xx	Byte 4 returns the type of the originating command as specified in byte 4 of the command packet.
5	$xx	Byte 5 specifies the status type. The IDTC supports the status types COMPLETION ($70), COMMAND ABORT ($71), SOLICITED STATUS ($72).
6-7	$xxxx	Bytes 6-7 contain a status code which is cleared ($0000) if the command has been completed successfully. In the case of a malfunction, the status code contains a reference number indicating the type of error.
8-(n-1)	$xx..xx	The following bytes are optional and are used for passing additional parameters or status details to the system. They are explained in the detailed status message descriptions in the following paragraphs.
n	$03	The last byte of the status packet is marked with the end-of-text character ($03).

In the following paragraphs all IDTC status messages are described in detail in the order of ascending status types and codes.

To assist in tracing hard disk malfunctions, each message description contains a list of SASI/SCSI controller error messages that result in the described IDTC status message. For details of the SASI/SCSI protocol, refer to Chapter 4.

Read/Write Complete

The read/write complete message indicates the successful completion of a data transfer operation between disk or tape and system memory.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$13	status packet size: 19 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE)
5	$70	status type: COMPLETION
6-7	$0000	status code
8-9	$xxxx	number of blocks read or written
10-11	$xxxx	block size in bytes
12	$xx	memory address modifier
13-15	$xxxxxx	starting memory address
16-17	$0000	filler
18	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
0	$0	NO ERROR: The controller completed the previous command without detecting errors.

3

Head Position Complete

The head position complete message returns the current position of the read/write head. The head position is specified by the LSA of the first sector on the current track. This status message is only available for floppy disk operations.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$0D	status packet size: 13 bytes
3	$yz	controller and drive number (y=controller number, z=drive number)
4	$30	type of originating command: TEST
5	$70	status type: COMPLETION
6-7	$0000	status code
8-11	$xxxxxxxx	LSA of first sector on the current track
12	$03	ETX mark

Other Commands Complete

The other commands complete message indicates the successful completion of other commands, such as format, restore, seek, verify, and initialize.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($30 = TEST, $40 = CONTROL, $50 = HALT)
5	$70	status type: COMPLETION
6-7	$0000	status code
8	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
0	$0	NO ERROR: The controller completed the previous command without detecting errors.

IDTC Self Test Complete

The IDTC self test complete message returns the results of the request IDTC self test command execution (refer to *Request IDTC Self Test* command description). For each part of the test (floppy disk/tape controller, hard disk controller) the message contains a bit in the status byte which is set if the corresponding test failed.

3

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$10	status packet size: 16 bytes
3	$00	filler
4	$30	type of originating command: TEST
5	$70	status type: COMPLETION
6-7	$0000	status code
8	$02	controller type: DISK/TAPE CONTROLLER
9-11	$000000	filler
12	$xx	status byte
13-14	$0000	filler
15	$03	ETX mark

Status Byte:

BIT	DESCRIPTION
7	%0 (not used)
6	%0 = floppy disk/tape controller test passed %1 = floppy disk/tape controller test failed
5-1	%00000 (not used)
0	%0 = hard disk controller test passed %1 = hard disk controller test failed

Data CRC/ECC Error

The data CRC/ECC error message indicates that a data CRC (floppy disk/tape) or ECC (hard disk) error has been detected during a read, write with verify, or format operation. This message is also returned when the controller did not find the address mark in a sector data field. The defective sector is specified by the block CBN (bytes 8-11).

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$0D	status packet size: 13 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0001	status code
8-11	$xxxxxxxx	CBN of defective block
12	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
1	$1	UNCORRECTABLE DATA ERROR: the controller detected an uncorrectable ECC error in the data field of the target sector during a read operation.

Disk Write Protected

The disk write protected message indicates an attempt to write on or format a write protected floppy disk or tape cartridge. This status message is only available for floppy disk and tape operations.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$xx	physical device number (4-7)
4	$xx	type of originating command ($20 = WRITE, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0002	status code
8	$03	ETX mark

Drive Not Ready

The drive not ready message indicates that no READY signal was received after selecting a disk or tape drive. Note that for 5-1/4 inch floppy disk drives without the READY signal, the INDEX signal may be used for detecting readiness. The tape drive is ready when a cartridge is inserted and the retention pass is completed.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0003	status code
8	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
0	$4	DRIVE NOT READY: after selecting the disk drive, the controller did not receive a ready signal.
0	$8	DRIVE STILL SEEKING: the disk drive did not complete a seek operation within a prescribed time.

Deleted Data Mark Read

The deleted data mark read message indicates that a sector with either a deleted data mark or a bad track flag has been encountered during a read or verify operation. This message is also returned when an ECC error has been detected during a verify pass. The defective sector is specified by the block CBN (byte 8-11).

3

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$0D	status packet size: 13 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $30 =TEST)
5	$70	status type: COMPLETION
6-7	$0004	status code
8-11	$xxxxxxxx	CBN of block containing the deleted data mark
12	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
1	$9	BAD TRACK: the controller detected a track with the bad track flag set (XEBEC S1410/S1410A).
1	$9	ECC ERROR DURING VERIFY: the controller detected an uncorrectable ECC error during a verify pass (ADAPTEC ACB 4000).

Invalid Drive Number

The invalid drive number message indicates that the controller or drive number specified in the command is invalid. For floppy disk and tape drives, the controller number must be 0, and valid drive numbers are 4 through 7. For hard disks, valid controller numbers are 0 through 7, valid drive numbers are 0 and 1. This message is also returned when an attempt is made to configure drive number 5 for a floppy disk while drive number 4 is still configured as FloppyTape.

3

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	invalid controller and drive number (y=controller number, z=drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0005	status code
8	$03	ETX mark

3

Invalid Disk Address

The invalid disk address message indicates that a CBN specified in the command or calculated by the controller exceeds the total number of blocks on the disk or tape.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$0D	status packet size: 13 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0006	status code
8-11	$xxxxxxxx	invalid CBN
12	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
2	$1	ILLEGAL SECTOR ADDRESS: the sector address specified in the controller command exceeds the total number of sectors on the disk.
2	$3	VOLUME OVERFLOW: the address of the last sector to be read or written exceeds the total number of sectors on the disk.

Restore Error

The restore error message indicates an unsuccessful attempt to detect the TRACK ZERO signal. This usually implies a severe disk or tape drive malfunction.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0007	status code
8	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
0	$6	TRACK ZERO NOT FOUND: after stepping the maximum number of cylinders, the controller did not receive the TRACK ZERO signal from the disk drive.

3

Record Not Found

The record not found message indicates that the sector specified by the CBN in the command cannot be found on the disk or tape.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$0D	status packet size: 13 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0008	status code
8-11	$xxxxxxxx	CBN of block not found
12	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
1	$2	ID ADDRESS MARK NOT FOUND: the controller did not detect the address mark in the ID field of the target sector.
1	$3	DATA ADDRESS MARK NOT FOUND: the controller did not detect the address mark in the data field of the target sector.
1	$4	SECTOR NOT FOUND: the controller found the correct cylinder and head, but not the specified sector.

Sector ID CRC/ECC Error

The sector ID CRC/ECC error message indicates that a CRC (floppy disk/tape) or ECC (hard disk) error has been detected in the sector ID field. The defective sector is specified by the block CBN (bytes 8-11).

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$0D	status packet size: 13 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0009	status code
8-11	$xxxxxxxx	CBN of defective block
12	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
1	$0	UNCORRECTABLE ID ERROR: the controller detected an uncorrectable ECC error in the ID field of the target sector.

VMEbus DMA Error

The VMEbus DMA error message indicates that the DMA controller could not access the specified system memory locations. This may happen when the VMEbus request being asserted by the IDTC has not been granted within 1 second (bus request time-out), or if a started system DMA operation has not been completed within 2 seconds (data transfer time-out). For details of time-out functions, refer to Chapter 4.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE)
5	$70	status type: COMPLETION
6-7	$000A	status code
8	$03	ETX mark

Controller Error

The controller error message indicates that an internal hardware or software malfunction has been detected on the IDTC or the hard disk controller or that no hard disk controller is connected. As this might lead to fatal errors in the mass storage devices, the IDTC should be reset and reinitialized before executing further commands.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$000F	status code
8	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
1	$D	SELF TEST FAILED: the controller detected a malfunction during execution of its self test program.
2	$0	INVALID COMMAND: the controller has received an invalid device control block from the IDTC.
3	$0	RAM ERROR: the controller detected a data error during the sector buffer RAM test.
3	$1	PROGRAM ERROR: the controller detected a checksum error during the program memory test.
3	$2	ECC POLYNOMINAL ERROR: the controller detected an error during the ECC generator hardware test.

3

Drive Error

The drive error message indicates that either the index signal cannot be detected or a condition at the disk or tape drive exists which might cause improper writing. This message usually implies a severe disk or tape malfunction.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	controller and drive number (y=controller number, z=drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0010	status code
8	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
0	$1	NO INDEX SIGNAL: the controller did not detect an index signal from the disk drive.
0	$2	NO SEEK COMPLETE: the controller did not receive a seek complete signal from the disk drive after a seek operation.
0	$3	WRITE FAULT: the controller received a write fault signal from the disk drive. This indicates a condition which might cause improper writing on the disk.

Seek Error

The seek error message indicates that the disk side or track or the tape segment being specified in the command cannot be found.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$70	status type: COMPLETION
6-7	$0011	status code
8	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
0	$2	NO SEEK COMPLETE: the controller did not receive a seek complete signal from the disk drive after a seek operation.
1	$5	SEEK ERROR: the current cylinder or side number is not identical to the cylinder or side information in the sector ID.
1	$A	FORMAT ERROR: the disk is either unformatted or the format is incorrect.
1	$C	BAD DRIVE FORMAT: the disk is either unformatted or the format is incorrect.

3

I/O DMA Error

The I/O DMA error message indicates that a data transfer on the peripheral bus between local RAM and SASI/SCSI bus or floppy disk/tape controller was not terminated within 5 seconds (local bus time-out). This usually indicates a severe hardware problem.

3

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE)
5	$70	status type: COMPLETION
6-7	$0019	status code
8	$03	ETX mark

Invalid Command

The invalid command message indicates that invalid command types, function codes, or parameters have been used.

In the simple command channel protocol, this message is also used for indicating an incorrect command format or syntax. In the extended command channel protocol, such errors are flagged in the CMD ACK/NAK byte.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$09	status packet size: 9 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($10 = READ, $20 = WRITE, $30 = TEST, $40 = CONTROL)
5	$71	status type: COMMAND ABORT
6-7	$0001	status code
8	$03	ETX mark

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
2	$0	INVALID COMMAND: the controller has received an invalid device control block from the IDTC.
2	$2	INVALID PARAMETER: a parameter specified in the device control block is not valid.
2	$4	BAD ARGUMENT: a parameter specified in the device control block is not valid.
2	$5	INVALID LOGICAL UNIT NUMBER: the logical unit number specified in the device control block is not valid.

3

Drive Status

The drive status message is the result of a request drive status command. It returns device type, device status, and disk or tape parameters.

IDTC Command Message:

BYTE	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID from originating command
2	$0F	status packet size: 15 bytes
3	$yz	controller and drive number (y=controller number, z= drive number)
4	$xx	type of originating command ($30 = TEST)
5	$72	status type: SOLICITED STATUS
6-7	$0000	status code
8	$xx	status byte (see below)
9	$00	filler
10-13	$xxxxxxxx	total number of blocks on disk (floppy disk only)
14	$03	ETX mark

Status Byte:

BIT	DESCRIPTION
7	%0 = drive ready %1 = drive not ready
6	%0 (not used)
5	%0 = disk/tape not write protected %1 = disk/tape write protected
4	%0 (not used)
3	%0 (not used)
2-0	device type: %001 = floppy disk %010 = hard disk %101 = floppy tape

SASI/SCSI Controller Error:

CLASS	CODE	DESCRIPTION
0	$0	NO ERROR: the controller completed the previous TEST DRIVE READY command without detecting errors.

CHAPTER 4 - FUNCTIONAL DESCRIPTION

INTRODUCTION

This chapter provides the functional description of the MVME319 (IDTC) at block level. The functional description provides an overview of the module, followed by a detailed description of each section of the module. The block diagram of the MVME319 is shown in Figure 4-1.

The MVME319 provides a universal interface between hard and floppy disk drives, a FloppyTape drive and the VMEbus. It contains a complete microcomputer which relieves the system processor(s) from disk/tape controlling tasks. The IDTC is capable of controlling up to eight hard disk controllers via the SASI/SCSI host adapter and either four floppy disk drives or one FloppyTape drive plus two floppy disk drives directly.

The IDTC receives macro commands through a dual ported RAM which is shared between the operating system and the local processor. The onboard firmware resident program interprets and executes the macro commands and returns status messages into the command channel after completion. The IDTC supports two alternative protocols for the command and status transmission between the operating system and the IDTC firmware.

To minimize VMEbus usage, data to be written or read on the disks or tape is intermediately stored in local RAM and transferred from or to VME system memory in blocks of up to 17Kb. All data transfers between disk/tape drives, local RAM, and system RAM are performed by a high speed DMA controller, concurrent with the local processor program execution.

The IDTC firmware contains self test routines which are executed after system reset or by IDTC command. Controller malfunctions are indicated by a FAIL LED on the front panel.

HARDWARE OVERVIEW

For the following description, the IDTC module is regarded as consisting of functional blocks as shown in the block diagram. For hardware details, Chapter 5 includes the schematic diagrams and an assembly drawing.

The microprocessing unit on the IDTC is the MC68121 Intelligent Peripheral Controller (IPC). This very large scale integrated circuit contains an 8-bit microprocessor, a 16-bit timer, a serial port, and 128 bytes of dual ported RAM on a single chip. The timer generates the time-slices for the multitasking IDTC firmware. The serial port provides a 9600 baud asynchronous RS-232C communication interface for Motorola field service diagnostics (it is not intended for user applications). The dual ported RAM serves as the command channel between the IPC and the VMEbus.

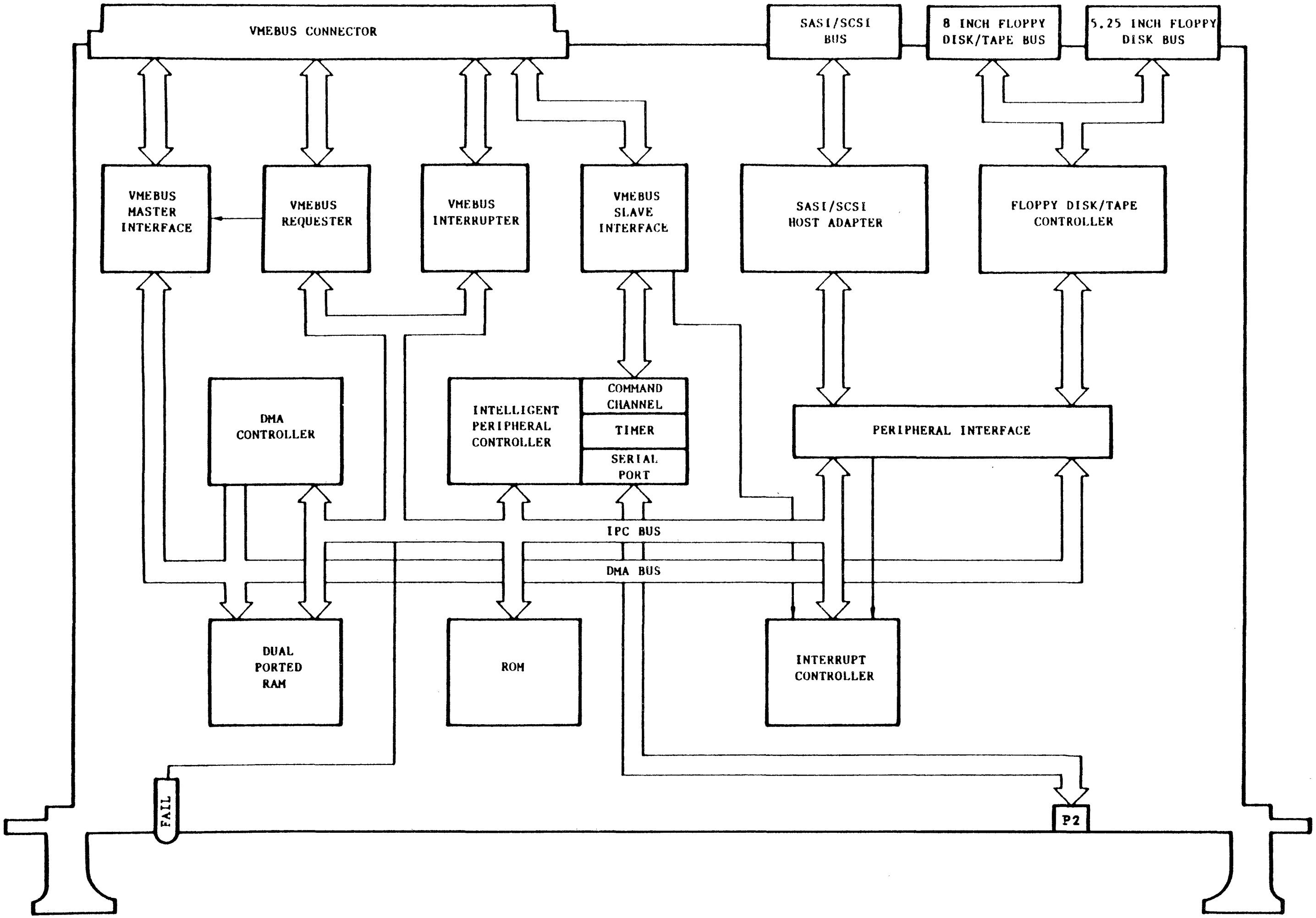


FIGURE 4-1. MVME319 Block Diagram

The IDTC firmware is resident in one 166Kb EPROM. The program is described later in this chapter.

Two 32Kb static RAMs (only the lower 16Kb of each RAM is used) are available for IDTC program data and for intermediate storage of up to 17Kb of tape, floppy, and hard disk I/O data. The RAM is dual ported and is accessed from the IDTC processor and from the DMA controller concurrently. This allows the IDTC firmware to be executed unhindered by peripheral data I/O.

The AM9517A Direct Memory Access (DMA) controller performs all data transfers between the VME system memory and local RAM, as well as between local RAM and, through the peripheral interface, the SASI/SCSI host adapter and the floppy disk/tape controller.

The VMEbus interface is compatible with the VMEbus specification. It provides an A16:D8 slave interface to the command channel and an A24:D16 master interface for system DMA. The command channel is accessed with the address modifier code $2D (short supervisory I/O access). The address modifier code for the system DMA is specified in the IDTC command by the calling program and may be any in the whole range from $00 to $3F.

Prior to accessing the VME system memory, the IDTC performs a bus arbitration sequence. This is done by the option Release-When-Done (RWD) VMEbus requester which asserts a bus request upon the IDTC processor demand on a jumper selectable priority level. When bus master, the IDTC executes pending system DMA tasks and then releases the bus after a maximum time of 64 microseconds.

Whenever the IDTC has placed a message into the command channel, it asserts a VMEbus interrupt request on a jumper selectable level. The status/ID byte supplied during the interrupt acknowledge cycle is specified during IDTC initialization by a command from the operating system.

A detailed description of all VMEbus operations, including the bus requester and interrupter functions, is given in the *VME System Interface* paragraph in this chapter.

The AM9519A Universal Interrupt Controller provides for fast interrupt response times by supplying a unique interrupt vector to the IPC processor for each local interrupt source.

