



TECHNICAL MANUAL
FOR

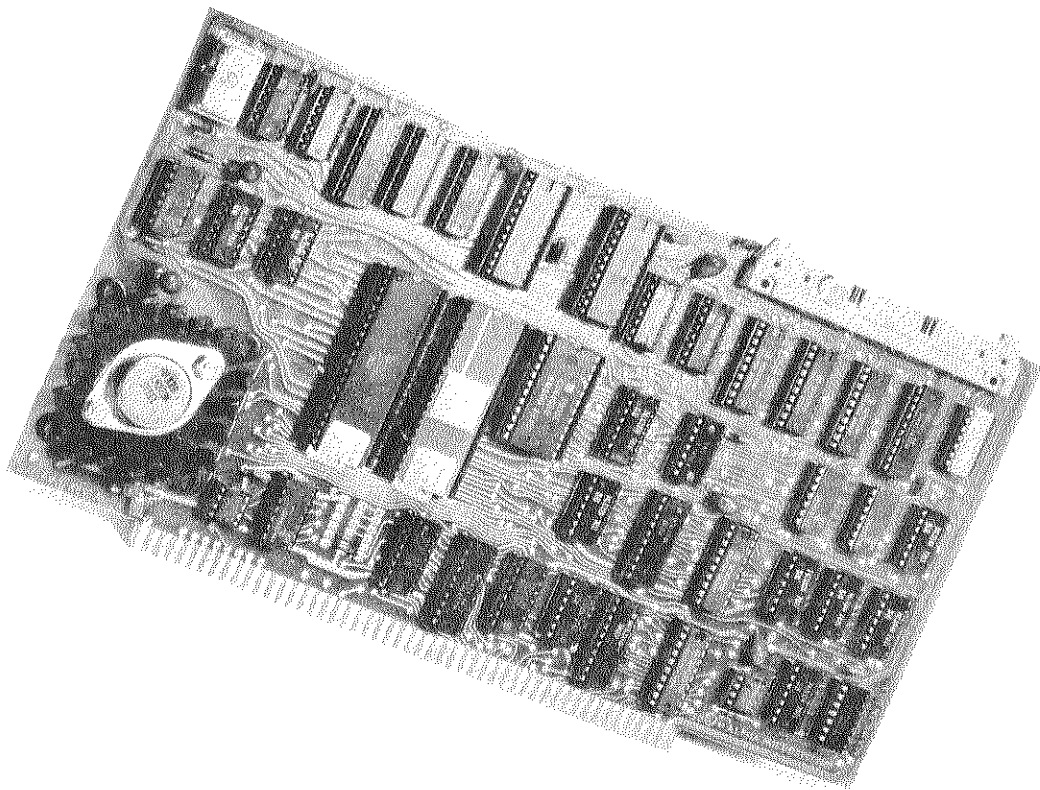
AM-500

HARD-DISK
CONTROLLER



TECHNICAL MANUAL
FOR

AM-500
HARD-DISK
CONTROLLER



Manufactured By
ALPHA MICROSYSTEMS
17881 SKY PARK NORTH
IRVINE, CALIFORNIA 92714

PROPRIETARY NOTICE

This document and the information herein disclosed is the proprietary property of ALPHA MICRO, 17881 Sky Park North, Irvine, California 92714. Any person or entity to whom this document is furnished or having possession thereof, by acceptance, assumes custody thereof and agrees that the document is given in confidence and will not be copied or reproduced in whole or in part, nor used or revealed to any person in any manner except to meet the purposes for which it was delivered. Additional rights and obligations regarding this document and its contents may be defined by a separate written agreement with ALPHA MICRO-SYSTEMS, and if so, such separate written agreement shall be controlling.

TABLE OF CONTENTS

Paragraph		Page
SECTION 1 GENERAL DESCRIPTION		
1.0	Introduction	1-1
1.1	Circuit Board Description	1-1
1.2	Application	1-1
SECTION 2 OPERATING DATA		
2.0	Introduction	2-1
2.1	Capabilities and Specifications	2-1
2.2	Interface Description and Wiring	2-3
2.2.1	S-100 Bus Interface	2-3
2.2.2	Disk Drive Interface	2-3
2.2.3	Wiring and Connections	2-6
2.3	User Options	2-6
2.3.1	Address Code	2-9
2.3.2	Interrupt Lines	2-9
2.3.3	Phantom Feature	2-10
2.3.4	Boot Options	2-10
2.3.4.1	The AM-500 as a Peripheral to the AM-200 or AM-210	2-11
2.3.4.2	Operating the AM-500 with the AM-400 Trident	2-11
SECTION 3 PROGRAMMING		
3.0	Introduction	3-1
3.1	Addressing	3-1
3.2	Initialization	3-1
3.3	Command Register	3-2
3.4	Status Register	3-2
3.5	Interrupt Mode	3-4
3.6	Handshake Sequence	3-4
3.7	Relkey Description	3-5

TABLE OF CONTENTS (Continued)

SECTION 3 PROGRAMMING (Continued)

3.8	DMA Buffer	3-6
3.8.1	DMA Buffer Operation for Disk Write	3-6
3.8.2	DMA Buffer Operation for Disk Read	3-10
3.9	Restore Command	3-10
3.10	Disk Formatting Commands	3-11
3.11	Status Check	3-11
3.12	Error Code Reporting	3-11

SECTION 4 FUNCTIONAL THEORY OF OPERATION

4.0	Introduction	4-1
4.1	Circuit Board Operation	4-1
4.1.1	Power-Up Reset	4-1
4.1.2	Addressing	4-13
4.1.3	CPU Output	4-14
4.1.3.1	Command Register	4-14
4.1.3.2	Command Types	4-15
4.1.3.3	Data Out Register	4-18
4.1.4	CPU Input	4-19
4.1.4.1	Status Register	4-19
4.1.4.2	Status Word Formats	4-19
4.1.4.3	Data In Register	4-20
4.1.5	Boot Load Procedure	4-21
4.1.6	Interrupts	4-21
4.1.7	Output to Disk Drive	4-22
4.1.7.1	Data Write Logic	4-22
4.1.7.2	Drive Registers	4-23
4.1.8	Input from Disk Drive	4-23
4.1.8.1	Data Read Logic	4-24
4.1.8.2	Drive Status Buffer	4-24

TABLE OF CONTENTS (Continued)

SECTION 4 FUNCTIONAL THEORY OF OPERATION (Continued)

4.1.9	FIFO Memory and Control	4-25
4.1.10	DMA Buffer	4-25
4.2	Circuit Module Description	4-26
4.2.1	CPU Microprocessor (U19)	4-26
4.2.1.1	Instruction Op-Code Fetch	4-31
4.2.1.2	Memory Read or Write Cycles	4-32
4.2.1.3	Input or Output Cycles	4-33
4.2.1.4	Interrupt Request/Acknowledge Cycle	4-33
4.2.1.5	CPU Instruction Set	4-34
4.2.2	DMA Controller (U20)	4-39
4.2.2.1	DMA Controller Functional Description	4-46
4.2.2.2	DMA Operation	4-47
4.2.2.3	Idle Cycle	4-48
4.2.2.4	Active Cycle	4-48
4.2.2.5	Transfer Types	4-50
4.2.2.6	Register Description	4-52
4.2.3	First-In First-Out (FIFO) Buffer Memory (U7, U8)	4-61
4.2.3.1	Functional Description	4-62
4.2.3.2	Input Register (Data Entry)	4-63
4.2.3.3	Transfer to the Fall-Through Stack	4-64
4.2.3.4	Output Register (Data Extraction)	4-65
4.2.3.5	Horizontal Expansion	4-67
4.2.4	Cyclic Redundancy Check (CRC) Generator/ Checker (U23)	4-72
4.2.5	16K (2K x 8) UV Erasable PROM (U21)	4-75
4.2.5.1	Erase Characteristics	4-75
4.2.5.2	Device Operation	4-76
4.2.6	1024 x 4 Bit Static Random Access Memory (U5, U6)	4-78

TABLE OF CONTENTS (Continued)

SECTION 4 FUNCTIONAL THEORY OF OPERATION (Continued)

4.2.7	Tri-State Buffers (U39)	4-79
4.2.8	Bus Comparator (U37)	4-80
4.2.9	Schmitt Trigger (U27).	4-81
4.2.10	D Positive-Edge-Triggered Flop-Flops with Preset and Clear (U2, U16, U25, U26, U36). . .	4-8
4.2.11	Tri-State Buffers (U14, U38)	4-83
4.2.12	Decoder (U10, U17)	4-84
4.2.13	Decoder/Demultiplexer (U22).	4-85
4.2.14	Quad 2-Line to 1-Line Data Selector/Multi- plexer (U9)	4-86
4.2.15	D-Type Transparent Latches (U42)	4-87
4.2.16	Dual 4-Bit Binary Counter (U45).	4-88
4.2.17	Tri-State D Flip-Flops (U4, U11, U12, U13, U30, U31, U41)	4-89

SECTION 5 MAINTENANCE AND TROUBLESHOOTING

5.0	Introduction	5-1
5.1	Circuit Board Checkout	5-1
5.2	Warranty Procedures	5-2
5.3	Troubleshooting Procedures	5-2
5.3.1	Test Description - Exercise Mode	5-2
5.3.2	Test Description - Adjustment Mode	5-5
5.3.3	Test Operation	5-5
5.3.3.1	Running the Program	5-5
5.3.3.2	Operator Inputs	5-5
5.3.3.3	User Options	5-6
5.3.3.4	Error Handling	5-7
5.3.3.5	Error Types	5-7
5.3.3.6	Test Termination	5-7
5.3.3.7	Buffer Presets	5-7

TABLE OF CONTENTS (Continued)

SECTION 5 MAINTENANCE AND TROUBLESHOOTING (Continued)

5.3.4	Worst Case Data Patterns	5-7
5.3.5	Disk Error Codes	5-8
5.3.6	Recommended Test Procedure	5-9

SECTION 6 SCHEMATIC AND PARTS LIST

LIST OF ILLUSTRATIONS

Figure	Title	Page
1-1	AM-500 Simplified Block Diagram	1-3
2-1	Standard Jumper Connections	2-9
3-1	AM-500 Commands	3-3
3-2	DMA Buffer Operation	3-7
3-3	Handshake Sequence	3-8
3-4	DMA Transfer Sequence	3-9
4-1	AM-500 Block Diagram	4-3
4-2	CPU Block Diagram	4-27
4-3	CPU Registers	4-27
4-4	CPU Pin Configuration	4-27
4-5	DMA Controller Connections	4-39
4-6	DMA Controller Block Diagram	4-46
4-7	DMA Controller Command Bits	4-54
4-8	DMA Controller Mode Register Bits	4-56
4-9	DMA Controller Request Register Bits	4-57
4-10	DMA Controller Mask Register Bits	4-58
4-11	DMA Controller Status Register Bits	4-59
4-12	FIFO Buffer Connections	4-61
4-13	FIFO Block Diagram	4-62
4-14	FIFO Conceptual INPUT Section	4-63
4-15	Final Positions In A FIFO Resulting from A 64-Bit Serial Train	4-64
4-16	FIFO Conceptual Output Section	4-65
4-17	FIFO Horizontal Expansion Scheme	4-67
4-18	FIFO Timing-Serial Input, Unexpanded or Master Operation	4-68
4-19	FIFO Timing-Serial Input, Expanded Slave Operation	4-68
4-20	FIFO Timing-Serial Output, Unexpanded or Master Operation	4-69
4-21	FIFO Timing-Serial Output, Slave Operation	4-69

LIST OF ILLUSTRATIONS (Continued)

Figure	Title	Page
4-22	FIFO Timing-Parallel Output, 4-Bit Word or Master In Parallel Expansion	4-70
4-23	FIFO Fall Through Time	4-70
4-24	FIFO Timing-Parallel Load Mode, 4-Bit Word (Unexpanded) or Master In Parallel Expansion	4-71
4-25	FIFO Timing-Parallel Load, Slave Mode . . .	4-71
4-26	CRC Checker Connections	4-73
4-27	CRC Equivalent Circuit for $X^{16} + X^{15} + X^2 + 1$.	4-74
4-28	CRC Check Word Generation	4-74
4-29	PROM Connections	4-75
4-30	RAM Connections	4-78
4-31	Tri-State Buffer Connections	4-79
4-32	Bus Comparator Connections	4-80
4-33	Schmitt Trigger Connections	4-81
4-34	D Flip-Flop Connections	4-82
4-35	Tri-State Buffer Connections	4-83
4-36	Decoder Connections	4-84
4-37	Decoder/Demultiplexer Connections	4-85
4-38	Quad 2-Line to 1-Line Data Selector/Multi- plexer Connections	4-86
4-39	D-Type Transparent Latch Connections	4-87
4-40	4-Bit Binary Counter Connections	4-88
4-41	Tri-State D Flip-Flop Connections	4-89
5-1	Disk Drive I/O Circuit Board Switch Location	5-4

LIST OF TABLES

Table	Title	Page
2-1	AM-500 Specifications	2-2
2-2	S-100 Bus Interface Signals List	2-4
2-3	Disk Drive Interface Signals List.	2-6
3-1	Status Word Formats	3-4
4-1	AM-500 Signals List	4-7
4-2	Non-Disk Command Bits	4-15
4-3	Non-Disk Command Formats	4-16
4-4	Disk Command Bits	4-17
4-5	Disk Command Codes	4-18
4-6	Status Word Formats	4-20
4-7	CPU Signals List	4-28
4-8	CPU Instruction Set	4-35
4-9	DMA Controller Signal List	4-40
4-10	DMA Controller Internal Registers	4-46
4-11	DMA Controller Word Count and Address Register Command Codes	4-55
4-12	DMA Controller Register Codes	4-58
4-13	DMA Controller Software Command Codes	4-60
4-14	CRC Polynomial Select Codes	4-74
4-15	PROM Mode Selection	4-76
6-1	Component Crossreference List	6-1

