MOTOROLA

MVME319/D1

MVME319
Intelligent Disk/Tape Controller
User's Manual

MICROSYSTEMS

QUALITY • PEOPLE • PERFORMANCE

M V M E 3 1 9

I N T E L L I G E N T D I S K / T A P E C O N T R O L L E R

U S E R ' S M A N U A L

First Edition, April 1985

TABLE OF CONTENTS Page

TABLE OF CONTENTS

Chapter 4 OPERATING INSTRUCTIONS (continued)

TABLE OF CONTENTS Page

LIST OF TABLES Page

LIST OF FIGURES Page

CHAPTER 1

GENERAL INFORMATION

1.1 INTRODUCTION

The MVME319 Intelligent Disk/Tape Controller (IDTC) is a VMEmodule 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 on-board 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 UNIX SYSTEM V/68.

This manual provides general information, functional description, installation instructions, operating instructions and maintenance information for the MVME319 Intelligent Disk/Tape Controller, which is shown in Figure 1.1.

1.2 SPECIFICATIONS

The specifications of the MVME319 Intelligent Disk/Tape Controller (IDTC) are given in Table 1.1.

1.3 REFERENCE MANUALS

The following manuals may be used for further information about the VMEbus, the command channel software interface, the SASI/SCSI bus, the XEBEC and ADAPTEC hard disk controllers and the Cipher FloppyTape drive:

- MVMEBS — VMEbus Specification Manual
- M68KIPCS — M68000/IPC Command Channel Software Interface Ref. Manual
- Shugart Ass. — SASI Bus Specification
- XEBEC Inc. — XEBEC S1410 5.25 Inch Winchester Disk Controller Manual
- XEBEC Inc. — XEBEC S1410A 5.25 Inch Winchester Disk Controller Manual
- ADAPTEC Inc. — ACB 4000 5 1/4" Winchester Disk Controller OEM Manual
- CIPHER Inc. — Series 525 FloppyTape Cartridge Tape Drive Product Description

Figure 1.1: The MVME319 Intelligent Disk/Tape Controller

Table 1.1: MVME319 Specifications

CHARACTERISTIC	SPECIFICATION
General	The MVME319 Intelligent Disk/Tape Controller is a VMEmodule for interfacing hard and floppy disk drives and a FloppyTape drive with a VME system.
Local Microcomputer	The local microcomputer consists of an MC68121 Intelligent Peripheral Controller with built-in 8-bit microprocessor, 16-bit timer, serial port and dual ported RAM. This microcontroller is supported by 16K ROM, 32K RAM, an interrupt controller and a DMA controller.
VMEbus Interface	The IDTC command channel is accessed from the VMEbus through an option A16:D8 slave interface. The IDTC accesses system memory through an option A24:D16 master interface and supervises data transfers with a 64 microseconds time-out function. The option RWD bus requester is selectable to operate on any one of the four priority levels. The interrupter is selectable to operate on any one of the seven priority levels.
SASI/SCSI Host Adapter	The SASI/SCSI disk controller interface is capable of controlling up to eight hard disk controllers. Each controller may be connected with one or two hard disk drives. For each hard disk drive, the operating system may specify a unique set of parameters.
Supported Hard Disks and Controllers	The IDTC supports the XEBEC S1410/S1410A and Adaptec ACB 4000 Winchester Disk Controllers and a wide range of 5.25 inch hard disk drives.
Floppy Disk/Tape Controller	The floppy disk/tape controller is capable of controlling up to four floppy disk drives or one FloppyTape drive plus two floppy disk drives.
Supported Floppy Disk Drives	For each floppy disk drive the user may independently select media size (5.25 or 8 inch), number of media sides (1 or 2), data density (FM or MFM), sector size (128, 256, 512 or 1024 bytes), track format (IBM or Motorola) and several other parameters (number of sectors per track, number of cylinders, write pre-compensation or read post-compensation, interleave factor, spiral offset, stepping rate).
Supported Tape Drive	One Cipher 525 FloppyTape using 1/4" tape cartridges with a maximum formatted capacity of 25MB. The drive is controlled via the 8" floppy interface of the MVME319.

Table 1.1: MVME319 Specifications (continued)

CHARACTERISTIC	SPECIFICATION
Operating System Interface	A dual ported RAM provides the path for commands and status messages to be transferred between the operating system and the IDTC. This command channel appears to the operating system as 256 memory mapped byte locations in the short supervisory I/O address segment. The command and status message transmissions between the operating system and the IDTC follow a specified command channel protocol.
IDTC Commands	The IDTC commands include: - IDTC configuration and diagnostics - tape, hard disk and floppy disk parameter initialization - multiple sector read and write - multiple sector verification - track and disk formatting - seek and restore operations - disk and drive diagnostics
IDTC Status Messages	The IDTC status messages include: - operation complete - IDTC self-test status - disk and tape drive status - data and sector ID errors - seek and restore errors - invalid disk and tape accesses - invalid commands - disk and tape drive malfunctions - disk controller malfunctions
Mechan. Dimensions	Double height VME board with front panel Board size: 233 mm * 160 mm Front panel size: 262 mm * 20 mm
Connectors	One 96 pole DIN 41612 connector for VMEbus, one 50 pole connector for SASI bus, one 50 pole connector for 8" floppy disk drives, one 34 pole connector for 5" floppy disk drives.
Power Requirements	+ 5 V DC (+/- 5%), 3.7 A (typ), 4.6 A (max) +12 V DC (+/- 5%), 30 mA (typ), 50 mA (max) -12 V DC (+/- 5%), 20 mA (typ), 35 mA (max)
Environmental Conditions	Operating temperature: 0 to 55 deg C Storage temperature: -40 to 100 deg C Operating humidity: 0% to 90% non condensing

1.4 MANUAL TERMINOLOGY

1.4.1 Address and Data Formats

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

%	percent	specifies a binary number
&	ampersand	specifies a decimal number
$	dollar	specifies a hexadecimal number
	none	specifies a decimal number (default)

1.4.2 Electrical Signal Levels

A signal line is always assumed to be in one of two levels, or in transition between these levels. Whenever the term "high" is used, it refers to a high TTL voltage level (> +2.0 V). The term "low" refers to a low TTL voltage level (< +0.8 V).

There are two possible transitions which can appear on a signal line, and these will be referred to as "edges". A "rising edge" is defined as the time period during which a signal line makes its transition from a low level to a high level. The "falling edge" is defined as the time period during which a signal line makes its transition from a high level to a low level.

A signal is defined as "active low" if the function associated with the signal line is valid or initiated by either a low level or a falling edge on the signal line. The mnemonics of active low signals are marked with the suffix "*".

A signal is defined as "active high" if the function associated with the signal line is valid or initiated by either a high level or a rising edge on the signal line.

1.4.3 Logic Signal States

The terms "assert" and "negate" describe the logic state of a signal without indicating the associated voltage level. An active low signal is asserted when its voltage level is low, it is negated when its voltage level is high. An active high signal is asserted when its voltage level is high, it is negated when its voltage level is low.

For signals which are driven by three-state or open-collector outputs, the term "release" describes the high impedance state of the corresponding driver. Typically these signal lines are driven to a high voltage level by pull-up resistors when all drivers on the line are turned off.

CHAPTER 2

FUNCTIONAL DESCRIPTION

2.1 INTRODUCTION

The MVME319 Intelligent Disk/Tape Controller (IDTC) 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 on-board 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 17K bytes. 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's program execution.

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

The following paragraphs provide a functional description of the IDTC hardware and firmware on a block level and detailed specifications of the various interfaces (VMEbus, SASI/SCSI bus, floppy disk/tape interface).

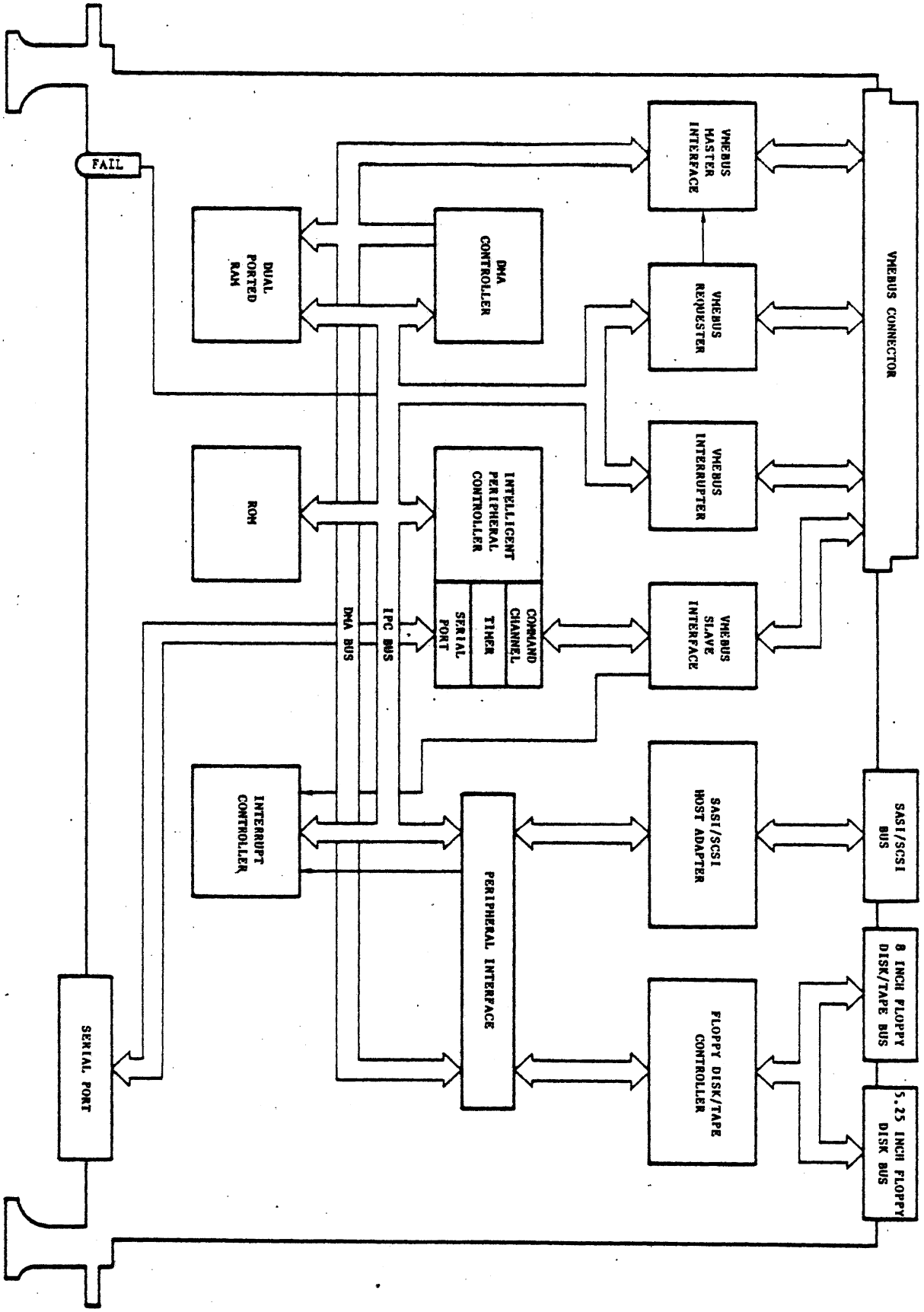
2.2 HARDWARE OVERVIEW

For the following description, the IDTC module is regarded as consisting of functional blocks, as shown in Figure 2.1. 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 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 RS232C communication interface for Motorola field service diagnostics (i.e. it is not intended for user applications). The dual ported RAM serves as the command channel between the IPC and the VMEbus.

The IDTC firmware is resident in two 8K-byte EPROMs. The program is described in Paragraph 2.3.

Figure 2.1: MVME319 Block Diagram



Four 8K-byte static RAMs are available for IDTC program data and for intermediate storage of up to 17K bytes 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 controller performs all data transfers between 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 Rev.B. 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 an option RWD (release when done) VMEbus requester which asserts a bus request upon the IDTC processor's demand on a jumper selectable priority level. Once being 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 Paragraph 2.4.

The AM9519A Universal Interrupt Controller provides for fast interrupt response times by surrendering 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 Paragraph 2.5 and the floppy disk/tape controller in Paragraph 2.6.

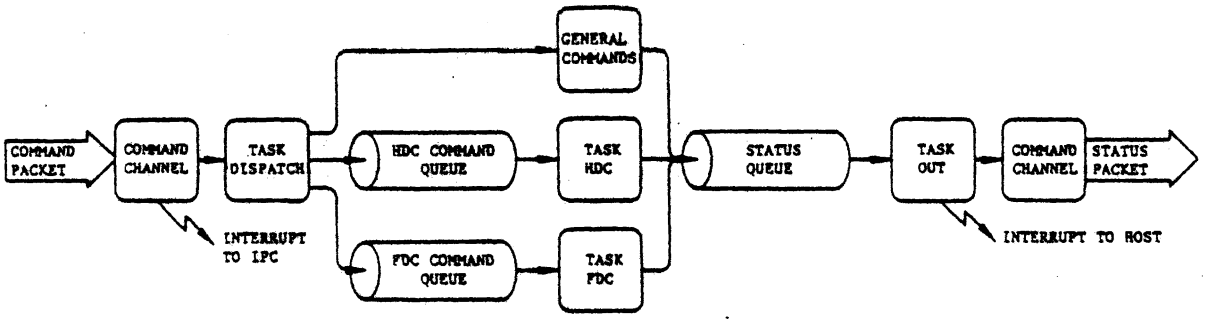
2.3 FIRMWARE OVERVIEW

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

- receive commands from the VMEbus through the command channel;
- analyze and validate the commands;
- transform the commands into micro operation sequences;
- supervise hard disk controller, floppy disk/tape controller, DMA controller, VMEbus requester and VMEbus interrupter operations;
- place status messages into the command channel.

The firmware consists of a general purpose multi-tasking executive and several specific tasks for disk/tape controller functions. Figure 2.2 shows the block structure of the IDTC firmware.

Figure 2.2: MVME319 Firmware Structure



Each time a command is written from the VMEbus into the command channel, the IDTC processor will be 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 will be 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 which 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 will be aborted and an error message will be 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.

2.4 VME SYSTEM INTERFACE

This section supplies detailed specifications of all VME system operations supported by the IDTC. Includes descriptions of VMEbus signals, system memory access, VMEbus requester and interrupter, reset and self-test.

2.4.1 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 Rev.B.

Table 2.1 identifies all VMEbus signals used by the IDTC on mnemonics and functional descriptions. The locations of the VMEbus signals at connector P1 are shown in Table 2.2.

Table 2.1: VMEbus Signal Description

SIGNAL	SIGNAL DESCRIPTION
A01..A23	ADDRESS BUS 23 address lines. In master mode, A01..A31 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.
AM0..AM5	ADDRESS MODIFIERS Six lines providing additional address information. 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.
D00..D15	DATA BUS 16 data lines. In master mode, D00..D15 are used for data transfer to and from system memory. 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.
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.

Table 2.1: VMEbus Signal Description (continued)

AS*	ADDRESS STROBE An active low bidirectional signal. The falling edge 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.
DS0*	DATA STROBE 0 An active low bidirectional signal that indicates a data transfer on the data lines D00..D07. In master mode, DS0* is a three-state output; in slave mode, it is an input.
DS1*	DATA STROBE 1 An active low bidirectional signal that indicates a data transfer on the data lines D08...D15. In master mode, DS1* is a three-state output; in slave mode, it is an input.
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.
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.
BG0IN*...BG3IN* BG0OUT*..BG3OUT*	BUS GRANT INPUTS LEVEL 0-3 BUS GRANT OUTPUTS LEVEL 0-3 Four active low signal pairs that form bus grant daisy-chains 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.

Table 2.1: VMEbus Signal Description (continued)

BBSY*	BUS BUSY An active low open-collector output signal indicating that the module is the current bus master.
IRQ1*..IRQ7*	INTERRUPT REQUEST LEVEL 1-7 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.
IACK*	INTERRUPT ACKNOWLEDGE An active low input signal that indicates an interrupt status/ID byte fetch on the data transfer bus.
IACKIN* IACKOUT*	INTERRUPT ACKNOWLEDGE INPUT INTERRUPT ACKNOWLEDGE OUTPUT An active low signal pair that forms an interrupt acknowledge 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. 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.
SYSFAIL*	SYSTEM FAILURE An optional active low open-collector output signal indicating that a severe malfunction has occurred, and that no further commands can be executed.
SYSRESET*	SYSTEM RESET An active low input signal that causes a complete hardware and firmware reset of the module.
GND	GROUND
+5V	+ 5 VOLTS POWER
+12V	+ 12 VOLTS POWER
-12V	- 12 VOLTS POWER

Table 2.2: Connector P1 Signal Locations

PIN NO.	ROW A SIGNALS	ROW B SIGNALS	ROW C SIGNALS	PIN NO.
1	D00	BBSY*	D08	1
2	D01	(BCLR*)	D09	2
3	D02	(ACFAIL*)	D10	3
4	D03	BG0IN*	D11	4
5	D04	BG0OUT*	D12	5
6	D05	BG1IN*	D13	6
7	D06	BG1OUT*	D14	7
8	D07	BG2IN*	D15	8
9	GND	BG2OUT*	GND	9
10	(SYSCLK)	BG3IN*	SYSFAIL*	10
11	GND	BG3OUT*	(BERR*)	11
12	DS1*	BR0*	SYSRESET*	12
13	DS0*	BR1*	(LWORD*)	13
14	WRITE*	BR2*	AM5	14
15	GND	BR3*	A23	15
16	DTACK*	AM0	A22	16
17	GND	AM1	A21	17
18	AS*	AM2	A20	18
19	GND	AM3	A19	19
20	IACK*	GND	A18	20
21	IACKIN*	(SERCLK)	A17	21
22	IACKOUT*	(SERDAT)	A16	22
23	AM4	GND	A15	23
24	A07	IRQ7*	A14	24
25	A06	IRQ6*	A13	25
26	A05	IRQ5*	A12	26
27	A04	IRQ4*	A11	27
28	A03	IRQ3*	A10	28
29	A02	IRQ2*	A09	29
30	A01	IRQ1*	A08	30
31	-12V	(5V STBY)	+12V	31
32	+5V	+5V	+5V	32

Note: Signals in parentheses are not used by the IDTC.

2.4.2 System Memory Access

Peripheral data transfers between the disk/tape drives and 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 16M byte address map and all address modifier codes.

When the IDTC receives a disk/tape read or write command, it will load 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. (The bus arbitration will be described in Paragraph 2.4.3)

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 17K bytes of peripheral data can be intermediately 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 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 will abort the DMA operation and return an error message in the command channel. The occurence 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.

2.4.3 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 the jumper areas K1, K2 and K3, as described in Paragraph 3.4.1.

The flow diagram in Figure 2.3 illustrates the operation sequence of the VMEbus requester.

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 is not ackowledged within 1 second, the execution of the command which has initiated the system DMA request will be aborted, and an error message will be returned in the command channel.

The VMEbus requester however 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.

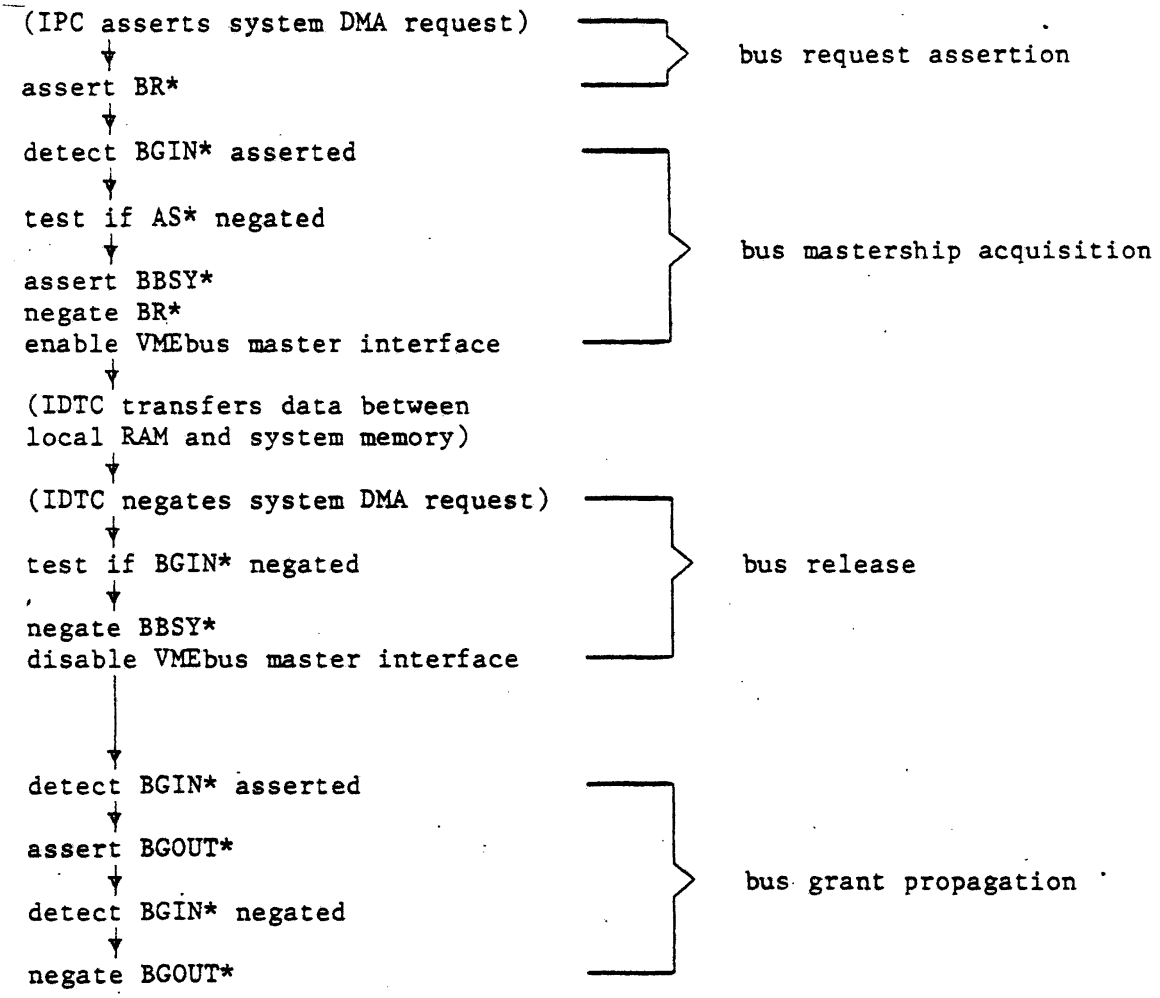
When the VMEbus requester has a bus request pending and it receives a bus grant on the same priority level, it acquires bus mastership. After the previous bus master has finished its last VMEbus cycle and negated AS*, the VMEbus requester acknowledges the bus grant by asserting BBSY* and negating BR*, 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 micro-seconds. As the VMEbus requester on the MVME319 module is of the release-when-done (RWD) type, the system DMA operation will neither be aborted by bus requests of other modules nor by assertion 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 negates 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 will be propagated to the next bus requester. The VMEbus requester keeps BGOUT* asserted as long as BGIN* is low.

Figure 2.3: VMEbus Arbitration Flow Diagram



2.4.4 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 anyone of the seven interrupt levels. This is done by setting the appropriate jumpers on the jumper areas K6 and K8, as described in Paragraph 3.4.2.

The interrupt status/ID byte which will be 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 (see Paragraph 4.3.1.1).