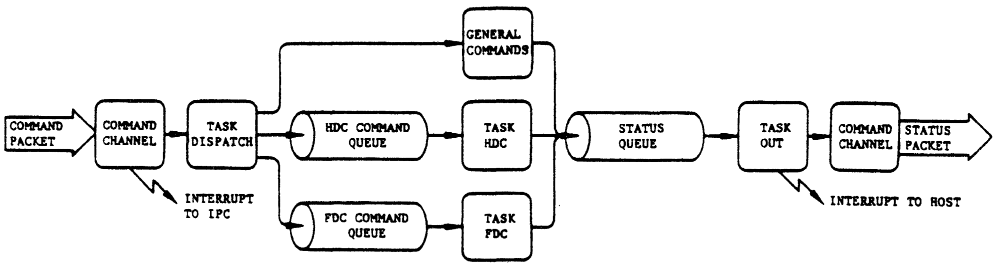
Both the SASI/SCSI host adapter and the floppy disk/tape controller are accessed through the local peripheral interface bus either from the IDTC processor (for control operations) or from the DMA controller (for peripheral data transfers). The SASI/SCSI bus interface is described in the *SASI/SCSI Bus Host Adapter* paragraph and the floppy disk/tape controller in the *Floppy Disk/Tape Controller* paragraph in this chapter.

FIRMWARE OVERVIEW

The program resident in the IDTC firmware performs the following functions:

- Receives commands from the VMEbus through the command channel.
- Analyzes and validates the commands.
- Transforms the commands into micro operation sequences.
- Supervises hard disk controller, floppy disk/tape controller, DMA controller, VMEbus requester, and VMEbus interrupter operations.
- Places status messages into the command channel.

The firmware consists of a general purpose multitasking executive and several specific tasks for disk/tape controller functions. The figure below shows the block structure of the IDTC firmware.



Each time a command is written from the VMEbus into the command channel, the IDTC processor is interrupted. This interrupt starts a service routine in the executive which checks the syntax of the received command. If the syntax is invalid, an error message is placed in the command channel, a VMEbus interrupt is asserted, and no further actions are taken. If the syntax is valid, the command is forwarded to the task DISPATCH.

The task DISPATCH classifies the commands in general commands, hard disk commands, and floppy disk/tape commands. Hard disk commands are transferred to the task HDC, floppy disk/tape commands are transferred to the task FDC. The tasks HDC and FDC contain input queues that allow commands to be stacked up and executed sequentially.

The tasks HDC and FDC first check all parameters in the received commands to avoid peripheral malfunctions. Then the commands are transformed into the appropriate micro operation sequences and executed.

VMEbus requester and DMA controller operations are supervised by time-out functions. If a VMEbus request is not granted within 1 second or if a DMA operation is not terminated within 2 seconds, the command execution is aborted and an error message is returned to the command channel. Hard disk and floppy disk/tape controller operations are supervised by checking their status after the command execution.

After the completion of each command a status message is assembled which contains information about the results and eventual errors. These messages are transferred to the input queue of the task OUT.

When messages are put in the queue, the task OUT checks if any previous message has been fetched by the operating system, and if so, places the next status message into the command channel. Then a VMEbus interrupt is asserted to indicate that a new status message is available.

VME SYSTEM INTERFACE

The following paragraphs supply detailed specifications of all VME system operations supported by the IDTC and includes descriptions of VMEbus signals, system memory access, VMEbus requester and interrupter, reset, and self test.

VMEbus Signals

The VMEbus interface provides the signal path between the IDTC and the VMEbus backplane. The interface complies with all requirements for the signal driver/receiver characteristics and bus operation protocols, as specified in the VMEbus specification. The VMEbus signal descriptions are given in Chapter 5.

System Memory Access

Peripheral data transfers between the disk/tape drives and the VME system memory are performed by the DMA controller and its support logic. The DMA controller accesses the VMEbus through an A24:D16 master interface and supports the complete 16Mb address map and all address modifier codes.

When the IDTC receives a disk/tape read or write command, it loads the DMA controller with the starting system memory address and the address modifier code specified in the command and with a cycle count evaluated from the number of blocks to be read or written. Then it asserts a VMEbus request. Refer to *Bus Arbitration* paragraph in this chapter.

As soon as the bus requester indicates bus availability, the VMEbus master interface is enabled and the DMA controller starts transferring the data in 16-bit word read or write cycles. Up to 17Kb of peripheral data can be immediately stored in the IDTC local RAM and are transferred to or from system memory in multiple bursts with a maximum duration of 64 microseconds per DMA operation (typical duration is 53 microseconds).

All data transfers on the VMEbus are supervised by data transfer time-out functions. If the accessed memory does not respond with a DTACK* signal, the IDTC aborts the current DMA cycle and releases the VMEbus after a maximum of 64 microseconds. If the transfer of the total data block to be moved between local RAM and system memory cannot be terminated within 2 seconds, the IDTC aborts the DMA operation and returns an error message in the command channel. The occurrence of a data transfer time-out indicates that the specified addresses or address modifiers do not exist in the system, or that the addressed memory location has responded with a bus error signal.

VMEbus Requester

The VMEbus requester is responsible for the following tasks:

. Assert a bus request when the IDTC indicates a system DMA request.
. Acquire bus mastership when the bus request is granted.
. Release the bus when the system DMA operation is terminated.
. Propagate not requested bus grants to the next bus requester.

The VMEbus requester can be configured to operate on any one of the four bus arbitration levels. This is done by setting the appropriate jumpers on headers K1, K2, and K3 as described in Chapter 2.

The IDTC firmware initiates a bus request by setting a local system DMA request signal. This causes the VMEbus requester to assert BR*.

The bus request is supervised by a bus request time-out counter. If the bus request in not acknowledged within 1 second, the execution of the command which has initiated the system DMA request is aborted and an error message is returned in the command channel.

However, the VMEbus requester keeps BR* asserted until it receives a bus grant on the same priority level, regardless of the local system DMA request status. This is necessary to obey the bus arbitration protocol as specified for the VMEbus.

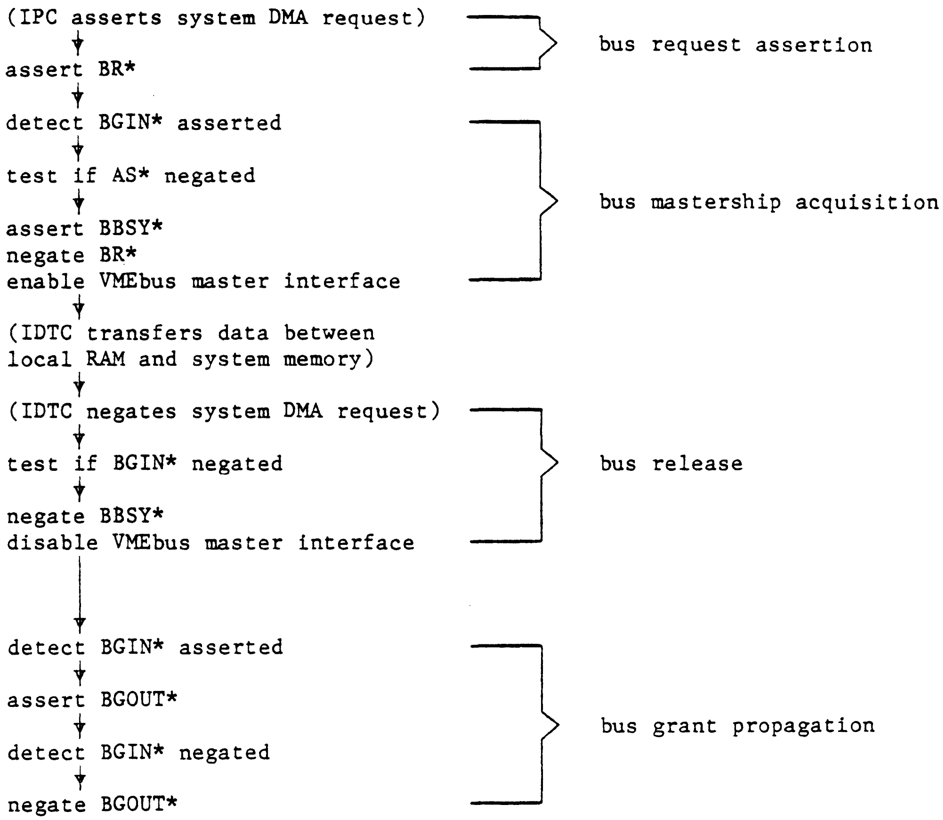
When the VMEbus requester has a bus request pending and it receives a bus grant on the same priority level, it acquires bus mastership. After the previous bus master has finished its last VMEbus cycle and negated AS*, the VMEbus requester acknowledges the bus grant by asserting BBSY* and negating BR*, and it enables the VMEbus master interface.

Now the DMA controller starts transferring data between the local RAM and system memory. The maximum duration of each DMA operation is 64 microseconds. Because the VMEbus requester on the MVME319 is of the RWD type, the system DMA operation is neither aborted by bus requests of other modules nor by activation of BCLR*.

When the system DMA operation is terminated, the IDTC negates the local system DMA request signal. Upon detecting this, the VMEbus requester tests if BGIN* is negated and then asserts BBSY* and disables the VMEbus master interface.

Any bus grant which is received at a time when the IDTC has no system DMA request pending is propagated to the next bus requester. The VMEbus requester keeps BGOUT* asserted as long as BGIN* is low.

The flow diagram below illustrates the operation sequence of the VMEbus requester.



4

VMEbus Interrupter

The VMEbus interrupter is responsible for the following tasks:

- Assert a bus interrupt when the IDTC requests a system interrupt.
- Supply a status/ID byte during the interrupt acknowledge cycle.
- Propagate not requested acknowledgements to the next interrupter.

The VMEbus interrupter can be configured to operate on any one of the seven interrupt levels. This is done by setting the appropriate jumpers on headers K6 and K8 as described in Chapter 2.

4

The interrupt status/ID byte which is supplied by the VMEbus interrupter in the interrupt acknowledge cycle must be specified in the *Configure IDTC* command by the operating system during system initialization (refer to Chapter 3).

The IDTC firmware initiates a bus interrupt by setting a local system interrupt request signal. This causes the interrupter to assert IRQ*.

When the VMEbus interrupter has an interrupt request pending and it receives an interrupt acknowledge on the same priority level, it negates IRQ* and supplies the status/ID byte to the interrupt handler.

Any interrupt acknowledge which the VMEbus interrupter receives on a level different from its own is propagated to the next interrupter. The VMEbus interrupter keeps IACKOUT* asserted as long as IACKIN* is low.

The flow diagram below illustrates the operation sequence of the VMEbus interrupter.

```
(IPC requests system interrupt)        ─┐
        ↓                                > interrupt request assertion
assert IRQ*                            ─┘
        ↓
detect IACK* asserted                  ─┐
detect AS* asserted                     │
compare interrupt level                 │
        ↓                               │
if interrupt level is equal:            │
detect IACKIN* asserted                 │
negate IRQ*                             > status/ID byte supply
detect DSO* asserted                    │
place status/ID byte on data bus        │
assert DTACK*                           │
        ↓                               │
detect DSO* negated                     │
release data bus                        │
negate DTACK*                          ─┘
        │
        ↓
detect IACK* asserted                  ─┐
detect AS* asserted                     │
compare interrupt level                 │
        ↓                               │
if interrupt level is different:        │
detect IACKIN* asserted                 > interrupt ack. propagation
assert IACKOUT*                         │
        ↓                               │
detect IACKIN* negated                  │
        ↓                               │
negate IACKOUT*                        ─┘
```

Reset, Self Test, and Failure

The MVME319 is reset when the VMEbus SYSRESET* signal is asserted. This is usually done automatically after power up or manually by a RESET switch which must be provided on a system controller module on the VMEbus.

After a system reset the IDTC performs an extensive local self test of all large scale integrated circuits (memory, command channel, timer, DMA controller, interrupt controller). If the self test program detects any hardware malfunctions, the IDTC lights the FAIL LED on the front panel. The VMEbus SYSFAIL* signal is connected with the local failure signal via header K7 as described in Chapter 2.

A system failure is also indicated in the case of severe IDTC firmware malfunctions. All IDTC functions are supervised by a program time-out counter which is reset periodically by the IDTC firmware executive. The system failure signal is asserted if a program error occurs and the counter is not reset within 2.5 seconds.

When the IDTC has detected a system failure and turned the FAIL LED on, it immediately stops executing and does not accept further commands. The system must be reset for restarting the MVME319 module.

During the self test execution the IDTC is not capable of processing interrupts and any information in the command channel is destroyed. Therefore, the operating system must not transmit any commands to the IDTC within 10 seconds after a system reset.

The successful completion of the IDTC self test and initialization can be checked by reading location $1FF in the command channel. Data $01 indicates that the IDTC is ready for the *Configure IDTC* command. All other command channel locations are cleared.

SASI/SCSI BUS HOST ADAPTER

The IDTC supports the SASI (Shugart Associates System Interface) bus and SCSI (Small Computer System Interface) bus. Both provide a physical and logical standard for controlling peripheral devices, especially mass storage units. The host adapter implemented on the IDTC supports the command and status message protocols for driving up to eight non-arbitrating hard disk controllers (XEBEX S1410/S1410A, ADAPTEC ACB 4000), each of them connected with one or two hard disk devices.

SASI/SCSI Bus Host Adapter Hardware

The SASI/SCSI bus host adapter provides the interface between the peripheral interface bus on the IDTC and the SASI/SCSI bus. A block diagram of the SASI/SCSI bus host adapter is shown in Figure 4-2.

The host adapter contains separate input and output registers for the SASI/SCSI bus data and control signals, and an address decoder and interface control logic for selecting the registers. The input/output registers are accessed either by the IDTC processor (for selection, command and status transfer, and completion sequences) or by the DMA controller (for data transfer sequences). All SASI/SCSI bus signals (except ACK*) are monitored and driven under control of the IDTC firmware which creates the signal sequences according to the SASI/SCSI protocols. To improve the data transfer rate on the SASI/SCSI bus, the REQ*/ACK* handshake is hardware generated.

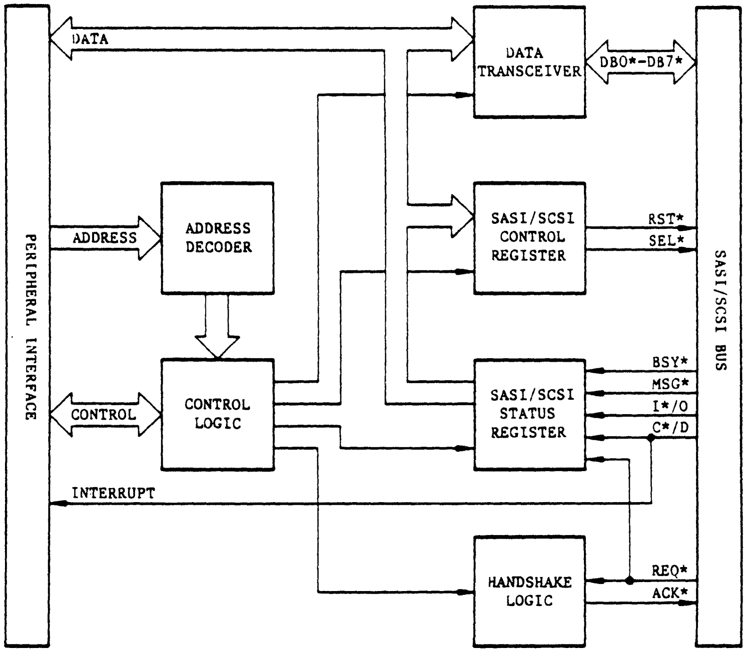


FIGURE 4-2. SASI/SCSI Host Adapter Block Diagram

SASI/SCSI Bus Signals

All SASI/SCSI bus signals are available at connector P3. Refer to Chapter 5 for signal descriptions.

SASI/SCSI Bus Sequences

The interactions between the IDTC and the hard disk controllers obey the protocol for the SASI/SCSI bus sequences which are described in the following paragraphs.

Controller Selection and Command Transfer

Each disk operation is started with a controller selection. The IDTC first checks if the SASI/SCSI bus is available (i.e., if the signals BSY*, REQ*, I*/O, and MSG* are high).

Then the IDTC presents the controller ID byte on the data bus and asserts SEL*. In systems containing multiple controllers, each controller is selected by setting one of the eight bits of the controller ID, while all other bits are cleared. The following list shows the assignment of controller IDs to controller numbers:

CONTROLLER NUMBER	CONTROLLER ID
0	$01
1	$02
2	$04
3	$08
4	$10
5	$20
6	$40
7	$80

When receiving its ID byte and the SEL* signal, the selected hard disk controller asserts BSY*. Upon detecting BSY* being low, the IDTC terminates the selection sequence by negating SEL* and releasing the data bus.

After being selected, the hard disk controller requests the command transfer from the IDTC by driving C*/D low and asserting REQ*. The IDTC acknowledges this request by presenting the first byte of the command descriptor block on the data bus and asserting ACK*. When receiving the ACK* signal, the controller reads the command byte and negates REQ*. Upon detecting REQ* being high, the IDTC terminates the transfer of the first command byte by negating ACK* and releasing the data bus.

Then the IDTC waits for the next activation of REQ* to present the second command byte. The above described handshake sequence continues until the complete command descriptor block has been transferred from the IDTC to the controller byte by byte.

A detailed description of the command descriptor blocks is given in the *Command Descriptor Blocks* paragraph in this chapter.

The timing diagram of the controller selection and command sequence is shown in Figure 4-3.

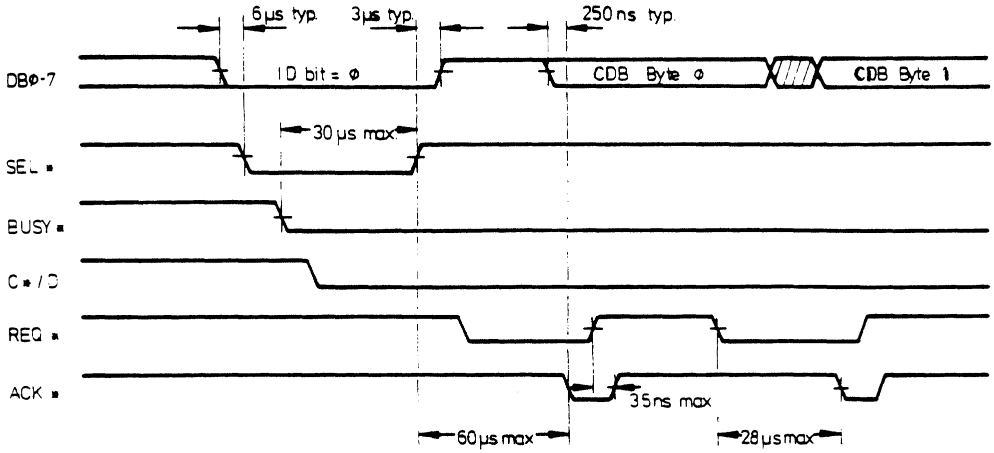


FIGURE 4-3. **Controller Selection** and Command Transfer Timing

Data Transfer

The hard disk controller initiates data transfers over the SASI/SCSI bus by driving the C*/D signal high and specifying the direction of the transfer in the I*/O signal.

If data is to be read by the IDTC, the controller drives I*/O low, presents the first data byte on the data bus, and asserts REQ*. The IDTC acknowledges this request by reading the data byte and asserting ACK*. When receiving the ACK* signal, the controller negates REQ* and releases the data bus. Upon detecting REQ* being high, the IDTC terminates the transfer of the first data byte by negating ACK*. Then the IDTC waits for the next activation of REQ* to read the second data byte. The described handshake sequence continues until the complete data block has been transferred from the controller to the IDTC byte by byte.

The timing diagram of the read data transfer sequence is shown in Figure 4-4.

If data is to be written by the IDTC, the controller drives I*/O high and asserts REQ*. The IDTC acknowledges this request by presenting the first data byte and asserting ACK*. When receiving the ACK* signal, the controller reads the data byte and negates REQ*. Upon detecting REQ* being high, the IDTC terminates the transfer of the first data byte by releasing the data bus and negating ACK*. Then the IDTC waits for the next activation of REQ* to write the second data byte. The described handshake sequence continues until the complete data block has been transferred from the IDTC to the controller byte by byte.

The timing diagram of the read data transfer sequence is shown in Figure 4-5.