USER COMMENTS		TECHNICAL PUBLICATIONS FILE REFERENCE
<p>FROM: NAME _____</p> <p>ADDRESS _____</p> <p>CITY _____</p> <p>STATE _____ ZIP _____</p>		<p>NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.</p> <p>If necessary, call us at:</p> <p>(714) 957-6076</p>

USER COMMENTS		TECHNICAL PUBLICATIONS FILE REFERENCE
<p>FROM: NAME _____</p> <p>ADDRESS _____</p> <p>CITY _____</p> <p>STATE _____ ZIP _____</p>		<p>NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.</p> <p>If necessary, call us at:</p> <p>(714) 957-6076</p>

USER COMMENTS		TECHNICAL PUBLICATIONS FILE REFERENCE
<p>FROM: NAME _____</p> <p>ADDRESS _____</p> <p>CITY _____</p> <p>STATE _____ ZIP _____</p>		<p>NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.</p> <p>If necessary, call us at:</p> <p>(714) 957-6076</p>

USER COMMENTS		TECHNICAL PUBLICATIONS FILE REFERENCE
<p>FROM: NAME _____</p> <p>ADDRESS _____</p> <p>CITY _____</p> <p>STATE _____ ZIP _____</p>		<p>NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.</p> <p>If necessary, call us at:</p> <p>(714) 957-6076</p>

USER COMMENTS		TECHNICAL PUBLICATIONS FILE REFERENCE
<p>FROM: NAME _____</p> <p>ADDRESS _____</p> <p>CITY _____</p> <p>STATE _____ ZIP _____</p>		<p>NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.</p> <p>If necessary, call us at:</p> <p>(714) 957-6076</p>

USER COMMENTS		TECHNICAL PUBLICATIONS FILE REFERENCE
<p>FROM: NAME _____</p> <p>ADDRESS _____</p> <p>CITY _____</p> <p>STATE _____ ZIP _____</p>		<p>NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.</p> <p>If necessary, call us at:</p> <p>(714) 957-6076</p>

USER COMMENTS		TECHNICAL PUBLICATIONS FILE REFERENCE
<p>FROM: NAME _____</p> <p>ADDRESS _____</p> <p>CITY _____</p> <p>STATE _____ ZIP _____</p>		<p>NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.</p> <p>If necessary, call us at:</p> <p>(714) 957-6076</p>

USER COMMENTS		TECHNICAL PUBLICATIONS FILE REFERENCE
<p>FROM: NAME _____</p> <p>ADDRESS _____</p> <p>CITY _____</p> <p>STATE _____ ZIP _____</p>		<p>NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.</p> <p>If necessary, call us at:</p> <p>(714) 957-6076</p>

USER COMMENTS		TECHNICAL PUBLICATIONS FILE REFERENCE
<p>FROM: NAME _____</p> <p>ADDRESS _____</p> <p>CITY _____</p> <p>STATE _____ ZIP _____</p>		<p>NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.</p> <p>If necessary, call us at:</p> <p>(714) 957-6076</p>

STAPLE

STAPLE

FOLD

FOLD

PLACE
STAMP
HERE



17881 Sky Park North
Irvine, California
92714

ATTN: EDUCATIONAL SERVICES DIVISION

FOLD

FOLD

CUT ALONG LINE

SECTION 1

GENERAL DESCRIPTION

1.0 INTRODUCTION.

This Manual provides operating and maintenance instructions for the AM-500 Hard Disk Controller Circuit Board manufactured by Alpha Microsystems Inc., located in Irvine, California. Circuit board description, operating and usage instructions, programming, theory of operation, and maintenance instructions are included to provide the user with the information necessary to utilize this circuit board to its full capability.

1.1 CIRCUIT BOARD DESCRIPTION.

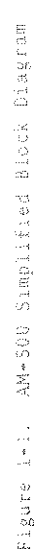
The AM-500 Hard Disk Controller Circuit Board provides data processing, control and interface capability between a standard S-100 bus system and one to four CDC 9427H (Hawk) hard disk drives. A CPU module, a DMA module, a 16 x 8 First-In-First-Out (FIFO) memory and 1K of Random Access Memory (RAM) provide the sophisticated data processing necessary for control of up to four ten-megabyte disk drives. A 2K Read Only Memory (ROM) is contained on the board to provide a bootstrap load program and also to contain the microcode necessary for the CPU module operation.

A simplified block diagram of the circuit board is shown in Figure 1-1. For a complete detailed description of circuit board operation, see Section 4 of this manual. For programming requirements, see Section 3 of this manual.

1.2 APPLICATION.

This circuit board provides the data processing and interface capability necessary for operation of up to four CDC 9427H (Hawk) hard disk drives for a maximum of ten megabytes of memory capacity per drive. This circuit board is not recommended for any other disk drives at this time. See Section 2

of this manual for wiring instructions and system interface information. For complete information on the disk drive, see the Control Data Corporation 9427H Hardware Maintenance Manual.



SECTION 2

OPERATING DATA

2.0 INTRODUCTION.

This Section contains information on the use of the AM-500 Hard Disk Controller circuit board. Capabilities, specifications, interface wiring, and user option descriptions are provided for the successful integration of the AM-500 into the user's system.

2.1 CAPABILITIES AND SPECIFICATIONS.

This circuit board operates from the standard S-100 bus structure to interface with the CDC 9427H (Hawk) hard disk drive with 10 megabyte capacity (five fixed/five removable). A single AM-500 board can support up to four drives for a total capacity of 40 megabytes of hard disk memory. Detailed specifications are listed in Table 2-1.

Table 2-1. AM-500 Specifications

PARAMETER	SPECIFICATION
CPU Interface	Standard S-100 Bus - Interrupt Driven
Storage Capacity, Each Drive	10 Megabyte Total; 5 Megabytes on fixed surface, 5 Megabytes on removable cartridge.
Maximum Storage capacity from each AM-500 Controller Board	40 Megabytes with 4 Drives connected.
AM-500 to disk Bit Transfer Rate.	2.5 MHz.
Sector Size	512 Bytes/Sector plus CRC Error Code and Sentinel Check byte.
CPU Interface Description	Multilevel interrupt driven; full sector block transfers from or to internal sector buffer.
Circuit Board	Standard 5" x 10" with 100 pin connector.

2.2 INTERFACE DESCRIPTION AND WIRING.

The AM-500 Hard Disk Controller Circuit Board provides interface capability between the Standard S-100 bus and CDC 9427H (Hawk) hard disk drives.

2.2.1 S-100 BUS INTERFACE.

The AM-500 circuit board is fully S-100 bus compatible. The board and its associated disk drives are addressed through the address lines and data is transferred through the standard data in and data out ports. The S-100 bus connections are made via the bottom edge connector and are described in Table 2-2.

2.2.2 DISK DRIVE INTERFACE.

The AM-500 circuit board accommodates a maximum of four hard disk drives. Unit select outputs from the circuit board select any one of the four drives individually and in arbitrary sequence. All disk drive interface signals are described in Table 2-3

Table 2-2. S-100 Bus Interface Signals List

MNEMONIC	NAME	PIN	DESCRIPTION
A0	Address 0	79	16 bits of S-100 addressing.
A1	Address 1	80	
A2	Address 2	81	
A3	Address 3	31	
A4	Address 4	30	
A5	Address 5	29	
A6	Address 6	82	
A7	Address 7	83	
A8	Address 8	84	
A9	Address 9	34	
A10	Address 10	37	
A11	Address 11	87	
A12	Address 12	33	
A13	Address 13	85	
A14	Address 14	86	
A15	Address 15	32	
DATAIN 0	Input Data Bit 0	95	Data input port. Bus master input from slaves.
DATAIN 1	Input Data Bit 1	94	
DATAIN 2	Input Data Bit 2	41	
DATAIN 3	Input Data Bit 3	42	
DATAIN 4	Input Data Bit 4	91	
DATAIN 5	Input Data Bit 5	92	
DATAIN 6	Input Data Bit 6	93	
DATAIN 7	Input Data Bit 7	43	
DATAOUT 0	Output Data Bit 0	36	Data output port. Bus master output to slaves.
DATAOUT 1	Output Data Bit 1	35	
DATAOUT 2	Output Data Bit 2	88	
DATAOUT 3	Output Data Bit 3	89	
DATAOUT 4	Output Data Bit 4	38	
DATAOUT 5	Output Data Bit 5	39	
DATAOUT 6	Output Data Bit 6	40	
DATAOUT 7	Output Data Bit 7	90	

Table 2-2 (Cont.). S-100 Bus Interface Signals List

MNEMONIC	NAME	PIN	DESCRIPTION
PDBIN	Data Bus In	78	Read enable. Used by bus master to request addressed slave to place data on input port.
$\overline{\text{PHANTOM}}$	Phantom	67	Disables phantom controlled memories when asserted.
$\overline{\text{PWR}}$	Write Strobe	77	Write strobe. Generated by bus masters as write command to slaves.
$\overline{\text{RESET}}$	Reset	75	AM-500 reset input.
SINP	I/O Input Cycle Status Line	46	Current bus cycle is a bus master input from an I/O address.
SMEMR	Memory Cycle Input Status Line	47	Current bus cycle is a bus master input from a memory address.
SOUT	I/O Output Cycle Status Line	45	Current bus cycle is a bus master output to an I/O address.
$\overline{\text{VI}}\ 0$	Vectored Interrupt 0	4	Jumper selected interrupts. Standard interrupt level = 4.
$\overline{\text{VI}}\ 1$	Vectored Interrupt 1	5	
$\overline{\text{VI}}\ 2$	Vectored Interrupt 2	6	
$\overline{\text{VI}}\ 3$	Vectored Interrupt 3	7	
$\overline{\text{VI}}\ 4$	Vectored Interrupt 4	8	
$\overline{\text{VI}}\ 5$	Vectored Interrupt 5	9	
$\overline{\text{VI}}\ 6$	Vectored Interrupt 6	10	
$\overline{\text{VI}}\ 7$	Vectored Interrupt 7	11	

Table 2-3. Disk Drive Interface Signal List

MNEMONIC	NAME	PIN	DESCRIPTION
<u>CA 0</u>	Cylinder Address Bit 1	39	The absolute address location of the cylinder to which the heads are to be positioned.
<u>CA 1</u>	Cylinder Address Bit 2	17	
<u>CA 2</u>	Cylinder Address Bit 4	40	
<u>CA 3</u>	Cylinder Address Bit 8	33	
<u>CA 4</u>	Cylinder Address Bit 16	28	
<u>CA 5</u>	Cylinder Address Bit 32	35	
<u>CA 6</u>	Cylinder Address Bit 64	31	
<u>CA 7</u>	Cylinder Address Bit 128	11	
<u>CA 8</u>	Cylinder Address Bit 256	24	
<u>CYLSTR</u>	Cylinder Address Strobe	18	Gates cylinder address information into the cylinder address register and initiates servo controlled head positioning.
<u>DRVRDY</u>	Drive Ready	30	Unit is ready to seek, read, or write.
<u>EGATE</u>	Erase Gate	37	Enables erase current to flow through selected head.
<u>FAULT</u>	Fault	22	Attempted to write or erase when not on cylinder. Attempted to read when write or erase is enabled. Current fault during read, write, or erase.
<u>HDSEL</u>	Head Select 1	26	1=Select upper head. 0=Select lower head.

Table 2-3 (Cont.). Disk Drive Interface Signal List

MNEMONIC	NAME	PIN	DESCRIPTION
INDEX	Index Mark	27	Index mark once per disk revolution (25 msec).
ONCYL	On Cylinder	45	Seek operation is complete and unit is ready to read or write.
$\overline{\text{PLSEL}}$	Platter Select	13	1=Select fixed disk; 0=Select removable cartridge.
READ CLOCK	Read Clock	44	Separated clocks concurrent with read data.
READ DATA	Read Data	42	Separated data input.
$\overline{\text{RESTOR}}$	Return to Zero Seek	14	Positions the heads to cylinder zero (clears seek error).
$\overline{\text{RGATE}}$	Read Gate	41	Enables separated data and clock to the interface lines.
SECTOR	Sector Mark	29	Sector mark pulse (2 msec).
SEKERR	Seek Error	5	Seek operation did not complete within 500 ms.
$\overline{\text{SEL 1}}$	Unit Select 1	46	Selects any one of the 4 disk units individually and in arbitrary sequence.
$\overline{\text{SEL 2}}$	Unit Select 2	47	
$\overline{\text{SEL 3}}$	Unit Select 3	49	
$\overline{\text{SEL 4}}$	Unit Select 4	48	

Table 2-3 (Cont.). Disk Drive Interface Signal List

MNEMONIC	NAME	PIN	DESCRIPTION
$\overline{\text{WGATE}}$	Write Gate	23	Enables write current to flow through selected head.
WPROT	Write Protected	34	Inhibits erase and write currents. 1=Inhibit; 0=Enable.
WRITE DATA	Write Data	50	Double frequency encoded clock and data.
+5VDC	+5VDC (not used)	43	Power and ground.
GND	GND	1	
GND	GND	15	
GND	GND	32	

2.2.3 WIRING AND CONNECTIONS.

When the AM-500 circuit board is received, it is ready for use. No adjustment or calibration is required for operation. The hardware requirements for use are described in this Section and the software requirements are described in Section 3.

First ensure that the proper power wiring is available and the correct voltages are connected to the various pins as shown on sheet 1 of the schematic. All power connections are made through the bottom edge connector.

All connections to the S-100 bus are made through the bottom edge connector (Table 2-2) and connections to the disk drive are made through the top connector (Table 2-3). Ensure that these connections are correct before plugging the AM-500 circuit board into the system.

2.3 USER OPTIONS.

Several features of the AM-500 circuit board can be jumper selected at the user's option. These jumpers must be installed before the AM-500 circuit board is installed in the system. The standard jumper connections are shown in Figure 2-1.