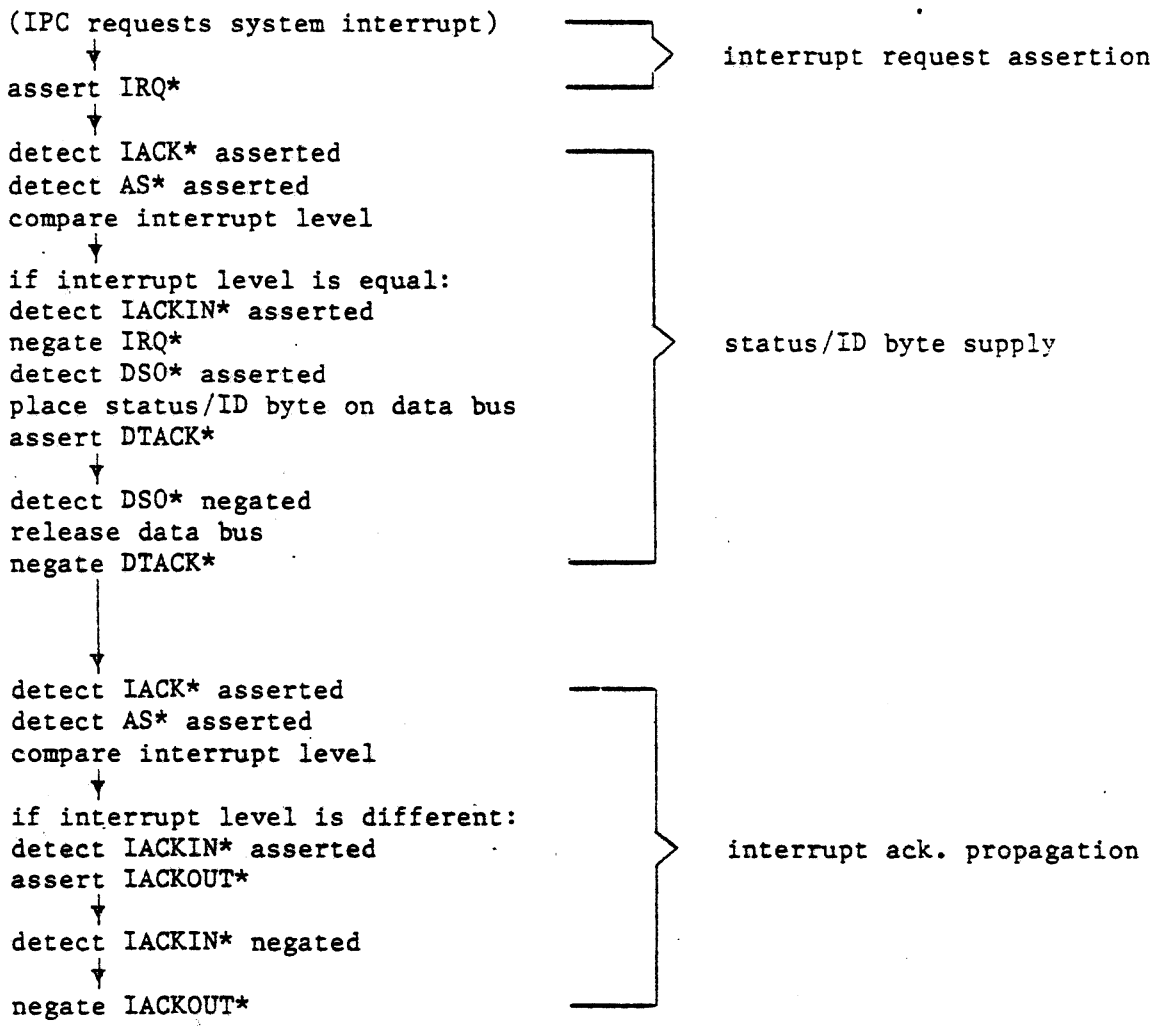
The flow diagram in Figure 2.4 illustrates the operation sequence of the VMEbus interrupter.

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 to its own will be propagated to the next interrupter. The VMEbus interrupter keeps IACKOUT* asserted as long as IACKIN* is low.

Figure 2.4: VMEbus Interrupt Flow Diagram



2.4.5 Reset, Self-test, and Failure

The MVME319 module 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 illuminates the FAIL LED on the front panel. The VMEbus SYSFAIL* signal is connected with the local failure signal via jumper K7, as described in Paragraph 3.4.4.

A system failure will also be 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 will 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 will be destroyed. Therefore the operating system must not transmit any commands to the IDTC within 10 seconds after a system reset.

The successful completion of 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 will be cleared.

2.5 SASI/SCSI BUS HOST ADAPTER

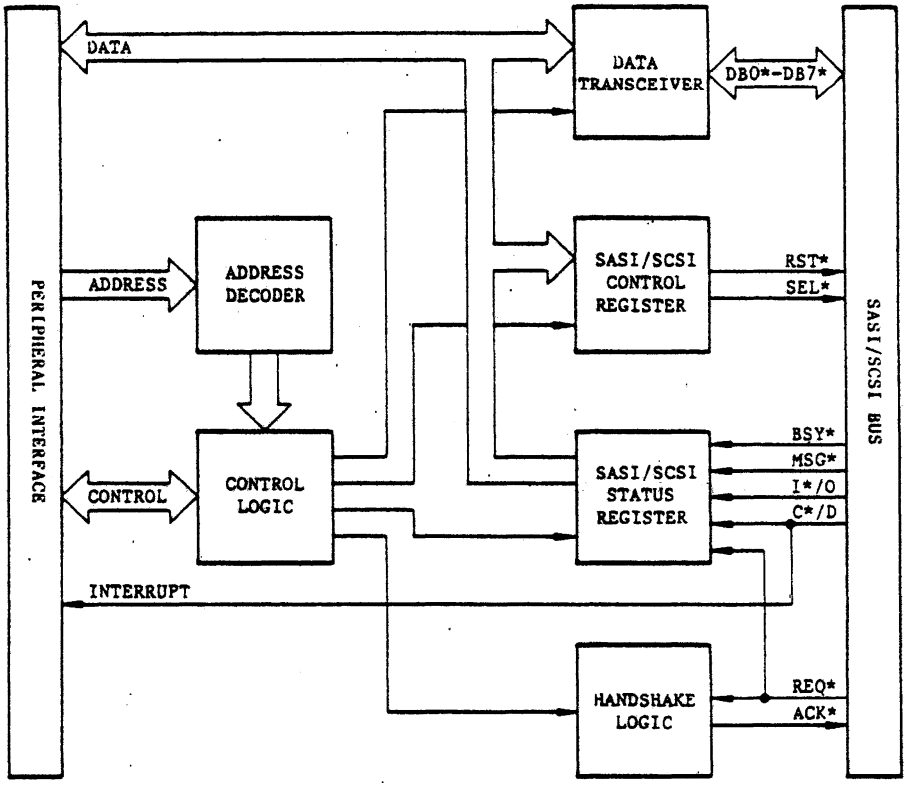
The IDTC supports the SASI (Shugart Associates System Interface) bus and SCSI (Small Computer System Interface) bus. Both provide a physical and logical standard for controlling peripheral devices, especially mass storage units. The host adapter implemented on the IDTC supports the command and status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

2.5.1 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 2.5.

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 2.5: SASI/SCSI Host Adapter Block Diagram



2.5.2 SASI/SCSI Bus Signals

All .SASI/SCSI bus signals are available at connector P3. Table 2.4 identifies all these signals by mnemonics and functional descriptions. The locations of the SASI/SCSI bus signals at connector P3 are shown in Table 2.3.

The SASI/SCSI bus host adapter provides TTL compatible high current drivers which are capable of sinking at least 48mA at 0.5V and sourcing at least 3mA at 2.4V. All input signals are terminated with a 220/330 ohms resistor network on the MVME319 module. All output signals have to be terminated on the last hard disk controller on the SASI/SCSI bus to ensure proper operation.

Table 2.3: Connector P3 Signal Locations

PIN	SIGNALS		PIN
1	GND	DB0*	2
3	GND	DB1*	4
5	GND	DB2*	6
7	GND	DB3*	8
9	GND	DB4*	10
11	GND	DB5*	12
13	GND	DB6*	14
15	GND	DB7*	16
17	GND	(DBP*)	18
19	GND	—	20
21	GND	—	22
23	GND	—	24
25	GND	—	26
27	GND	—	28
29	GND	—	30
31	GND	(ATN*)	32
33	GND	(SPARE)	34
35	GND	BSY*	36
37	GND	ACK*	38
39	GND	RST*	40
41	GND	MSG*	42
43	GND	SEL*	44
45	GND	C*/D	46
47	GND	REQ*	48
49	GND	I*/O	50

Note: Signals in parentheses are not used by the IDTC.

Table 2.4: SASI/SCSI Bus Signal Description

SIGNAL	SIGNAL DESCRIPTION
DB0..DB7	DATA BUS 8 bit bidirectional data bus for transferring data between the IDTC and the hard disk controllers.
SEL*	SELECT An active low output indicating that a valid controller ID is placed on the data bus.
BSY*	BUSY An active low input indicating that the selected hard disk controller is ready for SASI/SCSI bus operations.
I*/O	INPUT/OUTPUT This input 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.
C*/D	CONTROL/DATA This input specifies the information on the data bus. A low level indicates command and status bytes, a high level indicates data bytes.
MSG*	MESSAGE An active low input indicating that the hard disk controller has completed the previous command.
REQ*	REQUEST An active low input that initiates 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.
ACK*	ACKNOWLEDGE An active low output 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.
RST*	RESET An active low output that causes all hard disk controllers on the bus to be reset and to enter an idle state.
GND	GROUND

2.5.3 SASI/SCSI Bus Sequences

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

2.5.3.1 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 #	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.

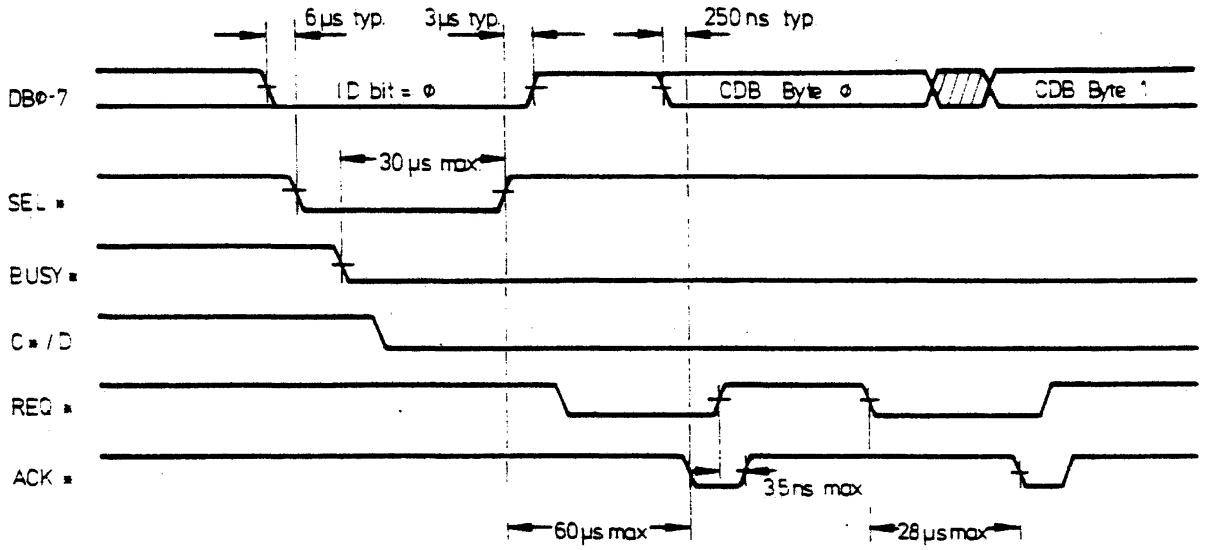
After being selected, the hard disk controller requests the command transfer from the IDTC by driving C*/D low and asserting REQ*. The IDTC acknowledges this request by presenting the first byte of the command descriptor block on the data bus and asserting ACK*. When receiving the ACK* signal, the controller 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 assertion of REQ* to present the second command byte. The above described handshake sequence will continue 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 Paragraph 2.5.4.

Figure 2.6 shows the timing diagram of the controller selection and command transfer sequence.

Figure 2.6: Controller Selection and Command Transfer Timing



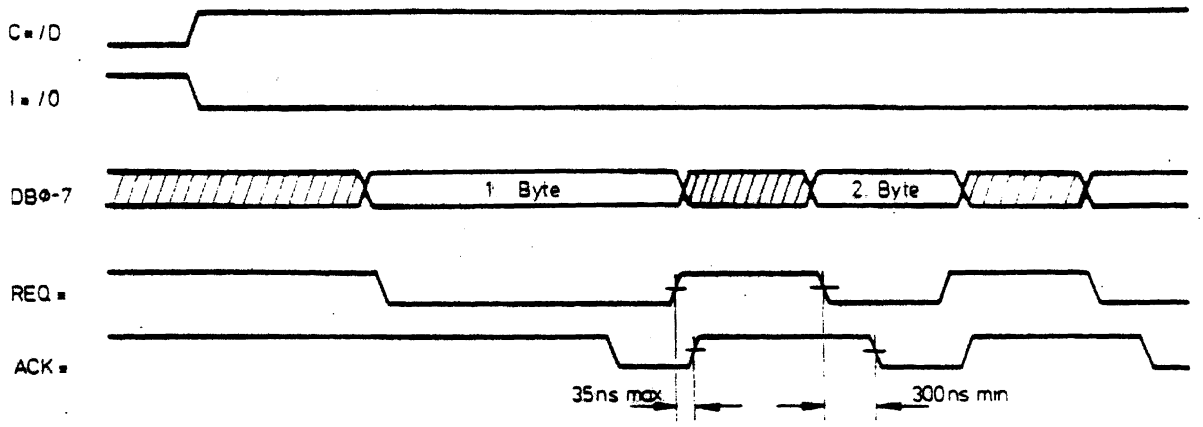
2.5.3.2 Data Transfer

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 on the I*/O signal.

If data has 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 assertion of REQ* to read the second data byte. The described handshake sequence will continue until the complete data block has been transferred from the controller to the IDTC byte by byte.

Figure 2.7 shows the timing diagram of the read data transfer sequence.

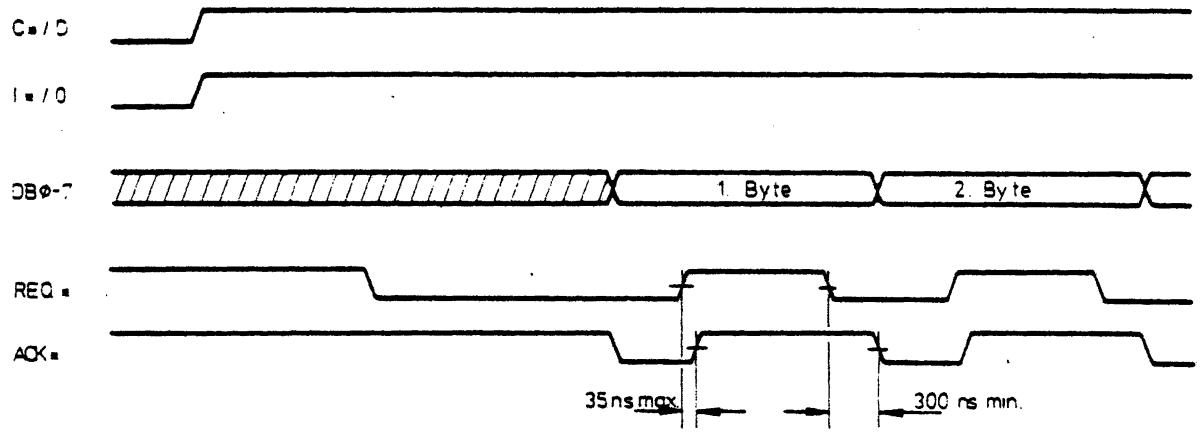
Figure 2.7: Read Data Transfer Timing



If data has 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 on the data bus 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 assertion of REQ* to write the second data byte. The described handshake sequence will continue until the complete data block has been transferred from the IDTC to the controller byte by byte.

Figure 2.8 shows the timing diagram of the write data transfer sequence.

Figure 2.8: Write Data Transfer Timing



2.5.3.3 Status and Completion Message Transfer

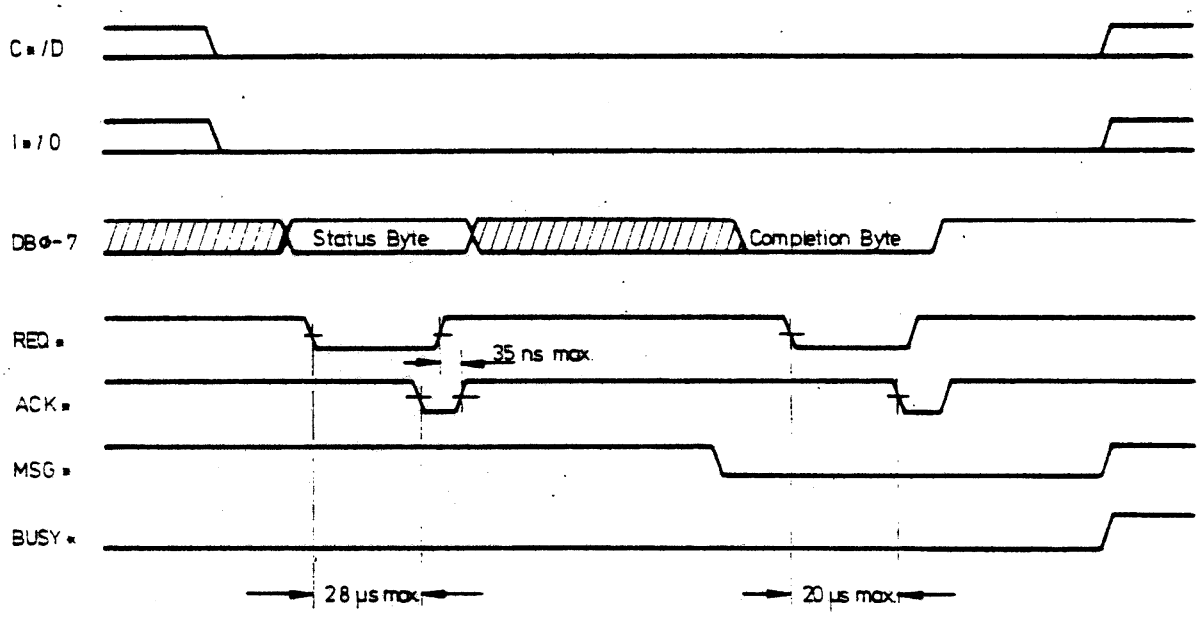
Following the execution of a command, the hard disk controller initiates a status and completion message transfer by driving the signals C*/D and I*/O low. Then the controller presents the status byte on the data bus and asserts REQ*. The IDTC acknowledges this request by reading the status byte and 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 Paragraph 2.5.5.

Figure 2.9 shows the timing diagram of the status and completion message transfer sequence.

Figure 2.9: Status and Completion Message Timing



2.5.4 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 (diagnostic) commands. Both command classes have a length of 6 bytes and the format described below.

Table 2.5: 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.

Bits 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 on the disk 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 a 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.