4

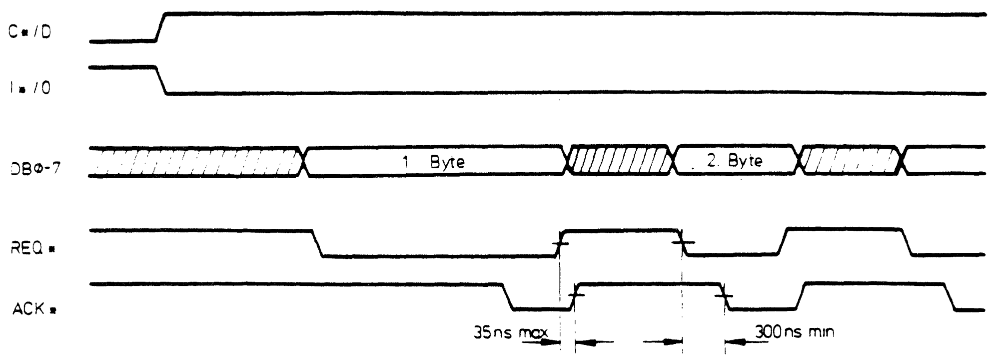


FIGURE 4-4. Read Data Transfer Timing

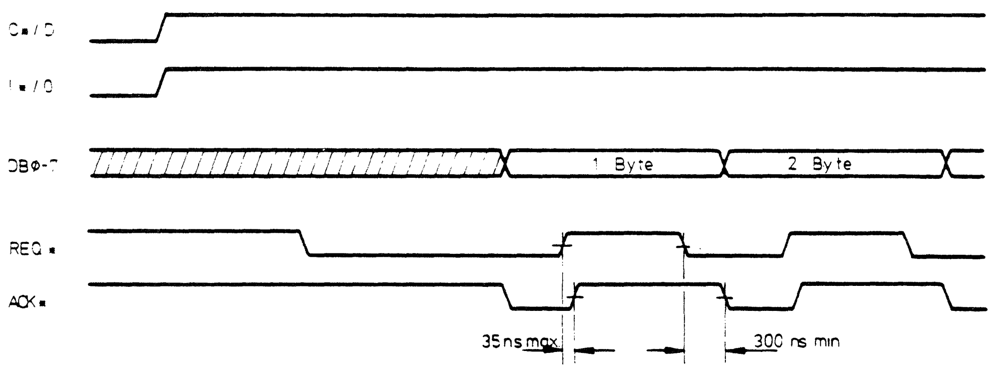


FIGURE 4-5. Write Data Transfer Timing

Status and Completion Message Transfer

Following the execution of a command, the hard disk controller initiates a status and completion message transfer by driving the signals C*/D and I*/O low. Then the controller presents the status byte on the data bus and asserts REQ*. The IDTC acknowledges this request by reading the status byte and asserting ACK*. When receiving the ACK* signal, the controller negates REQ* and releases the data bus. Upon detecting REQ* being high, the IDTC terminates the transfer of the status byte by negating ACK*.

Then the controller indicates the termination of the command execution by asserting the MSG* signal, presenting the completion byte ($00) on the data bus, and asserting REQ*. The IDTC acknowledges this request by reading the completion byte and asserting ACK*. When receiving the ACK* signal, the controller negates REQ* and releases the data bus. Upon detecting REQ* being high, the IDTC terminates the transfer of the status byte by negating ACK*. At last the controller releases the SASI/SCSI bus and enters an idle loop waiting for the next selection.

A detailed description of the status bytes is given in the *Status and Completion Bytes* paragraph in this chapter.

The timing diagram of the status and completion message transfer sequence is shown in Figure 4-6.

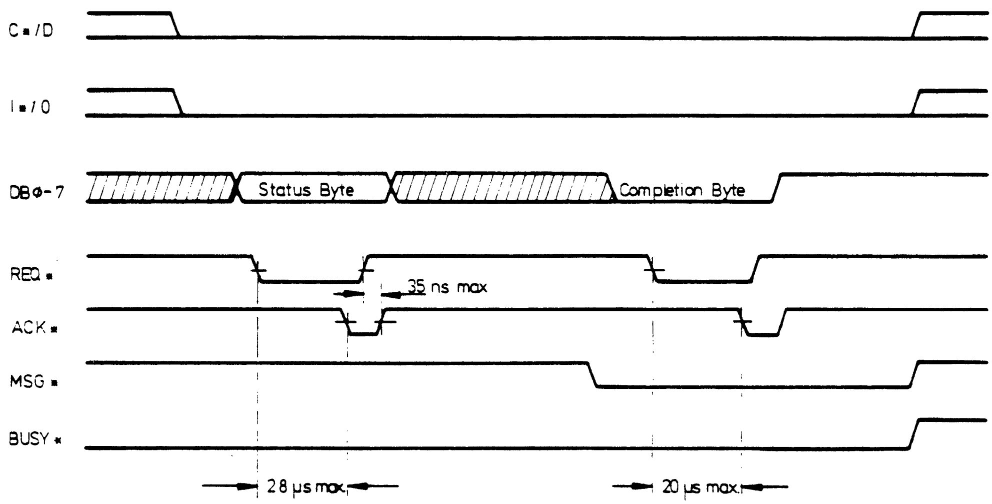


FIGURE 4-6. Status and Completion Message Timing

Command Descriptor Blocks

After being selected by the IDTC, the hard disk controller requests the transfer of a Command Descriptor Block (CDB) which specifies the operation to be performed. SASI/SCSI controller CDBs are categorized in 8 classes. The IDTC supports class 0 (data transfer, non-data transfer, status) commands and class 7 (diagnostics) commands. Both command classes have a length of 6 bytes and the format described below.

SASI/SCSI Command Descriptor Block

BIT	7	6	5	4	3	2	1	0
Byte 0	command class			command opcode				
Byte 1	log. unit no.			logical sector addr (high)				
Byte 2	logical sector addr (mid)							
Byte 3	logical sector addr (low)							
Byte 4	number of sectors/interleave factor							
Byte 5	control byte							

Bits 5-7 of byte 0 define the command class. Valid classes are 0 and 7.

Bite 0-4 of byte 0 contain the command opcode.

Bits 5-7 of byte 1 specify the Logical Unit Number (LUN) of the disk drive to be accessed. The IDTC supports one or two disk drives being connected with each controller (i.e., valid LUNs are 0 and 1).

For all commands involving disk operations (read, write, format, seek) bytes 1, 2, and 3 specify the Logical Sector Address (LSA) of the first sector to be accessed. Bits 0-4 of byte 1 is the high order byte, byte 2 is the mid order byte, and byte 3 is the low order byte of the LSA. The hard disk controller organizes the data in sectors of 256, 512, or 1024 bytes and assigns a continuous logical sector address to each sector. The operating system organizes the data on the disk in blocks of 256, 512, or 1024 bytes and assigns Continuous Block Number (CBN) to each block. The block size must be at least the sector size. In the IDTC commands, the disk access is specified by block size and CBN of the first block to be read or written. The IDTC translates the CBN given in the IDTC command into the LSA contained in the SASI/SCSI command according to the formula:

$$\text{logical sector address} = \text{continuous block number} * \frac{\text{block size}}{\text{sector size}}$$

For all commands involving data transfers (read, write), byte 4 specifies the number of sectors to be transferred. The IDTC translates the number of blocks given in the IDTC command into the number of sectors contained in the SASI/SCSI command according to the formula:

$$\text{number of sectors} = \text{number of blocks} * \frac{\text{block size}}{\text{sector size}}$$

For all disk format commands, byte 4 specifies the interleave factor. Interleaving is a method of mapping logically contiguous sectors on a track onto non-adjacent physical sectors. The interleaving factor is equal to the number of physical sectors between two contiguous logical sectors plus one. It should be adjusted for maximum system performance and may be in the range from one up to the number of sectors per track minus one.

4

The best interleave factor for the XEBEC S1410 is 5, for the XEBEC S1410A is 4, and for the ADAPTEC ACB 4000 is 2.

Byte 5 is a control field which selects imbedded servo and stepping rate options for the XEBEC S1410/S1410A disk controller:

```
bit 7-5 : %000 = not used
bit 4   : imbedded servo : %0 = drive without imbedded servo
                           %1 = drive with imbedded servo (S1410A only)
bit 3   : %0 = not used
bit 2-0 : stepping rate : %000 =   3 ms not buffered
                          %100 = 200 us buffered
                          %101 =  70 us buffered
                          %110 =  30 us buffered
                          %111 =  15 us buffered
```

For the ADAPTEC ACB 4000 disk controller byte 5 is not used and is always $00.

The following table lists all SASI/SCSI commands supported by the IDTC.

IDTC Supported SASI/SCSI Commands

CLASS	CODE	DESCRIPTION
0	$00	test drive ready
0	$01	recalibrate
0	$03	request sense status
0	$04	format drive
0	$06	format track (XEBEC S1410/S1410A)
0	$08	read
0	$0A	write
0	$0B	seek
0	$0C	init drive characteristics (XEBEC S1410/S1410A)
0	$15	mode select (ADAPTEC ACB 4000)
7	$04	controller diagnostics

Status and Completion Bytes

After having executed a command, the hard disk controller transmits a status and completion message to the IDTC. This message has a length of 2 bytes and the format described below.

Expected SASI/SCSI Status and Completion Message

BIT	7	6	5	4	3	2	1	0
Byte 0	0	0	DC.	0	DC.	DC.	error	0
Byte 1	0	0	0	0	0	0	0	0

DC. = don't care

Because XEBEC and ADAPTEC controllers send different status and completion messages in byte 0, only bit 1 in byte 0 is monitored by the IDTC.

Bit 1 of byte 0 is set if the hard disk controller detected an error during the command execution. Otherwise, it is cleared.

Byte 1 is a zero byte which indicates that the command execution is complete and the SASI/SCSI bus is released.

Controller Sense Blocks

If the hard disk controller flags an error in the status byte, the IDTC transmits the *Request Sense Status* command to the controller. This command causes the controller to return a Controller Sense Block (CSB) which describes the error being detected. The CSB has a length of 4 bytes and the format described below.

Expected SASI/SCSI Controller Sense Block

BIT	7	6	5	4	3	2	1	0
Byte 0	a.v.	0	error type		error code			
Byte 1	DC.			logical sector addr (high)				
Byte 0	logical sector addr (mid)							
Byte 1	logical sector addr (low)							

DC. = don't care

Bit 7 of byte 0 is the address valid bit. It is set if the controller sense block contains a valid address. Otherwise, it is cleared.

Bit 6 of byte 0 is not used and is always cleared.

Errors are categorized in four types: drive errors (type 0), controller errors (type 1), command errors (type 2), and miscellaneous errors (type 3). Bits 4-5 of byte 0 define the error type.

Bits 0-3 of byte 0 contain the error code.

For all errors occurring during disk accesses (read, write, format, seek), bytes 1, 2, and 3 specify the logical sector address of the defective sector. Bits 0-4 of byte 1 is the high order byte, byte 2 is the mid order byte, and byte 3 is the low order byte of the LSA.

The following table lists all error types and codes accepted by the IDTC.

Accepted SASI/SCSI Errors

TYPE	CODE	DESCRIPTION
0	$0	No error
0	$1	No index signal
0	$2	No seek complete
0	$3	Write fault
0	$4	Drive not ready
0	$6	Track zero not found
0	$8	Drive still seeking
1	$0	Uncorrectable ID error
1	$1	Uncorrectable data error
1	$2	ID address mark not found
1	$3	Data address mark not found
1	$4	Sector not found
1	$5	Seek error
1	$9	Bad track/ECC error during verify
1	$A	Format error
1	$C	Bad drive format
1	$D	Self test failed
2	$0	Invalid command
2	$1	Illegal sector address
2	$2	Invalid parameter
2	$3	Volume overflow
2	$4	Bad argument
2	$5	Invalid logical unit number
3	$0	RAM error
3	$1	Program error
3	$2	ECC polynominal error

FLOPPY DISK/TAPE CONTROLLER

The IDTC is capable of controlling four floppy disk drives or one CIPHER 525 FloppyTape drive with SA 850 interface plus two floppy disk drives.

For each floppy disk drive the user may independently select media size (5-1/4 inch or 8-inch), number of media sides (one or two), data density (FM or MFM), format (Motorola or IBM), and several other parameters (sector size, number of sectors per track, number of cylinders, write precompensation or read post compensation, interleave factor, spiral effect, stepping rate).

The FloppyTape recording technique is MFM, with a format similar to the IBM 34 floppy disk format. The data organized into sectors which are contained in fixed length segments on 6 streams on the tape. Sector length, number of sectors per segment, and segment skip factor are user selectable.

Floppy Disk/Tape Controller Hardware

The floppy disk/tape controller provides the interface between the peripheral interface bus on the IDTC and the floppy disk and tape drives. A block diagram is shown in Figure 4-7.

The disk/tape operations are controlled by the Western Digital FD1793 floppy disk controller chip and the supporting components for data separation and write precompensation. In addition, a control register is used for drive, side, and data density selection, and motor control. The floppy disk/tape controller circuits are connected to the peripheral interface bus on the IDTC through an address decoder and interface control logic. The controller is accessed either by the IDTC processor (for selection and control operations) or by the DMA controller (for data transfers).

Floppy Disk/Tape Signals

The floppy disk/tape controller provides two peripheral connectors: the signals for 8-inch floppy disk or a FloppyTape drive are available at connector P4. The signals for 5-1/4 inch drives are available at connector P5. The signals for both connectors are described in Chapter 5.

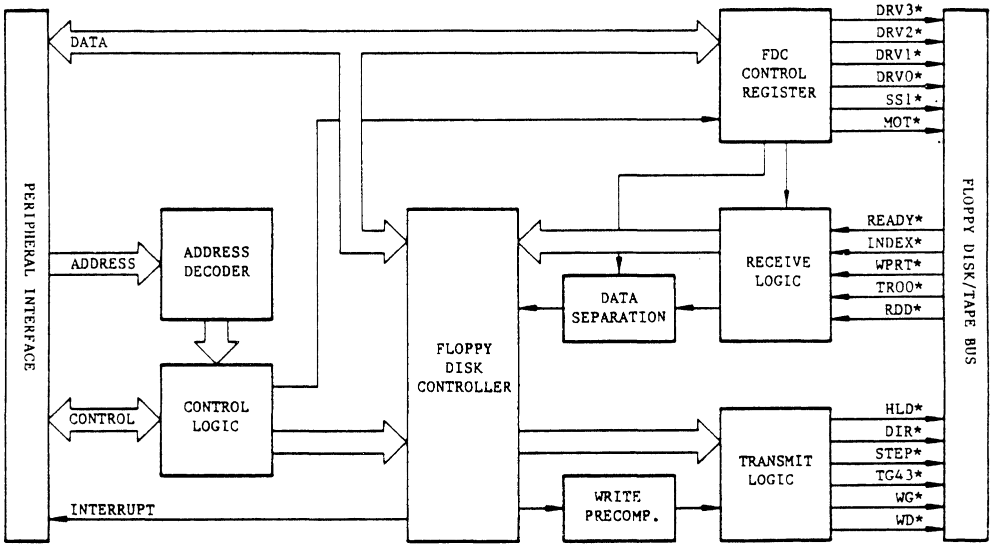


FIGURE 4-7. Floppy Disk/Tape Controller Block Diagram

FLOPPY DISK FUNCTIONS

The following paragraphs describe the floppy disk controller operations, functions, and signals supported by the IDTC.

Drive Selection

Each floppy disk operation is started with a drive selection. The IDTC translates the logical drive number specified in the IDTC command into the physical drive number on the floppy disk interface and asserts the corresponding drive select signal (DRV0*-DRV3*) as shown below:

DRIVE NUMBER	DRIVE SELECT
4	DRV0*
5	DRV1*
6	DRV2*
7	DRV3*

Note that not all 5-1/4 inch floppy disk drives support the DRV3* signal, but use pin 6 of the drive connector for the READY* signal. If such drives are used, the DRV3* output of the IDTC must be disabled by removing the jumper on header K16, and only up to three 5-1/4 inch floppy disk drives may be installed.

To minimize the access time between successive operations on one drive, the IDTC keeps the select signal asserted for 2 seconds after the completion of any command, unless another drive has to be selected immediately.

Drive Motor Control

Most 5-1/4 inch floppy disk drives provide a motor control input (MOT*) for turning the drive motor on and off. To ensure a maximum motor and media lifetime, the IDTC provides a timer function for the motor control: MOT* is asserted or retriggered with each drive selection, and it is negated if no drive is selected within 30 seconds after a command is completed.

Drive Ready Detection

After selecting a drive and before performing any further operations, the floppy disk controller checks if the drive is ready. All 8-inch and most 5-1/4 inch floppy disk drives provide a READY* signal for this purpose.

If 5-1/4 inch floppy disk drives have to be used which do not supply the READY* signal at pin 34 of connector P5, the IDTC provides an optional method for detecting drive readiness: the floppy disk controller may use the INDEX* pulses for retriggering a monostable flip-flop which generates a READY* signal for the controller. This option is selected in the *Initialize Floppy Disk Parameters* command.

However, for maximum system performance it is strongly recommended to use the READY* signal itself, whenever available at connector P5 pin 34.

Head Load Control

Most modern floppy disk drives provide a head load input (HLD*) for loading the read/write head on the disk. The floppy disk controller asserts HLD* before any disk operation and keeps the head loaded as long as the drive is selected. Seek, read, and write operations on the disk are not started until the head load time of 50 milliseconds has elapsed after the activation of HLD*.

Stepping

Each time a low pulse is issued at the step output (STEP*) of the floppy disk controller, the read/write head of the selected drive is moved one cylinder step in the direction determined by the DIR* output. The step pulse length depends on the specified data density and is 2 microseconds for MFM and 4 microseconds for FM. The DIR* signal is high for stepping outward and low for stepping inward. DIR* is valid 12 microseconds before the first stepping pulse is issued.

The stepping rate is the time interval between successive step pulses. It may be defined for each drive in the attribute byte 1 of the *Initialize Floppy Disk Parameters* command according to the specifications of the floppy disk drives used (refer to Chapter 3). The following list shows the available stepping rates for 5-1/4 inch and 8-inch floppy disk drives.

5-1/4 INCH	8-INCH
6 ms	3 ms
12 ms	6 ms
20 ms	12 ms
30 ms	15 ms

Read, write, and verify operations on the disk is not started until the head settling time of 15 milliseconds has elapsed after the last step pulse.

Compensation

When using 8-inch floppy disks with double density (MFM), it might be necessary to compensate the flux shift on the media. This may be achieved by two methods: write precompensation in the disk controller or read post compensation in the disk drive. The IDTC supports both methods.

The method to be used may be specified for each floppy disk drive in the *Initialize Floppy Disk Parameters* command (refer to Chapter 3). This command also specifies the number of the first cylinder with compensation. The write precompensation is adjusted to a value of 200 nanoseconds.

When the read/write head is positioned on a cylinder requiring compensation, the floppy disk controller asserts the output signal TG43*. This causes the drive to reduce the write current or to enable the read post compensation.

Physical Disk Organization

The IDTC supports a wide range of different physical data organizations on the floppy disks. For each disk, the number of disk sides, the number of cylinders, the data density, the sector size, and the number of sectors per track may be specified in the *Initialize Floppy Disk Parameters* command (refer to Chapter 3). The following tables summarizes the valid values for these parameters and illustrates their dependencies.

Physical Floppy Disk Organizations

DISK ORGANIZATION	DISK SIZE 5-1/4 INCH						DISK SIZE 8-INCH					
Number of disk sides	1 or 2						1 or 2					
Number of cylinders	40(48TPI) or 80(96TPI)						77					
Data density	single (FM)			double (MFM)			single (FM)			double (MFM)		
Number bytes/sector	128	256	512	256	512	1024	128	256	512	256	512	1024
Number sectors/track	16	9	5	16	9	5	26	15	8	26	15	8
Number sectors/track 0	16	9	5	16	8	4	26	15	8	26	14	8

Disk Capacity

The formatted floppy disk capacity (total number of data bytes per disk) depends on the number of disk sides, the number of cylinders, the number of sectors per track, and the sector size. It can be calculated using the following formula:

$$\frac{\#\ \text{bytes}}{\text{disk}} \quad \frac{\#\ \text{bytes}}{\text{sector}} \quad \frac{\#\ \text{sectors}}{\text{track}} * \#\ \text{cylinders} * \#\ \text{sides}$$

Interleaving

Interleaving is a method of mapping logically contiguous sectors on a track onto non-adjacent physical sectors. This technique is used for optimizing the speed of systems which require additional processing time between transferring successive sectors.