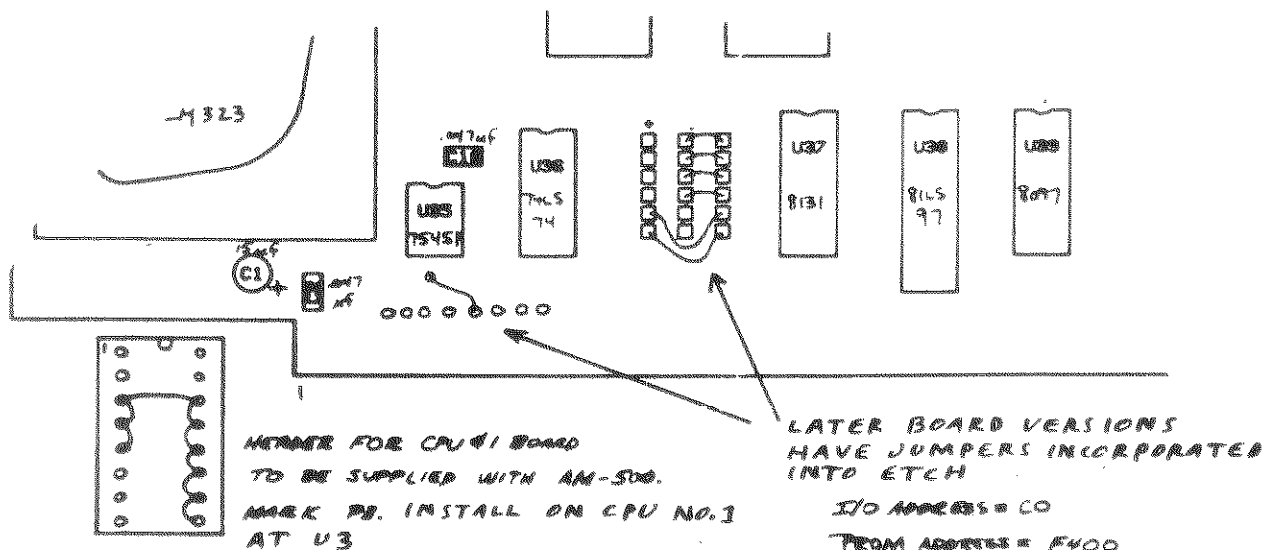


Figure 2-1. Standard Jumper Connections

2.3.1 ADDRESS CODE.

Circuit board addressing can be selected at the user's option for any address block (in increments of four) on the address lines A0-A7. Connect jumper wires to +5V or ground for the desired address. Jumper pads are located on the circuit board between U36 and U37 (schematic sheet 2). The standard I/O address is C0-C3.

2.3.2 INTERRUPT LINES.

Interrupt compatibility for any S-100 bus system is provided with jumpers to any of the vectored interrupt lines VI0-VI7. Attach the jumper wire from the pad connected to pin 3 of U35 to the desired interrupt line (schematic sheet 11). Jumper pads are located on the circuit board directly below U35. The standard interrupt level is VI4.

2.3.3 PHANTOM FEATURE.

A phantom feature can be used with the AM-500 Hard Disk Controller. This feature is supported in operating system software release 3.0 and later and will not operate if a video display module (VDM) is being used for software program DYSTAT because the system memory overlaps the VDM memory (see Alpha Micro Operating System AMOS manual). Also, the memory board must support the phantom feature.

The AM-500 circuit board comes from the factory configured to enable the phantom feature. If it is desired to disable the phantom feature in order to use the disk drive as a peripheral device, it will be necessary to cut the etch that runs between J1-67 and U35-5. Cut the etch between the two feed thru holes on the solder side of the board. The S-100 pins on the back of the card start at 51 on the right and count up to 100 on the left.

The AM-500 already has the phantom signal connected to pin 67 of the S-100 bus. During bootstrap, the line goes low disabling the upper memory board or any board that monitors the phantom line. The CPU starts executing the bootstrap PROM at location F400 and moves it into location 76000 octal or 7C00 hex. The bootstrap then starts executing from memory. Any memory board that overlaps the bootstrap PROM must have phantom enabled and all the others must have it disabled.

2.3.4 BOOT OPTIONS.

Since the AM-200, AM-210 and AM-400 also contain internal bootstrap loader PROMS, the user may desire to disable the bootstrap from the AM-500 and boot from one of these other circuit boards. The following paragraphs describe the requirements for these options.

2.3.4.1 THE AM-500 AS A PERIPHERAL TO THE AM-200 OR AM-210.
Remove the 74LS30 IC (U40) from the AM-500 (bottom row, 6th chip from the right). If the AM-200 or AM-210 are using phantom (phantom must be used if you have more than 56K of memory), the phantom etch must be cut on the AM-500. Turn the AM-500 board over (component side down) and cut the etch going to pin 67. Cut the etch between the two eyelets so a wire may be inserted between the two eyelets at a later date to reenable phantom. The S-100 pins on the back side start at 51 on the right and go to 100 on the left.

Boot the system with both the AM-500 and AM-200 or AM-210 in the system.

Log into area 1,4 on the floppy disk and modify your SYSTEM.INI to add the following:

```
BITMAP HWK,606,0,1
```

Be sure that the DEVTBL command in the SYSTEM.INI contains HWK0 and HWK1.

Log into area 1,6 and do the following:

```
COPY HWK.DVR=HWK500.DVR
```

Now reboot the system again and by mounting HWK0 and HWK1, you may access them as peripherals to the AM-200 or AM-210. Remember when in this mode the fixed platter is always HWK0.

2.3.4.2 OPERATING THE AM-500 WITH THE AM-400/TRIDENT.
When running these two in combination, phantom must be disabled on the controller that is not the system device. If it is desired to remove phantom on the AM-400, remove Z32

pin 5 from the socket so it sticks out. If you are booting from the AM-500, disable the PROM on the AM-400 by removing Z29, which is the IC to the left of the header in the bottom row.

The proper bitmap sizes and device names must be in the SYSTEM.INI file to accommodate the proper configuration. Be sure the proper driver in area 1,6 is renamed to the same name as the device you defined in the DEVTBL. The bitmap sizes are 562 for the T-25, 1022 for the T-50, 1534 for the T-80, and 1630 for the T-300.

If booting is off the AM-500, initialization must be from the AM-400. This is done by running a program. For the T-25, T-50, and T-200 use TRIINI.PRG, and for the T-80 and T-300 use T80INI.PRG. To utilize it, just type TxxINI followed by a carriage return. The program should return a status less than 3 octal. If you do not run this initialization program, the drives will show as NOT READY on a SYSTAT.

SECTION 3 PROGRAMMING

3.0 INTRODUCTION.

This section describes the programming requirements for the AM-500 circuit board. Circuit board addressing, bootstrap loader, and AM-500 internal programming are described for complete system compatibility.

When the AM-500 circuit board is received, it is ready for use with the CDC 9427H (Hawk) disk drive as described in Section 2 of this manual. The bootstrap loader and the microcode contained in internal firmware is designed only for this disk drive.

3.1 ADDRESSING.

The AM-500 and associated disk drives are addressed through the S-100 bus address lines. The circuit board address is jumper selectable by the jumpers connected to U37. Jumpers to either +5V or ground select the board address on S-100 bus address lines AD2-AD7. The board always occupies four I/O ports even though only two are actually used.

3.2 INITIALIZATION.

On initial power-up or reset, the AM-500 goes into a dormant state waiting for initialization. With the phantom option enabled, the phantom signal is activated, disabling any memory. The S-100 bus is enabled to access the bootstrap loader contained in the AM-500 PROM.

The 16K PROM (U21) contains both the bootstrap routine and the controller microcode. The microcode resides in the first 1K bytes, and the bootstrap routine resides in the second 1K bytes of this PROM. The current bootstrap routine is written in

AM-100 code, and other user codes may be used if the microcode in the first 1K bytes is duplicated exactly.

NOTE

If the user code contained in the AM-500 PROM is changed from the current boot routine, the microcode in the first 1K of memory must be duplicated exactly.

Initialization of the AM-500 is accomplished by writing a zero byte into the AM-500 command register during system initialization. The CPU finishes initialization of the disk controller and releases the Phantom signal. See paragraph 2.3.4 for boot options.

3.3 COMMAND REGISTER.

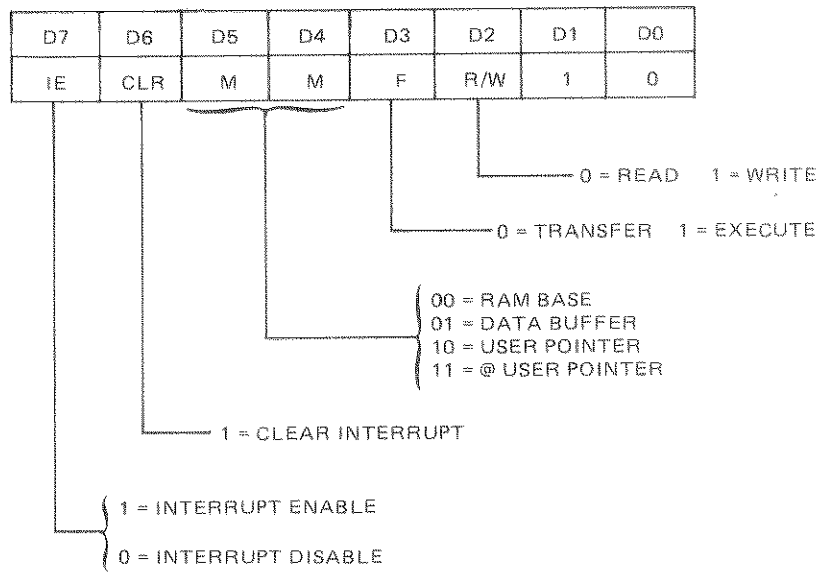
The command register receives AM-500 commands from the S-100 bus. It is accessed by writing to the selected address (see paragraph 2.3.1). The legal base addresses are in multiples of four from the first address 177400 (octal), 177404 (octal), 177408 (octal), through 177774 (octal). A software handshake sequence must be used for the command transfer (see paragraph 3.6).

There are two basic types of commands used for operation of the AM-500 system; non-disk commands and disk commands. Both types are described in Figure 3-1.

3.4 STATUS REGISTER.

The status register is used to determine the current status of the AM-500 controller. It is accessed from the S-100 bus by reading from the user selected base address of the circuit board. The format of the status register is described in Table 3-1.

NON-DISK COMMANDS:



DISK COMMANDS:

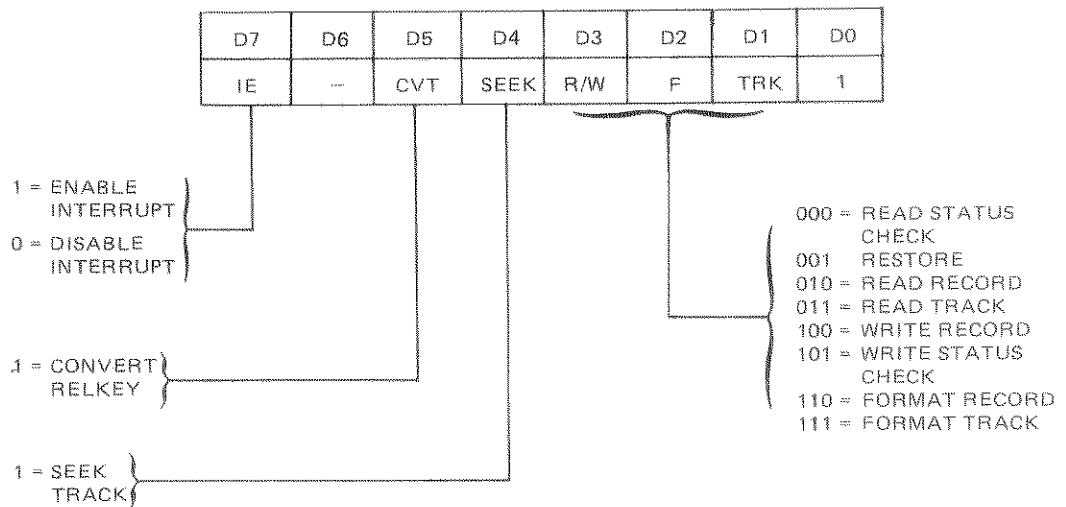


Figure 3-1. AM-500 Commands

Table 3-1. Status Word Formats

BIT								FUNCTION
7*	6	5	4	3	2	1	0	
CR	0	0	0	0	1	0	1	Disk Not Ready
CR	0	0	0	1	0	1	1	Write Protected
CR	1	0	0	0	0	0	1	Seek Error
CR	1	0	0	0	0	1	1	Sector Not Found
CR	1	0	0	0	1	0	0	CRC Error
CR	1	0	0	0	1	0	1	Sentinel Field Error

* CR=COMMAND RECEIVED, 0=CLEARED, 1=RECEIVED.

The sentinel field error indicates that data was not correctly transferred within the AM-500. The sentinel feature checks byte 513 for the correct code 5A (hex) after a 512 byte transfer.

3.5 INTERRUPT MODE.

The interrupt mode in the AM-500 can be selected by both disk and non-disk commands. If the interrupt mode is selected, the interrupt occurs at the completion of the current AM-500 command. The interrupt must be cleared by the user by setting bit 6=1 of any non-disk command.

3.6 HANDSHAKE SEQUENCE.

A handshaking type interaction is required between the AM-500 circuit board and the CPU any time a command is sent to the AM-500. This handshaking is a software operation that uses bit 7 of the Status Register. Handshake operations proceed in the following sequence:

1. A command is sent from the S-100 bus to the Command Register.
2. The system software waits for bit 7 of the Status Register to become a 1 to guarantee that the

AM-500 has received the command.

3. The system software writes a zero into the Command Register to clear the previous command.
4. The system software waits until bit 7 of the Status Register becomes a 0 indicating that the Command Register has been cleared.

This completes the handshake operation and the program continues.