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 controllers:

```
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 always $00.

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

Table 2.6: 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

2.5.5 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.

Table 2.7: 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 will be set if the hard disk controller detected an error during the command execution. Otherwise it will be cleared.

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

2.5.6 Controller Sense Blocks

If the hard disk controller flags an error in the status byte, the IDTC will transmit 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.

Table 2.8: 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 2	logical sector addr (mid)							
Byte 3	logical sector addr (low)							

DC. = don't care

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

Bit 6 of byte 0 is not used and will always be 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 occuring during disk accesses (read, write, format, seek), bytes 1, 2 and 3 specify the the logical sector address (LSA) 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.

Table 2.9: 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

2.6 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 (5.25 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 pre- or read post-compensation, interleave factor, spiral offset, stepping rate).

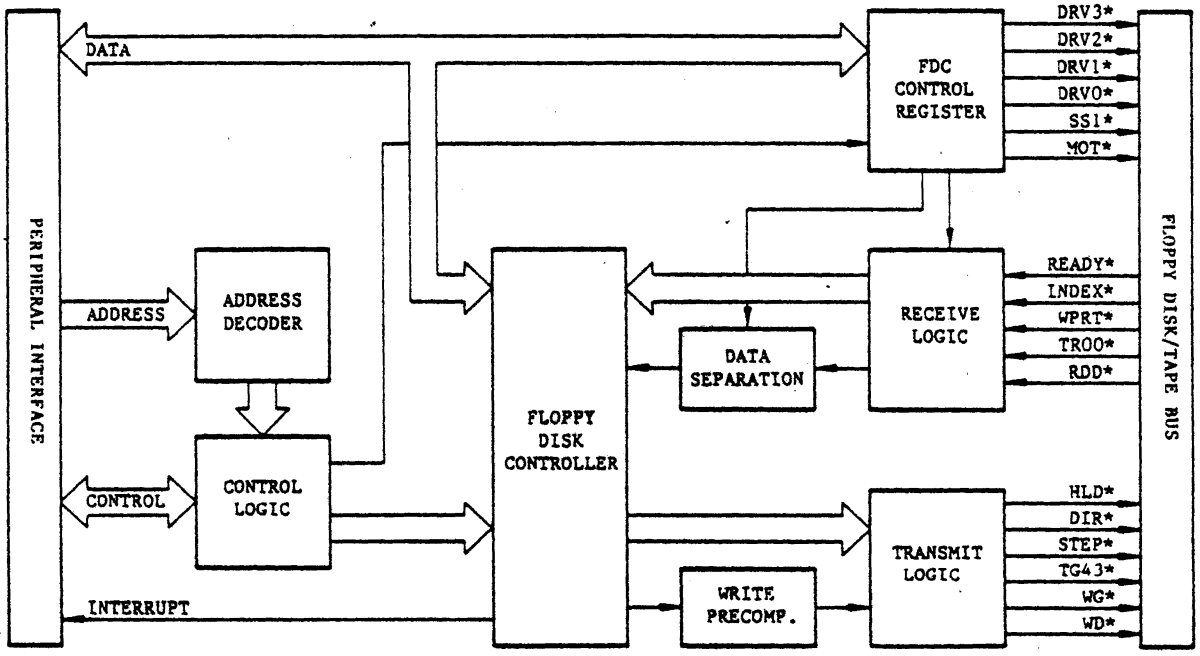
The FloppyTape recording technique is MFM, with a format similar to the IBM 34 floppy disk format. The data is organized into sectors which are contained in fixed length segments on 6 streams on the tape. Sector length, number of sectors per segment, and segment skip factor are user selectable.

2.6.1 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 2.10.

The disk/tape drive operations are controlled by the Western Digital FD1793 Floppy Disk Controller 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 with 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).

Figure 2.10: Floppy Disk/Tape Controller Block Diagram



2.6.2 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 5.25 inch drives are available at P5. Table 2.10 identifies all floppy disk/tape signals by mnemonics and functional descriptions. The locations of the 8 inch drive signals at connector P4 are shown in Table 2.11, the locations of the 5.25 inch drive signals at connector P5 are shown in Table 2.12.

The floppy disk/tape interface provides TTL compatible high current drivers which are capable of sinking at least 48mA at 0.5V and sourcing at least 3mA at 2.4V. All input signals are terminated with a 220/330 Ohms resistor network on the MVME319 module. All output signals have to be terminated on the last drive in the line to ensure proper operation.

Table 2.10: Floppy Disk/Tape Signal Description

SIGNAL	SIGNAL DESCRIPTION
DRV0* DRV1* DRV2* DRV3*	DRIVE SELECT 0-3 Four active low outputs 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. At P5, the DRV3* signal is optional and must be disabled if a drive is connected which uses pin 6 for the READY* signal.
MOT*	MOTOR ON This active low output turns the drive motor on. MOT* is only used by 5.25 inch floppy disk drives.
READY*	READY Floppy disk drives indicate at this active low input that the drive is ready for performing operations. Note that some 5.25 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.

Table 2.10: Floppy Disk/Tape Signal Description (continued)

SIGNAL	SIGNAL DESCRIPTION
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*.
DUAL*	DUAL SIDED DISK An active low input indicating that a dual sided disk is inserted in the drive. This signal is only supported by 8 inch disk drives.
WPRT*	WRITE PROTECT An active low input indicating that the floppy disk or tape cartridge in the drive is write protected.
HLD*	HEAD LOAD An active low output that may be used by floppy disk drives to load the read/write head on the disk. FloppyTape drives use HLD* to start the tape drive motor.
SS1*	SIDE SELECT For floppy disk drives, this output specifies which side of the disk will be accessed. A high level selects side 0, a low level selects side 1. FloppyTape drives use SS1* together with DRV0* and DRV1 as a stream select signal.
DIR*	DIRECTION For floppy disk drives, this output specifies the direction in which the 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 towards segment 0 and a low level towards segment 254.

Table 2.10: Floppy Disk/Tape Signal Description (continued)

SIGNAL	SIGNAL DESCRIPTION
STEP*	STEP For floppy disk drives, a falling edge on this output 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*.
TR00*	TRACK ZERO Floppy disk drives indicate at this active low input 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.
TG43*	TRACK GREATER 43 This active low output 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.
WG*	WRITE GATE This output 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.
WD*	WRITE DATA This output 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.
RDD*	READ DATA This input 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*.
GND	GROUND

Table 2.11: Connector P4 Signal Locations

PIN	SIGNALS		PIN
1	GND	—	2
3	GND	—	4
5	GND	—	6
7	GND	TG43*	8
9	GND	DUAL*	10
11	GND	—	12
13	GND	SS1*	14
15	GND	—	16
17	GND	HLD*	18
19	GND	INDEX*	20
21	GND	READY*	22
23	GND	—	24
25	GND	DRV0*	26
27	GND	DRV1*	28
29	GND	DRV2*	30
31	GND	DRV3*	32
33	GND	DIR*	34
35	GND	STEP*	36
37	GND	WD*	38
39	GND	WG*	40
41	GND	TR00*	42
43	GND	WPRT*	44
45	GND	RDD*	46
47	GND	—	48
49	GND	—	50

Table 2.12: Connector P5 Signal Locations

PIN	SIGNALS		PIN
1	GND	HLD*	2
3	GND	—	4
5	GND	DRV3*	6
7	GND	INDEX*	8
9	GND	DRV0*	10
11	GND	DRV1*	12
13	GND	DRV2*	14
15	GND	MOT*	16
17	GND	DIR*	18
19	GND	STEP*	20
21	GND	WD*	22
23	GND	WG*	24
25	GND	TR00*	26
27	GND	WPRT*	28
29	GND	RDD*	30
31	GND	SS1*	32
33	GND	READY*	34

Note: Some 5.25 inch floppy disk drives use pin 6 for the READY* signal.

2.7 FLOPPY DISK FUNCTIONS

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

2.7.1 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 5.25 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 jumper K16, and only up to three 5.25 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.

2.7.2 Drive Motor Control

Most 5.25 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 will be negated if no drive is selected within 30 seconds after a command is completed.

2.7.3 Drive Ready Detection

After selecting a drive and before performing any further operations, the floppy disk controller will check if the drive is ready. All 8 inch and most 5.25 inch floppy disk drives provide a READY* signal for this purpose.

If 5.25 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 flipflop which generates a READY* signal for the controller. This option is selected in the "Initialize Floppy Disk Parameters" command (See Paragraph 4.3.3.14).

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

2.7.4 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 will keep the head loaded as long as the drive is selected. Seek, read and write operations on the disk will not be started until the head load time of 50 milliseconds has been elapsed after the assertion of HLD*.

2.7.5 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 will be 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 outwards and low for stepping inwards. 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 (See Paragraph 4.3.3.14). The following list shows the available stepping rates for 5.25 inch and 8 inch floppy disk drives.

5.25 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 will not be started until the head settling time of 15 milliseconds has been elapsed after the last step pulse.

2.7.6 Compensation

When using 8 inch floppy disks with double data 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-postcompensation 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 Parameter" command (see Paragraph 4.3.3.14). 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 postcompensation.

2.7.7 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 (see Paragraph 4.3.3.14). The following table summarizes the valid values for these parameters and illustrates their dependencies.

Table 2.13: Physical Floppy Disk Organizations

DISK SIZE	5.25 inch						8 inch					
# DISK SIDES	1 or 2						1 or 2					
# CYLINDERS	40(48TPI) or 80(96TPI)						77					
DATA DENSITY	single (FM)			double (MFM)			single (FM)			double (MFM)		
# BYTES/SECTOR	128	256	512	256	512	1024	128	256	512	256	512	1024
# SECTORS/TRACK	16	9	5	16	9	5	26	15	8	26	15	8
# SECTORS/TRACK 0	16	9	5	16	8	4	26	15	8	26	14	8

2.7.8 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:

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

2.7.9 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 (see paragraph 4.3.3.14) and will affect 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.

Table 2.14: 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

As the IDTC provides a local buffer capable of storing one complete track, it will reach 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.

2.7.10 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 been elapsed. To save one disk revolution when stepping from the last sector on a track to the first sector on the next 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 (see paragraph 4.3.3.14) and will affect 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.

Table 2.15: 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

2.7.11 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 (see Paragraph 4.3.3.14).

2.7.11.1 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 will be 16, 9 or 5 for 5.25 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. In Motorola FM format, however, 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).

Table 2.16 describes one track in IBM 3740 and Motorola FM formats.

2.7.11.2 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 will be 16, 9 or 5 for 5.25 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 will always be 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 will be the specified number. If an odd number of sectors per track is specified for the disk, the number of sectors on track 0 will be the specified number minus one. This formatting method guarantees that track zero, where the operating system usually stores the disk ID and the format parameters, can be read in any system for defining the disk parameters.

Table 2.17 describes one track in IBM System 34 and Motorola MFM formats.

Table 2.16: IBM 3740 and Motorola FM Formats

NO. OF BYTES	CONTENTS	DESCRIPTION	
40	$FF..FF	gap 0	
6	$00..00	"	
1	$FC	index mark	
26	$FF..FF	gap 1	
6	$00..00	"	
1	$FE	ID address mark	
1	$xx	track number: $00..$27 for 5.25" 48 TPI $00..$4F for 5.25" 96 TPI $00..$4C for 8" 48 TPI	
1	$xx	side number: $00, $01 for IBM format $00 for Motorola format	this field repeated 5, 9 or 16 times for 5.25 inch, 8, 15 or 26 times for 8 inch disks
1	$xx	sector number: $01..$10 for 5.25" IBM format $01..$20 for 5.25" Mot format $01..$1A for 8" IBM format $01..$34 for 8" Mot format	
1	$xx	sector size: $00 for 128 bytes $01 for 256 bytes $10 for 512 bytes	
2	$xxxx	ID CRC bytes	
11	$FF..FF	gap 2	
6	$00..00	"	
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 2.17: IBM System 34 and Motorola MFM Format

NO. OF BYTES	CONTENTS	DESCRIPTION	
80	$4E..4E	gap 0	
12	$00..00	"	
3	$C2..C2	"	
1	$FC	index mark	
50	$4E..4E	gap 1	
12	$00..00	"	
3	$A1..A1	"	
1	$FE	ID address mark	
1	$xx	track number: $00..$27 for 5.25" 48 TPI $00..$4F for 5.25" 96 TPI $00..$4C for 8" 48 TPI	
1	$xx	side number: $00, $01 for IBM format $00 for Motorola format	this field repeated
1	$xx	sector number: $01..$10 for 5.25" IBM format $01..$20 for 5.25" Mot format $01..$1A for 8" IBM format $01..$34 for 8" Mot format	5, 9 or 16 times for 5.25 inch, 8, 15 or 26 times for 8 inch disks
1	$xx	sector size: $01 for 256 bytes $10 for 512 bytes $11 for 1024 bytes	
2	$xxxx	ID CRC bytes	
22	$4E..4E	gap 2	
12	$00..00	"	
3	$A1..A1	"	
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	

2.7.12 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 will always be 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 the logical sector address is as follows:

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

The following examples illustrate the floppy disk organization. Table 2.18 shows the organization of a single sided, single density 5.25 inch floppy disk with a sector size of 128 bytes and a block size of 256 bytes. Table 2.19 shows the organization of a double sided, double density 5.25 inch floppy disk with a sector size of 256 bytes and a block size of 512 bytes.

Table 2.18: 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 ...

Table 2.19: 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 ...				

2.8 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 on 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 signals supported by the IDTC.

2.8.1 Drive and Stream Selection

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

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 minimize the tape access time between successive operations on the same drive, the IDTC will keep the select signals asserted for two seconds after the completion of any command, unless another selection has to be made immediately.

2.8.2 Drive Motor Control

The Head Load line on the SA850 interface serves as the FloppyTape Motor On control signal. This signal is asserted in order to read or write data. After assertion of the signal, a 400ms delay is introduced prior to any data operation to allow the medium to reach full operating speed.

2.8.3 Drive Ready Detection