The interleave factor is equal to the number of physical sectors between two contiguous logical sectors plus one. It may be specified for each drive in the *Initialize Floppy Disk Parameters* command (refer to Chapter 3) and effects the logical sector numbering during format operations. The maximum value for the interleave factor is the number of sectors per track minus one. The following table illustrates the logical sector numbering on a track containing 16 sectors for several interleave factors.

Interleave Scheme

INTERLEAVE FACTOR	LOGICAL SECTOR NUMBERS ON THE TRACK															
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	1	9	2	10	3	11	4	12	5	13	6	14	7	15	8	16
3	1	12	7	2	13	8	3	14	9	4	15	10	5	16	11	6
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
13	1	6	11	16	5	10	15	4	9	14	3	8	13	2	7	12
14	1	9	8	16	7	15	6	14	5	13	4	12	3	11	2	10
15	1	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

Because the IDTC provides a local buffer capable of storing one complete track, it reaches its maximum speed with floppy disks being formatted without interleaving. Therefore, it is recommended to specify an interleave factor of 1 for floppy disks used mainly with the MVME319.

Spiral Offset

The spiral offset is a disk formatting technique which optimizes the system speed when multiple continuous sectors are read or written which overstep track boundaries. When the read/write head steps from one track to the next, data transfers cannot continue until the head settling time has elapsed. To save one disk revolution when stepping from the last sector on the track, the logical sector numbers on each track may be offset against the logical sector numbers of the previous track by a specified number of physical sectors.

The IDTC provides an optional spiral offset of two sectors. This option is selected in the *Initialize Floppy Disk Parameters* command and affects the logical sector numbering during format operations. The following table illustrates the logical sector numbering on a disk with 16 sectors per track without interleaving if the spiral offset is used.

Spiral Offset

TRACK NUMBER	LOGICAL SECTOR NUMBERS ON THE TRACKS															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	15	16	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	13	14	15	16	1	2	3	4	5	6	7	8	9	10	11	12
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
77	7	8	9	10	11	12	13	14	15	16	1	2	3	4	5	6
78	5	6	7	8	9	10	11	12	13	14	15	16	1	2	3	4
79	3	4	5	6	7	8	9	10	11	12	13	14	15	16	1	2

Record Formats

The IDTC supports four formats for the data recorded on a floppy disk: IBM 3740 and Motorola FM format for single density (FM) disks, and IBM System 34 and Motorola MFM format for double density (MFM) disks. The formats used have to be specified by the *Initialize Floppy Disk Parameters* command for each floppy disk drive in the system before the first disk access in a session.

Single Density Formats

Single density floppy disks may be formatted either in IBM 3740 or in Motorola FM format. In either format, the sector size may be 128, 256, or 512 bytes. Depending on the sector size, the number of sectors per track is 16, 9, or 5 for 5-1/4 inch disks or 26, 15, or 8 for 8-inch disks, respectively.

The difference between the two formats is the side and sector numbering scheme in the sector ID fields on double sided disks. In IBM 3740 format, the side number byte reflects the physical disk side (0 or 1), and on both disk sides the starting sector number is 1. However, in Motorola FM format on both disk sides the side number byte is 0, and the sectors are numbered from 1 to n on side 0, from n+1 to 2*n on side 1 (where n is the number of sectors per track).

One track in IBM 3740 and Motorola FM formats is shown in Table 4-1.

Double Density Formats

Double density floppy disks may be formatted either in IBM System 34 or in Motorola MFM format. In either format, the sector size may be 256, 512, or 1024 bytes. Depending on the sector size, the number of sectors per track is 16, 9, or 5 for 5-1/4 inch disks or 26, 15, or 8 for 8-inch disks, respectively. The difference between the two formats is the same as described for single density floppy disks.

On double density floppy disks, track 0 on side 0 is always formatted in single density with a sector size of half the value specified for the disk. If an even number of sectors per track is specified for the disk, the number of sectors on track 0 is the specified number. If an odd number of sectors per track is specified for the disk, the number of sectors on track 0 is the specified number minus 1. This formatting method guarantees that track 0, where the operating system usually stores the disk ID and format parameters, can be read in any system for defining the disk parameters.

One track in IBM System 34 and Motorola MFM formats is shown in Table 4-2.

TABLE 4-1. IBM 3740 and Motorola FM Formats

```
=========================================================================
 NO. OF BYTES   CONTENTS                    DESCRIPTION
=========================================================================

     40         $FF..FF     gap 0
      6         $00..00     gap 0
      1         $FC         index mark
     26         $FF..FF     gap 1
      6         $00..00     gap 1 ----------------------------+
      1         $FE         ID address mark                   |
      1         $xx         track number:                     |
                            $00..27 for 5-1/4 inch 48 TPI     |
                            $00..4F for 5-1/4 inch 96 TPI     |
                            $00..4C for 8-inch 48 TPI         |
      1         $xx         side number:                      |
                            $00, $01 for IBM format           |
                            $00 for Motorola format           |  this field
      1         $xx         sector number:                    |  repeated
                            $01..10 for 5-1/4 inch IBM format |  5, 9, or 16
                            $01..20 for 5-1/4 inch Mot format |- times for
                            $01..1A for 8-inch IBM format     |  5-1/4 inch,
                            $01..34 for 8-inch Mot format     |  8, 15, or 26
      1         $xx         sector size:                      |  times for
                            $00 for 128 bytes                 |  8-inch disks
                            $01 for 256 bytes                 |
                            $10 for 512 bytes                 |
      2         $xxxx       ID CRC bytes                      |
     11         $FF..FF     gap 2                             |
      6         $00..00     gap 2                             |
      1         $FB         data address mark                 |
  128/256/512   $xx..xx     data                              |
      2         $xxxx       data CRC bytes                    |
     27         $FF..FF     gap 3 ----------------------------+
     xxx        $FF..FF     gap 4
=========================================================================
```

TABLE 4-2. IBM System 34 and Motorola MFM Format

NO. OF BYTES	CONTENTS	DESCRIPTION	
80	$4E..4E	gap 0	
12	$00..00	gap 0	
3	$C2..C2	gap 0	
1	$FC	index mark	
50	$4E..4E	gap 1	
12	$00..00	gap 1 -----------------------+	
3	$A1..A1	gap 1	
1	$FE	ID address mark	
1	$xx	track number:	
		$00..27 for 5-1/4 inch 48 TPI	
		$00..4F for 5-1/4 inch 96 TPI	
		$00..4C for 8-inch 48 TPI	
1	$xx	side number:	
		$00, $01 for IBM format	
		$00 for Motorola format	this field
1	$xx	sector number:	repeated
		$01..10 for 5-1/4 inch IBM format	5, 9, or 16
		$01..20 for 5-1/4 inch Mot format	- times for
		$01..1A for 8-inch IBM format	5-1/4 inch,
		$01..34 for 8-inch Mot format	8, 15, or 26
1	$xx	sector size:	times for
		$01 for 256 bytes	8-inch disks
		$10 for 512 bytes	
		$11 for 1024 bytes	
2	$xxxx	ID CRC bytes	
22	$4E..4E	gap 2	
12	$00..00	gap 2	
3	$A1..A1	gap 2	
1	$FB	data address mark	
256/512/1024	$xx..xx	data	
2	$xxxx	data CRC bytes	
54	$4E..4E	gap 3 -----------------------+	
xxx	$4E..4E	gap 4	

Logical Disk Organization

The operating system organizes the data on the floppy disks in blocks of 256, 512, or 1024 bytes and assigns a Continuous Block Number (CBN) to each block. In the IDTC commands, the disk access is specified by block size and CBN of the first block to be read or written. For locating the target block on the floppy disk, the IDTC translates the CBN given in the command into side number, track number, and sector number, using an intermediate Logical Sector Address (LSA).

All sectors on the floppy disk are given logical addresses, starting with address 0 for sector 1 of track 0 on side 0, and assigning continuous addresses in the order of ascending sector numbers. When the last sector of a track on side 0 is reached, the addressing continues with sector 1 of the same track number on side 1. When the last sector of a track on side 1 is reached, the track number is incremented by one, and the addressing continues with sector 1 of the new track number on side 0. This method provides a unique logical sector address for each sector on the disk. Note that these addresses do not appear physically on the disk; they are only used by the IDTC firmware for the calculation process.

Remember that on double density (MFM) floppy disks, track 0 on side 0 is always formatted in single density (FM) with a sector size of half the value specified for the disk. To achieve a constant logical sector size throughout the disk, one logical sector on track 0, side 0 of such disks consists of two physical sectors.

The correlation between the continuous block number in the IDTC command and logical sector address is as follows:

$$\text{logical sector address} = \text{continuous block number} * \frac{\text{block size}}{\text{sector size}}$$

The following examples illustrate the floppy disk organization. The organization of a single sided , single density 5-1/4 inch floppy disk with a sector size of 128 bytes and a block size of 256 bytes is shown in the first table below. The organization of a double sided , double density 5-1/4 inch floppy disk with a sector size of 256 bytes and a block size of 512 bytes is shown in the second table below.

FM Floppy Disk Organization Example

```
      +---------------------------+---------------------------+---------------
TRACK |             0             |             1             |        2 ...
      +-----+-----+---+-----+-----+-----+-----+---+-----+-----+-----+---------
PSN   |  1  |  2  |...| 15  | 16  |  1  |  2  |...| 15  | 16  |  1  |.......
      +-----+-----+---+-----+-----+-----+-----+---+-----+-----+-----+---------
LSA   |  0  |  1  |...| 14  | 15  | 16  | 17  |...| 30  | 31  | 32  |.......
      +-----+-----+---+-----+-----+-----+-----+---+-----+-----+-----+---------
CBN   |     0     |...|     7     |     8     |...|    15     |    16     ...
      +-----------+---+-----------+-----------+---+-----------+---------------
```

MFM Floppy Disk Organization Example

```
      +-------------------------------------------------------+---------------
TRACK |                           0                           |        1 ...
      +---------------------------+---------------------------+---------------
SIDE  |             0             |             1             |        0 ...
      +--+--+--+--+---+--+--+--+--+-----+-----+---+-----+-----+-----+---------
PSN   | 1| 2| 3| 4|...|13|14|15|16|  1  |  2  |...| 15  | 16  |  1  |.......
      +--+--+--+--+---+--+--+--+--+-----+-----+---+-----+-----+-----+---------
LSA   |  0  |  1  |...|  6  |  7  |  8  |  9  |...| 22  | 23  | 24  |.......
      +-----+-----+---+-----+-----+-----+-----+---+-----+-----+-----+---------
CBN   |     0     |...|     3     |     4     |...|    11     |    12     ...
      +-----------+---+-----------+-----------+---+-----------+---------------
```

FLOPPYTAPE FUNCTIONS

The IDTC is capable of controlling a Cipher 525 FloppyTape with SA 850 interface, which appears to the controller like an 8-inch floppy disk drive. Because the data organization on a floppy tape is sector oriented as a floppy disk, single or contiguous blocks can be read or written using the tape in a quasi start/stop mode.

The following paragraphs describe the FloppyTape operations, functions, and signal supported by the IDTC.

Drive and Stream Selection

4

The FloppyTape is selected as logical drive number 4. The tape contains 6 streams, which can be considered as the equivalent of the disk drives. The IDTC accesses a specific stream by using the drive select lines DRV0* and DRV1* plus side select line SS1* as follow:

DRV0*	DRV1*	SS1*	STREAM
0	1	1	0
0	1	0	1
1	0	1	2
1	0	0	3
0	0	1	4
0	0	0	5

To optimize the tape access time between successive operations on the same drive, the IDTC keeps the selected signals asserted for two seconds after the completion of any command, unless another selection has to be made immediately.

Drive Motor Control

The head load line on the SA 850 interface serves as the FloppyTape motor on control signal. This signal is asserted in order to read or write data. After activation of the signal, 400 ms delay is introduced prior to any data operation to allow the medium to reach full operating speed.

Drive Ready Detection

After having selected a drive and before performing any further operations, the floppy disk/tape controller checks if the selected drive is ready. The FloppyTape drive provides a READY* signal for this purpose. This signal is asserted by the FloppyTape drive when a tape cartridge is inserted and the retention pass is completed.

Stepping

The streams on the tape are divided into fixed length segments, which can be considered as the equivalent of disk tracks. Each time the floppy disk/tape controller asserts the STEP* control line, the tape is moved one segment relative to the R/W head. The direction of movement is determined by the state of the DIR* control line. If the DIR* signal is low, the tape movement is downstream to the next segment; if the DIR* signal is high, the tape movement is upstream to the previous segment.

Compensation

The floppy disk/tape controller is factory adjusted to give a write precompensation of 200 ns. This value may not be changed.

Physical Tape Organization

The FloppyTape is physically divided into 6 streams, each consisting of 255 fixed length segments. A segment may be considered as a track on a floppy disk. Because the floppy disk/tape controller on the IDTC does not support track numbers above 245, only segments 0-244 are used. The total number of segments available is therefore 6 * 245 = 1470. Each segment is divided into sectors as with a track on a floppy disk. The IDTC firmware supports sector sizes of 256, 512, or 1024 bytes. The sector size determines the maximum number of sectors per segment.

The figure below shows the physical tape layout. The correlation between sector sizes, the resultant number of sectors per segment and tape, and the tape capacity is shown in Table 4-3.

FloppyTape Layout

```
          +------------+-------------------------+-------------+------------
Stream 0  |  Segment 0 | - - - - - - - - - - - ->| Segment 244 | - - - - - ->
          +------------+--+-------------+--------+-------------+--+---------
Stream 1  | <- - - - - -  | Segment 244 | <- - - - - - - - - - - - | Segment 0
          +------------+--+-------------+--------+-------------+--+---------
Stream 2  |  Segment 0 | - - - - - - - - - - - ->| Segment 244 | - - - - - ->
          +------------+--+-------------+--------+-------------+--+---------
Stream 3  | <- - - - - -  | Segment 244 | <- - - - - - - - - - - - | Segment 0
          +------------+--+-------------+--------+-------------+--+---------
Stream 4  |  Segment 0 | - - - - - - - - - - - ->| Segment 244 | - - - - - ->
          +------------+--+-------------+--------+-------------+--+---------
Stream 5  | <- - - - - -  | Segment 244 | <- - - - - - - - - - - - | Segment 0
          +------------+--+-------------+--------+-------------+--+---------
```

TABLE 4-3. Physical Tape Organization

BYTES/ SECTOR	SECTORS/ SEGMENT	SEGMENTS/ STREAM	SECTORS/ TAPE	TAPE CAPACITY
256	44	245	64680	16.6Mb
512	32	245	47040	24.1Mb
1024	17	245	24990	25.6Mb

Recording Format

The recording technique on the FloppyTape is MFM, with a format similar to the IBM System 34 floppy disk format. The data rate is 500 Kbits/s. Physical sector size and number of sectors per segment are user selectable and specified with the *Initialize FloppyTape Parameters* command. The FloppyTape recording format is shown in Table 4-4.

TABLE 4-4. FloppyTape Recording Format

NO. OF BYTES	CONTENTS	DESCRIPTION	
80	$4E..4E	gap 0	
12	$00..00	sync bytes	
3	$C2..C2	control bytes	
1	$FC	index character	
50	$4E..4E	gap 1	
12	$00..00	sector ID sync bytes	
3	$A1..A1	control character	
1	$FE	ID address mark	
1	$xx	segment number ($00..F5)	
1	$xx	stream number ($00..05)	
1	$xx	sector number ($00..35)	
1	$xx	sector size:	
		$01 for 256 bytes	this field
		$10 for 512 bytes	repeated
		$11 for 1024 bytes	for all
2	$xxxx	ID CRC bytes	sectors
22	$4E..4E	gap 2	
12	$00..00	sector data sync bytes	
3	$A1..A1	control characters	
1	$FB	data address mark	
256/512/1024	$xx..xx	data	
2	$xxxx	data CRC bytes	
54/54/84	$4E..4E	gap 3	
xxx	$4E..4E	gap 4	

Segment Skipping and Logical Streams

The Cipher 535 FloppyTape drive is provided for streaming operation. To keep the tape streaming, the IDTC reads and writes data via a 21K buffer which accepts one complete tape segment. Because in most cases the time interval between reading or writing a succeeding tape segment is too short for copying data from or to a hard disk, 1-3 tape segments can be skipped before transferring the next tape segment. The skipped segments are accessed in 1-3 succeeding passes. The skip factor is specified in the *Initialize FloppyTape Parameters* command.

The figure below gives an example for a tape read with a skip factor of 2 in two passes. Because each physical stream is read twice in this example, the IDTC firmware would deal with 12 logical streams. A physical stream may consist of 1, 2, 3, or 4 logical streams according to the skip factors 1, 2, 3, and 4.

Segment Skipping

```
=============================================================================
 LOGICAL                                                          PHYSICAL
 STREAM                     ACCESSED SEGMENTS                     STREAM
=============================================================================
             +---+---+---+---+---+--------+---+---+---+---+---+
    0        | 0 |-->| 2 |-->| 4 | - - - ->|240|-->|242|-->|244| --      0
             +---+---+---+---+---+--------+---+---+---+---+---+   )
    1     -- |<--|243|<--|241|<- - - - - - - - | 3 |<--| 1 |<--|<--     1
         (   +---+---+---+---+---+--------+---+---+---+---+---+
    2     -->| 0 |-->| 2 |-->| 4 | - - - ->|240|-->|242|-->|244| --      2
             +---+---+---+---+---+--------+---+---+---+---+---+   )
    3     -- |<--|243|<--|241|<- - - - - - - - | 3 |<--| 1 |<--|<--     3
         (   +---+---+---+---+---+--------+---+---+---+---+---+
    4     -->| 0 |-->| 2 |-->| 4 | - - - ->|240|-->|242|-->|244| --      4
             +---+---+---+---+---+--------+---+---+---+---+---+   )
    5     -- |<--|243|<--|241|<- - - - - - - - | 3 |<--| 1 |<--|<--     5
         (   +---+---+---+---+---------------+---+---+---+---+
    6     -->|-->| 1 |-->| 3 | - - - - - - - ->|241|-->|243|-->| --     0
             +---+---+---+---+---+--------+---+---+---+---+---+   )
    7     -- |244|<--|242|<--|240|<- - - - | 4 |<--| 2 |<--| 0 |<--     1
         (   +---+---+---+---+---+--------+---+---+---+---+---+
    8     -->|-->| 1 |-->| 3 | - - - - - - - ->|241|-->|243|-->| --     2
             +---+---+---+---+---+--------+---+---+---+---+---+   )
    9     -- |244|<--|242|<--|240|<- - - - | 4 |<--| 2 |<--| 0 |<--     3
         (   +---+---+---+---+---+--------+---+---+---+---+---+
   10     -->|-->| 1 |-->| 3 | - - - - - - - ->|241|-->|243|-->| --     4
             +---+---+---+---+---+--------+---+---+---+---+---+   )
   11        |244|<--|242|<--|240|<- - - - | 4 |<--| 2 |<--| 0 |<--     5
             +---+---+---+---+---+--------+---+---+---+---+---+
=============================================================================
```

4

Logical Tape Organization

The operating system organizes the data on the tape in blocks of 256, 512, or 1024 bytes and assigns a Continuous Block Number (CBN) to each block. In the IDTC commands, the tape access is specified by block size and CBN of the first block to be read or written. For locating the target block on the tape, the IDTC translates the CBN given in the command into stream number, segment number, and sector number, using an intermediate Logical Sector Address (LSA).

All sectors on the floppy disk are given logical addresses, starting with address 0 for sector 1 of segment 0 on stream 0, and assigning continuous addresses in the order of ascending sector numbers. When the last sector of a segment is reached, the specified number of segments is skipped, and the addressing continues with sector 1 of the following segment on the same stream. When the end of a stream is reached, the addressing continues with the first segment after skipping on the following logical stream. This method provides a unique logical sector address for each sector on the tape. Note that these addresses do not appear physically on the tape; they are only used by the IDTC firmware for the calculation process.