3.7 RELKEY DESCRIPTION.

The term RELKEY refers to a double byte number which points to one of 9696 possible sectors on the disk. The user loads a number from 0 to 9695 into the Relkey bytes of the DMA buffer to indicate which sector is to be operated on. If bit 5 of the disk command is a 1, the microcode converts this number along with the drive select byte into an absolute disk address and stores it back into the DMA buffer. The microcode replaces the drive number and Relkey bytes of the DMA buffer with the converted absolute disk address. Since all DMA buffer locations are accessible, the absolute disk address can be retrieved if necessary.

If the user has the desired absolute disk address, it can be placed in the DMA buffer, and the Relkey conversion can be skipped. This is accomplished as follows:

1. Place the actual drive number (1-3) into the second byte of the buffer.
2. Place the low byte of the track address in the third byte of the buffer.
3. Place the high byte of the track address (which specifies the surface, the platter select, and the sector address) into the fourth byte of the buffer as shown below.

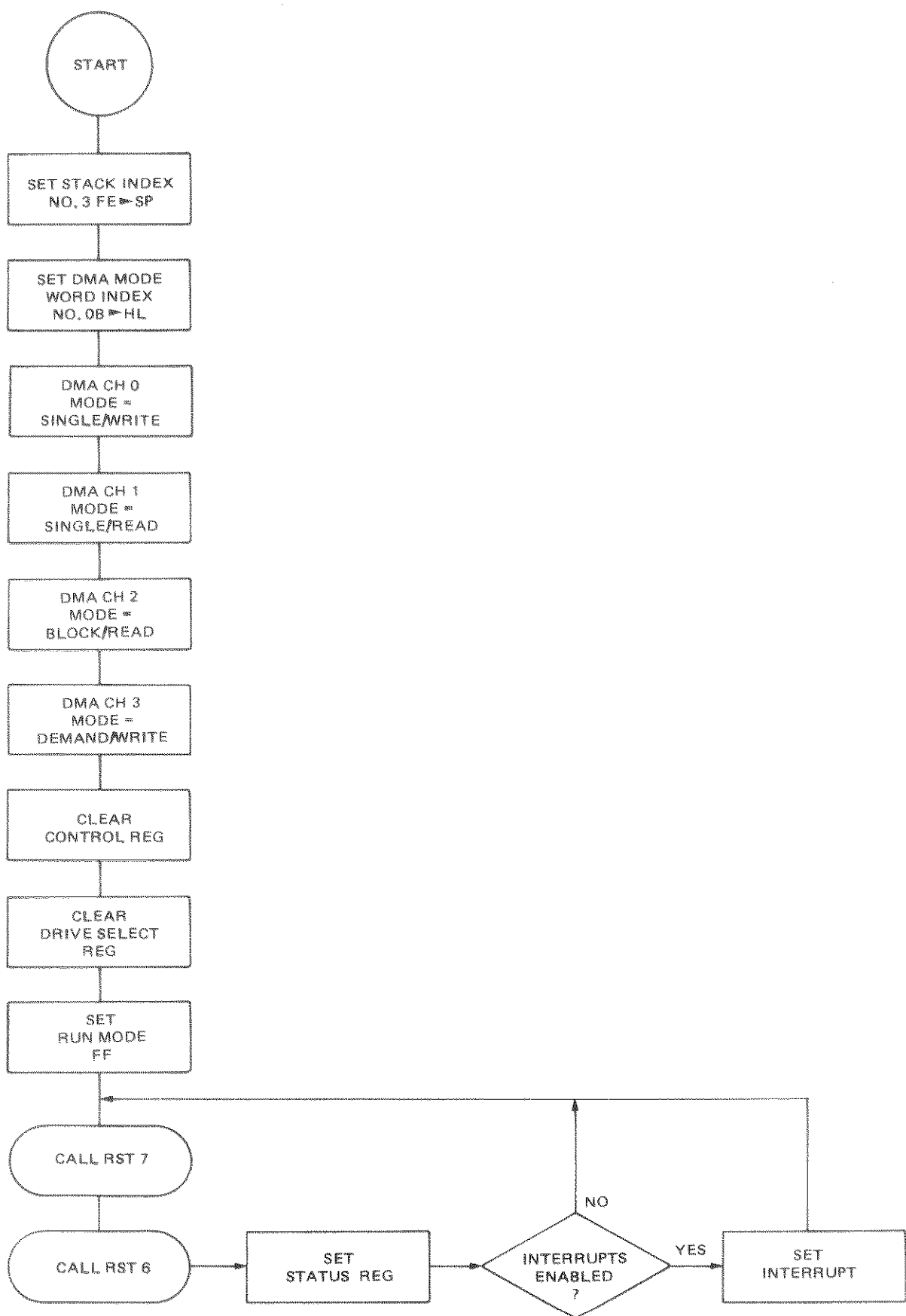


Figure 3-2. DMA Buffer Operation

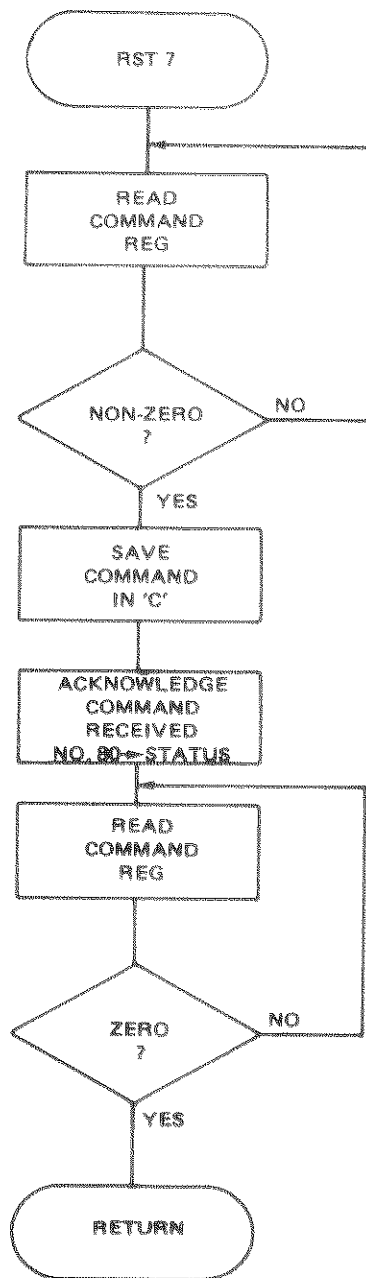


Figure 3-3. Handshake Sequence

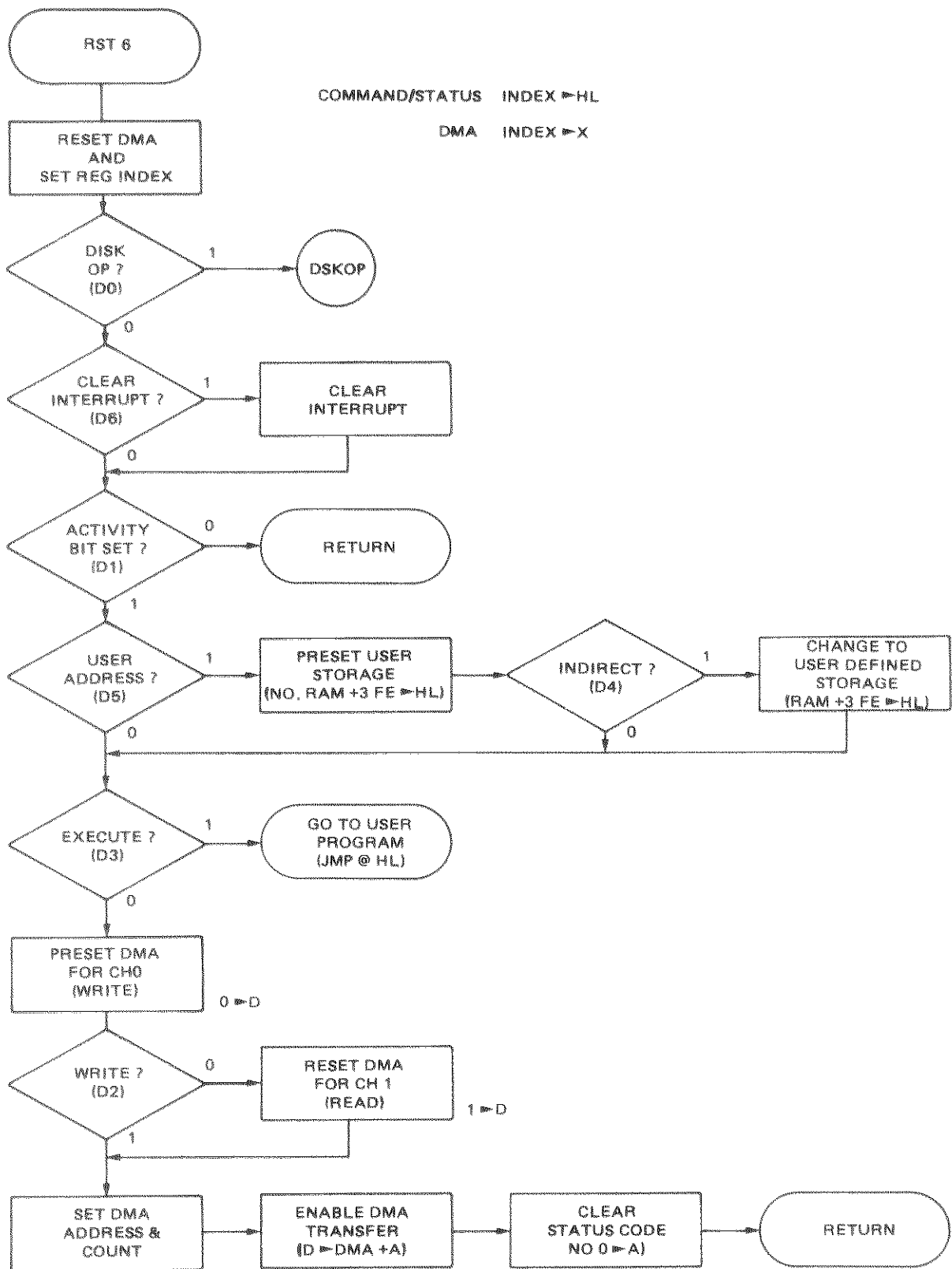


Figure 3-4. DMA Transfer Sequence

condition indicating no error.

9. Continue with next command if no errors were detected.

3.8.2 DMA BUFFER OPERATION FOR DISK READ.

Transferring data from the disk into the DMA buffer and then reading the DMA buffer is performed as follows:

1. Send DMA write command X06 (octal) to AM-500 command register and perform a handshake.
2. Write retry count into DMA buffer.
3. Write a disk number (0-7) into DMA buffer.
4. Write low byte of Relkey into DMA buffer.
5. Write high byte of Relkey into DMA buffer.
6. Send disk READ command X35 (octal) to AM-500 command register and perform handshake.
7. Read AM-500 status register and check for zero condition indicating no errors occurred.
8. Assuming no errors occurred during the disk read, send the DMA read command X12 (octal) to the AM-500 command register and perform the normal handshake.
9. Read one byte of data from the DMA buffer and discard it (the first byte is always invalid data).
10. Read valid data, 1 to 512 bytes, from the DMA buffer.

3.9 RESTORE COMMAND.

The restore command is used to reset the disk logic and force the disk head to return to track zero. The user may force a restore command with a disk command 3 (octal), or the microcode will exercise the restore command on its own following certain errors.

If a disk fault occurs, the microcode attempts one restore and then checks the fault status again. If the fault still exists, the microcode aborts the command operation and reports an error to the CPU.

3.10 DISK FORMATTING COMMANDS.

There are two disk formatting commands, Format Record and Format Track, that are identical in function. Both commands cause the currently addressed track to be formatted.

CAUTION

The Format Record and Format Track commands both destroy previously stored data on the addressed track by writing all new records with data fields of zeros.

3.11 STATUS CHECK.

The Read and Write Status check commands are used to access specific status information from the disk drive.

The Write Status check command 001 (octal) is used to determine if the currently addressed disk is write protected. It does not attempt to actually write any data. If the addressed disk is write protected, error code 13 (octal) is placed in the status register.

The Read Status check command (if SEEK bit 4 is on) seeks for the currently addressed track and then returns. Data is not read from the disk. If a seek error occurred, an error code 14 (octal) is placed in the status register.

3.12 ERROR CODE REPORTING.

The operating system does not normally report soft disk errors. A soft disk error is defined as a non-recoverable error after a set number of retries, usually 4 on the AM-500. If after 4 retries the error is non-recoverable, the system aborts the command and reports an error.

To enable reporting of the error type the operator should type:

SET DSKERR

The AM-500 error codes are as follows:

- 1 - SEEK ERROR
- 2 - NOT USED
- 3 - SECTOR NOT FOUND
- 4 - CRC CHECKSUM
- 5 - SOFTWARE CHECKSUM

SECTION 4

FUNCTIONAL THEORY OF OPERATION

4.0 INTRODUCTION.

The AM-500 Hard Disk Controller circuit board contains integrated circuit elements for the necessary data processing for the performance of the functions as described in Sections 1, 2 and 3 of this manual. This Section describes the functional theory of operation of the circuit board and also provides information for each of the integrated circuit elements.

4.1 CIRCUIT BOARD OPERATION.

This circuit board provides control and interface capability between the S-100 bus and the CDC 9427H (Hawk) hard disk drive. The functional block diagram of the circuit board is shown in Figure 4-1, and the circuit board schematic is contained in Section 6 of this manual. Table 4-1 contains a list of the signals used in this circuit board with definitions of their functions. For S-100 bus interface signals, see Table 2-2. For Hard Disk drive signals, see Table 2-3.

4.1.1 POWER-UP RESET.

On initial power-up, the circuit board is in a dormant state waiting for initialization. System clocks are generated from the 10 MHz oscillator on U1 and divided down by U2, U22 and U43, to generate CLOCK, ONE and CELL clock signals. The power-on reset circuit activates RESET to reset the INIT and RUN flip-flops. The RUN signal activates the PHANTOM signal to enable any memory under phantom control, and the INIT signal enables the boot logic so the S-100 bus can access the bootstrap program contained in the PROM U21.