After having selected a drive and before performing any further operations, the floppy disk/tape controller will check 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.

2.8.4 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 dowmstream to the next segment; if the DIR* signal is high, the tape movement is upstream to the previous segment.

2.8.5 Compensation

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

2.8.6 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.

Figure 2.11 illustrates the physical tape layout. Table 2.20 shows the correlation between sector sizes, the resultant number of sectors per segment and tape, and the tape capacity.

Figure 2.11: FloppyTape Layout

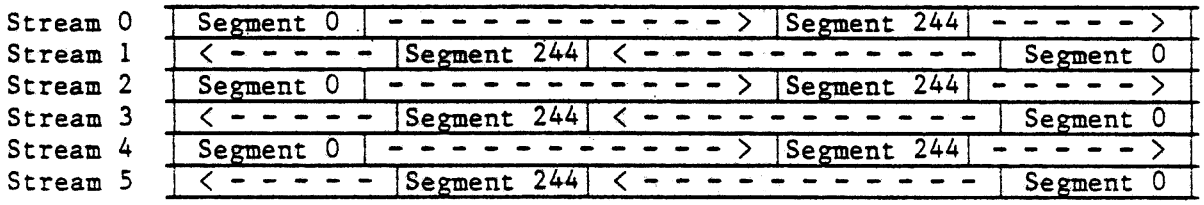


Table 2.20: Physical Tape Organization

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

2.8.7 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 (see Paragraph 4.3.4.12). Table 2.21 shows the FloppyTape recording format.

Table 2.21: FloppyTape Recording Format

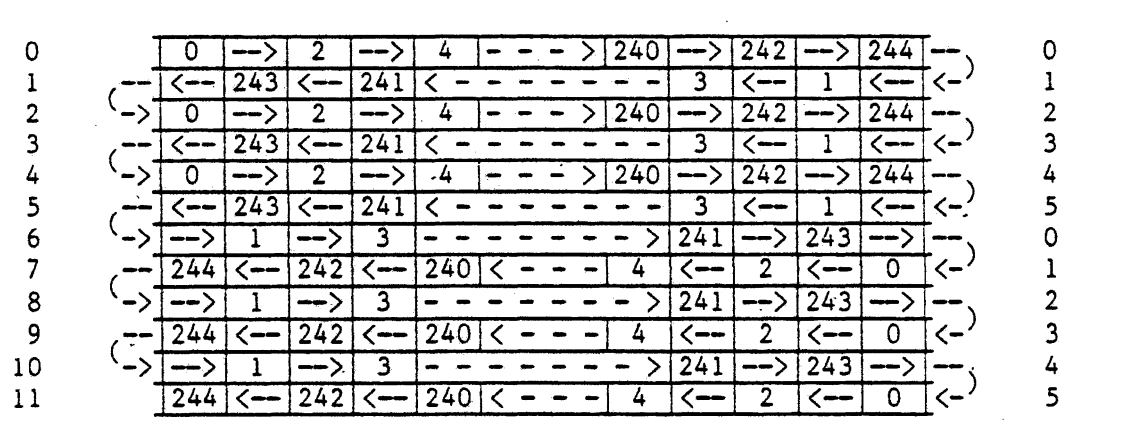
NO. OF BYTES	CONTENTS	DESCRIPTION	
80	$4E..4E	gap 0	
12	$00..00	sync bytes	
3	$C2..C2	control bytes	
1	$FC	index character	
50	$4E..4E	gap 1	
12	$00..00	sector ID sync bytes	this field repeated for all sectors
3	$A1..A1	control character	
1	$FE	ID address mark	
1	$xx	segment number ($00..$F5)	
1	$xx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

2.8.8 Segment Skipping and Logical Streams

The Cipher 525 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 (see Paragraph 4.3.4.12).

Figure 2.12 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.

Figure 2.12: Segment Skipping



2.8.9 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 tape 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 the 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 the logical sector address is as follows:

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

The block size must be at least the sector size.

2.8.10 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 2.22. 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 2.22: Bad Segment List

BYTE #	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 #3 is read, the bad segment list is loaded into the IDTC and the bad segments will be locked out for all further tape operations.

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

2.8.11 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 C or below 5 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 will deform 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 his 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.

- After the tape is formatted and verified, the bad segment list should be inspected by reading sector #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 #3 must be read. This loads the bad segment list into the IDTC and thereby ensures that bad segments on the tape will be 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 the there are no bad segments on the cartridge. This feature is useful when optimizing copy or backup times.

- The 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 identifiation.

CHAPTER 3

INSTALLATION INSTRUCTIONS

3.1 INTRODUCTION

This chapter provides the user of the MVME319 Intelligent Disk/Tape Controller with the unpacking, inspection, hardware preparation and installation procedures.

3.2 UNPACKING INSTRUCTIONS

IF THE SHIPPING CARTON IS DAMAGED UPON RECEIPT, REQUEST THAT THE CARRIER'S AGENT BE PRESENT DURING UNPACKING AND INSPECTION OF THE MODULE.

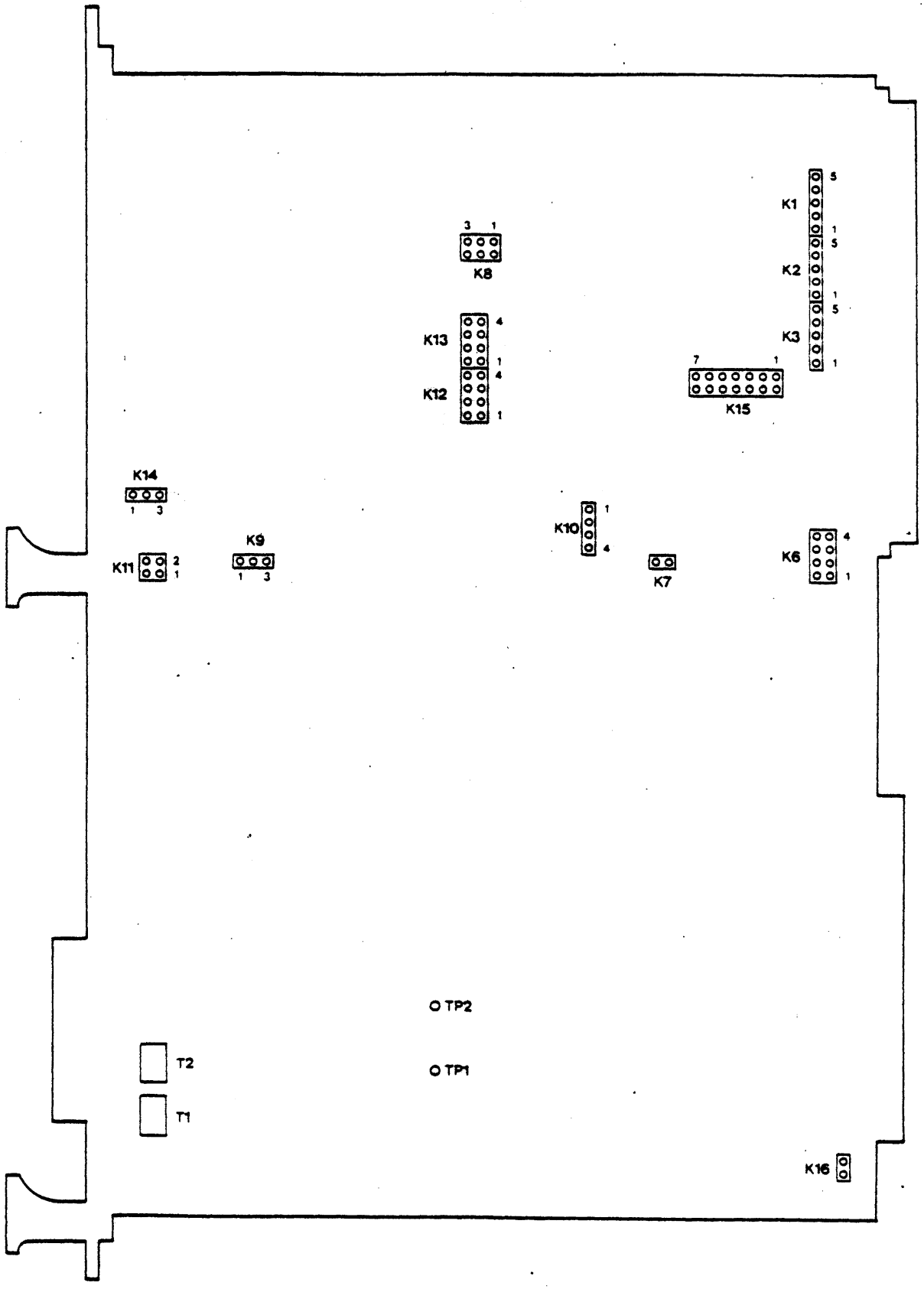
Unpack the MVME319 Intelligent Disk/Tape Controller from its shipping carton. Refer to the packing list and verify that all items are present. Save the packing material for storing or reshipping the module.

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

3.3 INSPECTION

The module should be inspected upon receipt for broken, damaged or missing parts and for physical damage to the printed circuit board.

Figure 3.1: MVME319 Jumper Area Locations



3.4 MODULE PREPARATION

This paragraph describes the hardware preparation of the MVME319 module prior to system installation. The module is shipped with factory-installed jumpers ready for use as a single disk controller in a VME system. However, the jumper areas which determine the VMEbus requester priority level, the VMEbus interrupter priority level, the command channel base address, the VMEbus system fail output and the 5.25 inch floppy disk DRV3* line must be configured according to the actual system requirements.

Figure 3.1 illustrates the locations of the jumper areas on the MVME319 module. Table 3.1 lists the function of each jumper area and refers to the detailed descriptions in Paragraphs 3.4.1 through 3.4.6.

Table 3.1: MVME319 Jumper Areas

JUMPER	FUNCTION	OPTIONS	PARA
K1, K2, K3	VMEbus Requester Priority Level	Select level 0, 1, 2 or 3	3.4.1
K6, K8	VMEbus Interrupter Priority Level	Select level 0, 1, 2, 3, 4, 5, 6 or 7	3.4.2
K15	Command Channel Base Address	Select any 512 byte boundary in the short I/O address segment	3.4.3
K7	System Fail Enable	Enable/disable SYSFAIL* output	3.4.4
K16	Drive Select 3 Enable	Select DRV3* or READY* on P5 pin 6	3.4.5
K10	Interrupt Contr. Configuration	No user options	3.4.6
K9, K11	Local RAM Configuration	No user options	3.4.6
K12, K13, K14	Local RAM Configuration	No user options	3.4.6

3.4.1 VMEbus Requester Priority Level

The jumper areas K1, K2 and K3 determine the priority level of the VMEbus requester. On K3 the bus request output of the VMEbus requester is connected with the appropriate bus request line, on K1 and K2 the VMEbus requester is placed in the corresponding bus grant daisy chain.

In VME systems containing an option ONE single level bus arbiter, the VMEbus requester must be placed on level 3. For use with a multilevel arbiter, any one of the four priority levels may be selected.

In multilevel arbitration systems, the bus grant daisy chain lines which are not used by the MVME319 should be jumpered on the VMEbus backplane.

Shipment configuration: VMEbus requester on level 3.

Figure 3.2: Jumper Areas K1, K2, K3

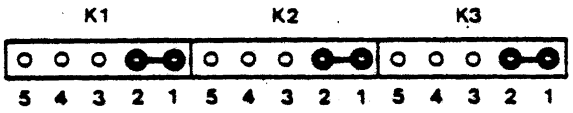


Table 3.2: VMEbus Requester Priority Selection

K1, K2, K3 CONNECTIONS	VMEBUS REQUESTER PRIORITY LEVEL
K1: 1-5; K2: 1-5; K3: 1-5	VMEbus requester on priority level 0
K1: 1-4; K2: 1-4; K3: 1-4	VMEbus requester on priority level 1
K1: 1-3; K2: 1-3; K3: 1-3	VMEbus requester on priority level 2
K1: 1-2; K2: 1-2; K3: 1-2	VMEbus requester on priority level 3

3.4.2 VMEbus Interrupter Priority Level

The jumper areas K6 and K8 determine the priority level of the VMEbus interrupter. On K6 the interrupt request output of the VMEbus interrupter is connected with the appropriate interrupt request line. K8 presents the binary code of the interrupt level to the level comparator.

Shipment configuration: VMEbus interrupter on level 3.

Figure 3.3: Jumper Areas K6, K8

Table 3.3: VMEbus Interrupter Priority Selection

K6, K8 CONNECTIONS	VMEBUS INTERRUPTER PRIORITY LEVEL
K6: 4-8; K8: 1-6, 2-5	VMEbus interrupter on priority level 1
K6: 5-8; K8: 1-6, 3-4	VMEbus interrupter on priority level 2
K6: 3-8; K8: 1-6	VMEbus interrupter on priority level 3
K6: 6-8; K8: 2-5, 3-4	VMEbus interrupter on priority level 4
K6: 2-8; K8: 2-5	VMEbus interrupter on priority level 5
K6: 7-8; K8: 3-4	VMEbus interrupter on priority level 6
K6: 1-8; K8: none	VMEbus interrupter on priority level 7

3.4.3 Command Channel Base Address

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

The jumper area K15 determines the base address of the MVME319 command channel relative to the base address of the VMEbus short supervisory I/O address segment. Each of the VMEbus address lines A09-A15 is represented by one pin on K15 and may be chosen to be low (if the jumper is set) or high (if the jumper is removed) for selecting the module. This allows the command channel to be placed on any 512 byte boundary within the 64K byte short supervisory I/O address segment.

Shipment configuration: command channel base address = $0000.

Figure 3.4: Jumper Area K15

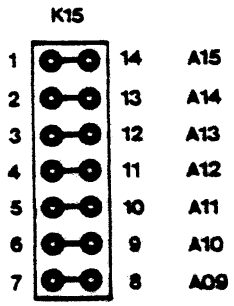


Table 3.4: Command Channel Base Address Selection

K15 CONNECTIONS	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

3.4.4 System Fail Enable

In the case of a selftest failure or a severe firmware malfunction the IDTC illuminates the FAIL LED on the front panel. This local failure signal is optionally output to the VMEbus SYSFAIL* line when a jumper is set on K7.

Shipment configuration: SYSFAIL* output enabled.

Figure 3.5: Jumper Area K7

Table 3.5: System Fail Options

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

3.4.5 Drive 3 Select Enable

With respect to 5.25 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 kind, the MVME319 module carries the drive select signal DRV3* via the jumper K16 to pin 6 of connector P5.

If 5.25 inch floppy disk drives with the READY* signal at pin 6 are used in the system, the jumper K16 must be removed, and only up to three 5.25 inch floppy disk drives may be installed.

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

Also note that whenever the READY* signal is not available, the IDTC must be initialized to use the INDEX* signal for detecting drive readiness. Refer to Paragraph 2.7.3 for details.

Shipment configuration: DRV3* enabled.

Figure 3.6: Jumper Area K16

Table 3.6: Drive 3 Select Options

K16 CONNECTION	CONNECTOR P5 CONFIGURATION
1-2	DRV3* enabled. Only valid if no 5.25 inch floppy disk drives with READY* at pin 6 are used.
none	DRV3* disabled. Jumper must be removed if any 5.25 inch floppy disk drive uses pin 6 for READY*.

3.4.6 Prescribed Jumper Configurations

The jumpers which configure the interrupt controller, the local RAM and the local ROM, are not available for the user. The factory installed configuration must not be altered. However, for inspection and service purposes the following table lists the proper jumper installations for K9 to K14.

Table 3.7: Prescribed Jumper Configurations

JUMPER CONNECTIONS	FUNCTIONS
K10: 1-4	Interrupt Controller Configuration. Any write operation from the VMEbus into the command channel interrupts the IDTC processor.
K9: 1-2 K11: 2-4	Local RAM Configuration. Memory sockets U54, U62, U78 and U79 configured for 8K*8 bit static RAMs.
K12: 1-7, 2-6, 3-4 K13: 1-4, 2-3 K14: 1-2	Local ROM Configuration. Memory sockets U6 and U14 configured for 8K*8 bit EPROMs.

3.5 HARD DISK CONFIGURATION

The MVME319 Intelligent Disk/Tape Controller 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 3.7.

A 50-pole flat ribbon cable provides the SASI/SCSI bus between connector P3 on the MVME319 module and the hard disk controllers. It is recommended to use a 3M P/N 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 3 m.

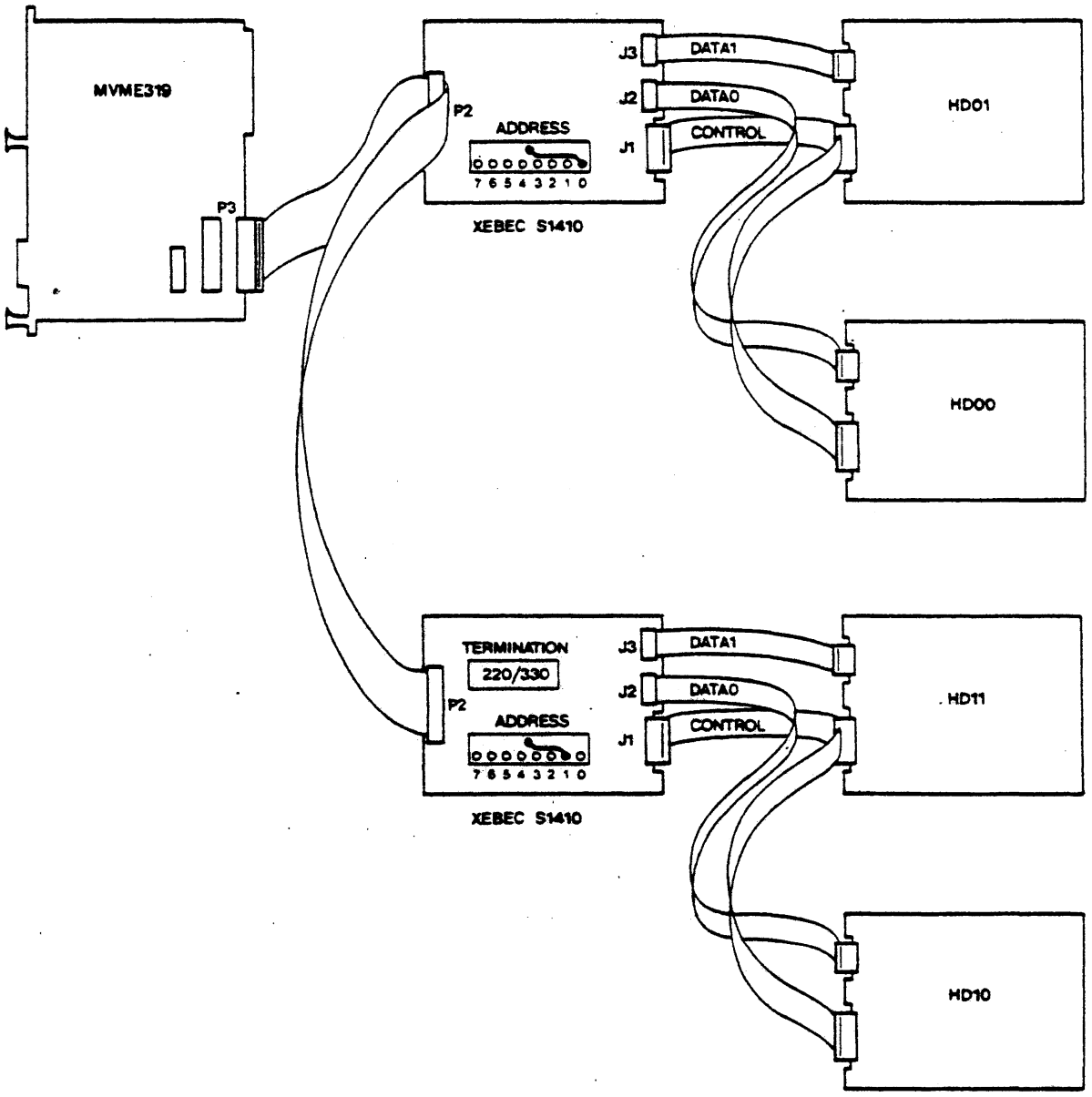
The SASI/SCSI bus signals driven by the MVME319 module must be terminated on the last hard disk controller in the line, using a 220/330 ohms resistor network.

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

Jumper ADDR0 inserted: Controller # = 0
Jumper ADDR1 inserted: Controller # = 1
Jumper ADDR2 inserted: Controller # = 2
Jumper ADDR3 inserted: Controller # = 3
Jumper ADDR4 inserted: Controller # = 4
Jumper ADDR5 inserted: Controller # = 5
Jumper ADDR6 inserted: Controller # = 6
Jumper ADDR7 inserted: Controller # = 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 3.7: Hard Disk Configuration



3.6 FLOPPY DISK/TAPE CONFIGURATION

The MVME319 Intelligent Disk/Tape Controller is capable of controlling up to four 5.25 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, 5.25 inch only, mixed size floppy disk and FloppyTape configurations with either 8 inch or 5.25 inch floppy disk drives.

3.6.1 8 Inch Floppy Disk Configuration

8 inch floppy disk drives are connected with the MVME319 module according to the example shown in Figure 3.8. A 50-pole flat ribbon cable provides the bus between connector P4 on the MVME319 module and the disk drives. The 50 pole dual row female connector used at P4 must not exceed a maximum height of 12 mm to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M P/N 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 3 m.

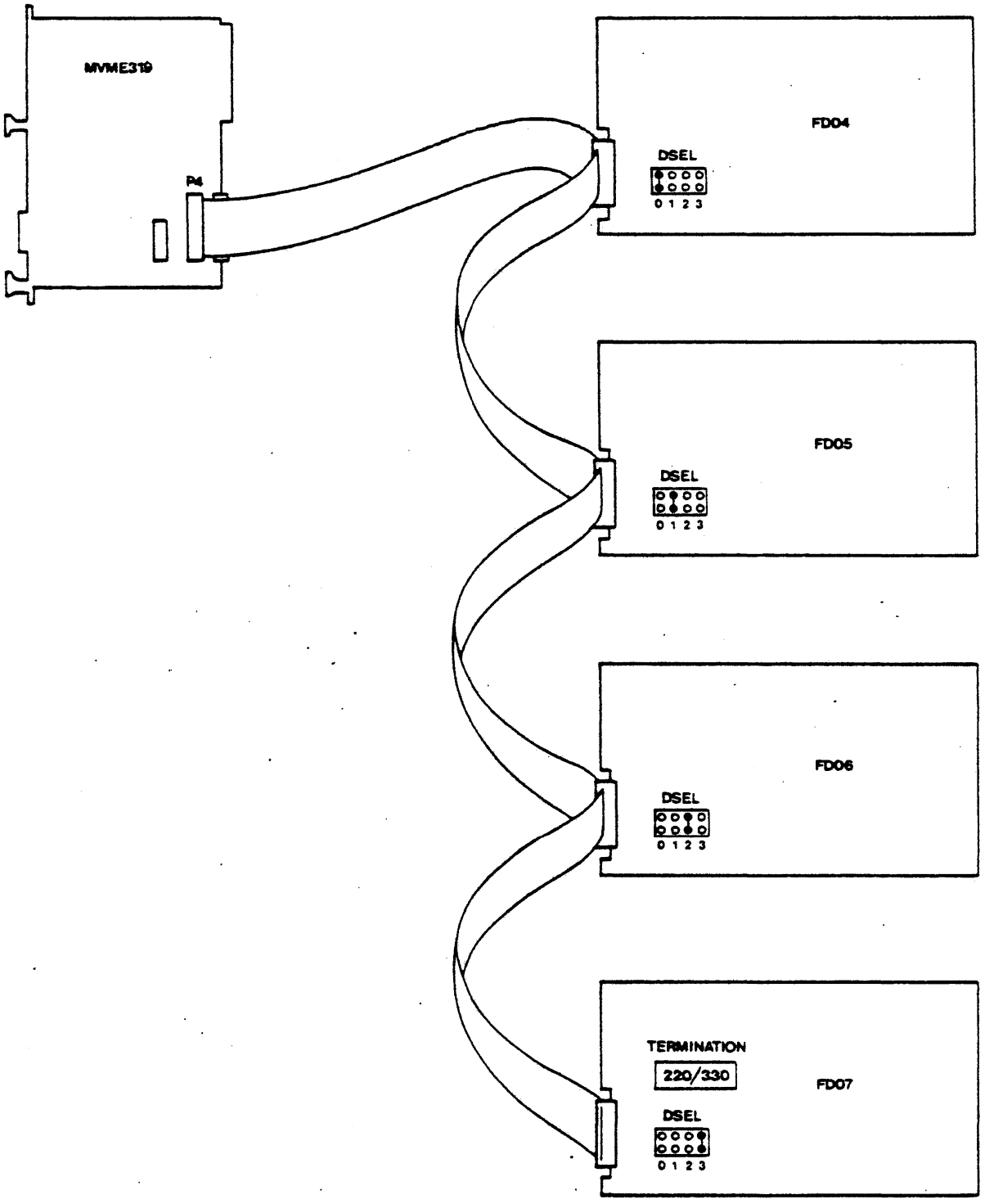
The signals driven by the MVME319 module must be terminated on the last disk drive in the line, using a 220/330 ohms resistor network. On all intermediate disk drives the termination resistors must be removed.

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

Jumper DSEL0 inserted: Floppy Disk Drive # = 4
Jumper DSEL1 inserted: Floppy Disk Drive # = 5
Jumper DSEL2 inserted: Floppy Disk Drive # = 6
Jumper DSEL3 inserted: Floppy Disk Drive # = 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 3.8: 8 Inch Floppy Disk Configuration



3.6.2 5.25 Inch Floppy Disk Configuration

5.25 inch floppy disk drives are connected with the MVME319 module according to the example shown in Figure 3.9. A 34-pole flat ribbon cable provides the bus between connector P5 on the MVME319 module and the disk drives. The 34-pole dual row female connector used at P5 must not exceed a maximum height of 12 mm to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M P/N 3414-3000 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 3 m.

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

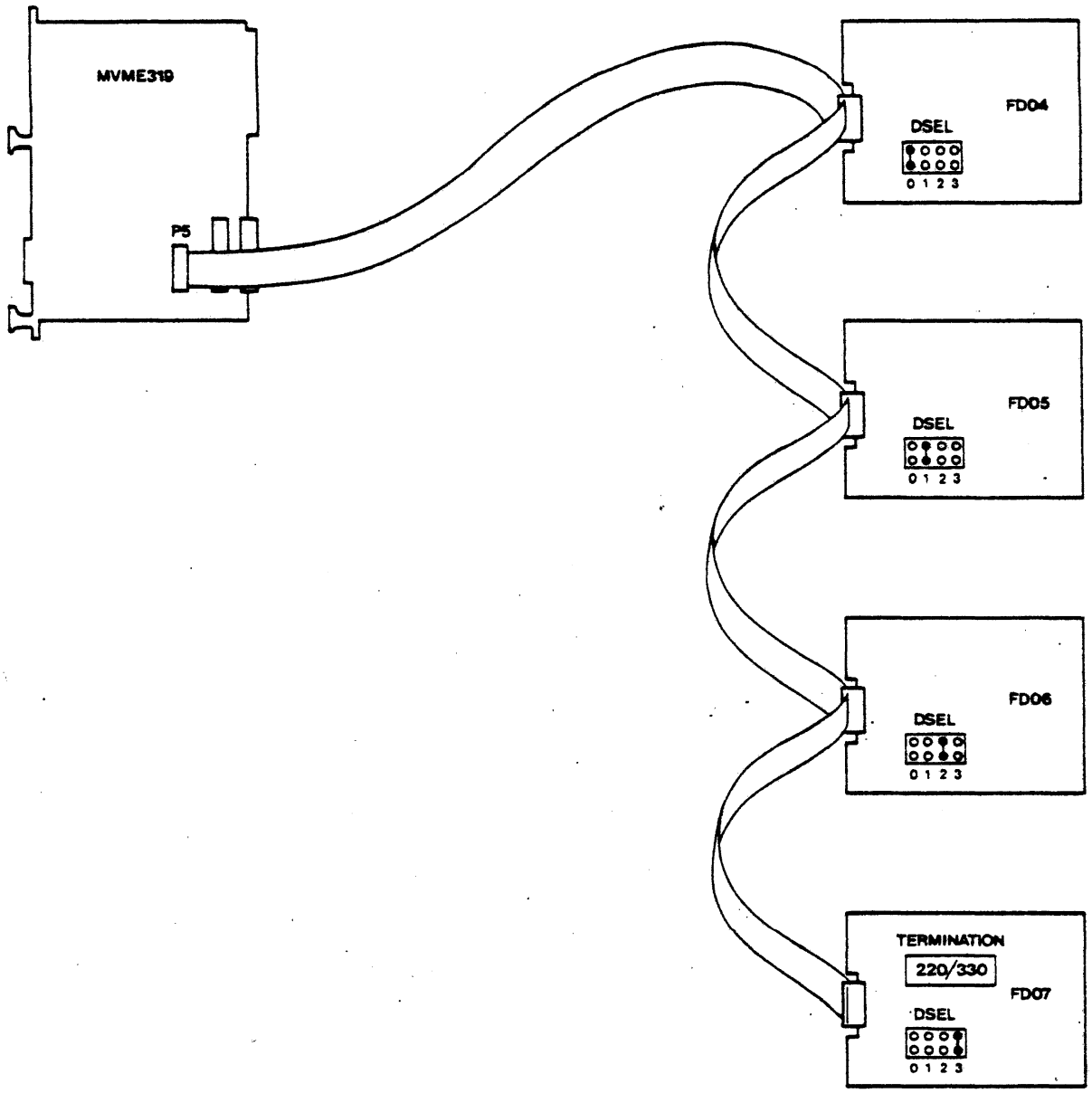
For each disk drive, the position of the drive select jumper specifies the logical drive number used in the IDTC commands as follows:

Jumper DSEL0 inserted: Floppy Disk Drive # = 4
Jumper DSEL1 inserted: Floppy Disk Drive # = 5
Jumper DSEL2 inserted: Floppy Disk Drive # = 6
Jumper DSEL3 inserted: Floppy Disk Drive # = 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 5.25 inch floppy disk drives shall be installed which use pin 6 of the connector for the READY* signal, refer to Paragraph 3.4.5 for further configuration details.