The correlation between the continuous block number in the IDTC command and logical sector address is as follows:

$$\text{logical sector address} = \text{continuous block number} * \frac{\text{block size}}{\text{sector size}}$$

The block size must be at least the sector size.

Handling of Media Defects on the Tape

After formatting the tape, an automatic verification pass is performed by reading all tape streams with the user specified segment skip factor. If a sector cannot be found or a CRC error is detected, the corresponding segment number is stored in a list in IDTC local RAM and the verification continues with the next segment. At the end of the verification pass, the controller writes this bad segment list into the fourth sector in segment 0 on stream 0 (logical sector address 3).

The bad segment list is shown in Table 4-5. It has a fixed size of 256 bytes and lists every bad segment by its physical segment number starting with logical stream 0. A total of 9 bad segments per logical stream are allowed and if this is exceeded, a drive error occurs. Note that a tape may have a maximum of 4 * 6 = 24 logical streams (segment skip factor = 4) and therefore 24 * 9 = 216 bad segments.

TABLE 4-5. Bad Segment List

BYTE NUMBER	DESCRIPTION
0-1	total number of good segments on tape
2-15	$00 (not used)
16	total number of bad segments on logical stream 0
17-25	bad segment numbers of logical stream 0 in ascending order
26	total number of bad segments on logical stream 1
27-35	bad segment numbers of logical stream 1 in ascending order
.	
.	
.	
246	total number of bad segments on logical stream 23
247-255	bad segment numbers of logical stream 23 in ascending order

Whenever sector number 3 is read, the bad segment list is loaded into the IDTC and the bad segments are locked out for all further tape operations.

Note that a tape cartridge cannot be used if sector number 3 is defective. Also the user should never overwrite this sector.

FloppyTape Operating Recommendations

The following recommendations result from experiences made by Motorola during design and test of the IDTC with the Cipher 525 FloppyTape.

The recommended tape cartridge is 3M DC600A.

Do not operate the FloppyTape at temperatures above 40 degrees C or below 5 degrees C.

Remove the cartridge from the FloppyTape drive (or unlock the lever) whenever the tape drive is not to be accessed for periods longer than 3 hours. If this is not done, the pressure of the head assembly on the tape material deforms the tape at that point and this leads to CRC or seek errors.

Before using a new tape cartridge for the first time, the tape must be formatted. The user can specify the tape format according to the specific requirements. The following tape format is suggested which offers a usable capacity of 24.1Mb.

sector size in bytes	=	512
number of sectors per segment	=	32
skip factor	=	2

A tape with this format is completely recorded in two passes after approximately 20 minutes. The skip factor of 2 allows a hard disk to be accessed between reading successive 16K segments. If a backup time of 10 minutes (skip factor = 1) is required and a lower tape capacity is acceptable, the number of sectors per segment can be reduced while retaining the same sector size. This generates gaps between the segments that enable intermediate hard disk accesses to be made.

4

After the tape is formatted and verified, the bad segment list should be inspected by reading sector number 3. This table normally contains no entries. If it does, and the tape drive has been proved to be operating correctly by checking with a reference cartridge, the formatted tape cartridge should be used with care because later media defects on other segments can be expected.

Each time a formatted tape cartridge is mounted, sector number 3 must be read. This loads the bad segment list into the IDTC and thereby ensures that bad segments on the tape are locked out for all further tape operations.

Because the skip factor which was specified at format time has no impact on the physical sector numbering, the skip factor can be changed on a formatted tape as long as there are no bad segments on the cartridge. This feature is useful when optimizing copy or at backup times.

Bytes 2-15 in the bad segment list can be used to store any user supplied information about the tape, such as format parameters or tape identification.

CHAPTER 5 - SUPPORT INFORMATION

INTRODUCTION

This chapter provides the interconnection signals, parts list with parts location illustration, calibration procedure, and schematic diagram for the MVME319 module.

INTERCONNECT SIGNALS

The MVME319 module interconnects with the VMEbus through connector P1. P3 is a 50-pin connector used for interconnection of SASI/SCSI internal controllers and hard disk drives. A 50-pin connector (P4) interconnects internal 8-inch floppy disk and FloppyTape drives with the controller. P5 is a 34-pin connector for interconnection of internal 5-1/4 inch floppy disk drives.

Connector P1 Interconnect Signals

Connector P1 is a standard DIN 41612 triple-row, 96-pin male connector. The MVME319 interconnects with the VMEbus through rows A, B, and C of P1. Each pin connection, signal mnemonic, and signal characteristic for the connector are listed in Table 5-1.

TABLE 5-1. Connector P1 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
A1-A8	D00-D07	Data bus (bits 0-7) - eight of 16 bidirectional data lines that are used for data transfer to and from system memory in master mode. In slave mode, D00-D07 are connected with the command channel. D00-D07 are also used by the interrupter when supplying the status ID byte.
A9	GND	GROUND
A10	SYSCLK	Not used.
A11	GND	GROUND
A12	DS1*	DATA STROBE 1 - an active low bidirectional signal that indicates a data transfer on data lines D08-D15. In master mode, DS1* is a three-state output; in slave mode; it is an input.

TABLE 5-1. Connector P1 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
A13	DS0*	DATA STROBE 0 - an active low bidirectional signal that indicates a data transfer on data lines D00-D07. In master mode, DS0* is a three-state output; in slave mode, it is an input.
A14	WRITE*	WRITE - an active low bidirectional signal that specifies the direction of a data transfer. A high level indicates a read operation, a low level indicates a write operation. In master mode, WRITE* is a three-state output; in slave mode, it is an input.
A15	GND	GROUND
A16	DTACK*	DATA TRANSFER ACKNOWLEDGE - an active low bidirectional signal that indicates the successful completion of a data transfer. In master mode, DTACK* is an input; in slave mode, it is an open-collector output.
A17	GND	GROUND
A18	AS*	ADDRESS STROBE - an active low bidirectional signal. The falling edge of this signal indicates that a valid address is placed on the bus. In master mode, AS* is a three-state output; in slave mode, it is an input. During bus arbitration, the rising edge of AS* indicates the end of the last bus cycle.
A19	GND	GROUND
A20	IACK*	INTERRUPT ACKNOWLEDGE - an active low input signal that indicates an interrupt status/ID byte fetch on the data transfer bus.
A21	IACKIN*	INTERRUPT ACKNOWLEDGE IN - IACKIN* and IACKOUT* are an active low signal pair that form a daisy-chain through all interrupters in the system. The IACKIN* input indicates that the interrupter may supply the status/ID byte if it has an interrupt request pending on the acknowledged priority level.

TABLE 5-1. Connector P1 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
A22	IACKOUT*	INTERRUPT ACKNOWLEDGE OUT - see IACKIN*. The IACKOUT* totem-pole output propagates the acknowledge signal further down the daisy-chain when the interrupter has no request pending on the acknowledged priority level.
A23	AM4	ADDRESS MODIFIER (bit 4) - one of six lines that provide additional information about the address bus. In master mode, AM0-AM5 are three-state outputs for addressing system memory. In slave mode, AM0-AM5 are inputs for addressing the command channel.
A24-A30	A07-A01	ADDRESS bus (bits 7-1) - seven of 23 address lines. In master mode, A01-A23 are three-state outputs for addressing system memory. In slave mode, A01-A15 are inputs for addressing the command channel. A01-A03 are also inputs to the interrupter for determining the priority level of an interrupt acknowledge cycle.
A31	-12 V	-12 Vdc power.
A32	+5 V	+5 Vdc power - used by the logic circuits on the MVME319.
B1	BBSY*	BUS BUSY - an active low open-collector output signal indicating that the module is the current bus master.
B2	BCLR*	Not used.
B3	ACFAIL*	Not used.
B4,B5	BG0IN* BG0OUT*	BUS GRANT IN (level 0) - BG0IN* and BG0OUT* are an active low signal pair that form a bus grant daisy-chain through all bus requesters in the system. One of these pairs is connected with an input and a totem-pole output of the bus requester. The priority level is jumper selectable. The bus grant input indicates to the bus requester that the module may be the next bus master if it has a bus request pending. The bus grant output propagates the bus grant signal further down the daisy-chain when the module does not request the bus.

5

TABLE 5-1. Connector P1 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
B6,B7	BG1IN* BG1OUT*	BUS GRANT IN (level 1) - same as BG0IN* and BG0OUT* on pin B4 and B5.
B8,B9	BG2IN* BG2OUT*	BUS GRANT IN (level 2) - same as BG0IN* and BG0OUT* on pin B4 and B5.
B10,B11	BG3IN* BG3OUT*	BUS GRANT IN (level 3) - same as BG0IN* and BG0OUT* on pin B4 and B5.
B12-B15	BR0*-BR3*	BUS REQUEST (level 0-3) - one of these active low signals is connected with an open-collector output of the VMEbus requester. It indicates that the module requests bus mastership. The priority level is jumper selectable.
B16-B19	AM0-AM3	ADDRESS MODIFIER (bits 0-3) - same as AM4 on pin A23.
B20	GND	GROUND
B21	SERCLK	Not used.
B22	SERDAT*	Not used.
B23	GND	GROUND
B24-B30	IRQ7*-IRQ1*	INTERRUPT REQUEST (7-1) - one of these active low open-collector output signals is used by the interrupter to generate an interrupt request. The priority level is jumper selectable.
B31	+5V STBY	Not used.
B32	+5 V	+5 Vdc power - same as +5 V on pin A32.
C1-C8	D08-D15	DATA bus (bits 8-15) - eight of 16 bidirectional data lines that are used for data transfer to and from system memory in master mode. Not used in slave mode.
C9	GND	GROUND
C10	SYSFAIL*	SYSTEM FAIL - an optional active low open-collector output signal indicating that a severe malfunction has occurred, and that no further commands can be executed.

TABLE 5-1. Connector P1 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
C11	BERR*	Not used.
C12	SYSRESET*	SYSTEM RESET - an active low input signal that causes a complete hardware and firmware reset of the module.
C13	LWORD*	Not used.
C14	AM5	ADDRESS MODIFIER (bit 5) - same as AM4 on pin A23.
C15-C30	A23-A08	ADDRESS bus (bits 23-08) - same as A01-A07 on pins A1-A8.
C31	+12 V	+12 Vdc power.
C32	+5 V	+5 Vdc power - same as +5 V on pin A32.

5

Connector P3 Interconnect Signals

Each pin connection, signal mnemonic, and signal characteristic for the connector is listed in Table 5-2.

TABLE 5-2. Connector P3 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1-49 (ODD)	GND	GROUND
2-16	DB0*-DB7*	DATA BUS (bits 0-7) - 8-bit bidirectional data bus for transferring data between the IDTC and the hard disk controllers.
18	DBP*	Not used.
20-30		Not used.
32	ATN*	Not used.

TABLE 5-2. Connector P3 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
32	SPARE	Not used.
36	BSY*	BUS BUSY - an active low input signal indicating that the selected hard disk controller is ready for SASI/SCSI bus operations.
38	ACK*	ACKNOWLEDGE - an active output signal that terminates a data transfer. When I*/O is low, ACK* indicates that the IDTC has accepted data from the hard disk controller. When I*/O is high, ACK* indicates that the IDTC has placed valid data on the bus.
40	RST*	RESET - an active low output signal that causes all hard disk controllers on the bus to be reset and to enter an idle state.
42	MSG*	MESSAGE - an active low input signal indicating that the hard disk controller has completed the previous command.
44	SEL*	SELECT - an active low output signal indicating that a valid controller ID is placed on the data bus.
46	C*/D	CONTROL/DATA - this input signal specifies the information on the data bus. A low level indicates command and status bytes, a high level indicates data bytes.
48	REQ*	REQUEST - an active low input signal that indicates a data transfer. When I*/O is low, REQ* indicates that the controller has placed valid data on the bus. When I*/O is high, REQ* indicates that the controller will accept data from the IDTC.
50	I*/O	INPUT/OUTPUT - this input signal specifies the data direction on the bus. A low level indicates that the IDTC receives data from the bus, a high level indicates that the IDTC transmits data on the bus.

5

Connector P4 Interconnect Signals

Each pin connection, signal mnemonic, and signal characteristic for the connector is listed in Table 5-3.

TABLE 5-3. Connector P4 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1-49 (ODD)	GND	GROUND
2,4,6		Not used.
8	TG43*	TRACK GREATER 43 - this active low output signal indicates that the read/write head is positioned on a cylinder requiring reduced write current (for write operations) or read postcompensation (for read operations). Note that the default number of the first cylinder with reduced write current (44) may be altered during disk parameter initialization. TG43* is only used by 8-inch floppy disk drives.
10	DUAL*	DUAL SIDED DISK - an active low input signal indicating that a dual sided disk is inserted in the drive. This signal is only supported by 8-inch disk drives.
12		Not used.
14	SS1*	SIDE SELECT - for floppy disk drives, this output signal specifies which side of the disk is accessed. A high level selects side 0, a low level selects side 1. FloppyTape drives use SS1* together with DRV0* and DRV1* as stream select signal.
16		Not used.
18	HLD*	HEAD LOAD - an active low output signal that may be used by the floppy disk drives to load the read/write head on the disk. FloppyTape drives use HLD* to start the tape drive motor.

5

TABLE 5-3. Connector P4 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
20	INDEX*	INDEX - floppy disk drives apply a low pulse at this input each time the index hole in the floppy disk is detected. The falling edge of INDEX* defines the start of a track. FloppyTape drives indicate the start of a segment with the falling edge of INDEX*.
22	READY*	READY - floppy disk drives indicate, at this active low input signal, that the drive is ready for performing operations. Note that some 5-1/4 inch drives use pin 6 for READY* or do not support READY* at all. In such cases, the IDTC can be configured to use the INDEX* signal for checking drive readiness. FloppyTape drives indicate with READY* that the cartridge is in and the retention pass is completed.
24		Not used.
26-32	DRV0*-DRV3*	DRIVE SELECT (0-3) - four active low output signals used for selecting one of up to four floppy disk drives. If a FloppyTape drive is connected, DRV0* and DRV1* serve as stream select signals together with SS1*. DRV2* and DRV3* can then be used as drive select signals for two additional floppy disk drives. On connector P5, the DRV3* signal is optional and must be disabled if a drive is connected which uses pin 6 for the READY* signal.
34	DIR*	DIRECTION - for floppy disk drives, this output signal specifies the direction in which read/write head moves when STEP* pulses are applied. A high level causes the head to step outwards, a low level causes the head to step inwards. For FloppyTape drives, DIR* specifies the direction of tape movement when STEP* pulses are applied. A high level moves the tape toward segment 0 and a low level toward segment 254.
36	STEP*	STEP - for floppy disk drives, a falling edge on this output signal causes the read/write head to be moved one cylinder in the direction specified by the DIR* signal. For FloppyTape drives, STEP* causes the tape to be moved one segment in the direction specified by DIR*.

TABLE 5-3. Connector P4 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
38	WD*	WRITE DATA - this output signal transmits the data to be recorded on the floppy disk or tape. Each low pulse of WD* causes the write current in the head to be reversed.
40	WG*	WRITE GATE - this output signal specifies whether data is read or written on the floppy disk or tape. A high level indicates a read operation and enables the read logic on the drive, a low level indicates a write operation and enables the write logic.
42	TR00*	TRACK ZERO - floppy disk drives indicate, at this active low input signal that the read/write head is positioned on cylinder number 0. FloppyTape drives indicate with TR00* that the read/write head is positioned on tape segment 0 of the selected stream.
44	WPRT*	WRITE PROTECT - an active low input signal indicating that the floppy disk or tape cartridge in the drive is write protected.
46	RDD*	READ DATA - this input signal receives the composite clock and data signal from the floppy disk or tape. Each flux reversal on the recording media causes a low pulse on RDD*.
48,50		Not used.

5

Connector P5 Interconnect Signals

Each pin connection, signal mnemonic, and signal characteristic for the connector is listed in Table 5-4.

TABLE 5-4. Connector P5 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1-49 (ODD)	GND	GROUND
2	HLD*	HEAD LOAD - an active low output signal that may be used by the floppy disk drives to load the read/write head on the disk.
4		Not used.
6	DRV3*	DRIVE SELECT (3) - an active low output signal used for selecting disk drive 3. The DRV3* signal is optional and must be disabled if a drive is connected which uses pin 6 for the READY* signal.
8	INDEX*	INDEX - floppy disk drives apply a low pulse at this input each time the index hole in the floppy disk is detected. The falling edge of INDEX* defines the start of a track.
10-14	DRV0*-DRV2*	DRIVE SELECT (0-2) - three active low output signals used for selecting up to three floppy disk drives.
16	MOT*	MOTOR ON - this active low output signal turns the drive motor on. MOT* is only used by 5-1/4 inch floppy disk drives.
18	DIR*	DIRECTION - for floppy disk drives, this output signal specifies the direction in which read/write head moves when STEP* pulses are applied. A high level causes the head to step outwards, a low level causes the head to step inwards.
20	STEP*	STEP - for floppy disk drives, a falling edge on this output signal causes the read/write head to be moved one cylinder in the direction specified by the DIR* signal.

TABLE 5-4. Connector P5 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
22	WD*	WRITE DATA - this output signal transmits the data to be recorded on the floppy disk. Each low pulse of WD* causes the write current in the head to be reversed.
24	WG*	WRITE GATE - this output signal specifies whether data is read or written on the floppy disk. A high level indicates a read operation and enables the read logic on the drive, a low level indicates a write operation and enables the write logic.
26	TR00*	TRACK ZERO - floppy disk drives indicate, at this active low input signal that the read/write head is positioned on cylinder number 0.
28	WPRT*	WRITE PROTECT - an active low input signal indicating that the floppy disk in the drive is write protected.
30	RDD*	READ DATA - this input signal receives the composite clock and data signal from the floppy disk. Each flux reversal on the recording media causes a low pulse on RDD*.
32	SS1*	SIDE SELECT - for floppy disk drives, this output signal specifies which side of the disk is accessed. A high level selects side 0, a low level selects side 1.
34	READY*	READY - floppy disk drives indicate, at this active low input signal, that the drive is ready for performing operations. Note that some 5-1/4 inch drives use pin 6 for READY* or do not support READY* at all. In such cases, the IDTC can be configured to use the INDEX* signal for checking drive readiness.

FLOPPY DISK/TAPE CONTROLLER CALIBRATION

The data separator and write precompensation circuits of the floppy disk/tape controller are adjusted at the factory and must not be altered by the user. If, for any reason, a calibration becomes necessary, please contact your nearest Motorola Field Service Office.

The following calibration instructions are for service purposes only.

The parts location diagram in this chapter illustrates the locations of the test points and potentiometers.

a. Establish the as-shipped factory jumper configuration on the MVME319.

b. Install the MVME319 in a VME/VERSAdos system.

c. Turn the VMEsystem power on.

d. Connect an oscilloscope to test point TP1 (data separator oscillator).

e. Adjust potentiometer T1 for a frequency of 4 MHz at TP1.

f. Connect a VERSAdos hard disk to the MVME319.

g. Connect a floppy disk drive to the MVME319.

h. Connect an oscilloscope to test point TP2 (write precompensation pulse).

i. Boot VERSAdos and log on.

j. Insert a scratch floppy disk into the floppy disk drive.

k. Initialize the parameters of the connected floppy disk.

l. Format the scratch floppy disk.

m. Adjust potentiometer T2 for a negative pulse width of 200 ns at TP2. (This value must be correctly adjusted in order to ensure error-free recording on the FloppyTape drive.)

PARTS LIST

The components of the MVME319 are listed in Table 5-5. The parts locations are shown in Figure 5-1. These parts reflect the latest issue of hardware at the time of printing.