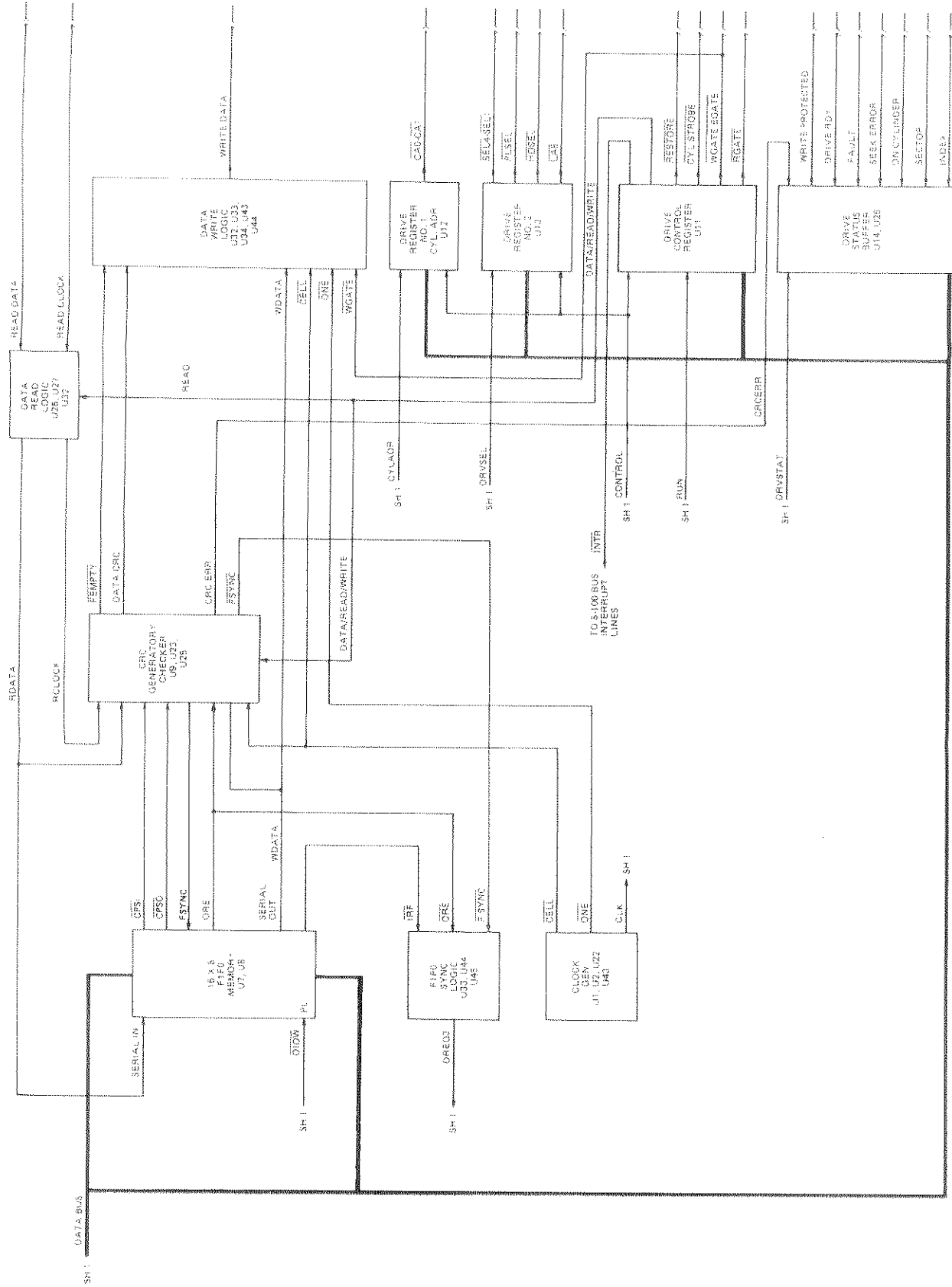


FIGURE 4-1. AM-506 Back Channel SCSI 2 v1.0
4-3-1-1 Block

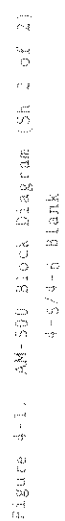


Figure 4-1. AM-500 Block Diagram (Sheet 1 of 2)

Table 4-1. AM-500 Signal List

SIGNAL	NAME	SCHEM PAGE OF SOURCE	FUNCTION
ADSTB	ADDRESS STROBE	7	Strobes upper address byte into external latch.
AEN	ADDRESS ENABLE	7	Active high that enables external latch which holds upper byte of address, and disables system during DMA cycle.
A0-A15	ADDRESS LINES	2, 6, 12	S-100 Bus Address Lines.
BOOT	BOOT LOAD	5	Enables a boot load from the PROM.
BUSAK	BUS ACKNOWLEDGE	6	Active low CPU output to indicate to the requesting device that the CPU address bus, data bus and tri-state control bus signals have been set to their high impedance state.
BUSRQ	BUS REQUEST	7	Request to the CPU for control of the system bus.
CELL	CELL CLOCK	3	Clock generated for clock and data output to disk drive.
CLOCK	2.5 MHz CLOCK	3	Clock generator output.

Table 4-1 (Cont.). AM-500 Signal List

SIGNAL	NAME	SCHEM PAGE OF SOURCE	FUNCTION
CMDWRT	COMMAND WRITE	2	Sets initialize flip-flop.
CNTRL	CONTROL	9	Clocks data from Data Bus (D0-07) into control register.
COMAND	COMMAND	9	Enables the output of the command register to place its contents on the Data Bus (D0-07).
CPS1	SERIAL INPUT CLOCK	10	Clock input for serial data input to FIFO memory stack.
CPS0	SERIAL OUTPUT CLOCK	9	(Input to FIFO.)
CRC	CYCLIC REDUN- DANCY CHECK	9	Data output of CRC checker.
CRC CLK	CRC CLOCK	9	Clock input to CRC flip-flop generated from either $\overline{\text{CELL}}$ or $\overline{\text{RCLOCK}}$. WRITE=1 selects $\overline{\text{CELL}}$ WRITE=0 selects $\overline{\text{RCLOCK}}$.
CRC DAT	CRC DATA	9	Data input to CRC checker.
CRC ERR	CRC ERROR	9	Error output of CRC checker.
CYL ADR	CYLINDER ADDRESS	9	Clocks Data from Data Bus into Drive Register 1.

Table 4-1 (Cont.). AM-500 Signal List

SIGNAL	NAME	SCHEM PAGE OF SOURCE	FUNCTION
DACK 0	DMA ACKNOWLEDGE 1	7	DMA controller output.
DACK 3	DMA ACKNOWLEDGE 3	7	DMA controller output.
DATA	DATA RESET	11	Master reset input to FIFO Memory and CRC checker.
DATRD	DATA READ	2	Detects Read Cycle.
DATWRT	DATA WRITE	2	Detects Write Cycle.
DIOR	DATA I/O READ	5	Enables data on S-100 bus D00-D07 to be placed on internal Data Bus and resets DREQ 0 to DMA controller.
DIOW	DATA I/O WRITE	5	Resets DREQ1 to DMA controller.
DMASEL	DMA SELECT	9	Chip Select input to enable DMA Controller.
D0-D7	DATA LINES	5	Tri-State bus internal to AM-500.
DREQ 0	DMA REQUEST 0	7	DMA Request input.
DREQ 1	DMA REQUEST 1	7	DMA Request input 1.

Table 4-1 (Cont.). AM-500 Signal List

SIGNAL	NAME	SCHEM PAGE OF SOURCE	FUNCTION
DREQ 3	DMA REQUEST 3	9	DMA Request input 3.
DRLCLR	DMA REQUEST LATCH CLEAR	9	Clears DMA request latches and presets the RUN flip-flop.
DRVSEL	DRIVE SELECT	9	Clock for Drive Register 1.
DRVSTS	DRIVE STATUS	9	Gates status data from the disk drive onto the internal data bus.
FEMPTY	FIFO MEMORY EMPTY	9	Enables the CRC checker and write logic when the FIFO memory is not empty.
FSYNC	FIFO MEMORY SYNC	9	Enables clock input to FIFO memory.
INIT	INITIALIZE	3	Reset signal to CPU.
IOR	I/O READ	6	Bidirectional, active low, three state line.
IOSEL	I/O SELECT	2	Enables the circuit board when address lines AD2-AD7 match the address code on the address jumpers.
IOW	I/O WRITE	6	Bidirectional, active low, three state line.

Table 4-1 (Cont.). AM-500 Signal List

SIGNAL	NAME	SCHEM PAGE OF SOURCE	FUNCTION
IRF	INPUT REGISTER FULL	10	Active low output of the upper FIFO memory when its input register is full.
LIRF	LOWER INPUT REGISTER FULL	10	Active low output of the lower FIFO memory when its input register is full.
LORE	LOWER OUTPUT REGISTER EMPTY	10	Active low output of the lower FIFO memory when its output register is empty (FIFO ORE).
MSL	MEMORY SELECT	9	Enables the memory select decoder.
ONE	ONE CLOCK	3	Clock generated for clock and data output to disk drive.
ORE	OUTPUT REGIS- TER EMPTY	10	Active low output of the upper FIFO memory when its output register is empty.
PL	PARALLEL LOAD	10	Parallel load signal to the FIFO memory.
RAMSEL	RAM MEMORY SELECT	9	Chip Select signal to enable the two Random Access Memory modules.
RCLOCK	READ CLOCK	3	Clock input from disk.

Table 4-1 (Cont.). AM-500 Signal List

SIGNAL	NAME	SCHEM PAGE OF SOURCE	FUNCTION
RDATA	READ DATA	3	Latched data input from disk.
READ	READ	11	Output of control register for disk read mode.
REGSEL	REGISTER SELECT	9	Enables the decoder to select a disk drive register control.
RD	MEMORY READ	6, 7	Tri-state CPU output, active low indicates that the CPU wants to read from memory.
RESET	RESET	3	Reset signal from S-100 bus.
RMSL	ROM SELECT	8	Chip select signal to ROM Memory generated from either ROMSEL or BOOT.
ROMSEL	ROM SELECT	9	Output of memory select decoder to select ROM memory.
RUN	RUN	3	Disk drive cycle run signal.
SIO	SELECTED I/O PORT	2	S-100 bus input generated from SINP or SOUT signals.
STATUS	STATUS SELECT	9	Clock input for Status Register.
STSRD	STATUS READ	2	Enables contents of status register to be placed on S-100 data lines DIO-DI7.

Table 4-1 (Cont.). AM-500 Signal List

SIGNAL	NAME	SCHEM PAGE OF SOURCE	FUNCTION
TOP	TRANSFER OUT PARALLEL	10	FIFO memory input to transfer parallel data into its output register.
WDATA	WRITE DATA	10	Serial output of FIFO memory to be written onto disk.
WGATE	WRITE GATE	11	Write enable output to disk.
WR	MEMORY WRITE	6, 7	Tri-state CPU output, active low indicates that CPU holds valid data to be stored into the RAM.
WRITE	WRITE	11	Control register output for disk write mode.

4.1.2 ADDRESSING.

Address data is received from the S-100 bus on address lines A0-A15. This address data is transferred to a tri-state address bus for use by the controllers and memories within the circuit board and is also used for direct addressing of the circuit board.

Tri-State Buffers on U38 and U39 gate the address data to the internal bus. Initialize signal INIT controls these gates when set by the Command Write signal CMDWRT.

Address lines A2-A7 are wired directly to comparator U37 for circuit board addressing. The other inputs come from jumper wires to either +5V or ground to produce the desired circuit board address. Signal IOSEL is asserted when the data from the address lines compare with the address of the card. This generates I/O read and write commands when combined with the S-100 bus control signals in decoder U17. The board address is programmable from 00 to FC (hex) in increments of 4.

4.1.3 CPU OUTPUT.

CPU Output data is transmitted to the AM-500 by the S-100 data and control lines. It is received in the AM-500 by the Command Register and Data Out Register.

4.1.3.1 COMMAND REGISTER.

The command register is used by the AM-500 to receive commands from the S-100 bus and consists of the register on U30. The input to the command register comes from the S-100 bus data lines D00-D07; and the output, when enabled, goes to the internal AM-500 data bus D0-D7.

Data from the S-100 bus data lines is clocked into the command register by the Command Write signal CMDWRT. This comes from the I/O command register when the AM-500 is addressed and A0 and A1 are zero. When I/O Output signal (SOUT) from the S-100 bus is asserted, CMDWRT clocks the command word from the S-100 bus data lines into the command register and also sets the initialize flip-flop U36 to begin an AM-500 cycle. Flip-flop U36 remains set until the RESET pushbutton is pressed or power turned off. When the CPU issues a Memory Read Command (RD low and WR high) and address lines A0 and A1 are zero, the Register/Memory Select decoders issue the COMAND signal to transfer the command register contents to the AM-500 data bus.

4.1.3.2 COMMAND TYPES.

The non-disk commands issued to the AM-500 are used for control and operation of the logic internal to the circuit card. The bit functions are listed in Table 4-2 and the commands are listed in Table 4-3. Disk Commands are used to control the disk drive and access status information from the drive. Disk command bit functions are listed in Table 4-4, and disk command codes are listed in Table 4-5.