Figure 3.9: '5.25 Inch Floppy Disk Configuration



3.6.3 Mixed Size Floppy Disk Configuration

Multiple floppy disk drives including both 8 inch and 5.25 inch size are connected with the MVME319 module according to the example shown in Figure 3.10. A 50-pole flat ribbon cable provides the bus between connector P4 on the MVME319 module and the 8 inch disk drives, a 34-pole flat ribbon cable provides the bus between connector P5 and the 5.25 inch disk drives. The dual row female connectors used at P4 and P5 must not exceed a maximum height of 12 mm to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M P/N 3425-3000 connector at P4 and a 3M P/N 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 1.5 m.

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

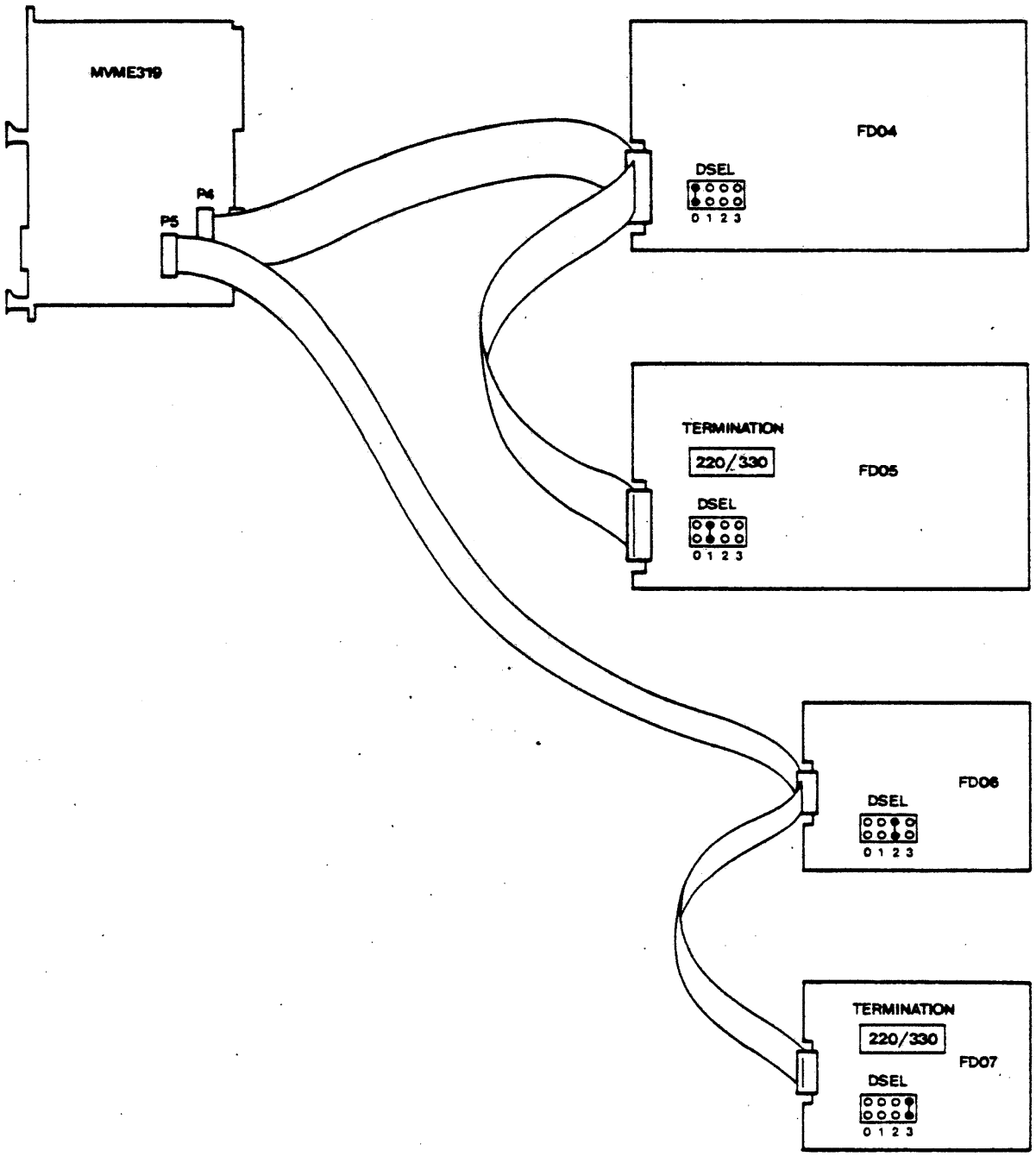
For each disk drive, the position of the drive select jumper specifies the logical drive number used in the IDTC commands as follows:

Jumper DSEL0 inserted: Floppy Disk Drive # = 4
Jumper DSEL1 inserted: Floppy Disk Drive # = 5
Jumper DSEL2 inserted: Floppy Disk Drive # = 6
Jumper DSEL3 inserted: Floppy Disk Drive # = 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 5.25 inch floppy disk drives shall be installed which use pin 6 of the connector for the READY* signal, refer to Paragraph 3.4.5 for further configuration details.

Figure 3.10: Mixed Size Floppy Disk Configuration



3.6.4 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 module in a daisy chain configuration according to the example shown in Figure 3.11. A 50 pole flat ribbon cable provides the bus between connector P4 on the MVME319 module and the floppy disk and FloppyTape drives. The 50 pole dual row female connector used at P4 must not exceed a maximum height of 12 mm to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M P/N 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 3 m.

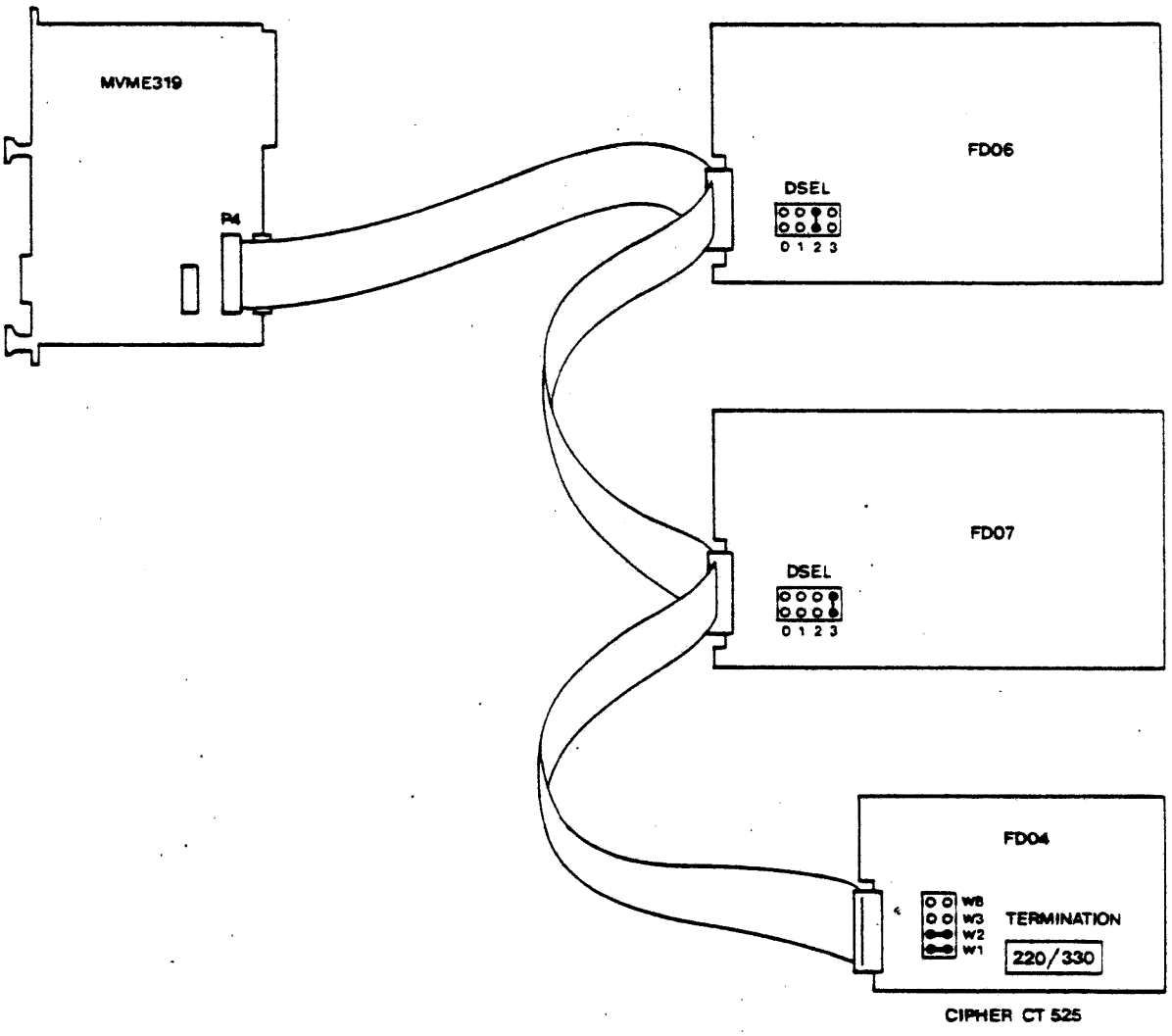
Note that the FloppyTape drive termination resistor network is difficult to remove from its socket. For this reason, it is recommended that the tape drive is the last drive in the chain. All termination networks on floppy disk drives must be removed.

The position of the drive select jumper in each floppy disk drive determines the logical disk drive number used for that drive in the IDTC commands as follows:

Jumper DSEL2 inserted: Floppy Disk Drive # = 6
Jumper DSEL3 inserted: Floppy Disk Drive # = 7

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

Figure 3.11: Tape Drive and 8 Inch Floppy Disk Configuration



3.6.5 FloppyTape and 5.25 Inch Floppy Disk Configuration

A Cipher CT 525 FloppyTape and up to two 5.25 inch floppy disk drives are connected to the MVME319 module according to the example shown in Figure 3.12. A 50 pole flat ribbon cable provides the bus between connector P4 on the MVME319 module and the FloppyTape drive and a 34 pole flat ribbon cable between connector P5 on the MVME319 and the 5.25 inch floppy disk drives. The 50 and 34 pole dual row female connectors used at P4 and P5 must not exceed a maximum height of 12 mm to avoid conflicts with the module adjacent to the MVME319. It is recommended to use a 3M P/N 3425-3000 connector at P4 and a 3M P/N 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 each connector/cable assembly should not exceed 1.5m.

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

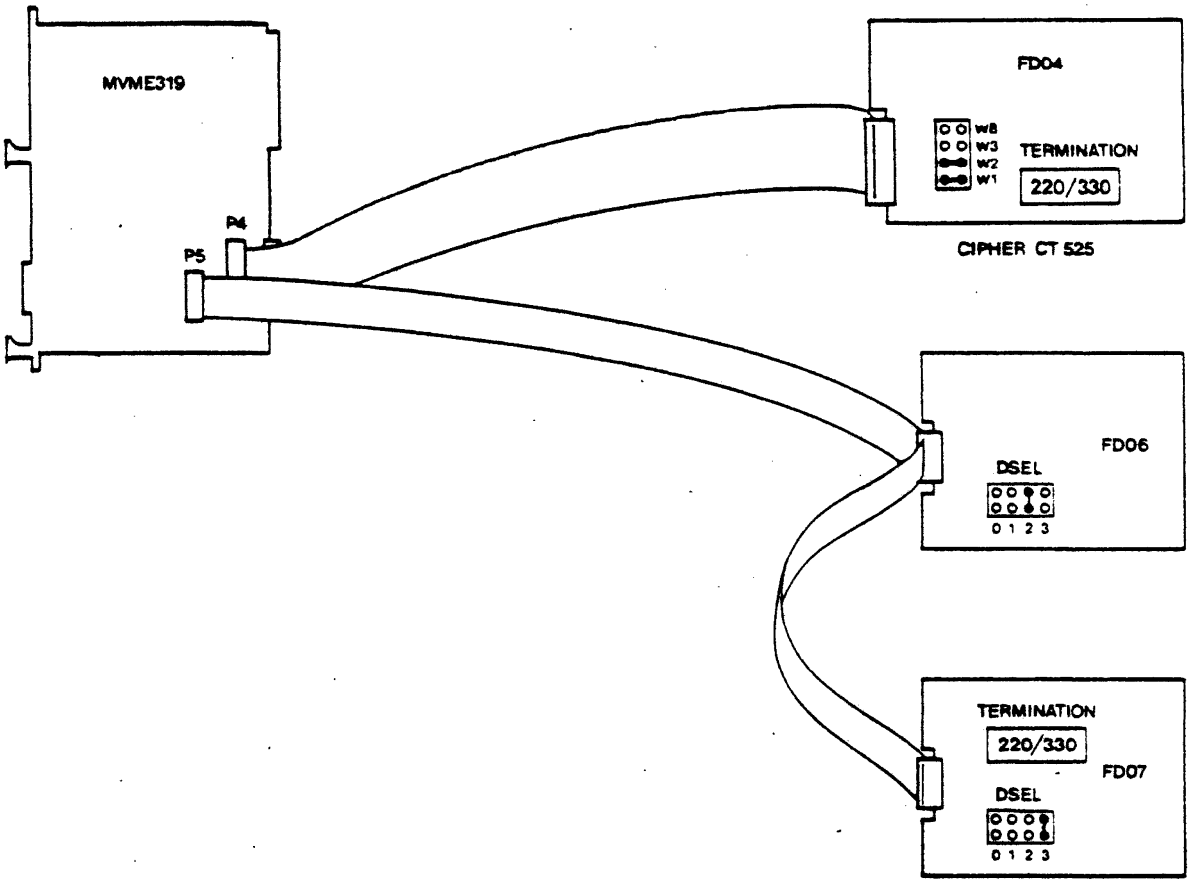
The position of the drive select jumper in each floppy disk drive determines the logical disk drive number used for that drive in the IDTC commands as follows:

Jumper DSEL2 inserted: Floppy Disk Drive # = 6
Jumper DSEL3 inserted: Floppy Disk Drive # = 7

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

If 5.25 inch floppy disk drives which use pin 6 of the connector for the READY* signal are to be installed, refer to Paragraph 3.4.5 for further configuration details.

Figure 3.12: Tape Drive and 5.25 Inch Floppy Disk Configuration



CHAPTER 4

OPERATING INSTRUCTIONS

4.1 INTRODUCTION

This Chapter provides the user of the MVME319 Intelligent Disk/Tape Controller with details of the IDTC command channel, commands for hard disk, floppy disk and Floppy-Tape drives and status messages generated by the IDTC.

4.2 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 jumper area K15 and may be any 512 byte boundary (see Paragraph 3.4.3). Figure 4.1 shows the command channel address map as seen from the VMEbus.

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.

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 (see Paragraph 4.3.1.1).

In the simple command channel protocol each command from the operating system will be 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.

Note that Motorola's 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 the Paragraphs 4.3 and 4.4.

Figure 4.1: IDTC Command Channel Address Map

ADDRESS	EVEN BYTES	ODD BYTES	ADDRESS
1FE :	Even bytes not used	status message buffer	1FF : 185
:		CMD ACK/NAK	183
:		MSG SENT	181
:		command message buffer	17F : 105
:		MSG ACK/NAK	103
:		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 segment. Use address modifier code $2D when addressing the IDTC command channel.

4.2.1 Simple Command Channel Protocol

A flow diagram of the simple IDTC command channel protocol is shown in Figure 4.2.

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.

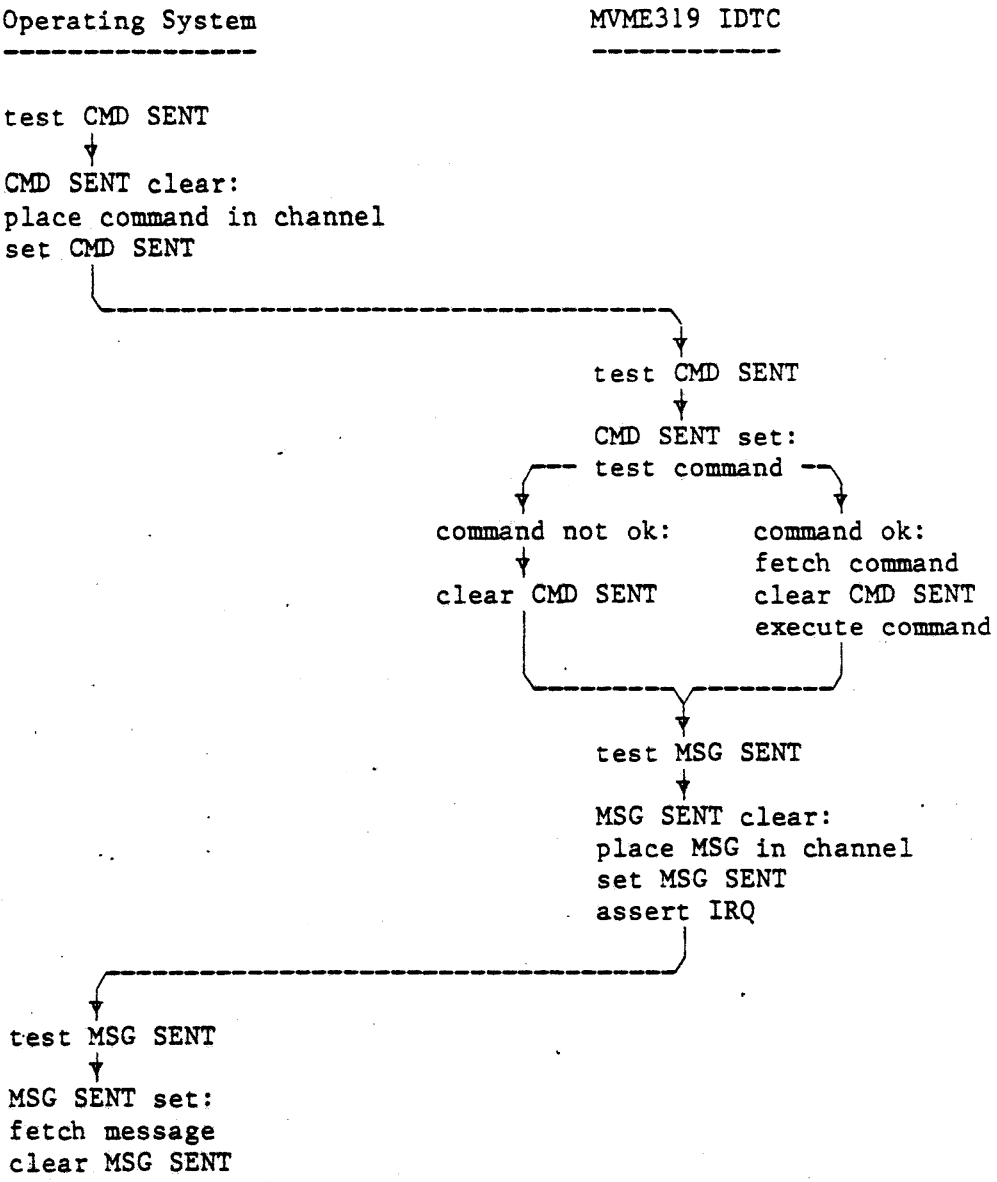
When the IDTC has found a correct command packet in the channel, it transfers the command to the appropriate task and clears the CMD SENT byte to allow new commands to be transferred.

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.

Figure 4.2: Simple IDTC Command Channel Protocol



4.2.2 Extended Command Channel Protocol

A flow diagram of the extended IDTC command channel protocol is shown in Figure 4.3.

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 CMD 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.

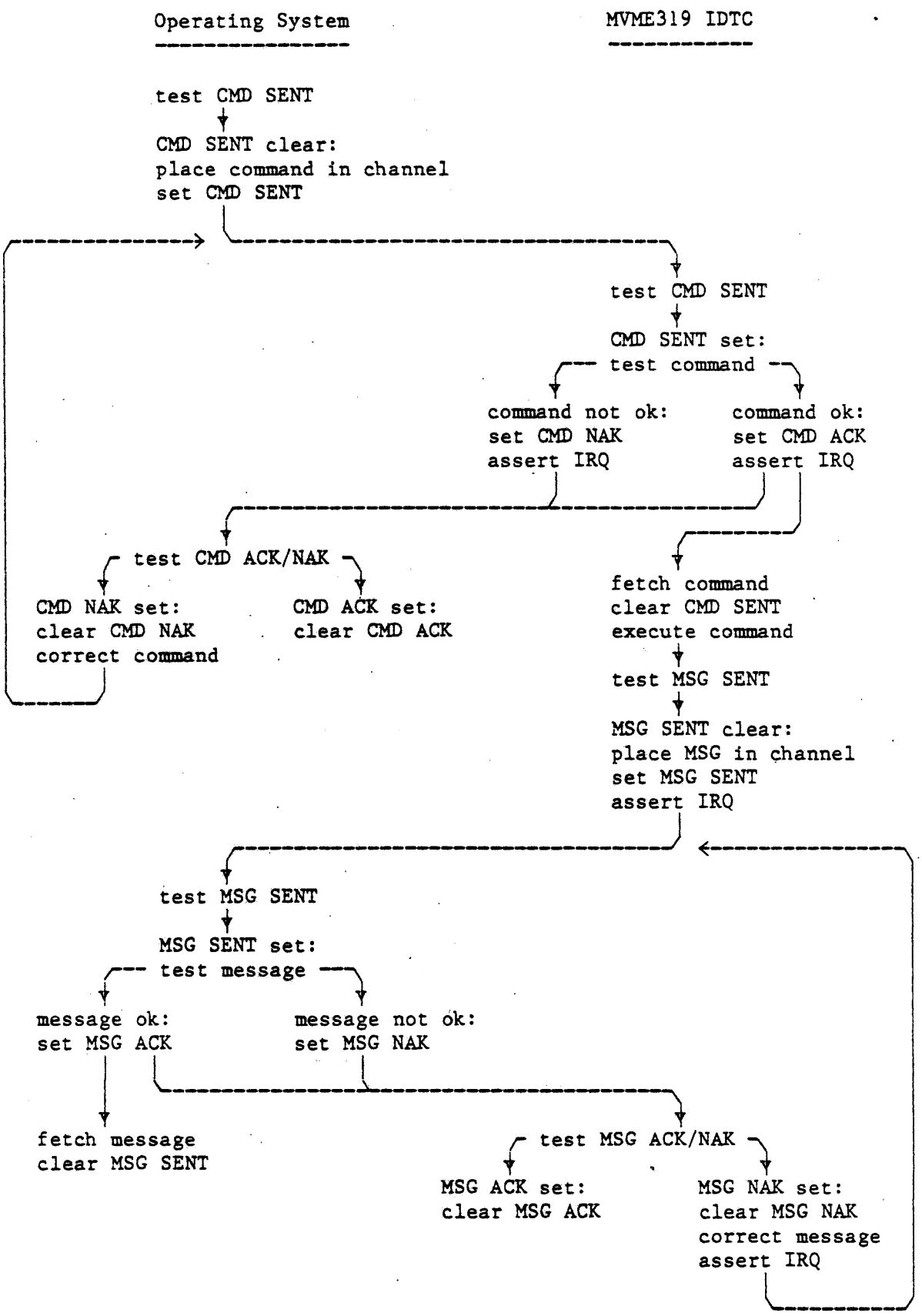
Prior to placing a status message into the channel, the IDTC tests the MSG SENT byte. A value of $00 indicates that any previous status message has been acknowledged by the operating system and that the status message buffer is available. Then the IDTC places a status message packet into the status message 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 will correct the status message and retransmit the packet. In either case the IDTC clears the MSG ACK/NAK byte to indicate that the message check result has been read.

When the operating system has found a correct command packet in the channel, it clears the MSG SENT byte to allow new status messages to be transferred.

Figure 4.3: Extended IDTC Command Channel Protocol



4.3 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 will be 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.
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 will translate the specified controller number into the according address on the SASI/SCSI bus when selecting the controller. The drive number defines the logical unit number 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 will assert 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.

BYTE #	CONTENTS	DESCRIPTION (continued)
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 will be translated by the IDTC into the according logical hard disk sector address, for floppy disks, into the according side, cylinder, and physical sector numbers and for FloppyTape into the according stream, segment and physical sector numbers. For further details of the disk and tape structures refer to Paragraphs 2.5.4, 2.7.7 and 2.8.6. 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 16 Mbyte 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 will be 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 Paragraph 2.5.3.

4.3.1 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.

4.3.1.1. Configure IDTC

The "Configure IDTC" command must be the first command sent to the IDTC after 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. (See also Paragraph 4.2.)

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 will read the specified status/ID byte during the interrupt acknowledge cycle for initiating the appropriate servicing routine.

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	%0000000 = reserved

4.3.1.2. Request IDTC Self-Test

The "Request IDTC Self-Test" command executes the following tests on the IDTC module:

1. Floppy disk/tape controller : Non-destructive test of the track, sector and data registers.

2. SASI/SCSI bus interface : Test of the control and data registers.

3. Hard disk controller : If the previous test indicated that a hard disk controller is connected, the "Controller Diagnostics" command will be transmitted. This causes the controller to perform a selftest and to return a status message. If no hard disk controller is connected, this part of the IDTC self-test will be skipped.

After completion of these tests the IDTC places a status message in the command channel which contains all results (see Paragraph 4.2).

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.

4.3.2 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 will translate it into the according disk controller command and supervise the hard disk controller operation.

4.3.2.1 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.

4.3.2.2 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	$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.

4.3.2.3 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.

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.

4.3.2.4 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 (see Paragraph 4.4.20).

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.

4.3.2.5 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 Paragraph 4.3.2.10. If the XEBEC disk controller does not indicate the completion of the operation within 1 minute, the command execution will be aborted and a "Controller Error" message will be 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 on the track ($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, write $6C into all data fields.

4.3.2.6 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 paragraph 4.3.2.10. This command is applicable for all supported hard disk controllers. When using the ADAPTEC ACB 4000 controller with automatic media defect handling, refer to Paragraph 4.3.2.7.

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 disk, clear "bad block" flag in all sectors, set ID fields according to the interleave factor in the CDB, write $6C into all data fields.

4.3.2.7 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 the formatting of a hard disk with media defects by utilising 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	start 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 #1
7	Head number of defect #1
8-11	Bytes from index
.	.
.	.
.	.
8*N-4 to 8*N+3	Cylinder number of defect #N Head number of defect #N Bytes from index

Note: 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 will result in an IDTC hang-up. Also, track 0 must be free of defects, as it is used for storing disk formatting information.

4.3.2.8 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 will be aborted and a "Controller Error" message will be 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, clear error status flag in the drive.

4.3.2.9 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 will be aborted and a "Controller Error" message will be 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: SEEK

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

4.3.2.10 Initialize Hard Disk Parameters

The "Initialize Hard Disk Parameters" command initializes the hard disk controller with the parameters of the disk drives used. Each controller connected with 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 with this controller, whereas 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 pages 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 on XEBEC S1410/S1410A controller.
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 currert ($0000-$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 S1410A 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
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. See the ACB 4000 manual for further details.
21-23	$000000	filler
24	$xx	stepping rate ($00 = 3ms, $01 = 28us, $02 = 12us)
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)

4.3.3 Floppy Disk Commands

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

4.3.3.1 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 device 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

4.3.3.2 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 device 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

4.3.3.3 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 device 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

4.3.3.4 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.

IDTC Command Message:

BYTE #	CONTENTS	DESCRIPTION
0	$02	STX mark
1	$xx	message ID
2	$15	command packet size: 21 bytes
3	$xx	physical device 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 written ($0000-$FFFF)
20	$03	ETX mark

4.3.3.5 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 device 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 written ($0000-$FFFF)
20	$03	ETX mark

4.3.3.6 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 device 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

4.3.3.7 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 device 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

4.3.3.8 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 (see Paragraph 4.4.20).

IDTC Command Message:

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

4.3.3.9 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 device number ($04-$07)
4	$30	command type: TEST
5	$03	function code
6	$03	ETX mark

4.3.3.10 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 Paragraph 4.3.3.14. After formatting, all sectors are read back for CRC verification. In the case of a seek error or a CRC 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 device 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

4.3.3.11 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 Paragraph 4.3.3.14. After formatting, all sectors are read back for CRC verification. In the case of a seek error or a CRC 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 device number ($04-$07)
4	$40	command type: CONTROL
5	$02	function code
6	$03	ETX mark

4.3.3.12 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:

1. 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.

2. 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 5.25 inch drive.

3. 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 5.25 inch disk or a double sided single density 8 inch disk are selected, according to the table below.

4. 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.

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

disk size	5.25"	5.25"	8"	8"
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	IBM 3740	IBM 34	Motorola	Motorola

Note: "Restore" does not change configuration between tape and disk.

IDTC Command Message:

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

4.3.3.13 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" will be executed and the "Seek Track" be 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 device 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 track to be sought ($0000-$FFFF)
14	$03	ETX mark

4.3.3.14 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 next page)
7	$xx	attribute byte 1 (see next page)
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)
12-13	$xxxx	number of cylinders per disk ($0000-$00FF)
14-15	$xxxx	first cylinder with compensation ($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 = #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 = 5.25 inch drive with READY signal %1 = 5.25 inch drive without READY signal (drive ready detection with INDEX signal)
2	%0 = 5.25 inch disk size %1 = 8 inch disk size
1	%0 = pre- or postcompensation %1 = no compensation
0	%0 = write precompensation (if command byte #15 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 (5.25 inch) %010 = 6 ms (8 inch), 12 ms (5.25 inch) %011 = 10 ms (8 inch), 20 ms (5.25 inch) %100 = 15 ms (8 inch), 30 ms (5.25 inch)

4.3.4 FloppyTape Commands

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

4.3.4.1 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

4.3.4.2 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

4.3.4.3 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

4.3.4.4 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 written ($0000-$FFFF)
20	$03	ETX mark

Note: Do not write to sector #3 (bad segment table)

4.3.4.5 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 written ($0000-$FFFF)
20	$03	ETX mark

Note: Do not write to sector #3 (bad segment table)

4.3.4.6 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

Note: Do not write to sector #3 (bad segment table)

4.3.4.7 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

4.3.4.8 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 (see Paragraph 4.4.20).

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

4.3.4.9 Format Tape

The "Format Tape" command formats all 6 streams of the tape. Unless otherwise defined by a previous "Initialize Tape Drive Parameters" command, the recording format defaults to the parameters described in Paragraph 4.3.4.12. 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 #3) of the first tape segment of stream 0, at the end of the command (see Paragraph 2.8.9).

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

4.3.4.10 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

4.3.4.11 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 will be rewound and the "Seek Segment" be 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 segment to be sought ($0000-$FFFF)
14	$03	ETX mark

4.3.4.12 Initialize FloppyTape Parameters

The IDTC firmware generates a tape parameter table for the FloppyTape drive. After system reset, drive 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 "Initiialize FloppyTape Parameters" command before accessing the tape for the first time after any 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

4.4 IDTC STATUS MESSAGES

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 returns 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), and "SOLICITED STATUS" ($72).
6-7	$xxxx	Byte 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 will contain 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 all SASI/SCSI controller error messages that result in the described IDTC status message. For details of the SASI/SCSI protocol, refer to Paragraph 2.5.3.

4.4.1 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 Status 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#, z=drive#)
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:

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

4.4.2 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 Status 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#, z=drive#)
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

4.4.3 Other Commands Complete

The "Other Commands Complete" message indicates the successful completion of other commands, such as format, restore, seek, verify, and initialize.

IDTC Status 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#, z=drive#)
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:

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

4.4.4 IDTC Self-Test Complete

The "IDTC Self-Test Complete" message returns the results of the "Request IDTC Self-Test" command execution (see Paragraph 4.3.1.2). For each part of the test (floppy disk/tape controller, hard disk controller) the message contains a bit in the status byte which will be set if the corresponding test failed.

IDTC Status 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

4.4.5 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 Status 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#, z=drive#)
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:

TYPE	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.

4.4.6 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 Status 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

4.4.7 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 5.25 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 Status 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#, z=drive#)
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:

TYPE	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.

4.4.8 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 (bytes 8-11).

IDTC Status 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#, z=drive#)
4	$xx	type of originating command: ($10=READ, $30=TEST)
5	$70	status type: COMPLETION
6-7	$0004	status code
8-11	$xxxxxxx	CBN of block containing the deleted data mark
12	$03	ETX mark

SASI/SCSI Controller Error:

TYPE	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).

4.4.9 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.

IDTC Status 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#, z=drive#)
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

4.4.10 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 Status 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#, z=drive#)
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:

TYPE	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.

4.4.11 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 Status 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#, z=drive#)
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:

TYPE	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.

4.4.12 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 Status 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#, z=drive#)
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:

TYPE	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.

4.4.13 Sector ID CRC/ECC Error

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

IDTC Status 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#, z=drive#)
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:

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

4.4.14 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 the time-out functions refer to Paragraphs 2.4.2 and 2.4.3.

IDTC Status 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#, z=drive#)
4	$xx	type of originating command ($10=READ, $20=WRITE)
5	$70	status type: COMPLETION
6-7	$000A	status code
8	$03	ETX mark

4.4.15 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 at all. As this might lead to fatal errors in the mass storage devices, the IDTC should be reset and reinitialized before executing further commands.

IDTC Status 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#, z=drive#)
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:

TYPE	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.

4.4.16 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 drive malfunction.

IDTC Status 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#, z=drive#)
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:

TYPE	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.

4.4.17 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 Status 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#, z=drive#)
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:

TYPE	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 with 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.

4.4.18 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.

IDTC Status 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#, z=drive#)
4	$xx	type of originating command ($10=READ, $20=WRITE)
5	$70	status type: COMPLETION
6-7	$0019	status code
8	$03	ETX mark

4.4.19 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 Status 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#, z=drive#)
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:

TYPE	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.

4.4.20 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 Status 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#, z=drive#)
4	$30	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:

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

CHAPTER 5

MAINTENANCE INFORMATION

5.1 INTRODUCTION

This chapter provides calibration instructions, parts list, assembly drawing and schematic diagrams for the MVME319 Intelligent Disk/Tape Controller.

5.2 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 recalibration becomes necessary, please contact your nearest Motorola Field Service Office.

The following calibration instructions are for service purposes only.

Figure 3.1 illustrates the locations of the test points and potentiometers.

- establish the shipment jumper configuration on the MVME319 module.
- install the MVME319 module in a VME/VERSAdos system.
- turn the VME system power on.
- connect an oscilloscope with testpoint TP1 (data separator oscillator).
- adjust potentiometer T1 for a frequency of 4 MHz at TP1.
- connect a VERSAdos hard disk with the MVME319 module.
- connect a floppy disk drive with the MVME319 module.
- connect an oscilloscope with testpoint TP2 (write precompenstion pulse).
- boot VERSAdos and log on.
- insert a scratch floppy disk into the floppy disk drive.
- initialize the parameters of the connected floppy disk.
- format the scratch floppy disk.
- 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)

5.3 PARTS LIST

Table 5.1 reflects the latest issue of hardware for the MVME319 module at the time of printing. All parts are identified by quantity, designation on the board, Motorola part number, and a short part description. The part locations are shown in Figure 5.1.

5.4 ASSEMBLY DRAWING, SCHEMATIC DIAGRAMS

Figure 5.1 is the assembly drawing, Figures 5.2 to 5.14 are the schematic diagrams of the MVME319 Intelligent Disk/Tape Controller.

Table 5.1: MVME319 Parts List

QU	DESIGNATION	PART NUMBER	DESCRIPTION
1	C47	21NW9702A40	27 pF, 50 V Ceramic Capacitor
1	C43	21NW9702A37	47 pF, 50 V Ceramic Capacitor
1	C49	21NW9709A05	100 pF, 50 V Ceramic Capacitor
1	C29	21NW9604A74	100 pF, 50 V Ceramic Capacitor
1	C27	21NW9604A75	330 pF, 50 V Ceramic Capacitor
2	C28,C45	21-G9632M01	0.01 uF, 50 V Ceramic Capacitor
55	C2-C13, C15-C26, C30-C41, C50-C67,C69	21NW9702A09	0.1 uF, 50 V Ceramic Capacitor
1	C42	23NW9704A97	0.33 uF, 16 V Electrolytic Capacitor
1	C68	23NW9704A96	1.0 uF, 16 V Electrolytic Capacitor
2	C44,C46	23NW9618A31	10 uF, 16 V Electrolytic Capacitor
1	C48	23NW9704A95	22 uF, 10 V Electrolytic Capacitor
1	C14	23NW9704A99	33 uF, 10 V Electrolytic Capacitor
1	C1	23NW9618A74	47 uF, 10 V Electrolytic Capacitor
2	D1,D2	48NW9616A03	1N4148 Rectifier
1	DS1	48NW9612A34	Light Emitting Diode Red
2	K7,K16	28NW9802D01	Header Single Row 1*2 Pins
2	K9,K14	28NW9802D04	Header Single Row 1*3 Pins
1	K10	28NW9802D56	Header Single Row 1*4 Pins
3	K1,K2,K3	28NW9802D57	Header Single Row 1*5 Pins
1	K11	28NW9802C29	Header Double Row 2*2 Pins
1	K8	28NW9802B21	Header Double Row 2*3 Pins
3	K6,K12,K13	28NW9802C43	Header Double Row 2*4 Pins
1	K15	28NW9802C36	Header Double Row 2*7 Pins
1	P1	28-G9802M03	DIN 41612 C 96 Male Connector
1	P2	28NW9802F59	Sub-D 25-pole Female Connector
1	P5	28NW9802F66	Dual Row 34-pole Male Connector
1	P4	28NW9802F67	Dual Row 50-pole Male Connector
1	P3	28NW9802F68	Dual Row 50-pole Male Connector