TABLE 5-5. MVME319 Module Parts List

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
	84-W8532B01B	Printed wiring board
C1,C101,C102	23NW9618A71	Capacitor, electrolytic, 47 uF @ 10 Vdc
C2-C13,C15-C26,C30-C41,C50-C67,C69,C100	21NW9632A03	Capacitor, fixed, ceramic, 0.1 uF @ 50 Vdc
C14	23NW9618A88	Capacitor, electrolytic, 33 uF @ 16 Vdc
C27	21NW9604A75	Capacitor, fixed, ceramic, 330 pF @ 50 Vdc
C28,C45	21SW992C014	Capacitor, fixed, ceramic, 0.01 uF @ 50 Vdc
C29	21NW9604A74	Capacitor, fixed, ceramic, 100 pF @ 50 Vdc
C42	23NW9704A97	Capacitor, tantalum, 0.33 uF @ 35 Vdc
C43	21NW9702A37	Capacitor, fixed, ceramic, 47 pF @ 50 Vdc
C44,C46	23NW9618A80	Capacitor, electrolytic, 10 uF @ 50 Vdc
C47	21NW9702A40	Capacitor, fixed, ceramic, 27 pF @ 50 Vdc
C48	23NW9618A82	Capacitor, electrolytic, 22 uF @25 Vdc
C49	21NW9709A05	Capacitor, fixed, ceramic, DIP, 100 pF @ 50 Vdc
C68	21NW9632A02	Capacitor, fixed, ceramic, 1 uF @ 50 Vdc
DS1	48NW9612A49	LED, red
K1-K3,K6-K8,K15,K16,P2,TP1,TP2	29NW9805C07	Pin, autoinsert, (52 required)
P1	28NW9802E51	Connector, 96-pin
P3	28NW9802F68	Header, 50-pin

TABLE 5-5. MVME319 Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
P4	28NW9802F67	Header, 50-pin
P5	28NW9802F66	Header, 34-pin
Q1	48NW9611A41	Transistor, 2N5086
RP1,RP4	51NW9626B55	Resistor network, SIP, 9/4.7k ohm
RP2,RP3,RP9	51NW9626C12	Resistor network, SIP, 9/1.2k ohm
RP5,RP6,RP7	51NW9626C01	Resistor network, SIP, 8/220/330 ohm
R1,R3,R5	06SW-124A59	Resistor, fixed, film, 2.7k ohm, 5%, 1/4 W
R2,R39-R41	06SW-124A73	Resistor, fixed, film, 10k ohm, 5%, 1/4 W
R6,R15,R16, R18,R19	06SW-124A65	Resistor, fixed, film, 4.7k ohm, 5%, 1/4 W
R7	06SW-124A39	Resistor, fixed, film, 390 ohm, 5%, 1/4 W
R8,R35	06SW-124A43	Resistor, fixed, film, 560 ohm, 5%, 1/4 W
R9,R10	06SW-124A45	Resistor, fixed, film, 680 ohm, 5%, 1/4 W
R11,R14	06SW-124A51	Resistor, fixed, film, 1.2k ohm, 5%, 1/4 W
R13	06SW-124A33	Resistor, fixed, film, 220 ohm, 5%, 1/4 W
R17	06SW-961B43	Resistor, fixed, film, 274 ohm, 1%, 1/4 W
R20	06SW-124A95	Resistor, fixed, film, 82k ohm, 5%, 1/4 W
R22	06SW-124A89	Resistor, fixed, film, 47k ohm, 5%, 1/4 W
R23	06SW-124A13	Resistor, fixed, carbon, 33 ohm, 5%, 1/4 W
R26	06SW-961B76	Resistor, fixed, film, 604 ohm, 1%, 1/4 W
R30	06SW-961D30	Resistor, fixed, film, 20k ohm, 1%, 1/4 W
R31	06SW-961D01	Resistor, fixed, film, 10k ohm, 1%, 1/4 W
R32	06SW-961D66	Resistor, fixed, film, 47.5k ohm, 1%, 1/4 W

TABLE 5-5. MVME319 Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
R33	06SW-961C68	Resistor, fixed, film, 99k ohm, 1%, 1/4 W
T1,T2	18NW9603A47	Resistor, variable, 5k ohm, 12 turn
U1,U2,U12, U19	51NW9615F53	I.C. SN74LS191N
U3	51NW9615G39	I.C. SN74LS133N
U4,U5,U13, U45,U73,U93	51NW9615G19	I.C. SN74LS257N
U6	(NOTE)	I.C. programmed
U7,U16,U71	51NW9615E93	I.C. SN74LS14N
U8	51NW9615M94	I.C. AM9519A-1PC
U9	51NW9615G49	I.C. MC14521BCP
U10,U32,U58, U67	51NW9615R30	I.C. SN74AS373N
U11,U39,U51	51NW9615H89	I.C. SN74LS645-1N
U14	Not used.	
U15	51NW9615F09	I.C. SN74LS266J
U17,U37	51NW9615C29	I.C. SN74LS174N
U18,U21,U49, U66	51NW9615E96	I.C. SN74LS245
U20,U68	51NW9615C95	I.C. SN74S74N
U22	(NOTE)	I.C. programmed
U23,U29,U38, U70	51NW9615E98	I.C. SN74LS373N
U24	(NOTE)	I.C. programmed

TABLE 5-5. MVME319 Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
U25,U31,U80	51NW9615C22	I.C. SN74LS08N
U26	51NW9615M95	I.C. AM9517-5PC
U27	51NW9615L08	I.C. MC68121L1
U28	51NW9615N12	I.C. SN74ALS10AN
U30,U63,U84	51NW9615C24	I.C. SN74LS32N
U33,U36	51NW9615F02	I.C. SN74LS244N
U34	(NOTE)	I.C. programmed
U35	51NW9615J50	I.C. 74LS367AN
U37	51NW9615N56	I.C. 74F174PC
U40	51NW9615C69	I.C. SN74LS138N
U41	51NW9615N10	I.C. SN74ALS00N
U42	(NOTE)	I.C. programmed
U43	51NW9615H41	I.C. SN74LS682N
U44,U52,U60, U77	51NW9615C21	I.C. SN74LS04N
U46	51NW9615F30	I.C. DM74S05N
U47,U57	51NW9615F85	I.C. SN74S38N
U48	51NW9615K48	I.C. 74F241PC
U50	(NOTE)	I.C. programmed
U53	51NW9615N09	I.C. SN74LS113AN
U54,U62	51NW9615U69	I.C. UPD43256C-10
U55	51NW9615N11	I.C. SN74ALS08N
U56	51NW9615E91	I.C. SN74LS00N
U59	51NW9615K49	I.C. 74F241PC

TABLE 5-5. MVME319 Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
U61,U65	51NW9615D26	I.C. SN74S113N
U64	51NW9615H92	I.C. SN74LS112N
U69	(NOTE)	I.C. programmed
U74,U79,U97, U100	51NW9615C25	I.C. SN74LS74AN
U75	51NW9615E77	I.C. SN74LS27N
U76,U82,U96	51NW9615C20	I.C. SN74LS02N
U81	(NOTE)	I.C. programmed
U83	51NW9615N35	I.C. FD1793PL-02
U85,U91, U101	51NW9615R27	I.C. SN74AS533N
U86	51NW9615F01	I.C. SN74LS86N
U87	51NW9615F79	I.C. SN74S240N
U88	51NW9615L87	I.C. WD1691-PE
U89	51NW9615N36	I.C. WD2143PD-03
U90,U94	51NW9615C26	I.C. SN74LS123N
U92	51NW9615N34	I.C. SN74LS628N
U98	51NW9615E99	I.C. SN74LS374N
U99	51NW9615D12	I.C. MC78L05CP
Y1	48AW1015B23	Crystal oscillator, 9.8304 MHz
Y2	48AW1014B03	Crystal oscillator, 4.0 MHz
	09NW9811A29	Socket, DIL, 14-pin (use at U92)
	09NW9811A86	Socket, DIL, 16-pin (use at U50)

5

TABLE 5-5. MVME319 Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
	09NW9811A87	Socket, DIL, 18-pin (use at U89)
	09NW9811A78	Socket, DIL, 20-pin (use at U24,U42,U69, U85,U87,U88,U91,U95,U98,U101)
	09NW9811A64	Socket, DIL, 28-pin (use at U78,U79)
	09NW9811A37	Socket, DIL, 40-pin (use at U83)
	09-W4659B10	Socket, SIL, 10-pin (use at RP7)
	09-W4659B14	Socket, SIL, 14-pin (use at U6,U8,U14,U54, U62)
	09-W4659B20	Socket, SIL, 20-pin (use at U26)
	09-W4659B24	Socket, SIL, 24-pin (use at U27)
	29NW9805B17	Jumper, shorting, insulated (22 required)
	64-W5736B01A	Front panel

NOTE: When ordering, use number labeled on part

5

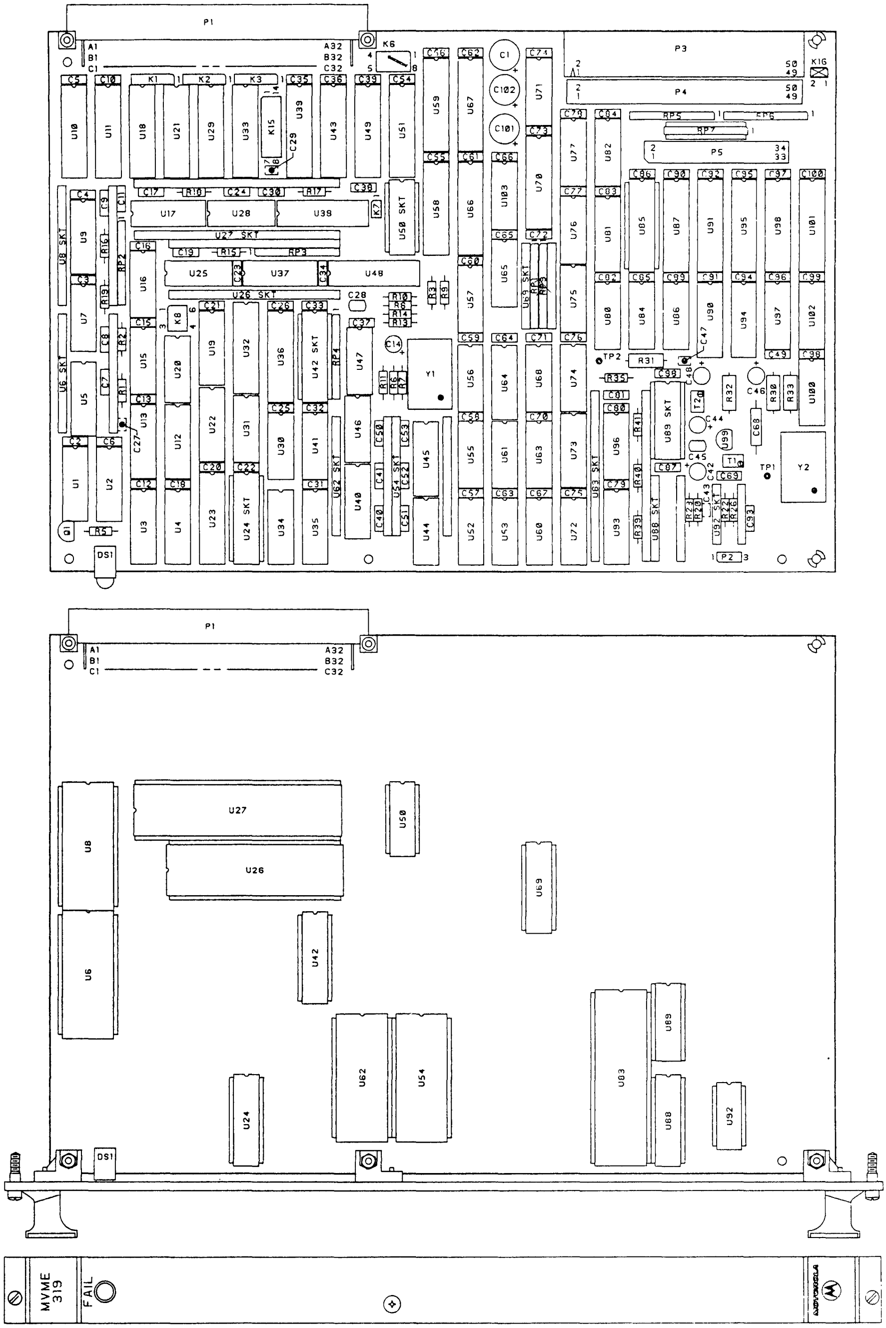


FIGURE 5-1. MVME319 Parts Location

SCHEMATIC DIAGRAM

The schematic diagram for the MVME319 is illustrated in Figure 5-2.

NOTES:

1. FOR REFERENCE DRAWINGS REFER TO BILL(S) OF MATERIAL 01-W3532B__, CURRENT REVISION/CONFIGURATION APPLIES.
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, WITH THE EXECPION OF LOGIC BLOCK SYMBOLOGY.