Table 4-2. Non-Disk Command Bits

BIT	NAME	FUNCTION																	
0	0	=0] Indicates non-disk command =1] Activity Bit.																	
1	1																		
2	R/W	Read/Write 0=read 1=write.																	
3	FN	Command Function 0=transfer 1=execute																	
4	BUF] Location in DMA buffer where data is to be retrieved or stored.*																	
5	DMA																		
6	CLR	Clear current interrupt 0=nop 1=clear																	
7	IE	Interrupt Enable 0=disable 1=enable																	
		* <table><tr><th colspan="2">Bit</th><th rowspan="2">Location</th></tr><tr><td>5</td><td>4</td></tr><tr><td>0</td><td>0</td><td>RAM Base (Direct)</td></tr><tr><td>0</td><td>1</td><td>Data Buffer</td></tr><tr><td>1</td><td>0</td><td>User Pointer</td></tr><tr><td>1</td><td>1</td><td>@ User Pointer (Indirect)</td></tr></table>	Bit		Location	5	4	0	0	RAM Base (Direct)	0	1	Data Buffer	1	0	User Pointer	1	1	@ User Pointer (Indirect)
Bit		Location																	
5	4																		
0	0	RAM Base (Direct)																	
0	1	Data Buffer																	
1	0	User Pointer																	
1	1	@ User Pointer (Indirect)																	

Table 4-3. Non-Disk Command Formats

BITS 7 6 5 4 3 2 1 0	FUNCTION
0 0 0 0 0 0 0 0 X X 0 0 0 1 1 0 X X 0 1 0 1 1 0	Initialization/Boot Load Write data to DMA buffer (RAM Base) Write data to DMA buffer (Data Base)
X X 0 0 0 0 1 0 X X 0 1 0 0 1 0	Read data from DMA buffer (RAM Base) Read data from DMA buffer (Data Base)
0 1 0 0 0 0 0 0 X 1 X X X X 1 0	Clear Interrupt (Clear Only) Clear Interrupt (Plus some other non-disk command)
X X 1 0 0 1 1 0	Load user microcode pointer
X X 1 1 0 1 1 0	Indirect load via user pointer
X X 1 1 1 0 1 0	Execute from users microcode pointer
X X 1 1 0 0 1 0	Indirect read via user pointer
X X 1 0 0 0 1 0	Read user microcode pointer

Table 4-4. Disk Command Bits

BIT	NAME	FUNCTION
0	DSK	Always 1 for disk type commands
1	TRK*	Part of command function
2	FN*	Command function comprised of bits 1, 2 and 3 taken as a whole
3	R/W*	0=Read 1=Write
4	SK	Seek
5	CVT	Convert Relkey 1=Convert 0=No Conversion
6	-	Not used
7	IE	Interrupt enable 0=Disable 1=Enable

* See Table 4-5

Table 4-5. Disk Command Codes

BITS			FUNCTION
3 R/W	2 FN	1 TRK	
0	0	0	Read Status Check
0	0	0	Restore to Cylinder Zero
0	1	0	Read Record
0	1	1	Read Track
1	0	0	Write Record
1	0	1	Write Status Check
1	1	0	Format Record (same result as 010)
1	1	1	Format Track (Same result as 011)

4.1.3.3 DATA OUT REGISTER.

The Data Out register controls the transfer of data from the S-100 bus data lines to the data bus internal to the AM-500 circuit board and consists of the register on U41. The input to the Data Out register comes from the S-100 bus data lines D00-D07; and the output, when enabled, goes to the internal AM-500 data bus D0-D7.

Data from the S-100 bus data lines is clocked into the Data Out register by the Data Write signal $\overline{\text{DATWRT}}$. This comes from the I/O command register when the AM-500 is addressed and A0=1 and A1=0. When I/O Output signal (SOUT) from the S-100

bus is asserted, $\overline{\text{DATWRT}}$ clocks the data word from the S-100 bus into the Data Out Register and also sets the DMA request latch for a DMA service request. The DMA controller issues an acknowledge ($\overline{\text{DACK0}}$) that is combined with address bit A15=1 and a Read Output from the CPU to generate Data I/O Read signal $\overline{\text{DIOR}}$. Signal $\overline{\text{DIOR}}$ transfers the contents of the Data Out Register onto the internal data bus and also resets the DMA request latch.

4.1.4 CPU INPUT.

CPU input data is transmitted from the AM-500 by the S-100 data and control lines. It is transmitted from the AM-500 by the Status Register and Data In Register.

4.1.4.1. STATUS REGISTER.

The Status Register is used to determine the current status of AM-500 Controller operation and consists of the register on U31. The input to the Status Register comes from the AM-500 internal data bus D0-D7; and the output, when enabled, goes to the S-100 bus data lines DI0-DI7.

Data from the internal data bus is clocked into the Status Register by the $\overline{\text{STATUS}}$ signal from the Register/Memory Select logic. This occurs when the board is addressed and lines A0, A1 and $\overline{\text{WR}}$ are all low. The Register/Memory Select decoders issue the $\overline{\text{STATUS}}$ signal to clock the data into the Status Register from the internal data bus. The contents of the Status Register are applied to the S-100 bus data lines by the Status Read signal ($\overline{\text{STSRD}}$) from the I/O Command Register. This signal is issued when address line A0=0, Write Strobe $\overline{\text{PWR}}$ is false (high), Read enable is True (high), and Activate Input (SINP) is high.

4.1.4.2 STATUS WORD FORMATS.

Status words contain information regarding the operation of

the AM-500 controller. Information and formats of the status words are contained in Table 4-6. Bit 7 is the Command Received (CR) bit and 1=Received, 0=Cleared.

Table 4-6. Status Word Formats

BIT								FUNCTION
7*	6	5	4	3	2	1	0	
CR	0	0	0	0	1	0	1	Disk Not Ready
CR	0	0	0	1	0	1	1	Write Protected
CR	1	0	0	0	0	0	1	Seek Error
CR	1	0	0	0	0	1	1	Sector Not Found
CR	1	0	0	0	1	0	0	CRC Error
CR	1	0	0	0	1	0	1	Sentinel Field Error

* CR = Command Received 0=Cleared, 1=Received.

4.1.4.3 DATA IN REGISTER.

The Data In Register controls the transfer of data from the data bus internal to the AM-500 to the S-100 bus data lines and consists of the register on U42. The input to this register comes from the internal data lines D0-D7 and the output, when enabled, goes to the S-100 bus data lines DI0-DI7.

Data from the internal bus is clocked into the register when either Data I/O Write ($\overline{\text{DIOW}}$) is true or RUN is false. Signal $\overline{\text{DIOW}}$ comes from Bus Acknowledge ($\overline{\text{BUSAK}}$) and Write ($\overline{\text{WR}}$) from the CPU when DMA is selected by the Memory/Register Select logic. Data is placed on the S-100 bus data lines when in the Bootstrap Mode or when the I/O Command Register issues a Data Read $\overline{\text{DATRD}}$ command. The $\overline{\text{DATRD}}$ is issued when the board is addressed and A0=1, A1=0, Read Enable PDBIN is true (high) and Write Strobe ($\overline{\text{PWR}}$) is false (high).

4.1.5 BOOT LOAD PROCEDURE.

The AM-500 circuit board contains a bootstrap load routine stored in the ROM on U21. The microcode for internal AM-500 operation is also contained in the first 1K of ROM memory. The bootstrap program is contained in the second 1K of memory.

NOTE

If the user code is changed from the current boot routine, the microcode in the first 1K of memory must be duplicated exactly.

The CPU reads the bootstrap load program by addressing the circuit card and reading from 172000 (octal) up. This address, combined with SMEMR from the S-100 bus and a high $\overline{\text{INIT}}$ signal which takes place on initial start-up, generates the $\overline{\text{BOOT}}$ signal. The $\overline{\text{BOOT}}$ signal enables the PROM (U21) and enables the output of the Data In Register. The clock input of the Data In Register is supplied by the RUN signal through gate U32. The RUN signal was set low by initial start-up and remains so until DRLCLR presets the RUN flip-flop. The Data In Register consists of transparent latches so that under these conditions, the boot data from the PROM enters the internal data bus and transfers directly to the S-100 bus data lines.

When the block data transfer of the boot loader is completed, the phantom is disabled and the CPU and DMA modules initialize the circuit board so that it can accept commands.

4.1.6 INTERRUPTS.

The interrupt mode can be selected for AM-500 operation when the software program sets bit 7 in either disk or non-disk commands. When the interrupt mode is selected, the interrupt occurs at the completion of the current AM-500 command. This

is taken from the internal data bus by the control register (U11) and connected to the S-100 bus vectored interrupt lines. Any one of the seven interrupt lines (VI0-VI7) may be selected by a jumper from pin 3 of U35 to the selected line.

The interrupt mode may be cleared by the software program so the interrupts do not occur. The program clears the interrupt mode by setting bit 6 in any non-disk command to a one (see Table 4-2).

4.1.7 OUTPUT TO DISK DRIVE.

The AM-500 circuit board transfers disk write data and controls to the disk drive through the write data logic and drive control registers. See Table 2-3 for a complete list of the signals between the AM-500 circuit board and the disk drive.

4.1.7.1 DATA WRITE LOGIC.

The data write logic provides the double frequency encoded clock and data output necessary for the disk drive write circuitry. When the AM-500 is in the write mode, the WRITE signal is high, which sets the multiplexer in the CRC logic (U9) to select the write data (WDATA) from the FIFO memory stack to the CRC checker module U23. The serial data is then sent to the FIFO sync flip-flop U25 and also to the data write logic. Should a CRC error occur, CRCERR signal goes high and sets bit 7 of the drive status buffer.

When the FIFO memory is not empty, $\overline{\text{FEMPTY}}$ is high enabling the CRC checker and gating the serial data output of the FIFO through the write logic where it is combined with the $\overline{\text{ONE}}$ clock and applied to the $\overline{\text{WRITEDATA}}$ output to the disk drive. When the write current is flowing through the disk write head from the WGate command, the CELL clock is also combined into the $\overline{\text{WRITEDATA}}$ command through OR gate U43. When the FIFO memory is empty, $\overline{\text{FEMPTY}}$ is low, gating CRC data to the $\overline{\text{WRITEDATA}}$ line.

4.1.7.2 DRIVE REGISTERS.

The disk drive registers control the selection and operation of the disk mechanism. Three flip-flop registers (U11, U12, U13) perform these functions when enabled by the RUN signal.

Drive Register No. 1 controls the address data to the disk drive. The address data is contained on the internal data bus and is transferred into Drive Register No. 1 by CYLADR from the register select logic. When the software selects the cylinder address, CYLADR clocks the data from the internal data bus into Drive Register No. 1.

Drive Register No. 2 controls the disk and head select data to the disk drive. This data is contained on the internal data bus when the software selects the drive commands and is clocked into Drive Register No. 2 by the DRVSEL signal from the register select logic.

The Drive Control Register handles the gate signals for the disk read and write operations. This includes Write Gate (WGATE), Erase Gate (EGATE), Read Gate (RGATE), Cylinder Address Strobe (CYLSTR) and return to Zero Seek (RESTOR). In addition, this register generates the READ, WRITE, and DATA signals for internal control and the S-100 bus interrupt signal. This data is clocked into the Drive Control Register by the CNTL signal from the register select logic.

4.1.8 INPUT FROM DISK DRIVE.

The AM-500 circuit board receives memory data and status from the disk drive through the data read logic and drive status register. See Table 2-3 for a complete list of the signals between the circuit board and the disk drive.

4.1.8.1 DATA READ LOGIC.

The data read logic receives the separate clock and data outputs of the disk drive and applies them to the clock and preset inputs of a flip-flop. When the AM-500 is not in the read mode, the READ signal holds this flip-flop in the clear state. The flip-flop output (RDATA) and the input clock ($\overline{\text{RCLOCK}}$) are sent to the CRC logic and also to the FIFO memory. When the AM-500 is in the Read mode, the $\overline{\text{RCLOCK}}$ and RDATA are selected by the multiplexer in the CRC logic (U9) for checking by the CRC module U23 and also entered into the serial input of the FIFO memory gated by the FSYNC output of the CRC logic.

4.1.8.2 DRIVE STATUS BUFFER.

The Drive Status Buffer receives status information from the disk drive and also the error signal from the CRC checker. When the CPU interrogates the disk status from the AM-500 (see Tables 4-4 and 4-5), the $\overline{\text{DRVSTS}}$ output of the register select logic is low which transfers the status data from the disk drive to the internal data bus for transfer to the CPU. The data format is as follows:

<u>BIT</u>	<u>STATUS</u>
0	<u>INDEX</u>
1	<u>SECTOR</u>
2	<u>ONCYLINDER</u>
3	<u>SEEKERROR</u>
4	<u>FAULT</u>
5	<u>DRIVEREADY</u>
6	<u>WRITEPROTECTED</u>
7	<u>CRCERR</u>

See Table 2-3 for a description of the disk interface signals.

4.1.9 FIFO MEMORY AND CONTROL.

The FIFO Memory and Control circuitry consists of two 4-bit First-In-First-Out memory modules connected for a stack of 16 8-bit bytes. See paragraph 4.2.3 for a detailed description of the module. The FIFO memory is used for both disk read and write operations. It converts the serial data from the disk to parallel data for internal processing and transfer, and converts parallel data to serial for disk write operations. The data is transferred between the FIFO and 1K of the Random Access Memory (RAM) by the DMA controller. For memory full or memory empty conditions, the FIFO synchronizing logic makes the DMA service requests to the DMA controller to ensure that the DMA controller runs at the disk speed.

4.1.10 DMA BUFFER.

The DMA buffer is contained in the Random Access Memory (RAM) modules U5 and U6. All data transfers to or from the disk memory must pass through this buffer.

DMA requests (DREQ0 and DREQ1) are generated by the DMA request latches on U16. For DREQ0, the Data Write (DATWRT) command is received from the I/O command register, setting the DREQ0 flip-flop. The acknowledge from the DMA controller ($\overline{\text{DACK0}}$) and the I/O Read ($\overline{\text{IOR}}$) are combined to reset the DREQ0 flip-flop by $\overline{\text{DIOR}}$. DREQ1 flip-flop operates in a similar way; set by Data Read ($\overline{\text{DATRD}}$) and clocked by $\overline{\text{DIOW}}$. Both request flip-flops are cleared by DMA Request Clear $\overline{\text{DRLCLR}}$.