1	Q1	48NW9611A35	BC307B PNP Transistor
1	R21		0 ohm Dummy Resistor
1	R17	06SW-961B43	270 ohm, 0.25 W, 1% Film Resistor
1	R26	06SW-964A18	620 ohm, 0.25 W, 1% Film Resistor
1	R33	06SW-964A20	5.1 kohm, 0.25 W, 1% Film Resistor
1	R31	06SW-961D01	10 kohm, 0.25 W, 1% Film Resistor
1	R30	06SW-961D47	30 kohm, 0.25 W, 1% Film Resistor
1	R32	06SW-961D73	56 kohm, 0.25 W, 1% Film Resistor
1	R13	06SW-124A01	10 ohm, 0.25 W, 5% Carbon Resistor
1	R23	06SW-124A13	33 ohm, 0.25 W, 5% Carbon Resistor
1	R36	06SW-124A33	220 ohm, 0.25 W, 5% Carbon Resistor
1	R34	06SW-124A37	330 ohm, 0.25 W, 5% Carbon Resistor
1	R7	06SW-124A39	390 ohm, 0.25 W, 5% Carbon Resistor
2	R8,R35	06SW-124A43	560 ohm, 0.25 W, 5% Carbon Resistor
4	R9-R12	06SW-124A45	680 ohm, 0.25 W, 5% Carbon Resistor
1	R14	06SW-124A51	1.2 kohm, 0.25 W, 5% Carbon Resistor
3	R1,R3,R5	06SW-124A59	2.7 kohm, 0.25 W, 5% Carbon Resistor
5	R6,R15,R16, R18,R19	06SW-124A65	4.7 kohm, 0.25 W, 5% Carbon Resistor
4	R2,R39-R41	06SW-124A73	10 kohm, 0.25 W, 5% Carbon Resistor
1	R22	06SW-124A89	47 kohm, 0.25 W, 5% Carbon Resistor

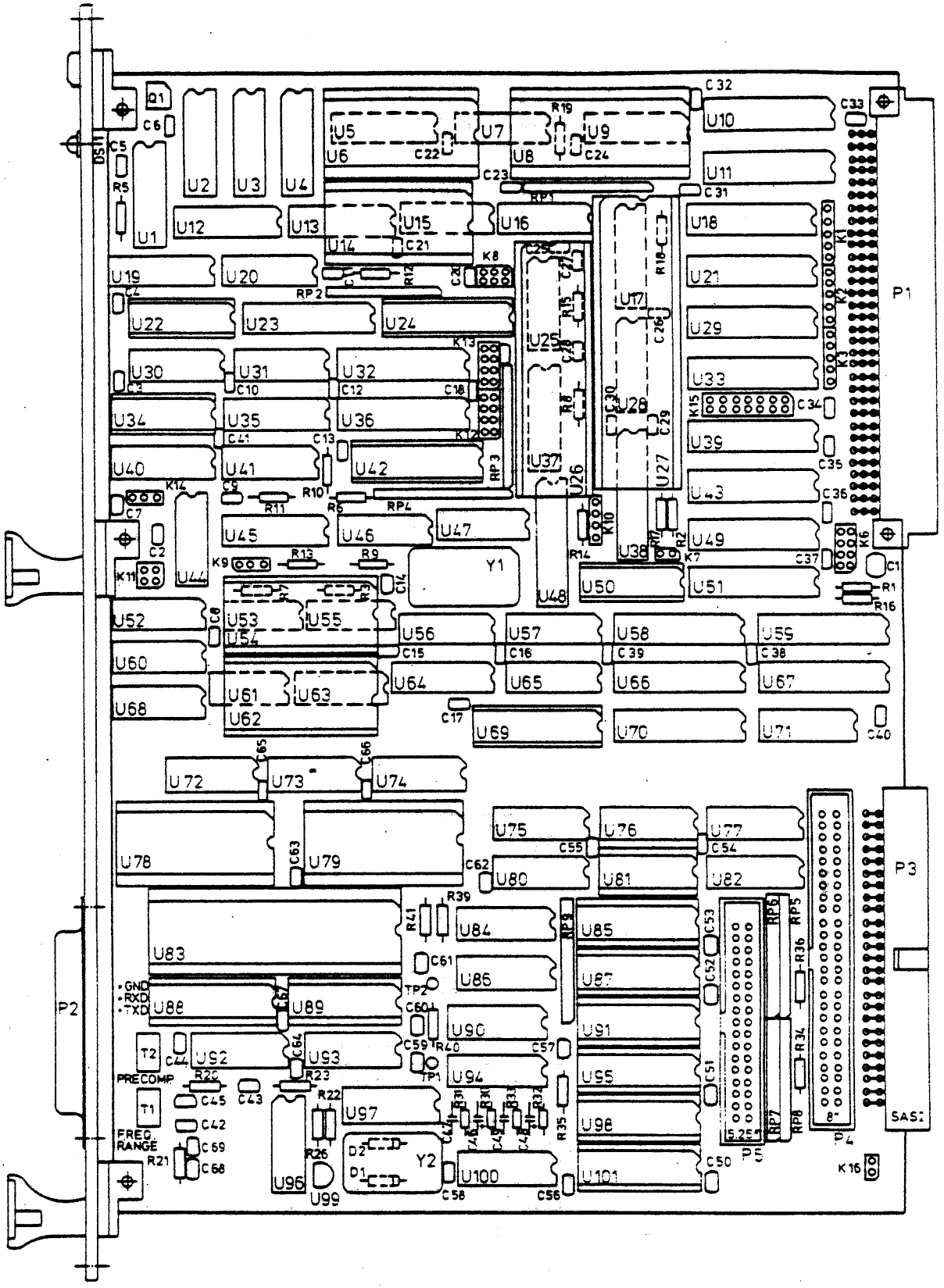
Table 5.1: MVME319 Parts List (continued)

QU	DESIGNATION	PART NUMBER	DESCRIPTION
1	R20	06SW-124A95	82 kohm, 0.25 W, 5% Carbon Resistor
2	RP5,RP7	51NW9626B45	9*220 ohm, 5% SIL Resistor Network
2	RP6,RP8	51NW9626B23	9*330 ohm, 5% SIL Resistor Network
3	RP2,RP3,RP9	51NW9626B41	9*1.2 kohm, 5% SIL Resistor Network
2	RP1,RP4	51NW9626A41	9*4.7 kohm, 5% SIL Resistor Network
2	T1,T2	18N..9603A47	4.7 kohm Variable Resistor
1	U48	51NW9615N39	AM25LS241P Octal Bus Driver TS
1	U26	51NW9615M95	AM9517A DMA Controller
1	U8	51NW9615M94	AM9519A Interrupt Controller
1	U83	51NW9615N35	FD1793B02 Floppy Disk Controller
4	U54,U62,U78 U79	51NW9615M85	HM6264P15 RAM 8K*8, Static
1	U50	51-G5011M01	HM7621A-5 PROM 512*4, Programmed
1	U6	51-G5018M05	HN482764G EPROM 8K*8, Programmed
1	U14	51-G5018M06	HN482764G EPROM 8K*8, Programmed
1	U9	51NW9615G49	MC14521B 24-Stage Counter
1	U27	51NW9615L08	MC68121L1 Intell. Periph. Controller
1	U99	51NW9615D12	MC78L05C +5V Voltage Regulator
1	U24	51-G5010M01	PAL16L8-2 PAL, Programmed
1	U42	51-G5010M02	PAL16L8-2 PAL, Programmed
1	U69	51-G5010M03	PAL16L8-2 PAL, Programmed
1	U41	51NW9615N10	SN74ALS00N Quad 2-Input NAND Gate
1	U55	51NW9615N11	SN74ALS08N Quad 2-Input AND Gate
1	U28	51NW9615N12	SN74ALS10N Triple 3-Input NAND Gate
4	U10,U32,U58, U67	51NW9615R30	SN74AS373N Octal Latch TS
3	U85,U91,U101	51NW9615R27	SN74AS533N Octal Latch Inverting TS
1	U56	51NW9615E91	SN74LS00N Quad 2-Input NAND Gate
3	U76,U82,U96	51NW9615C20	SN74LS02N Quad 2-Input NOR Gate
4	U44,U52,U60, U77	51NW9615C21	SN74LS04N Hex Inverter
3	U25,U31,U80	51NW9615C22	SN74LS08N Quad 2-Input AND Gate
3	U7,U16,U71	51NW9615E93	SN74LS14N Hex Schmitt-Trigger Inverter
1	U75	51NW9615E77	SN74LS27N Triple 3-Input NOR Gate
3	U30,U63,U84	51NW9615C24	SN74LS32N Quad 2-Input OR Gate
4	U72,U74,U97, U100	51NW9615C25	SN74LS74AN Dual D-Flip-Flop
1	U86	51NW9615F01	SN74LS86N Quad 2-Input EXOR Gate
1	U64	51NW9615H92	SN74LS112N Dual JK-Flip-Flop
1	U53	51NW9615N09	SN74LS113AN Dual JK-Flip-Flop
2	U90,U94	51NW9615C26	SN74LS123N Dual Monostable Flip-Flop
1	U3	51NW9615G39	SN74LS133N 13-Input NAND Gate
1	U40	51NW9615C69	SN74LS138N 3-Bit Binary Decoder
2	U17,U37	51NW9615C29	SN74LS174N Hex D-Flip-Flop
4	U1,U2,U12, U19	51NW9615F93	SN74LS191N Binary Up/Down-Counter
2	U33,U36	51NW9615F02	SN74LS244N Octal Bus Driver TS
4	U18,U21,U49,	51NW9615E96	SN74LS245N Octal Bus Transceiver TS
6	U4,U5,U13, U45,U73,U93	51NW9615G19	SN74LS257N Quad 2-Input Multiplexer
1	U15	51NW9615F09	SN74LS266N Quad 2-Input EXNOR Gate OC
1	U35	51NW9615J50	SN74LS367N Hex Bus Driver TS

Table 5.1: MVME319 Parts List (continued)

QU	DESIGNATION	PART NUMBER	DESCRIPTION
4	U23,U29,U38, U70	51NW9615E98	SN74LS373N Octal D-Latch TS
1	U98	51NW9615E99	SN74LS374N Octal D-Flip-Flop TS
1	U95	51NW9615J21	SN74LS534N Octal D-Flip-Flop Inv. OC
1	U92	51NW9615N34	SN74LS628N Voltage Contr. Oscillator
3	U11,U39,U51	51NW9615H89	SN74LS645-1N Octal Bus Transceiver TS
1	U43	51NW9615H41	SN74LS682N 8-Bit Magnitude Comparator
1	U46	51NW9615F30	SN74S05N Hex Inverter OC
2	U47,U57	51NW9615F85	SN74S38N Quad 2-Input NAND Driver
2	U20,U68	51NW9615C95	SN74S74N Dual D-Flip-Flop
2	U61,U65	51NW9615D26	SN74S113N Dual JK-Flip-Flop
1	U87	51NW9615F79	SN74S240N Octal Inv. Bus Driver TS
1	U59	51NW9615F65	SN74S241N Octal Bus Driver TS
1	U22	51-G5009M01	TBP18S030 PROM 32*8, Programmed
1	U34	51-G5009M02	TBP18S030 PROM 32*8, Programmed
1	U81	51-G5009M03	TBP18S030 PROM 32*8, Programmed
1	U88	51NW9615L87	WD1691 Disk Contr. Support Logic
1	U89	51NW9615N36	WD2143-01 Four Phase Clock Generator
1	Y2	48NW9606A29	K1115A 4.0000 MHz Crystal Osc.
1	Y1	48NW9606A51	K1115A 9.8304 MHZ Crystal Osc.
2	at Y1,Y2	09NW9811A46	4-Pin Oscillator Socket
1	at U92	09NW9811A02	14-Pin DIL IC Socket
4	at U22,U34, U50,U81	09NW9811A04	16-Pin DIL IC Socket
9	at U24,U42, U69,U85,U87, U91,U95,U98, U101	09NW9811A27	20-Pin DIL IC Socket
2	at U78,U79	09NW9811A21	28-Pin DIL IC Socket
1	at U83	09NW9811A22	40-Pin DIL IC Socket
10	at U6,U8, U14,U54,U62	09NW9811A75	14-Pin Socket Strip
2	at U26	09NW9811A76	20-Pin Socket Strip
2	at U27	09NW9811A74 +09NW9811A77	24-Pin Socket Strip (3-Pin + 21-Pin)
2	at P2	03SW992D306	DIN 84 M 3*6 Flat Head Screw
5	at P1, Front Panel	03SW993D210	DIN 84 M 2.5*10 Flat Head Screw
5	at P1, Front Panel	02SW990D001	DIN 934 M 2.5 Hexagonal Nut
16		29NW9805B17	Jumper Shorting Insulated
1		01-G3013M11	MVME319 Printed Circuit Board
1		64-G4079M01	MVME319 Front Panel

Figure 5.1: MVME319 Assembly Drawing



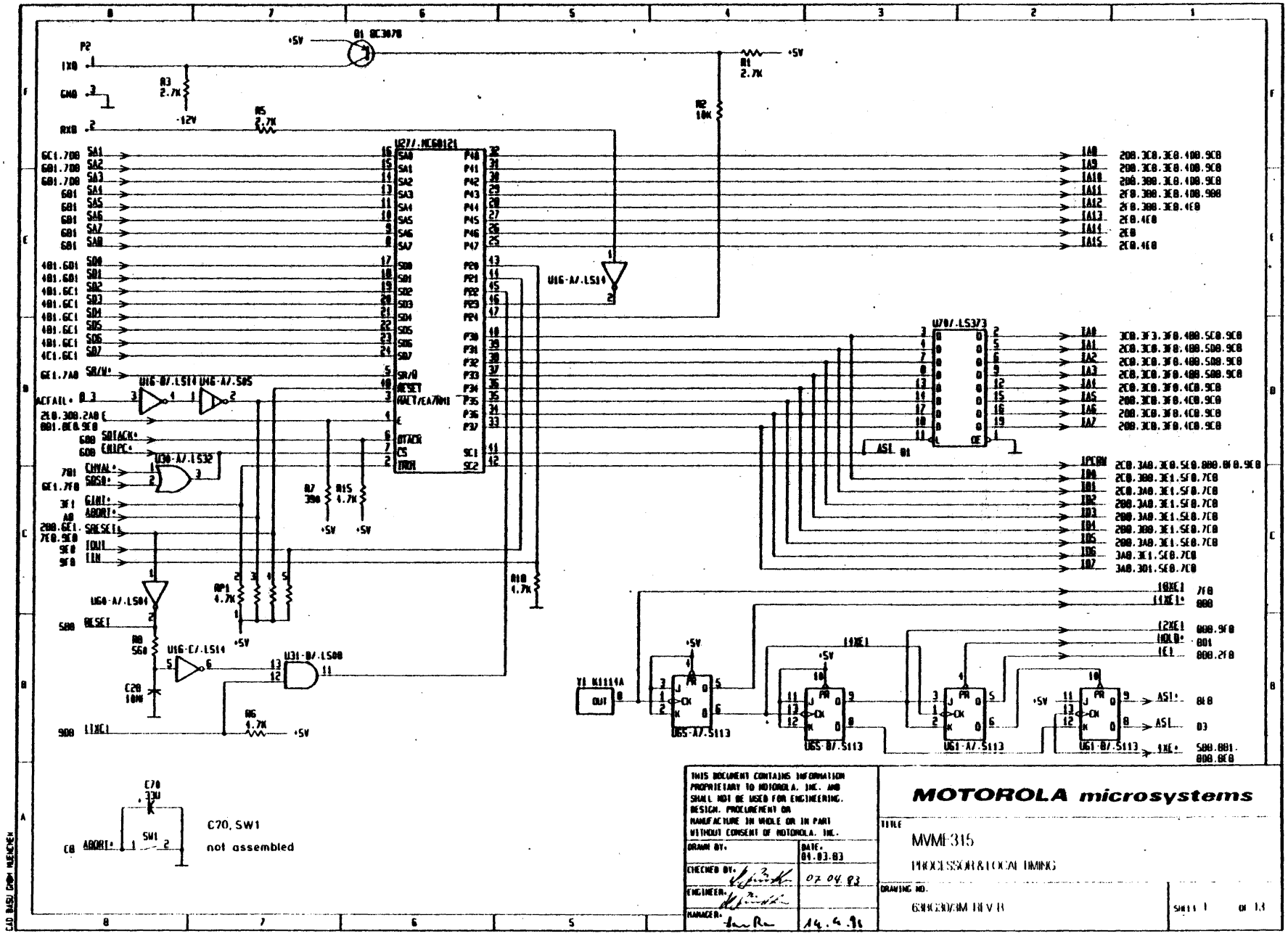


Figure 5.2: MVME319 Schematic Diagram Sheet 1/13

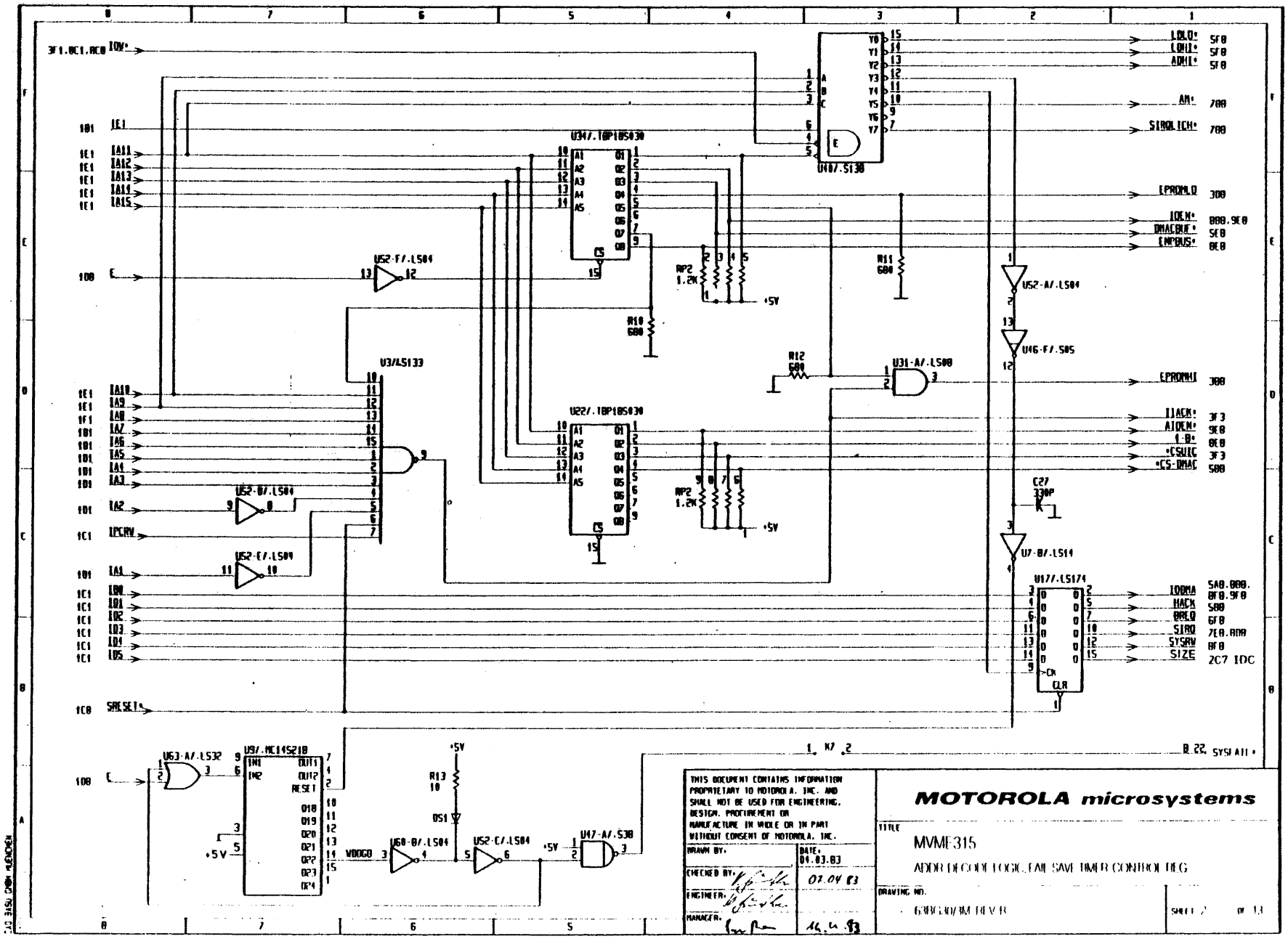


Figure 5.3: MVME319 Schematic Diagram Sheet 2/13

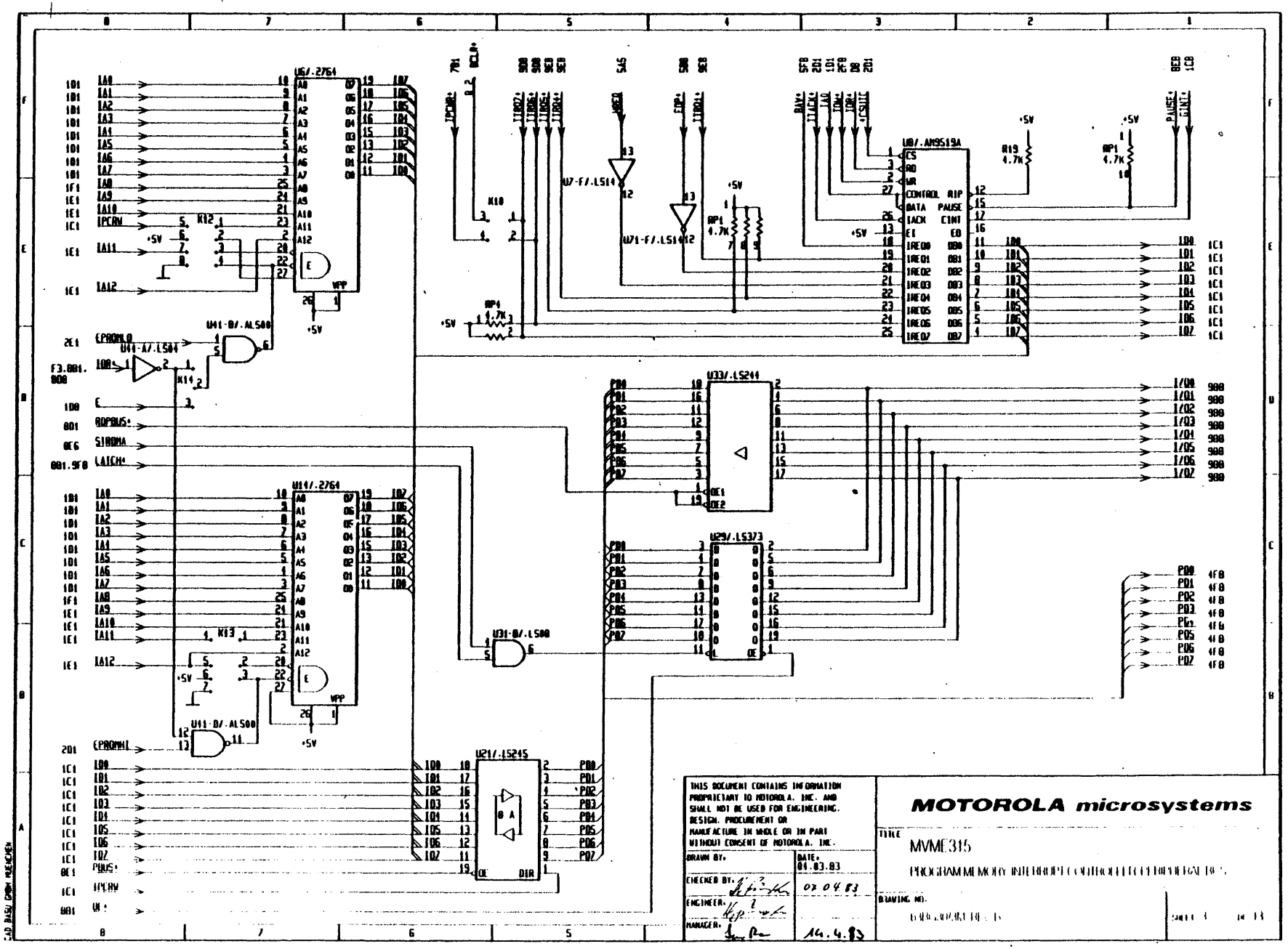


Figure 5.4: MVME319 Schematic Diagram Sheet 3/13

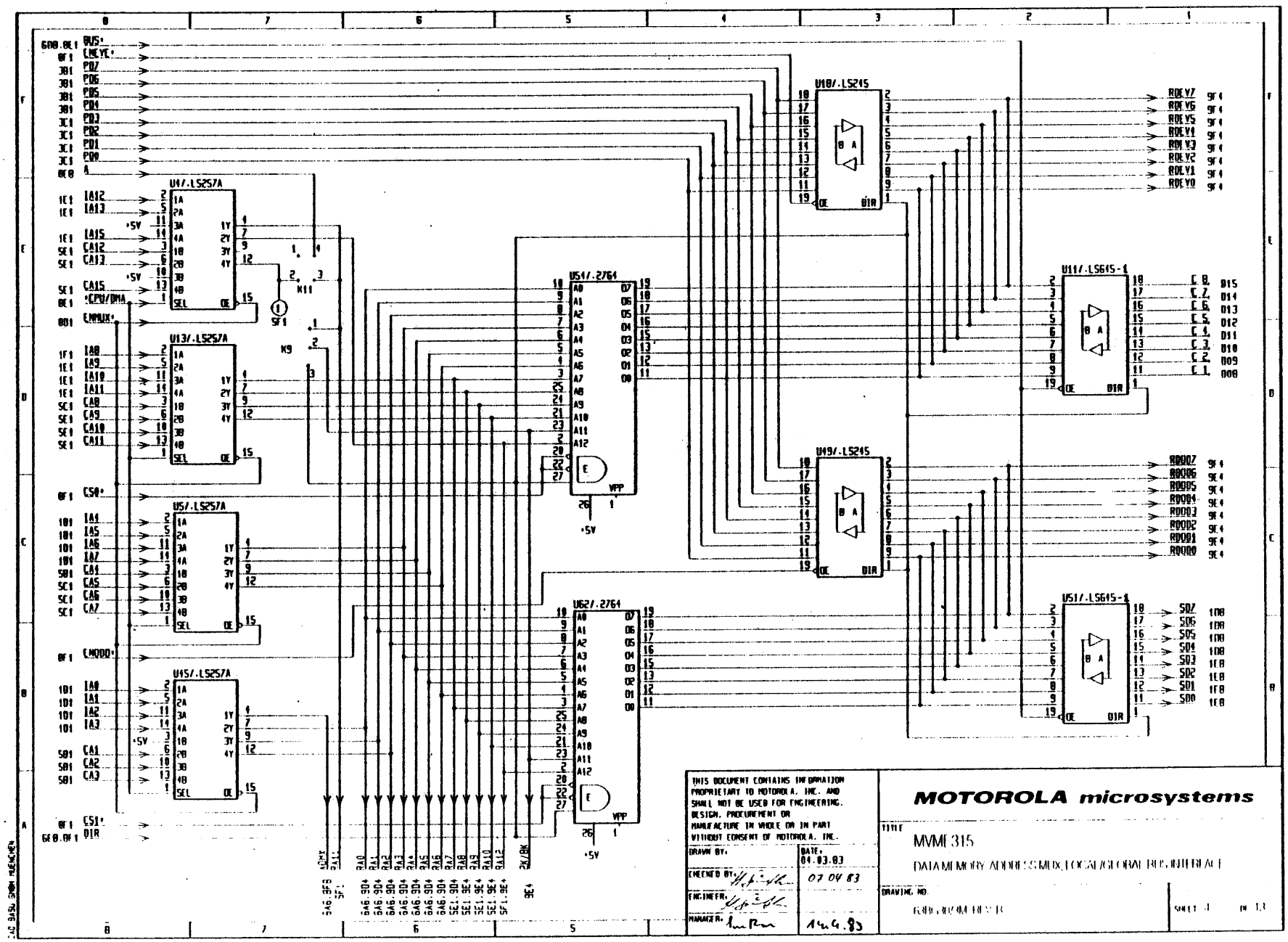


Figure 5.5: MVME319 Schematic Diagram Sheet 4/13

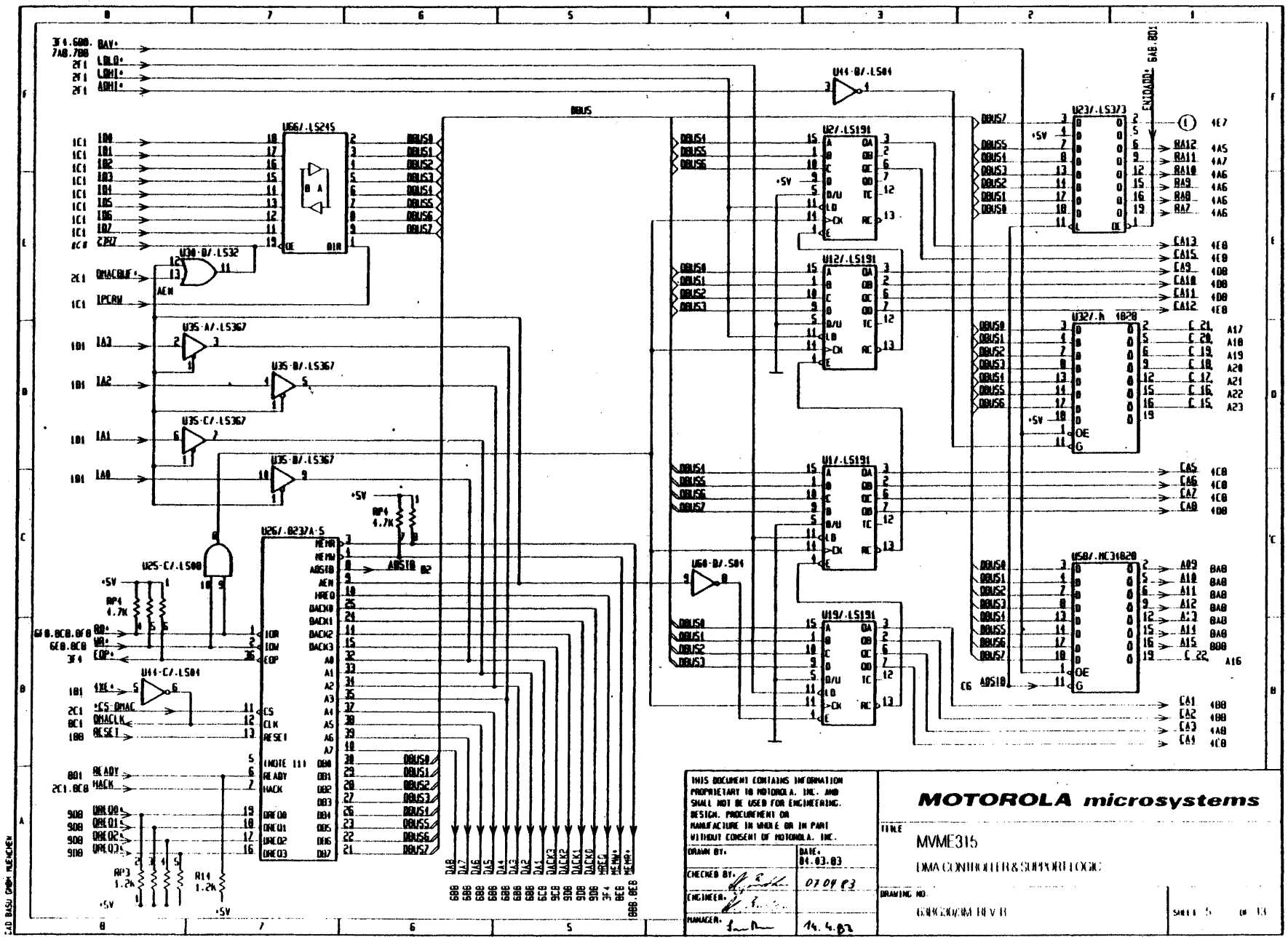


Figure 5.6: MVME319 Schematic Diagram Sheet 5/13

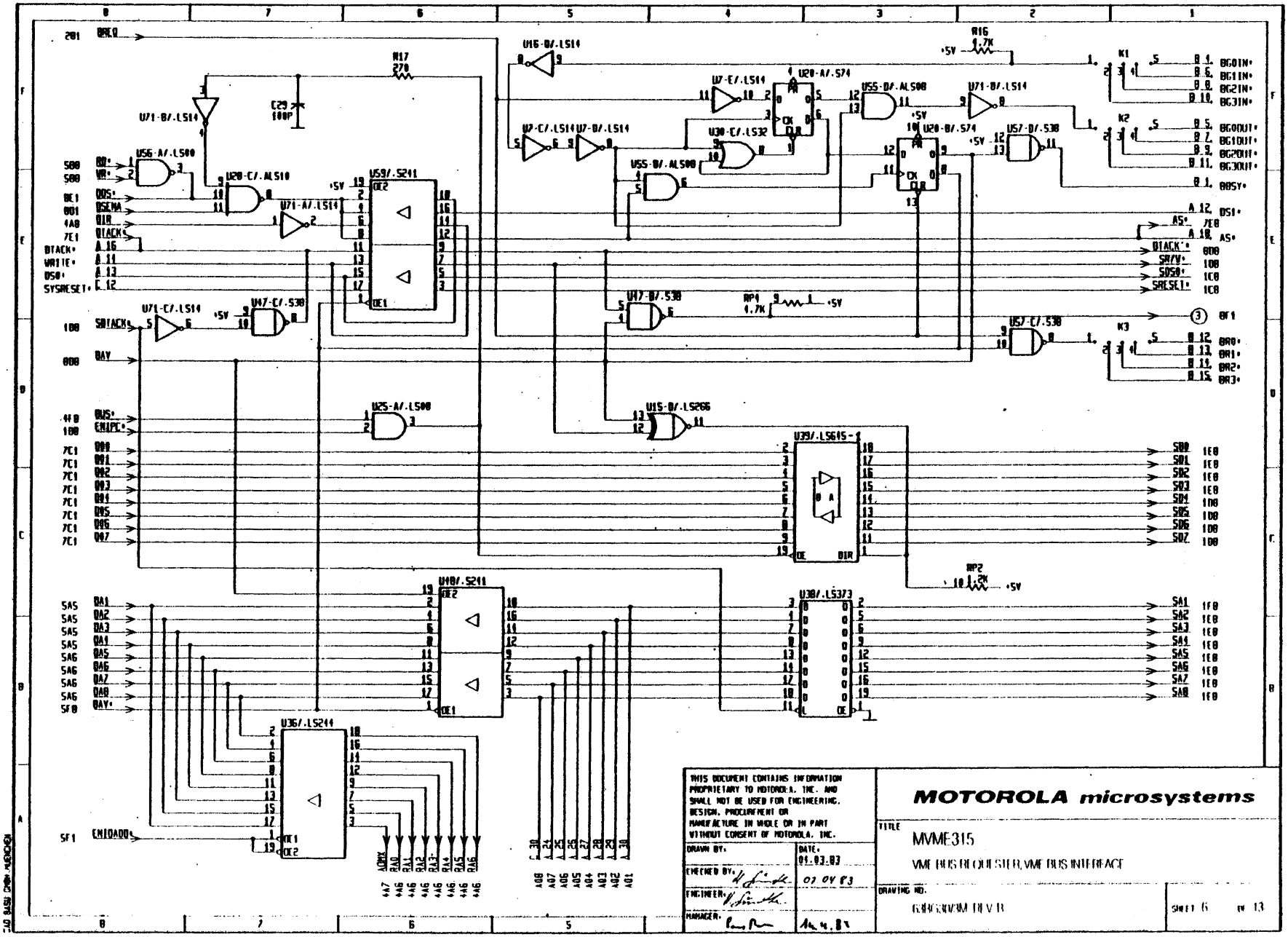


Figure 5.7: MVME319 Schematic Diagram Sheet 6/13

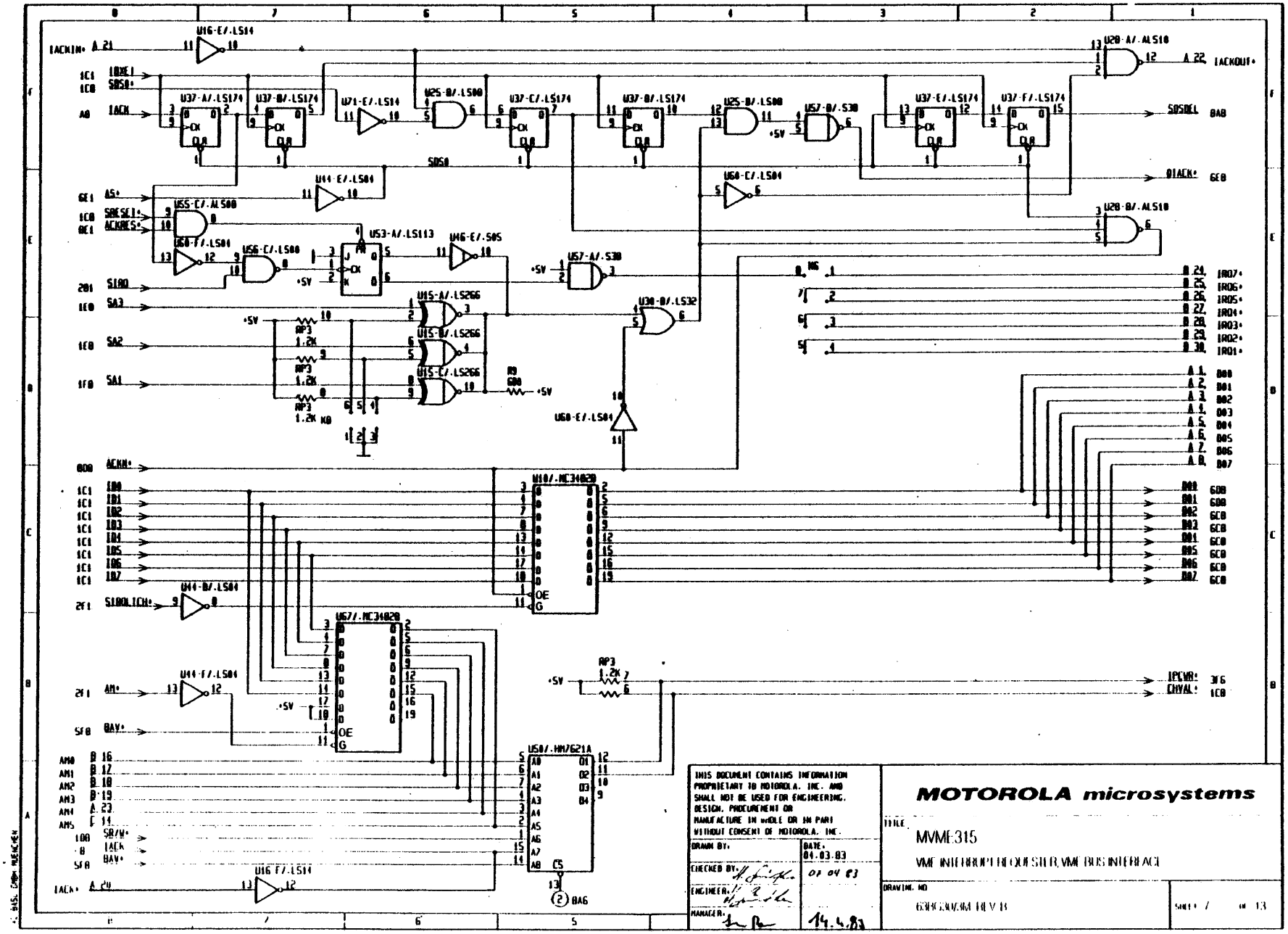


Figure 5.8: MVME319 Schematic Diagram Sheet 7/13

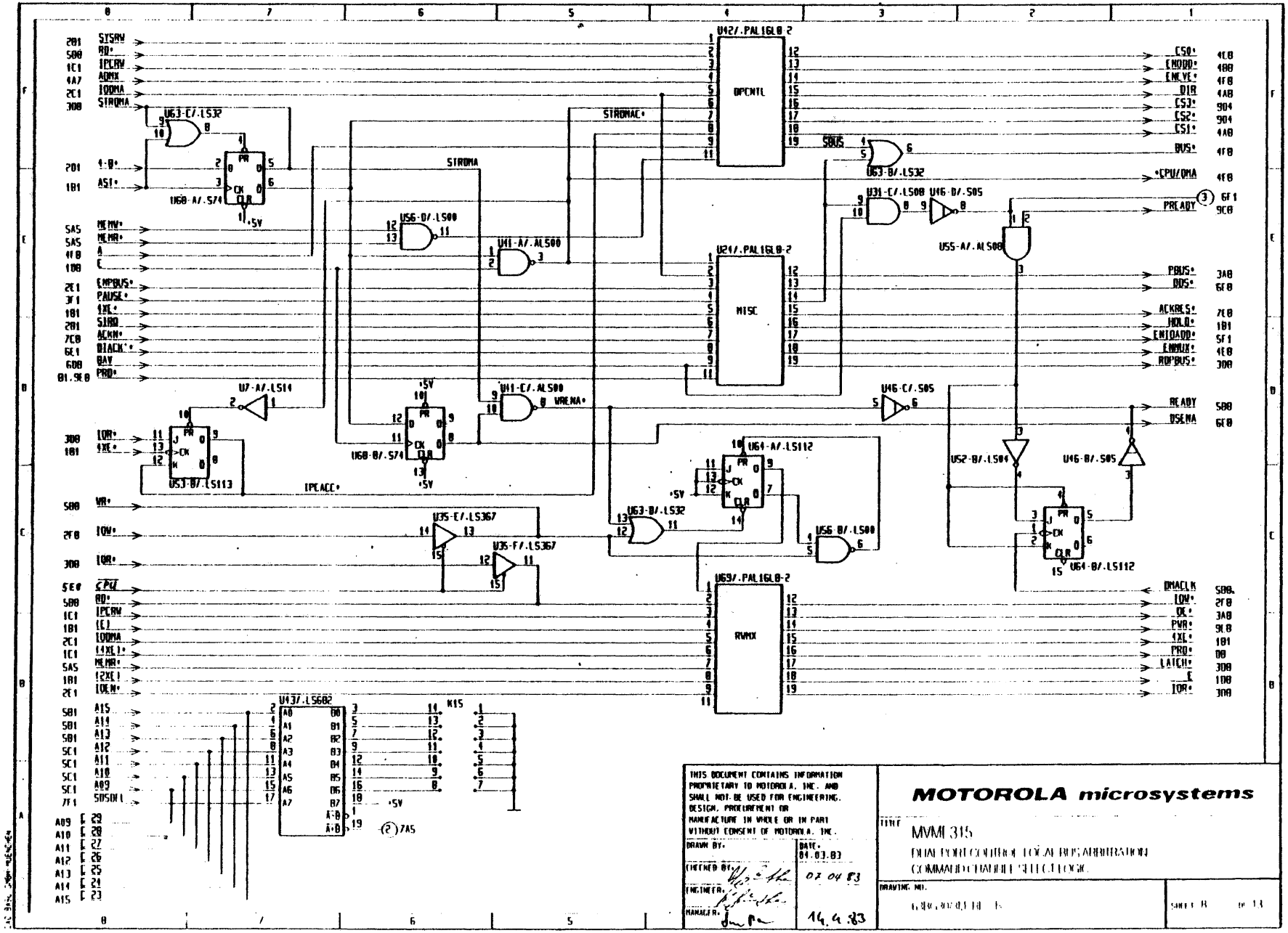


Figure 5.9: MVME319 Schematic Diagram Sheet 8/13

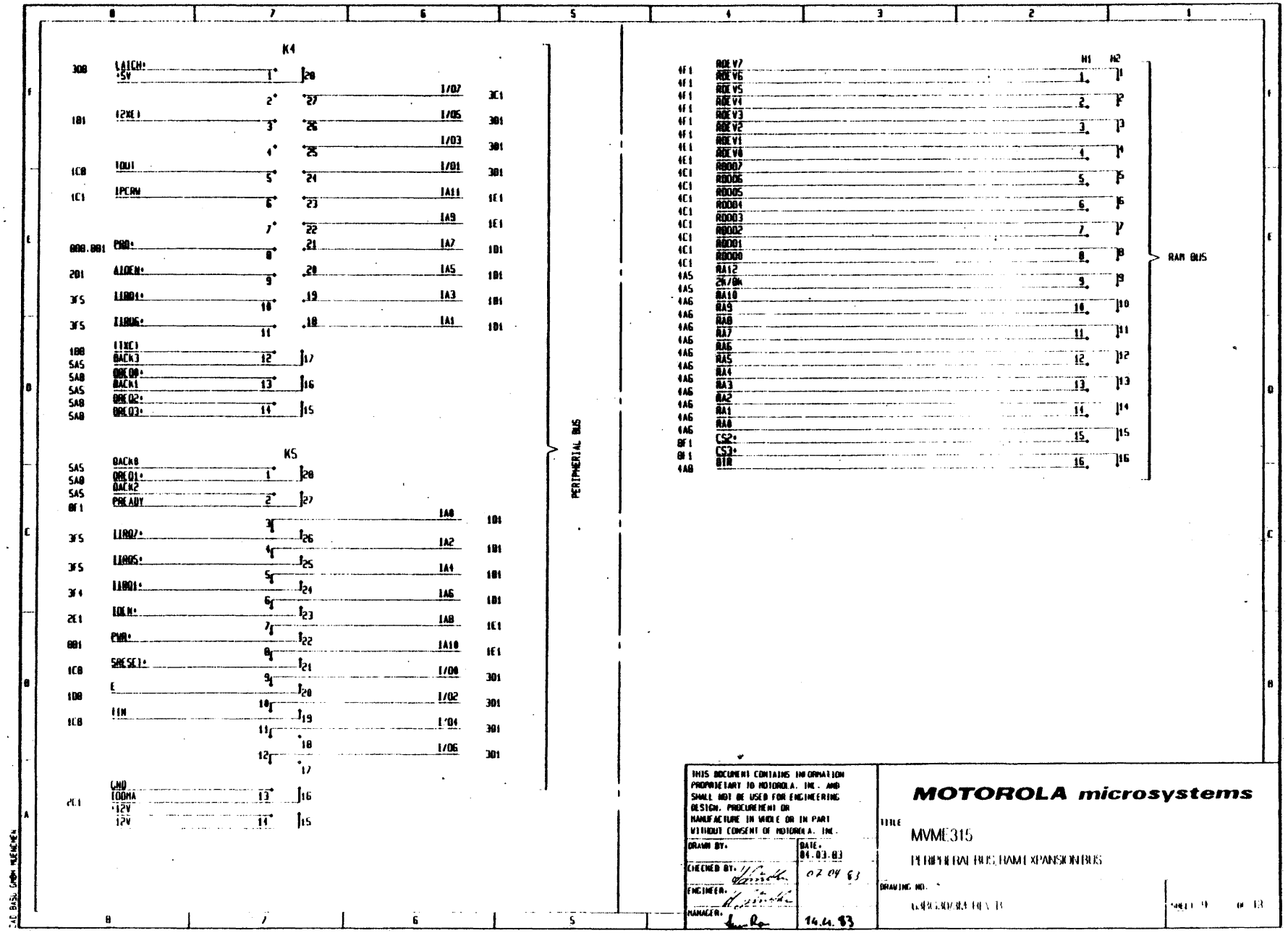
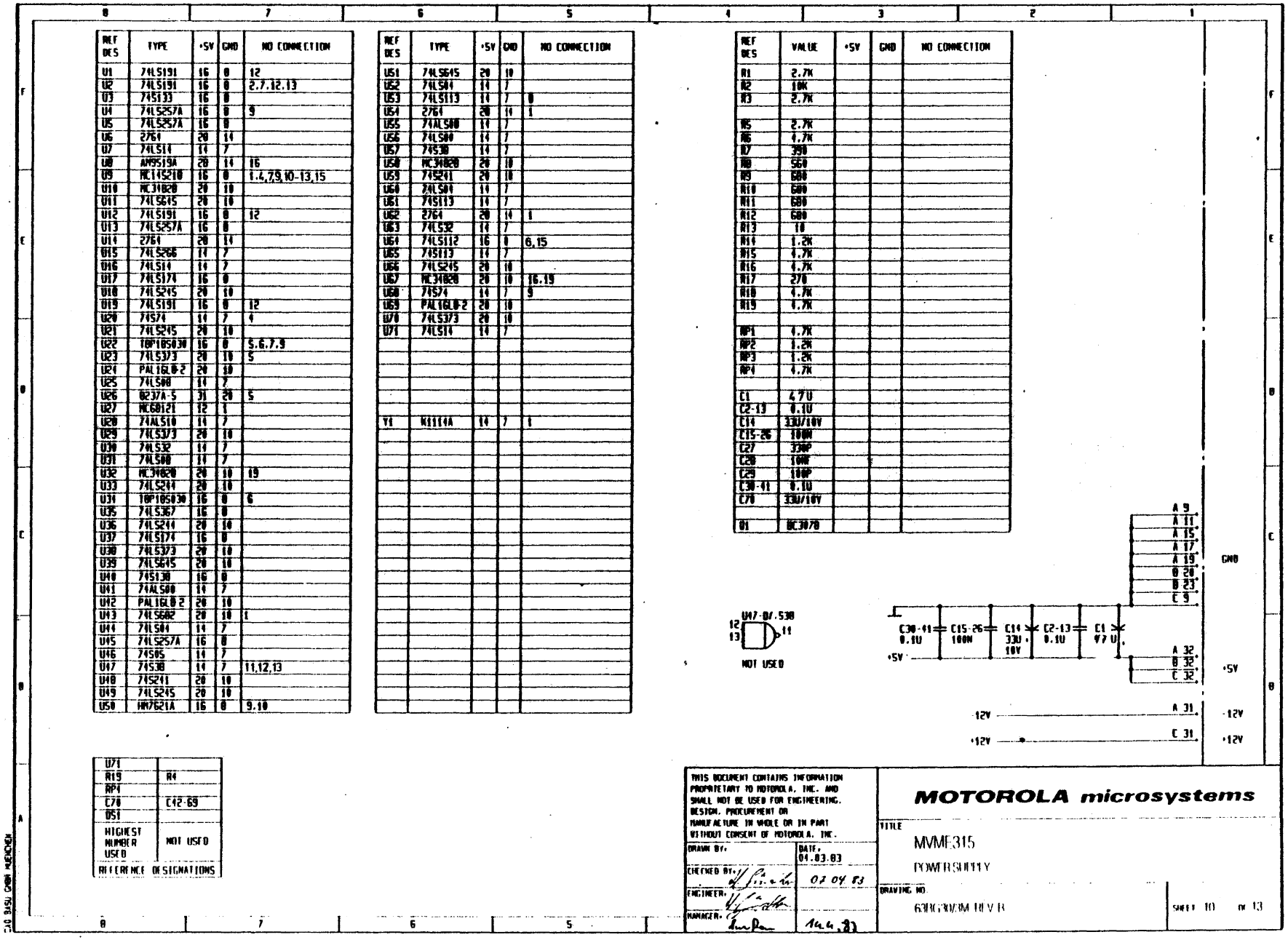


Figure 5.10: MVME319 Schematic Diagram Sheet 9/13



REF DES	TYPE	+5V	GND	NO CONNECTION
U1	74LS191	16	8	12
U2	74LS191	16	8	2.7.12.13
U3	745133	16	8	
U4	74LS257A	16	8	9
U5	74LS257A	16	8	
U6	2764	28	14	
U7	74LS14	14	7	
U8	AM9519A	28	14	16
U9	MC145218	16	8	1.4.7.9.10-13.15
U10	MC34020	20	10	
U11	74LS645	20	10	
U12	74LS191	16	8	12
U13	74LS257A	16	8	
U14	2764	28	14	
U15	74LS266	14	7	
U16	74LS14	14	7	
U17	74LS174	16	8	
U18	74LS245	20	10	
U19	74LS191	16	8	12
U20	74S74	14	7	4
U21	74LS245	20	10	
U22	TBP105030	16	8	5.6.7.9
U23	74LS373	20	10	5
U24	PAL16L8-2	20	10	
U25	74LS08	14	7	
U26	8237A-5	31	20	5
U27	MC68121	12	1	
U28	74ALS10	14	7	
U29	74LS373	20	10	
U30	74LS32	14	7	
U31	74LS08	14	7	
U32	MC34020	20	10	19
U33	74LS244	20	10	
U34	TBP105030	16	8	6
U35	74LS367	16	8	
U36	74LS244	20	10	
U37	74LS174	16	8	
U38	74LS373	20	10	
U39	74LS645	20	10	
U40	745138	16	8	
U41	74ALS08	14	7	
U42	PAL16L8-2	20	10	
U43	74LS682	20	10	1
U44	74LS04	14	7	
U45	74LS257A	16	8	
U46	74S05	14	7	
U47	74S38	14	7	11.12.13
U48	745241	20	10	
U49	74LS245	20	10	
U50	AM7621A	16	8	9.10

REF DES	TYPE	+5V	GND	NO CONNECTION
U51	74LS645	20	10	
U52	74LS04	14	7	
U53	74LS113	14	7	8
U54	2764	28	14	1
U55	74ALS08	14	7	
U56	74LS00	14	7	
U57	74S30	14	7	
U58	MC34020	20	10	
U59	745241	20	10	
U60	74LS04	14	7	
U61	745113	14	7	
U62	2764	28	14	1
U63	74LS32	14	7	
U64	74LS112	16	8	6.15
U65	745113	14	7	
U66	74LS245	20	10	
U67	MC34020	20	10	16.19
U68	74S74	14	7	9
U69	PAL16L8-2	20	10	
U70	74LS373	20	10	
U71	74LS14	14	7	
Y1	K1114A	14	7	1

REF DES	VALUE	+5V	GND	NO CONNECTION
R1	2.7K			
R2	10K			
R3	2.7K			
R5	2.7K			
R6	4.7K			
R7	390			
R8	560			
R9	680			
R10	680			
R11	680			
R12	680			
R13	10			
R14	1.2K			
R15	4.7K			
R16	4.7K			
R17	270			
R18	4.7K			
R19	4.7K			
RP1	4.7K			
RP2	1.2K			
RP3	1.2K			
RP4	4.7K			
C1	47U			
C2-13	0.1U			
C14	330/10V			
C15-26	100N			
C27	33NP			
C28	10NP			
C29	100P			
C30-41	0.1U			
C70	330/10V			
U1	MC3070			

U71	
R19	R4
RP4	
C70	C42-69
D51	
HIGHEST NUMBER USED	NOT USED

REFERENCE DESIGNATIONS

DRAWN BY:

CHECKED BY: 07 04 83

ENGINEER:

MANAGER: 14.4.83

DATE: 04.03.83

MOTOROLA microsystems

TITLE: MVME315 POWER SUPPLY

DRAWING NO: 68BG30E/M REV B

SHEET 10 OF 13

Figure 5.11: MVME319 Schematic Diagram Sheet 10/13

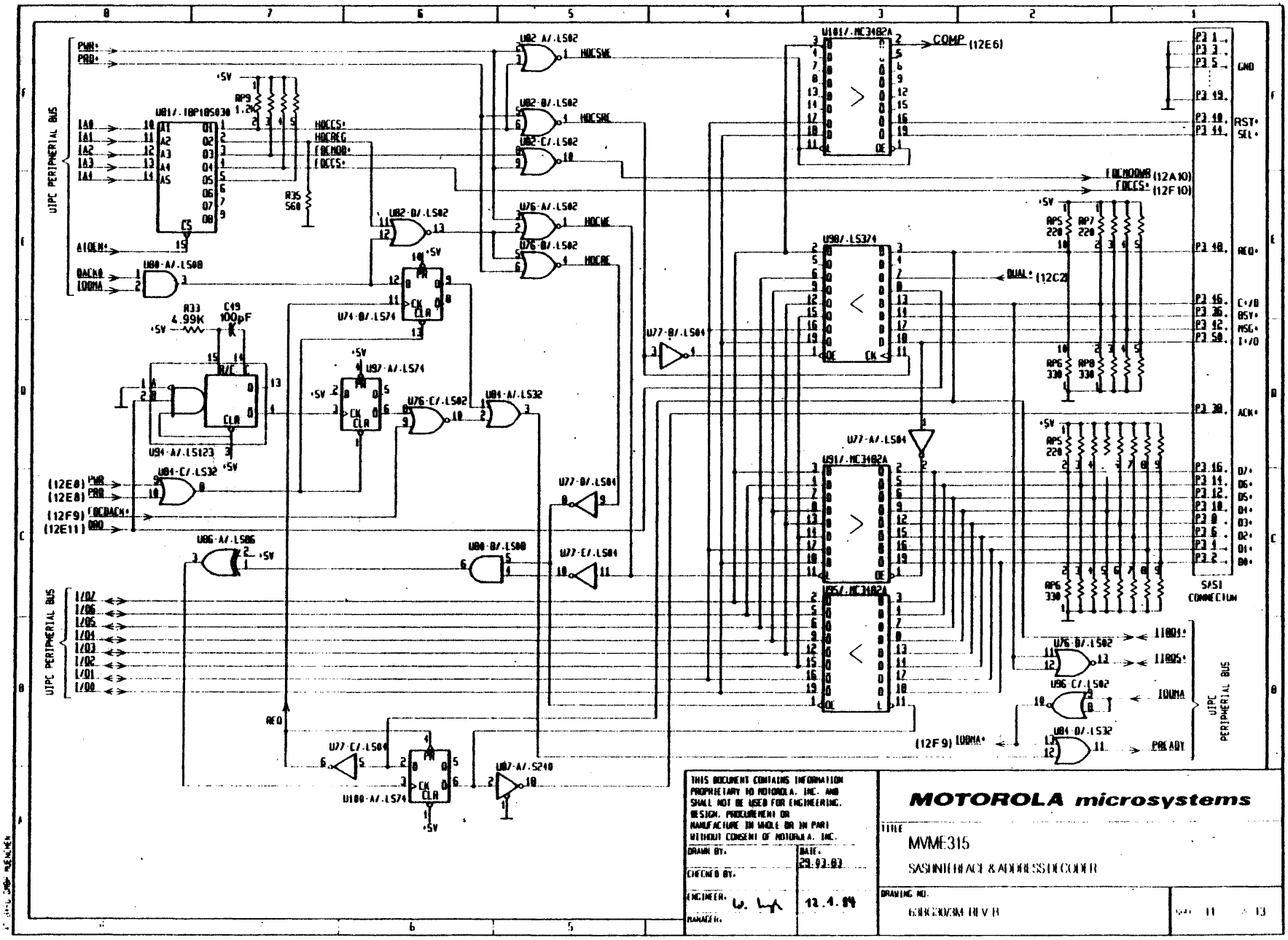
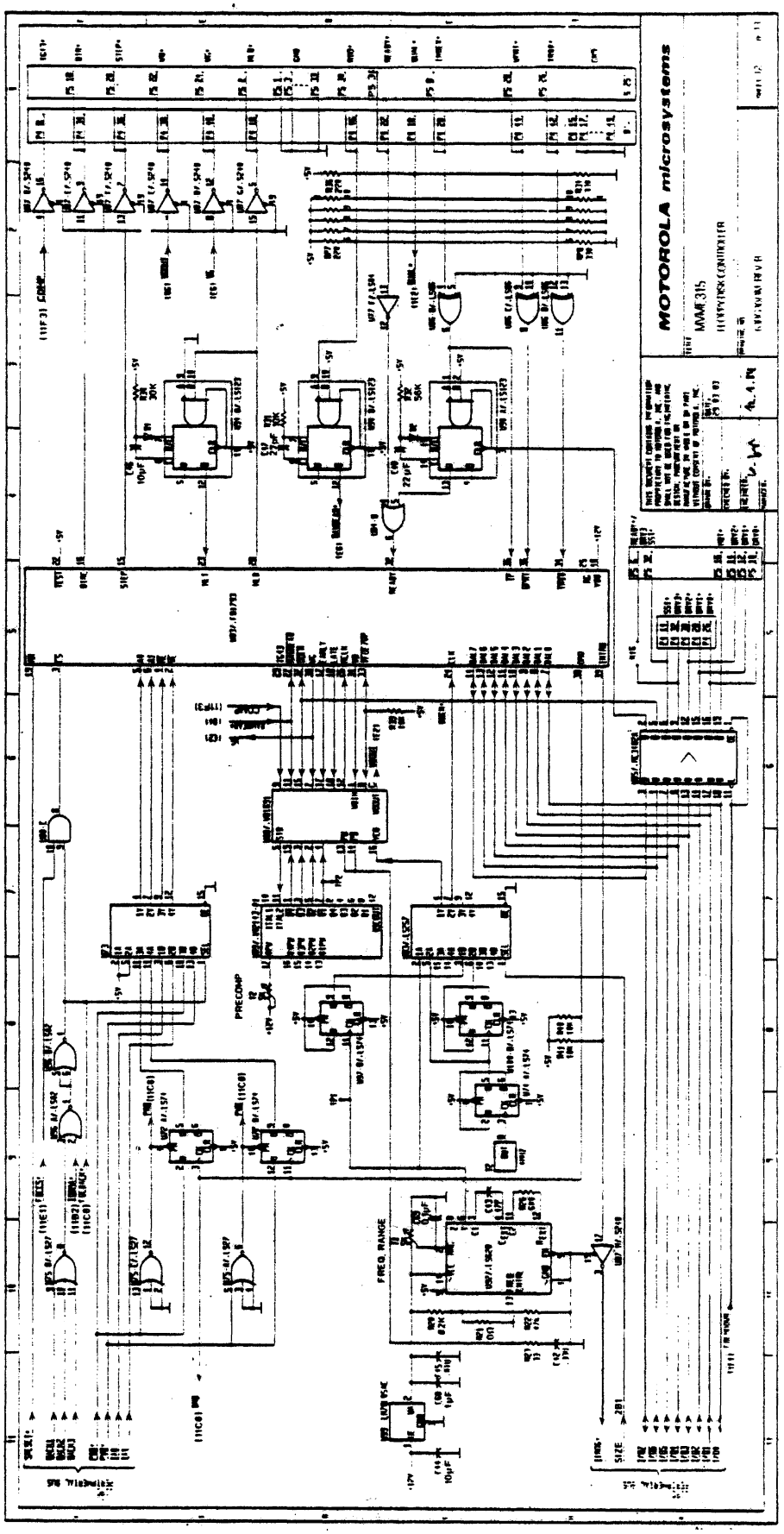
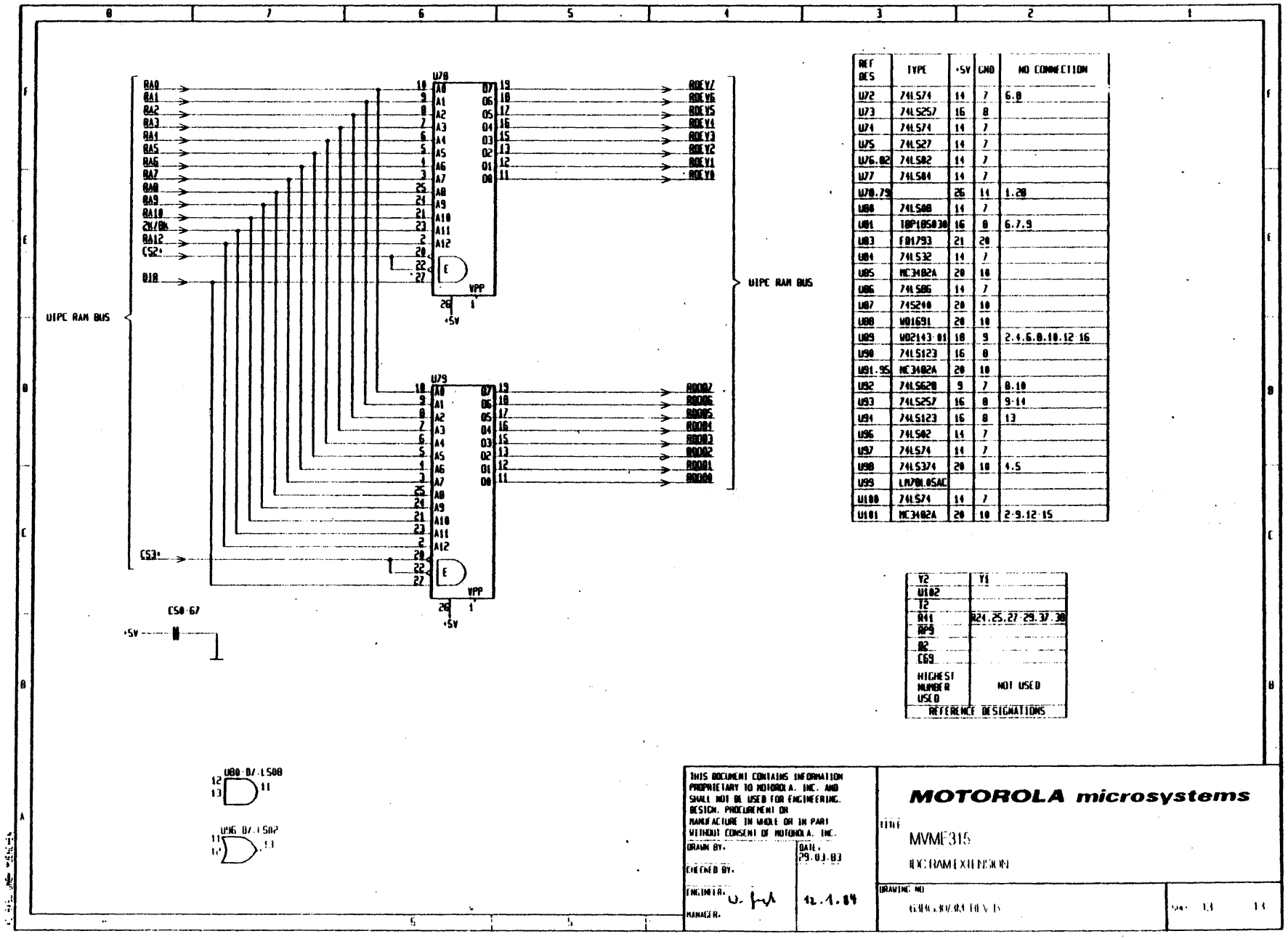


Figure 5.12: MVME319 Schematic Diagram Sheet 11/13

Figure 5.13: MVME319 Schematic Diagram Sheet 12/13





REF DES	TYPE	+5V	GND	NO CONNECTION
U72	74LS74	14	7	6,8
U73	74LS257	16	8	
U74	74LS74	14	7	
U75	74LS27	14	7	
U76,82	74LS02	14	7	
U77	74LS04	14	7	
U78,79		26	14	1,20
U80	74LS08	14	7	
U81	IDP1050-30	16	8	6,7,9
U83	FD1793	21	20	
U84	74LS32	14	7	
U85	MC3402A	20	10	
U86	74LS86	14	7	
U87	74S240	20	10	
U88	WD1691	20	10	
U89	WD2143-01	18	9	2,4,6,8,10,12,16
U90	74LS123	16	8	
U91-95	MC3402A	20	10	
U92	74LS620	9	7	8,10
U93	74LS257	16	8	9-14
U94	74LS123	16	8	13
U96	74LS02	14	7	
U97	74LS74	14	7	
U98	74LS374	20	10	4,5
U99	LM78L05AC			
U100	74LS74	14	7	
U101	MC3402A	20	10	2-9,12-15

Y2	Y1
U102	
T2	
R41	R24,25,27-29,37,38
AP3	
B2	
CR3	
HIGHEST NUMBER USED	NOT USED
REFERENCE DESIGNATIONS	

Figure 5.14: MVME319 Schematic Diagram Sheet 13/13

MVME319FW/L2
June 1985

M V M E 3 1 9 F I R M W A R E R E V . 1 . 0 0

C U S T O M E R L E T T E R

MOTOROLA INC.

1. INTRODUCTION

In addition to the MVME319 Intelligent Disk/Tape Controller User's Manual, this letter provides information related to the MVME319 module, the SYSTEM V/68 and VERSAdos drivers, and the supported mass storage devices.

The MVME319 module is shipped with Firmware Revision 1.00. The EPROM part numbers are 51-G5018M05 (in U6) and 51-G5018M06 (in U14).

2. RECOMMENDED FLOPPYTAPE PARAMETERS

The following configuration parameter values are recommended for the CIPHER 525 FloppyTape when used with the operating systems SYSTEM V/68 and VERSAdos:

PARAMETER	SYSTEM V/68	VERSAdos
Physical sector size in bytes	512	256
Number of sectors per segment	32	44
Segment skip factor	2	2
Block size in bytes	512	256

3. SUPPORTED HARD DISK CONTROLLERS AND DRIVES

The MVME319 supports the following hard disk controllers:

- XEBEC S1410 5.25 Inch Winchester Disk Controller
- XEBEC S1410A 5.25 Inch Winchester Disk Controller
- ADAPTEC ACB 4000 5 1/4" Winchester Disk Controller

The following 5.25 inch hard disk drives have been tested by Motorola with the MVME319 and are recommended for use:

- BASF	BASF 6188	(12 MByte)
- Computer Memories	CMI 5206, 5412, 5619	(5, 10, 15 MByte)
- Micropolis	M 1302, 1303, 1304	(20, 30, 40 MByte)
- Shugart	SA 706, 712	(5, 10 MByte)

4. MEDIA DEFECT HANDLING WITH ADAPTEC ACB 4000 REV.C

The ADAPTEC ACB 4000 Winchester Disk Controller is capable of handling disk drives with media defects. The defects are specified in a media defect list in the MVME319 command "Format Disk with Media Defect Handling" (see MVME319 User's Manual, section 4.3.2.7).

The ACB 4000 controller Rev.C contains a known bug which results in a misinterpretation of the defect location on the media. However, this problem can be overcome when the media defects, given as offsets relative to the index mark, are converted prior to being entered in the media defect list as follows:

$$BFI = \text{int} (ORGBFI * 1.00645 - 14)$$

where ORGBFI = original number of bytes from index supplied by the disk drive manufacturer,

and BFI = actual number of bytes from index entered in the media defect list.

The following example illustrates the correction:

Media defect location: cyl.= 178, head = 2, bytes from index = 6573

Defect conversion: BFI = int (6573 * 1.00645 - 14) = 6601

Entry in defect list: cyl.= 178, head = 2, bytes from index = 6601

Note that the described bug has been corrected by ADAPTEC and is no longer existent in the ACB 4000 controller Rev.H.

5. KNOWN MVME319 RESTRICTIONS AND PROBLEMS

The following restrictions and problems apply to the MVME319 module with Firmware Rev. 1.00:

- A media write protection is not indicated in the "Drive Status" message. Regardless of the media write protection, bit 5 of the status byte in the "Drive Status" message is always returned as %0 (see MVME319 User's Manual, section 4.4.20).

- A system power down with floppy disks located in closed floppy disk drives may destroy files unrecoverably, especially if separate disk drives with system independent power supply are used. Always take your floppy disks out of the drives prior to switching the system power off.

6. VERSAdos MVME315 DRIVER PATCH

The MVME319 Intelligent Disk/Tape Controller can be used with the MVME315 driver in a VERSAdos system. The VERSAdos MVME315 driver supports the MVME319 module with up to four floppy disks and one XEBEC S1410 or S1410A hard disk controller, connected with one or two identical hard disks. However note that neither the ADAPTEC ACB 4000 disk controller nor the CIPHER 525 FloppyTape are supported by the VERSAdos MVME315 driver.

The VERSAdos MVME315 driver was initially designed for early versions of the XEBEC S1410 controller which had a fixed stepping rate of 3 milliseconds. The driver can be patched to operate with different stepping rates when the S1410A controller or later versions of the S1410 controller shall be used.

The following procedure illustrates how the standard VME101.VERSADOS.SY file supplied with VERSAdos Release 4.4 is patched. Other VERSAdos system files for MVME101 based systems may be patched similarly. Refer to the SYSGEN and RMSGEN linkage listings for the according patch addresses.