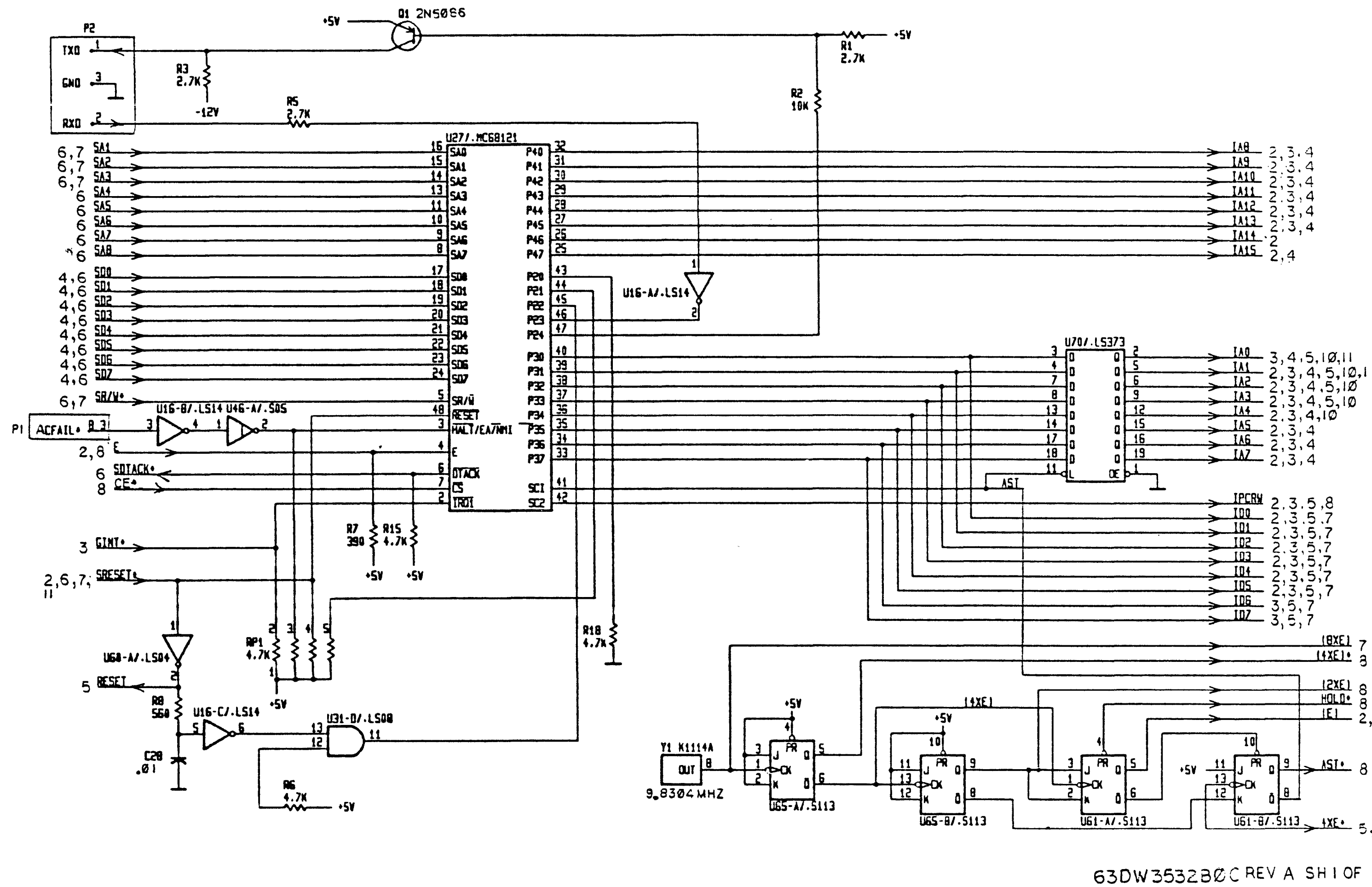


FIGURE 5-2. MVME319 Schematic Diagram

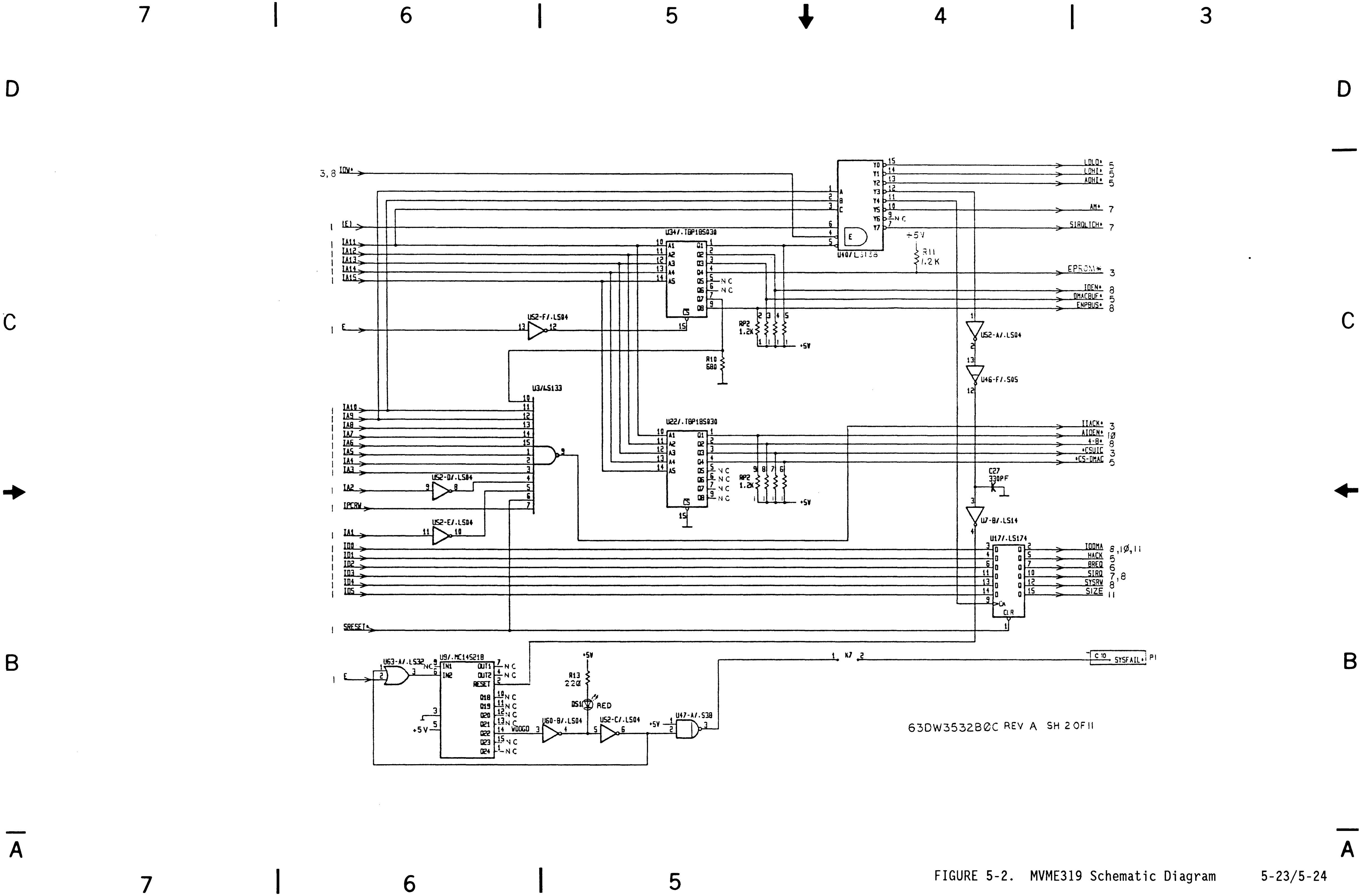


FIGURE 5-2. MVME319 Schematic Diagram

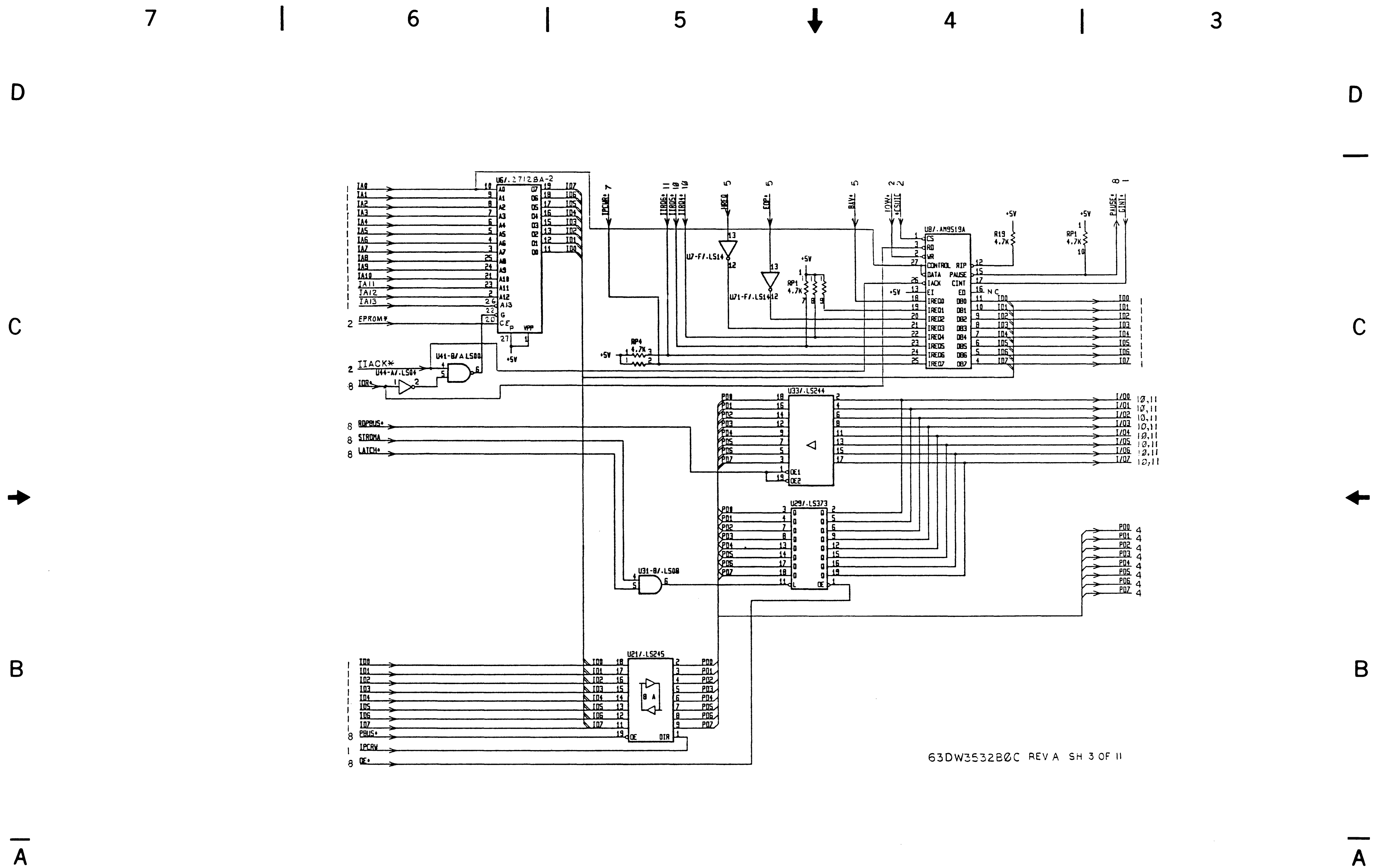
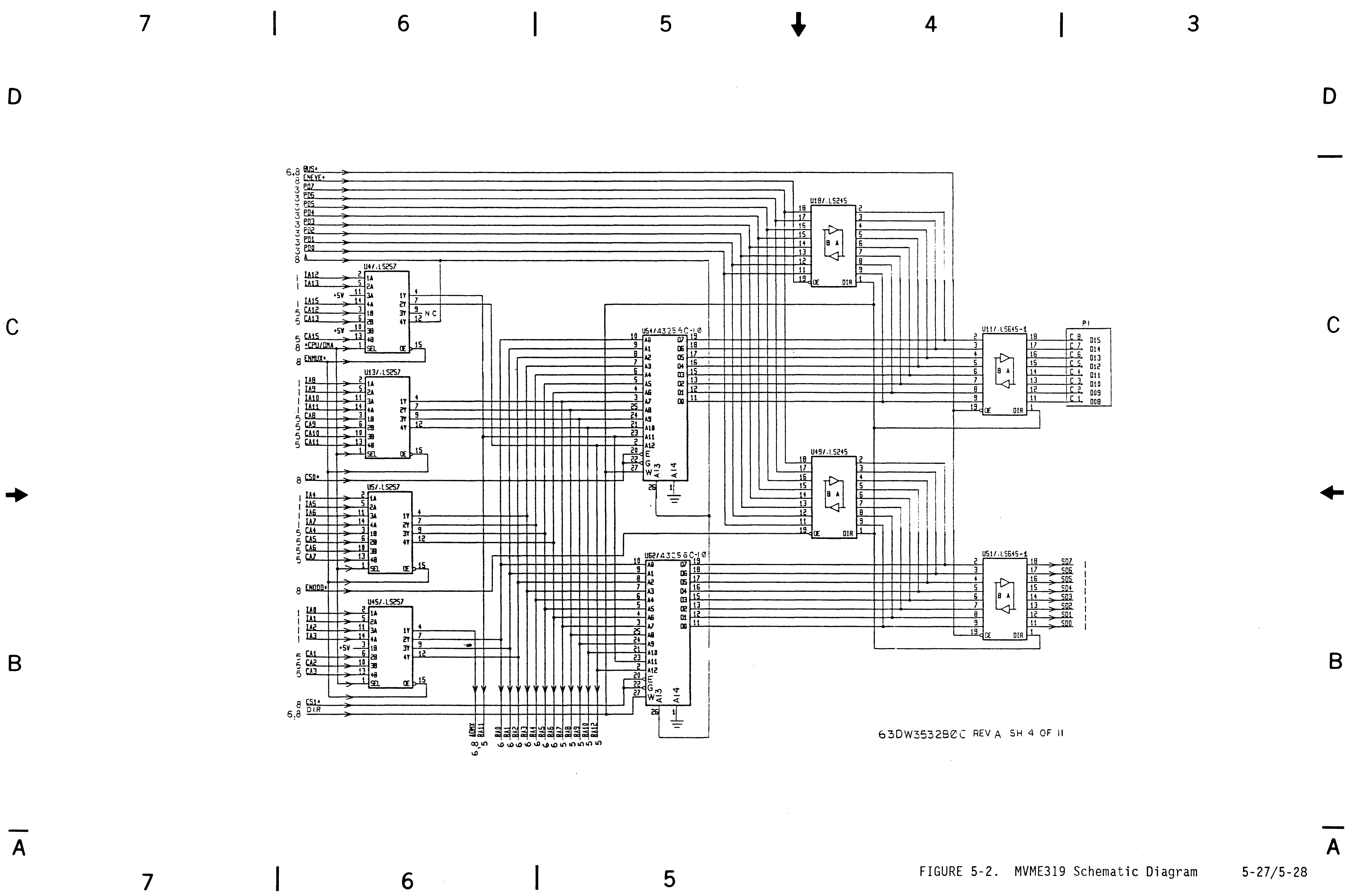


FIGURE 5-2. MVME319 Schematic Diagram



U18/.LS245
U47/.LS257
U13/.LS257
U5/.LS257
U45/.LS257
U54/43256C-10
U62/43256C-10
U49/.LS245
U11/.LS645-1
U51/.LS645-1
P1
BUS*
ENEYE*
PD7
PD6
PD5
PD4
PD3
PD2
PD1
PD0
CS0*
CS1*
ENODD*
ENMUX*
*CPU/DMA
DIR
NC
ADMX
63DW3532B0C REV A SH 4 OF 11