4.2 CIRCUIT MODULE DESCRIPTION.

This section describes the operation of the individual circuit packages (DIPS) contained on the AM-500 circuit board. Most of the processing is handled by the CPU and DMA controller modules, so these are described in detail. The control logic and interface modules are also described with logic and connection diagrams for each one.

4.2.1 CPU MICROPROCESSOR (U19).

The CPU microprocessor is a single DIP module that handles the data processing of the AM-500 circuit board.

Figure 4-2 is a block diagram of the CPU, and Figure 4-3 details the internal register configuration which contains 208 bits of Read/ Write memory that are accessible to the programmer. The registers include two sets of six general purpose registers that may be used individually as 8-bit registers or as 16-bit register pairs. There are also two sets of accumulator and flag registers. The programmer has access to either set of main or alternate registers through a group of exchange instructions. This alternate set allows foreground/background mode of operation or may be reserved for very fast Interrupt response. Each CPU also contains a 16-bit stack pointer which permits simple implementation of multiple level interrupts, unlimited subroutine nesting and simplification of many types of data handling.

The two 16-bit index registers allow tabular data manipulation and easy implementation of relocatable code. The Refresh register provides for automatic, totally transparent refresh of external dynamic memories. The I register is used in a powerful interrupt response mode to form the upper 8 bits of a pointer to an interrupt service address table, while the interrupting device supplies the lower 8 bits of the pointer. An indirect call is then made to this service address.

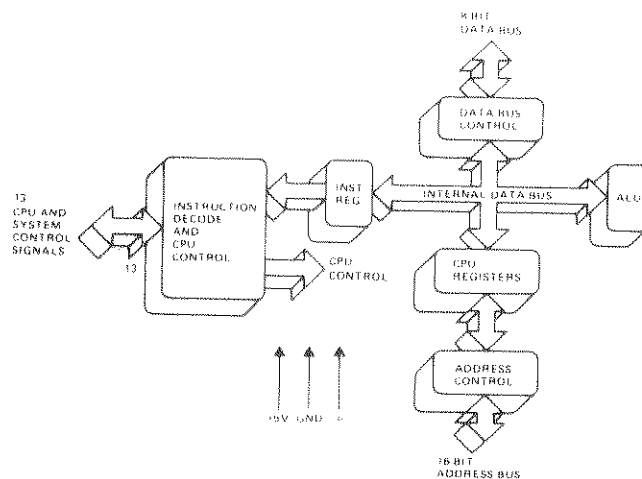


Figure 4-2. CPU Block Diagram

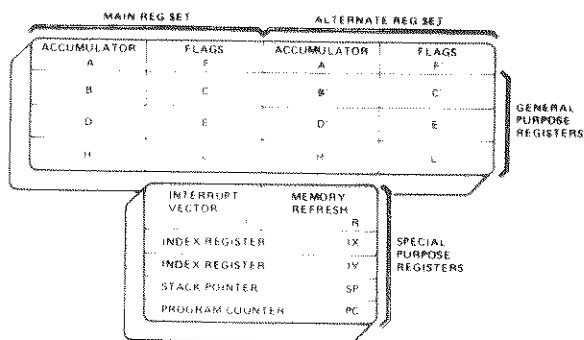


Figure 4-3. CPU Registers

Figure 4-4 shows the CPU pin in configuration and Table 4-7 contains a list of the CPU signals.

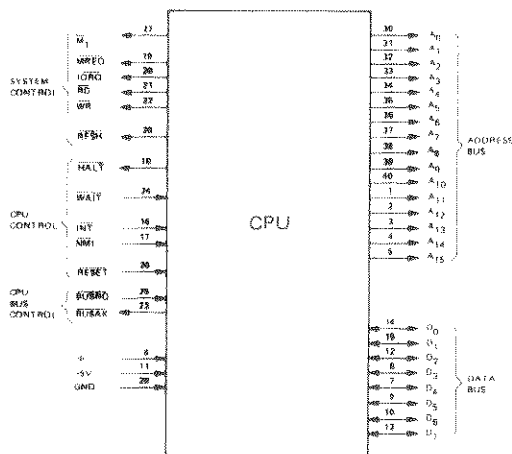


Figure 4-4. CPU Pin Configuration

Table 4-7. CPU Signal List

SIGNAL	PIN	FUNCTION
A0-A15 ADDRESS BUS		Tri-state output, active high. A0-A15 constitute a 16-bit address bus. The address bus provides the address for memory (up to 64K bytes) data exchanges and for I/O device data exchanges.
D0-D7 DATA BUS		Tri-state input/output, active high. D0-D7 constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.
$\overline{M1}$ MACHINE CYCLE ONE	27	Output, active low. $\overline{M1}$ indicates that the current machine cycle is the OP code fetch cycle of an instruction execution.
\overline{MREQ} MEMORY REQUEST	19	Tri-state output, active low. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.
\overline{IORQ} INPUT/OUTPUT REQUEST	20	Tri-state output, active low. The \overline{IORQ} signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An IORQ signal is also generated when an interrupt is being acknowledged to indicate that an

Table 4-7 (Cont.). CPU Signal List

SIGNAL	PIN	FUNCTION
		interrupt response vector can be placed on the data bus.
$\overline{\text{RD}}$ MEMORY READ	21	Tri-state output, active low. $\overline{\text{RD}}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.
$\overline{\text{WR}}$ MEMORY WRITE	22	Tri-state output, active low. $\overline{\text{WR}}$ indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.
$\overline{\text{RFSH}}$ REFRESH	28	Output, active low. $\overline{\text{RFSH}}$ indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and the current $\overline{\text{MREQ}}$ signal should be used to do a refresh read to all dynamic memories.
$\overline{\text{HALT}}$ HALT STATE	18	Output, active low. $\overline{\text{HALT}}$ indicates that the CPU has executed a $\overline{\text{HALT}}$ software instruction and is awaiting either a nonmaskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOP's to maintain memory refresh activity.

Table 4-7 (Cont.). CPU Signal List

SIGNAL	PIN	FUNCTION
$\overline{\text{WAIT}}$ WAIT	24	Input, active low. $\overline{\text{WAIT}}$ indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active.
$\overline{\text{INT}}$ INTERRUPT REQUEST	16	Input, active low. The Interrupt Request signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled.
$\overline{\text{NMI}}$ NON MASKABLE INTERRUPT	17	Input, active low. The non-maskable interrupt request line has a higher priority than $\overline{\text{INT}}$ and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. $\overline{\text{NMI}}$ automatically forces the CPU to restart to location 0066H.
$\overline{\text{RESET}}$	26	Input, active low. $\overline{\text{RESET}}$ initializes the CPU as follows: reset interrupt enable flip-flop, clear PC and registers I and R and set interrupt to 8080A mode. During reset time, the address and data bus go to a high impedance state

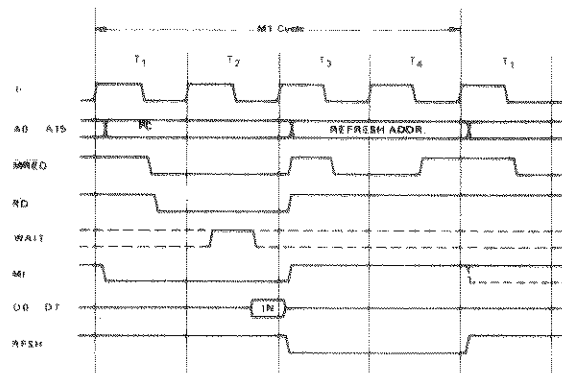
Table 4-7 (Cont.). CPU Signal List

SIGNAL	PIN	FUNCTION
		and all control output signals go to the inactive state.
$\overline{\text{BUSRQ}}$ BUS REQUEST	25	Input, active low. The bus request signal has a higher priority than $\overline{\text{NMI}}$ and is always recognized at the end of the current machine cycle and is used to request the CPU address bus, data bus and tri-state output control signals to go to a high impedance state so that other devices can control these busses.
$\overline{\text{BUSAK}}$ BUS ACKNOWLEDGE	23	Output, active low. Bus acknowledges is used to indicate to the requesting device that the CPU address bus, data bus and tri-state control bus signals have been set to their high impedance state and the external device can now control these signals.

4.2.1.1 INSTRUCTION OP-CODE FETCH.

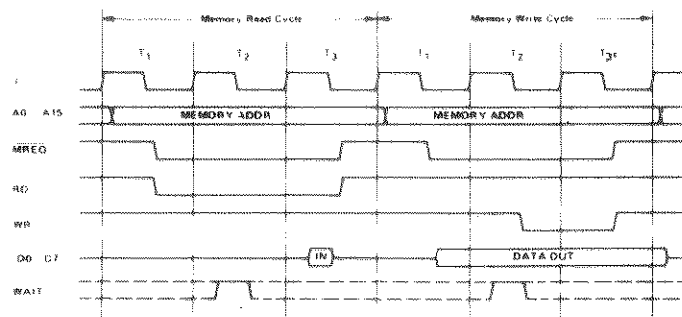
The program counter content (PC) is placed on the address bus immediately at the start of the cycle. One half clock time later $\overline{\text{MREQ}}$ goes active. The falling edge of $\overline{\text{MREQ}}$ can be used directly as a chip enable to dynamic memories. $\overline{\text{RD}}$ when active indicates that the memory data should be enabled onto the CPU data bus. The CPU samples data with the rising edge of the clock state T3. Clock states T3 and T4 of a fetch cycle are used to refresh dynamic memories while the CPU is internally decoding and executing the instruction. The

refresh control signal $\overline{\text{RFSH}}$ indicates that a refresh read of all dynamic memories should be accomplished.



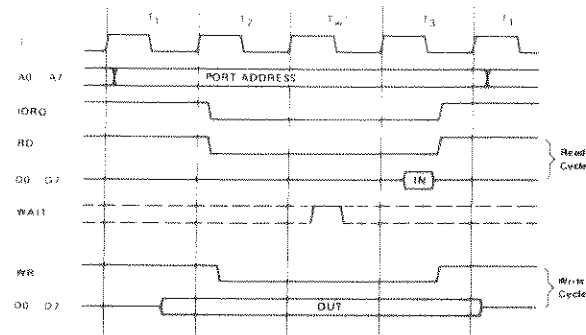
4.2.1.2 MEMORY READ OR WRITE CYCLES.

Illustrated here is the timing of memory read or write cycles other than an OP code fetch (M1 cycle). The $\overline{\text{MREQ}}$ and $\overline{\text{RD}}$ signals are used exactly as in the fetch cycle. In the case of a memory write cycle, the $\overline{\text{MREQ}}$ also becomes active when the address bus is stable so that it can be used directly as a chip enable for dynamic memories. The $\overline{\text{WR}}$ line is active when data on the data bus is stable so that it can be used directly as a R/W pulse to virtually any type of semiconductor memory.



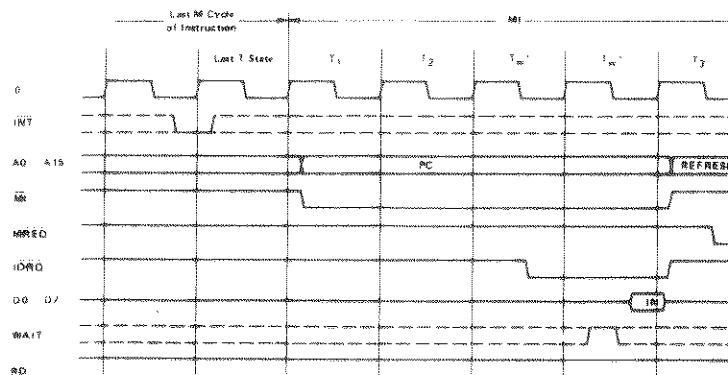
4.2.1.3 INPUT OR OUTPUT CYCLES.

Illustrated here is the timing for an I/O read or I/O write operation. Notice that during I/O operations a single wait state is automatically inserted (T_w^*). The reason for this is that during I/O operations this extra state allows sufficient time for an I/O port to decode its address and activate the $\overline{\text{WAIT}}$ line if a wait is required.



4.2.1.4 INTERRUPT REQUEST/ACKNOWLEDGE CYCLE.

The interrupt signal is sampled by the CPU with the rising edge of the last clock at the end of any instruction. When an interrupt is accepted, a special M1 cycle is generated. During this M1 cycle, the $\overline{\text{IORQ}}$ signal becomes active (instead of $\overline{\text{MREQ}}$) to indicate that the interrupting device can place an 8-bit vector on the data bus. Two wait states (T_w^*) are automatically added to this cycle so that a ripple priority interrupt scheme, such as the one used in the peripheral controllers, can be easily implemented.