```
=PATCH VME101.VERSADOS.SY
> O 16300                                  (start address of MVME315 driver)
> M 4F8;DI
000167F8  000004F8  1340002A      MOVE.B  D0,$002A(A1)
                                 >BRA.L   $14CE4
000167FC  000004FC  01C9002C      MOVEP.L D0,$002C(A1)
                                 >.

> O 14CE4                                  (start of RMS68K patch area)
> M 0;DI
00014CE4  00000000  00000000      OR.B    #0,D0
                                 >MOVE.B  D0,$2A(A1)
00014CE8  00000004  00000000      OR.B    #0,D0
                                 >MOVEP.W D0,$2C(A1)
00014CEC  00000008  00000000      OR.B    #0,D0
                                 >MOVE.B  #4,$30(A1)   (see note)
00014CF0  0000000C  00000000      OR.B    #0,D0
                                 >MOVE.B  D0,$32(A1)
00014CF4  00000010  00000000      OR.B    #0,D0
                                 >MOVEP.W D0,$34(A1)
00014CF8  00000014  00000000      OR.B    #0,D0
                                 >BRA.L   $16300+$504
00014CFC  00000018  00000000      OR.B    #0,D0
                                 >.

> Q
```

Note: This line specifies the actual code for the stepping rate:

```
  3 ms:  MOVE.B #0,$30(A1)
200 us:  MOVE.B #4,$30(A1)
 70 us:  MOVE.B #5,$30(A1)
 30 us:  MOVE.B #6,$30(A1)
 15 us:  MOVE.B #7,$30(A1)
```