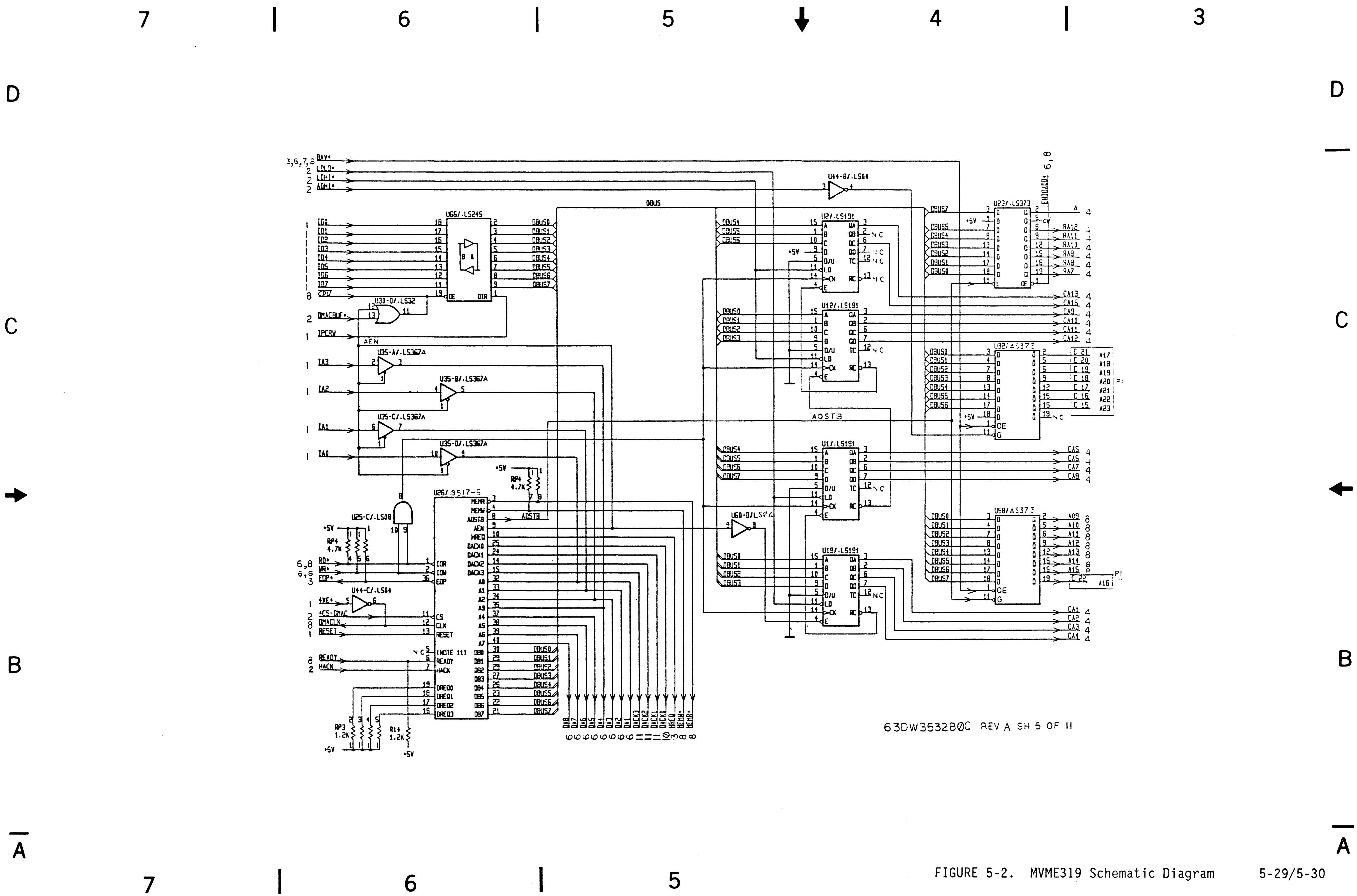


FIGURE 5-2. MVME319 Schematic Diagram

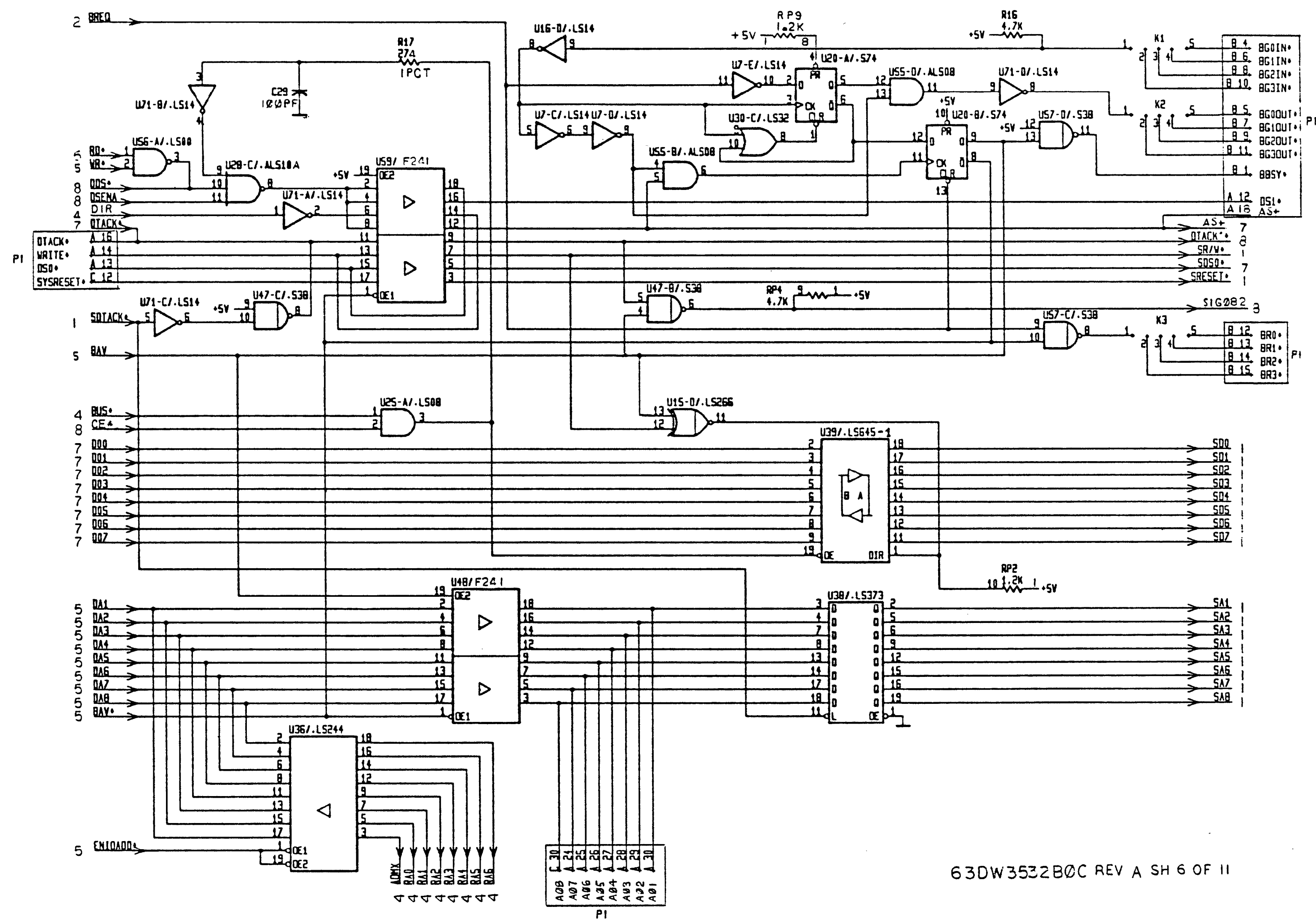


FIGURE 5-2. MVME319 Schematic Diagram

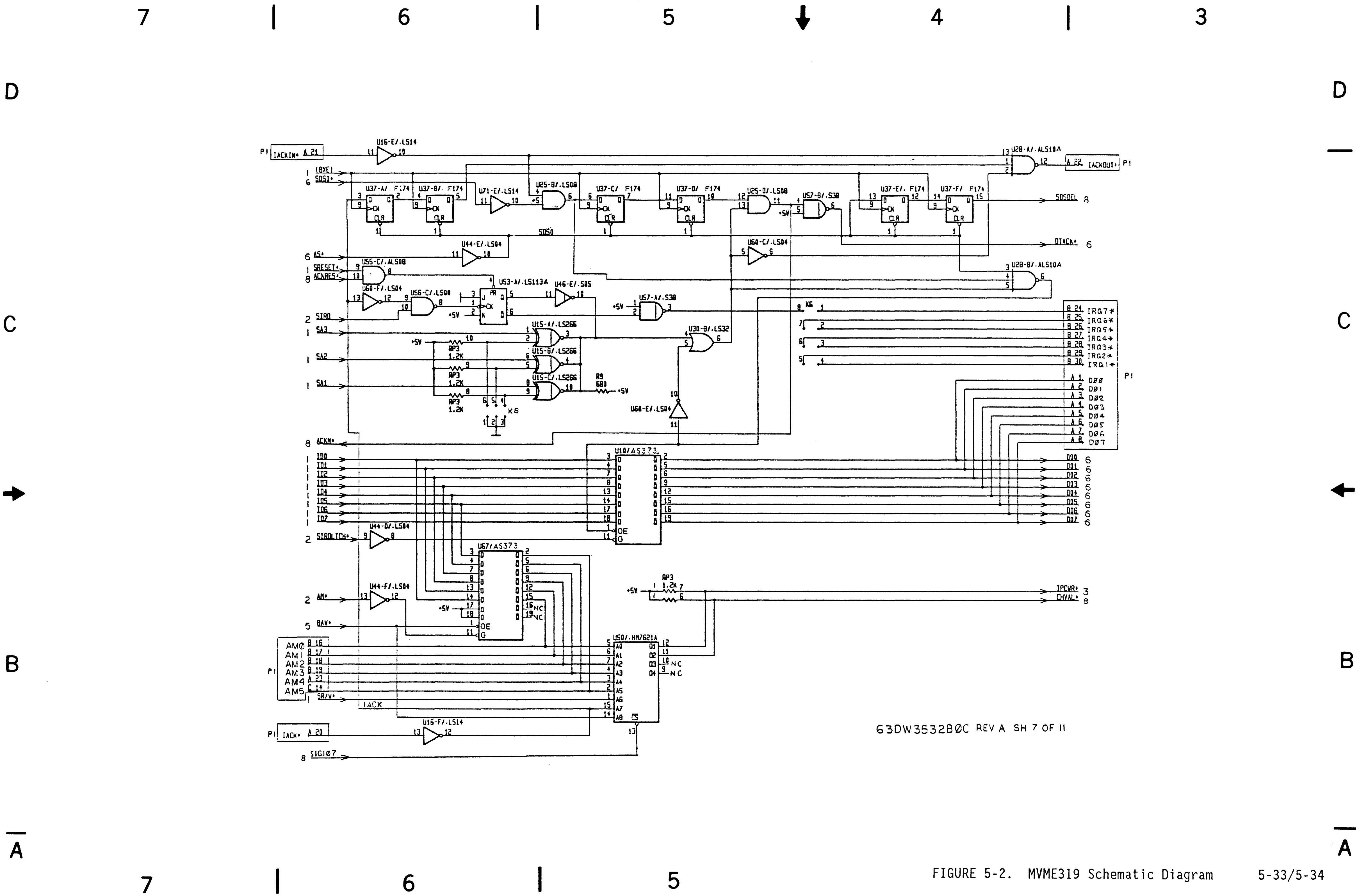


FIGURE 5-2. MVME319 Schematic Diagram

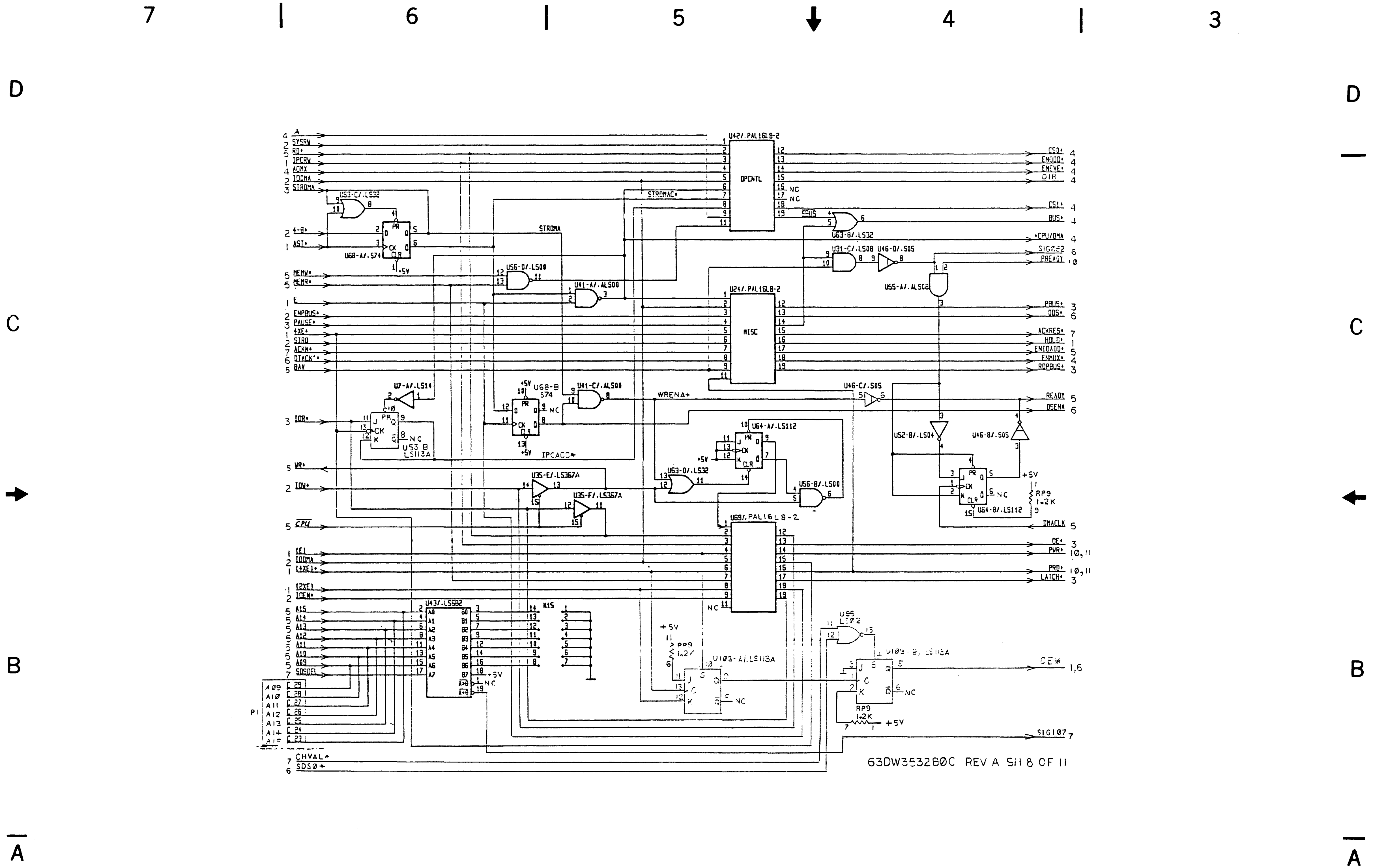


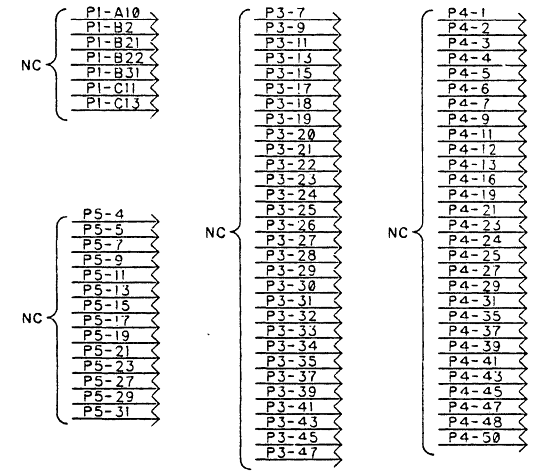
FIGURE 5-2. MVME319 Schematic Diagram

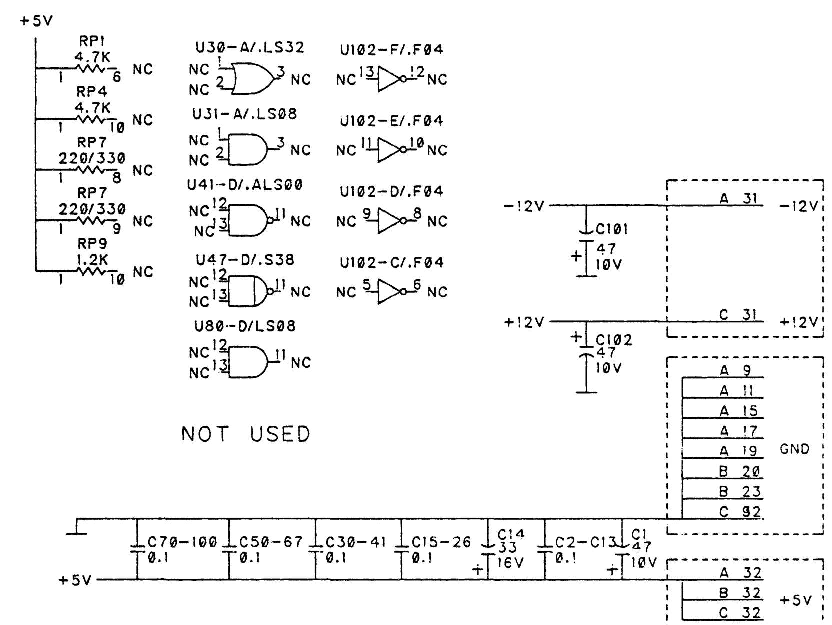
REF DES	TYPE	GND	+5V	
U1	74LS191	8	16	
U2	74LS191	8	16	
U3	74LS133	8	16	
U4	74LS257	8	16	
U5	74LS257	8	16	
U6	27128A-2	14	28	
U7	74LS14	7	14	
U8	AM9519A	14	28	
U9	MC14521B	8	16	
U10	MC34828	10	20	
U11	74LS645-1	10	20	
U12	74LS191	8	16	
U13	74LS257	8	16	
U15	74LS266	7	14	
U16	74LS14	7	14	
U17	74LS174	8	16	
U18	74LS245	10	20	
U19	74LS191	8	16	
U20	74S74	7	14	
U21	74LS245	10	20	
U22	TBP18S030	8	16	
U23	74LS373	10	20	
U24	PAL16L8-2	10	20	
U25	74LS08	7	14	
U26	8237A-5	20	31	
U27	MC68121	1	12	
U28	74ALS10A	7	14	
U29	74LS373	10	20	
U30	74LS32	7	14	
U31	74LS08	7	14	
U32	MC34828	10	20	
U33	74LS244	10	20	
U34	TBP18S030	8	16	
U35	74LS367A	8	16	
U36	74LS244	10	20	
U37	74F174	8	16	
U38	74LS373	10	20	
U39	74LS645-1	10	20	
U40	74LS138	8	16	
U41	74ALS00	7	14	
U42	PAL16L8-2	10	20	
U43	74LS682	10	20	
U44	74LS04	7	14	

Y2	
U103	U14,78,79
TP2	
T2	
R41	R4,12,21,24,25, 27-29,34,36-38
RP9	RP8
Q1	
P5	
K16	K4,5,9-14
DS1	
C102	
HIGHEST NUMBER USED	NOT USED
REFERENCE DESIGNATIONS	

REF DES	TYPE	GND	+5V	
U45	74LS257	8	16	
U46	74S05	7	14	
U47	74S38	7	14	
U48	74F241	10	20	
U49	74LS245	10	20	
U50	HM7621A	8	16	
U51	74LS645-1	10	20	
U52	74LS04	7	14	
U53	74LS113A	7	14	
U54	43256C-10	14	28	
U55	74ALS08	7	14	
U56	74LS00	7	14	
U57	74S38	7	14	
U58	MC34828	10	20	
U59	74S241	10	20	
U60	74LS04	7	14	
U61	74S113	7	14	
U62	43256C-10	14	28	
U63	74LS32	7	14	
U64	74LS112	8	16	
U65	74S113	7	14	
U66	74LS245	10	20	
U67	74AS373	10	20	
U68	74S74	7	14	
U69	PAL16L8-2	10	20	
U70	74LS373	10	20	
U71	74LS14	7	14	
U72	74LS74A	7	14	
U73	74LS257	8	16	
U74	74LS74A	7	14	
U75	74LS27	7	14	
U76	74LS02	7	14	
U77	74LS04	7	14	
U80	74LS08	7	14	
U81	TBP18S030	8	16	
U82	74LS02	7	14	
U83	FD1793	20	21	
U84	74LS32	7	14	
U85	74AS533	10	20	
U86	74LS86	7	14	
U87	74S240	10	20	
U88	WD1691	10	20	
U89	WD2143-03	9	18	
U90	74LS123	8	16	
U91	74AS533	10	20	
U92	74LS628	7	9	
U93	74LS257	8	16	
U94	74LS123	8	16	
U95	74LS534	10	20	
U96	74LS02	7	14	
U97	74LS74A	7	14	
U98	74LS374	10	20	
U100	74LS74A	7	14	
U101	74AS533	10	20	
U102	74F04	7	14	
U103	74S113A	7	14	
Y1	9.8304MHZ	7	14	
Y2	4.0MHZ	7	14	

REF DES	TYPE	GND	+5V	
R1	2.7K			
R2	10K			
R3	2.7K			
R5	2.7K			
R6	4.7K			
R7	390			
R8	560			
R9	680			
R10	680			
R11	1.2K			
R13	220			
R14	1.2K			
R15	4.7K			
R16	4.7K			
R17	274 1%			
R18	4.7K			
R19	4.7K			
R20	82K			
R22	47K			
R23	33			
R26	604 1%			
R30	20.0K 1%			
R31	10K 1%			
R32	47.5K 1%			
R33	4.99K 1%			
R35	560			
R39	10K			
R40	10K			
R41	10K			
RP1	4.7K			
RP2	1.2K			
RP3	1.2K			
RP4	4.7K			
RP5	220/330			
RP6	220/330			
RP7	220/330			
RP9	1.2K			
C1	47/10V			
C2-13	0.1			
C14	33/16V			
C15-26	0.1			
C27	330PF			
C28,C45	.01			
C29	100PF			
C30-41	0.1			
C42	.33			
C43	47PF			
C44,C46	10/50V			
C47	27PF			
C48	22/25V			
C49	100PF			
C50-67	0.1			
C68	1			
C69-100	0.1			
C101	47/10V			
C102	47/10V			
Q1	2N5086			





63DW3532B0C REV A SH 9 OF 11

FIGURE 5-2. MVME319 Schematic Diagram

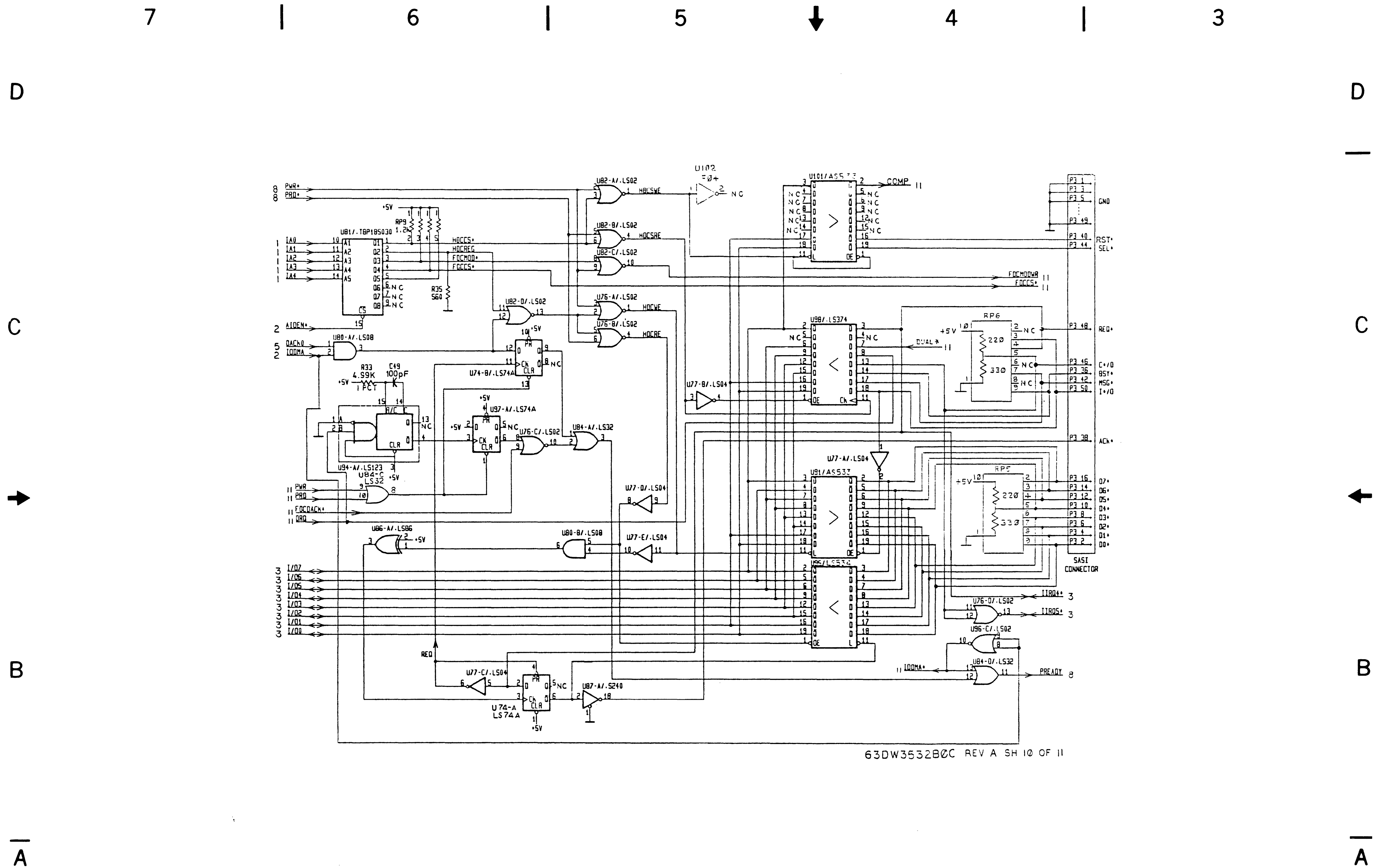
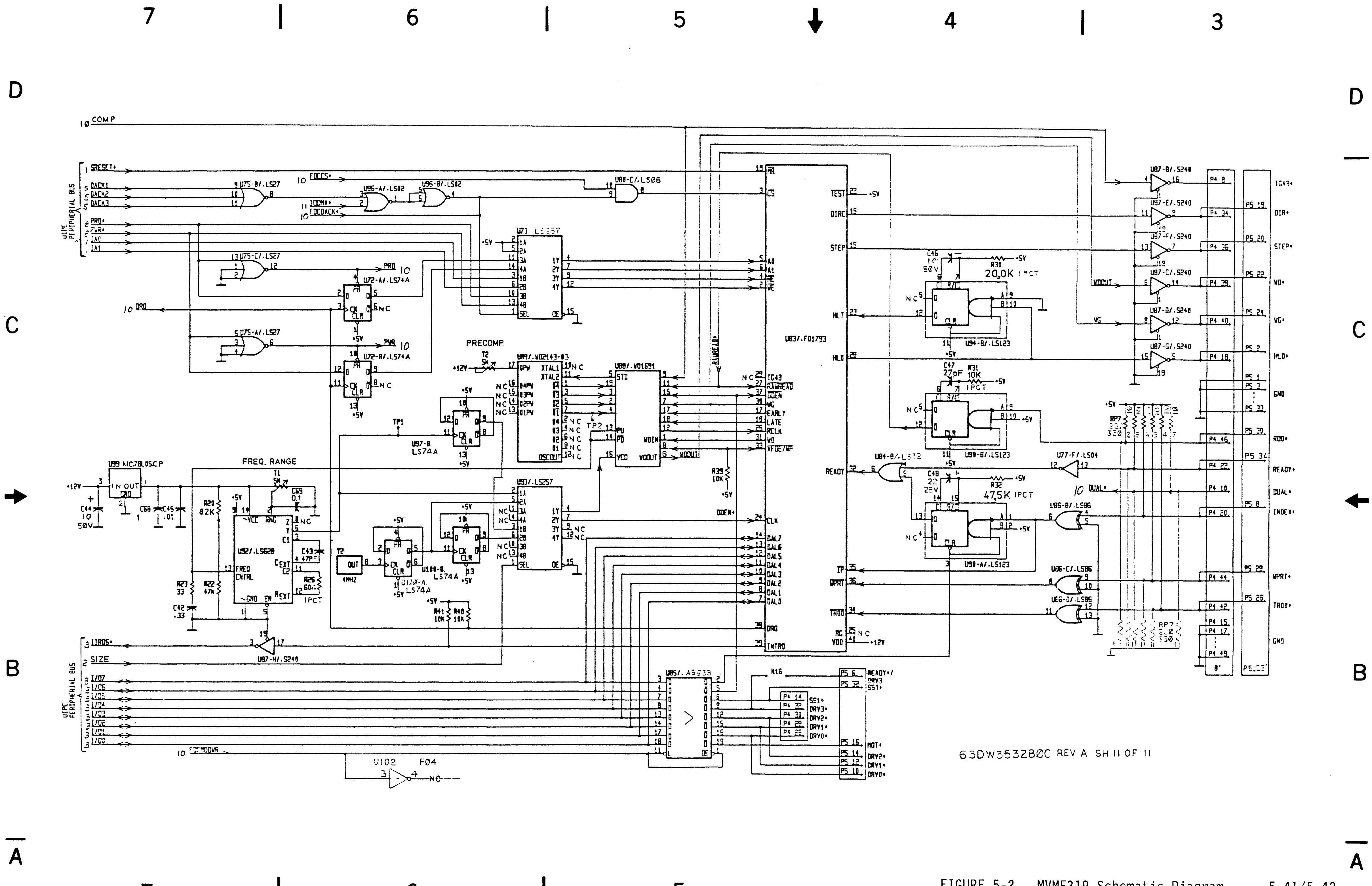


FIGURE 5-2. MVME319 Schematic Diagram



63DW3532B0C REV A SH 11 OF 11
U83/.FD1793
U88/.WD1691
U89/.WD2143-03
U73 LS257
U93/.LS257
U92/.LS628
U99 MC78L05CP
U85/.AS533
FREQ. RANGE
PRECOMP
UIPC PERIPHERIAL BUS

INDEX

In this index a page number indicates only where reference to a topic begins; the reference may extend to the following page or pages.

D

E

F

INDEX

INDEX

U

V

W

SUGGESTION/PROBLEM REPORT

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

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

Product: ______________________ Manual: ______________________

COMMENTS: ____

____

____

____

____

____

____

____

____

____

____

____

(For additional comments use other side)

Please Print

Name ______________________ Title ______________________

Company ______________________ Division ______________________

Street ______________________ Mail Drop ______________________

City ______________________ Phone ______________________

State ____________ Zip ________ Country ______________________

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

Motorola Field Service Division/Customer Support
(800) 528-1908
(602) 438-3100

MOTOROLA

COMMENTS:

MOTOROLA INC.

Microcomputer Division
2900 South Diablo Way
Tempe, Arizona 85282
P.O. Box 2953
Phoenix, Arizona 85062

10361 PRINTED IN USA (10/88) MPS 2M

68NW9209E61B