4.2.1.5 CPU INSTRUCTION SET.

The following is a summary of the CPU instruction set showing the assembly language mnemonic and the symbolic operation performed by the instruction. The instructions are divided into the following categories:

8-bit loads	Miscellaneous Group
16-bit loads	Rotates and Shifts
Exchanges	Bit Set, Reset and Test
Memory Block Moves	Input and Output
Memory Block Searches	Jumps
8-bit arithmetic and logic	Calls
16-bit arithmetic	Restarts
General purpose Accumulator	Returns
& Flag Operations	

In the table the following terminology is used.

b	≡ a bit number in any 8-bit register or memory location	d	≡ any 8-bit destination register or memory location
cc	≡ flag condition code	dd	≡ any 16-bit destination register or memory location
NZ	≡ non zero	e	≡ 8-bit signed 2's complement displacement used in relative jumps and indexed addressing
Z	≡ zero	L	≡ 8 special call locations in page zero. In decimal notation these are 0, 8, 16, 24, 32, 40, 48 and 56
NC	≡ non carry	n	≡ any 8-bit binary number
C	≡ carry	nn	≡ any 16-bit binary number
PO	≡ Parity odd or no over flow	r	≡ any 8-bit general purpose register (A, B, C, D, E, H, or L)
PE	≡ Parity even or over flow	s	≡ any 8-bit source register or memory location
P	≡ Positive	sb	≡ a bit in a specific 8-bit register or memory location
M	≡ Negative (minus)	ss	≡ any 16-bit source register or memory location
		subscript "L"	≡ the low order 8 bits of a 16-bit register
		subscript "H"	≡ the high order 8 bits of a 16-bit register
		()	≡ the contents within the () are to be used as a pointer to a memory location or I/O port number
			8-bit registers are A, B, C, D, E, H, L, I and R
			16-bit register pairs are AF, BC, DE and HL
			16-bit registers are SP, PC, IX and IY

Addressing Modes implemented include combinations of the following:

Immediate	Indexed
Immediate extended	Register
Modified Page Zero	Implied
Relative	Register Indirect
Extended	Bit

Table 4-8. CPU Instruction Set

	Mnemonic	Symbolic Operation	Comments
8-BIT LOADS	LD r, s	$r \leftarrow s$	$s \equiv r, n, (HL), (IX+e), (IY+e)$
	LD d, r	$d \leftarrow r$	$d \equiv (HL), r, (IX+e), (IY+e)$
	LD a, n	$d \leftarrow n$	$d \equiv (HL), (IX+e), (IY+e)$
	LD A, s	$A \leftarrow s$	$s \equiv (BC), (DE), (nn), I, R$
	LD d, A	$d \leftarrow A$	$d \equiv (BC), (DE), (nn), I, R$
16-BIT LOADS	LD dd, nn	$dd \leftarrow nn$	$dd \equiv BC, DE, HL, SP, IX, IY$
	LD dd, (nn)	$dd \leftarrow (nn)$	$dd \equiv BC, DE, HL, SP, IX, IY$
	LD (nn), ss	$(nn) \leftarrow ss$	$ss \equiv BC, DE, HL, SP, IX, IY$
	LD SP, ss	$SP \leftarrow ss$	$ss = HL, IX, IY$
	PUSH ss	$(SP-1) \leftarrow ss_H, (SP-2) \leftarrow ss_L$	$ss = BC, DE, HL, AF, IX, IY$
EXCHANGES	POP dd	$dd_L \leftarrow (SP), dd_H \leftarrow (SP+1)$	$dd = BC, DE, HL, AF, IX, IY$
	EX DE, HL EX AF, AF' EXX	$DE \leftrightarrow HL$ $AF \leftrightarrow AF'$ $\begin{pmatrix} BC \\ DE \\ HL \end{pmatrix} \leftrightarrow \begin{pmatrix} BC' \\ DE' \\ HL' \end{pmatrix}$	
	EX (SP), ss	$(SP) \leftrightarrow ss_L, (SP+1) \leftrightarrow ss_H$	$ss \equiv HL, IX, IY$
MEMORY BLOCK MOVES	LDI	$(DE) \leftarrow (HL), DE \leftarrow DE+1$ $HL \leftarrow HL+1, BC \leftarrow BC-1$	
	LDIR	$(DE) \leftarrow (HL), DE \leftarrow DE+1$ $HL \leftarrow HL+1, BC \leftarrow BC-1$ Repeat until $BC = 0$	
	LDD	$(DE) \leftarrow (HL), DE \leftarrow DE-1$ $HL \leftarrow HL-1, BC \leftarrow BC-1$	
	LDDR	$(DE) \leftarrow (HL), DE \leftarrow DE-1$ $HL \leftarrow HL-1, BC \leftarrow BC-1$ Repeat until $BC = 0$	

Table 4-8 (Cont.). CPU Instruction Set

	Mnemonic	Symbolic Operation	Comments
MEMORY BLOCK SEARCHES	CPI	$A \leftarrow (HL), HL \leftarrow HL+1$ $BC \leftarrow BC-1$	
	CPIR	$A \leftarrow (HL), HL \leftarrow HL+1$ $BC \leftarrow BC-1$. Repeat until $BC = 0$ or $A = (HL)$	$A \leftarrow (HL)$ sets the flags only. A is not affected
	CPD	$A \leftarrow (HL), HL \leftarrow HL-1$ $BC \leftarrow BC-1$	
	CPDR	$A \leftarrow (HL), HL \leftarrow HL-1$ $BC \leftarrow BC-1$. Repeat until $BC = 0$ or $A = (HL)$	
8-BIT ALU	ADD s	$A \leftarrow A + s$	
	ADC s	$A \leftarrow A + s + CY$	CY is the carry flag
	SUB s	$A \leftarrow A - s$	
	SBC s	$A \leftarrow A - s - CY$	$s \equiv r, n, (HL)$
	AND s	$A \leftarrow A \wedge s$	$(IX+e), (IY+e)$
	OR s	$A \leftarrow A \vee s$	
	XOR s	$A \leftarrow A \oplus s$	
	CP s	$A - s$	$s = r, n (HL)$ $(IX+e), (IY+e)$
	INC d	$d \leftarrow d + 1$	$d = r, (HL)$ $(IX+e), (IY+e)$
16-BIT ARITHMATIC	DEC d	$d \leftarrow d - 1$	
	ADD HL, ss	$HL \leftarrow HL + ss$	$\left. \begin{array}{l} ss \equiv BC, DE \\ HL, SP \end{array} \right\}$
	ADC HL, ss	$HL \leftarrow HL + ss + CY$	
	SBC HL, ss	$HL \leftarrow HL - ss - CY$	
	ADD IX, ss	$IX \leftarrow IX + ss$	$ss \equiv BC, DE, IX, SP$
	ADD IY, ss	$IY \leftarrow IY + ss$	$ss \equiv BC, DE, IY, SP$
	INC dd	$dd \leftarrow dd + 1$	$dd \equiv BC, DE, HL, SP, IX, IY$
GP ACC. AND FLAG	DEC dd	$dd \leftarrow dd - 1$	$dd \equiv BC, DE, HL, SP, IX, IY$
	DAA	Converts A contents into packed BCD following add or subtract.	Operands must be in packed BCD format
	CPL	$A \leftarrow \overline{A}$	
	NEG	$A \leftarrow 00 - A$	
	CCF	$CY \leftarrow \overline{CY}$	
	SCF	$CY \leftarrow 1$	

Table 4-8 (Cont.). CPU Instruction Set

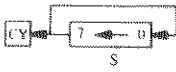
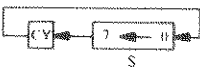
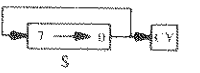
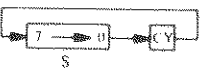

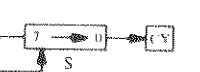



	Mnemonic	Symbolic Operation	Comments
MISCELLANEOUS	NOP	No operation	
	HALT	Halt CPU	
	DI	Disable Interrupts	
	EI	Enable Interrupts	
	IM 0	Set interrupt mode 0	8080A mode
	IM 1	Set interrupt mode 1	Call to 0038H
	IM 2	Set interrupt mode 2	Indirect Call
ROTATES AND SHIFTS	RLC s		
	RL s		
	RRC s		
	RR s		
	SLA s		$s \equiv r, (HL)$ $(IX+e), (IY+e)$
	SRA s		
	SRL s		
	RLD		
	RRD		
BIT S, R, AND T	BIT b, s	$Z \leftarrow \overline{s_b}$	Z is zero flag
	SET b, s	$s_b \leftarrow 1$	$s \equiv r, (HL)$
	RES b, s	$s_b \leftarrow 0$	$(IX+e), (IY+e)$
INPUT AND OUTPUT	IN A, (n)	$A \leftarrow (n)$	Set flags
	IN r, (C)	$r \leftarrow (C)$	
	INI	$(HL) \leftarrow (C), HL \leftarrow HL + 1$ $B \leftarrow B - 1$	
	INIR	$(HL) \leftarrow (C), HL \leftarrow HL + 1$ $B \leftarrow B - 1$ Repeat until $B = 0$	
	IND	$(HL) \leftarrow (C), HL \leftarrow HL - 1$ $B \leftarrow B - 1$	
	INDR	$(HL) \leftarrow (C), HL \leftarrow HL - 1$ $B \leftarrow B - 1$ Repeat until $B = 0$	

Table 4-8 (Cont.). CPU Instruction Set

	Mnemonic	Symbolic Operation	Comments
INPUT AND OUTPUT	OUT(n), A	$(n) \leftarrow A$	
	OUT(C), r	$(C) \leftarrow r$	
	OUTI	$(C) \leftarrow (HL), HL \leftarrow HL + 1$ $B \leftarrow B - 1$	
	OTIR	$(C) \leftarrow (HL), HL \leftarrow HL + 1$ $B \leftarrow B - 1$ Repeat until $B = 0$	
	OUTD	$(C) \leftarrow (HL), HL \leftarrow HL - 1$ $B \leftarrow B - 1$	
	OTDR	$(C) \leftarrow (HL), HL \leftarrow HL - 1$ $B \leftarrow B - 1$ Repeat until $B = 0$	
JUMPS	JP nn	$PC \leftarrow nn$	$cc \begin{cases} NZ & PO \\ Z & PE \\ NC & P \\ C & M \end{cases}$ $kk \begin{cases} NZ & NC \\ Z & C \end{cases}$ $ss = HL, IX, IY$
	JP cc, nn	If condition cc is true $PC \leftarrow nn$, else continue	
	JR e	$PC \leftarrow PC + e$	
	JR kk, e	If condition kk is true $PC \leftarrow PC + e$, else continue	
	JP (ss)	$PC \leftarrow ss$	
	DJNZ e	$B \leftarrow B - 1$, if $B \neq 0$ continue, else $PC \leftarrow PC + e$	
CALLS	CALL nn	$(SP-1) \leftarrow PC_H$ $(SP-2) \leftarrow PC_L, PC \leftarrow nn$	$cc \begin{cases} NZ & PO \\ Z & PE \\ NC & P \\ C & M \end{cases}$
	CALL cc, nn	If condition cc is false continue, else same as CALL nn	
RESTARTS	RST L	$(SP-1) \leftarrow PC_H$ $(SP-2) \leftarrow PC_L, PC_H \leftarrow 0$ $PC_L \leftarrow L$	
RETURNS	RET	$PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP+1)$	$cc \begin{cases} NZ & PO \\ Z & PE \\ NC & P \\ C & M \end{cases}$
	RET cc	If condition cc is false continue, else same as RET	
	RETI	Return from interrupt, same as RET	
	RETN	Return from non- maskable interrupt	

4.2.2 DMA CONTROLLER (U20).

This device is a multimode Direct Memory Access (DMA) controller for microprocessor systems. It enhances system performance by allowing other devices to directly transfer information to or from memory or to transfer data from one memory to another. Figure 4-5 shows the DMA controller connections and Table 4-9 lists these signals with their functions.

The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Autoinitialize to its original condition following an End of Process (\overline{EOP}).

Each channel has a full 64K address and word count capability. An external \overline{EOP} signal can terminate a DMA or memory-to-memory transfer. This is useful for block search or compare operations using external comparators or for intelligent peripherals to abort erroneous services.

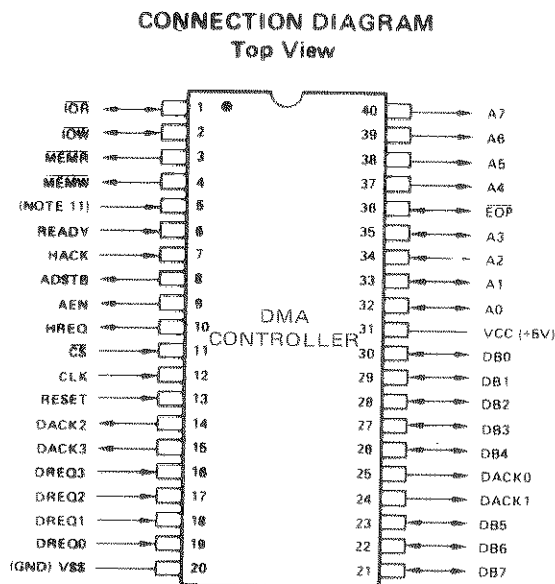


Figure 4-5. DMA Controller Connections

Table 4-9. DMA Controller Signal List

SIGNAL	PIN	FUNCTION
VCC	31	+5 Volt Supply
VSS	20	Ground
CLK CLOCK, INPUT	12	This input controls the internal operations of the DMA Controller and its rate of data transfers. The input may be driven at up to 3MHz for the standard DMA Controller and up to 4MHz for the DMA Controller.
$\overline{\text{CS}}$ CHIP SELECT, INPUT	11	Chip Select is an active low input used to select the DMA Controller as an I/O device during the Idle cycle. This allows CPU communication on the data bus.
RESET RESET, INPUT	13	Reset is an asynchronous active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip/flop and sets the Mask register. Following a Reset the device is in the Idle cycle.
READY READY, INPUT	6	Ready is an asynchronous input used to extend the memory read and write pulses from the DMA Controller to accommodate slow memories or I/O peripheral devices.

Table 4-9 (Cont.). DMA Controller Signal List

SIGNAL	PIN	FUNCTION
HACK HOLD ACKNOWLEDGE, INPUT	7	The active high Hold Acknowledge from the CPU indicates that control of the system buses has been relinquished.
DREQ0-DREQ3 DMA REQUEST, INPUT	19, 18, 17, 16	The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In Fixed Priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of the DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high.
DB0-DB7 DATA BUS, INPUT/OUTPUT	30, 29, 28, 27, 26, 23, 22, 21	The Data Bus lines are bidirectional three-state signals connected to the system data bus. The outputs are enabled in the Program Condition during the I/O Read to output the contents of an Address register, a Status register, the Temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O Write cycle when the CPU is programming the DMA controller control registers. During DMA cycles, the most significant eight bits of

Table 4-9 (Cont.). DMA Controller Signal List

SIGNAL	PIN	FUNCTION
		the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the DMA Controller on the data bus during the read-from-memory transfer. In the write-to-memory transfer, the data bus outputs place the data into the new memory location.
$\overline{\text{IOR}}$ I/O READ, INPUT/OUTPUT	1	I/O Read is a bidirectional, active low, three-state line. In the Idle cycle, it is an input control signal used by the CPU to read the control registers. In the Active cycle, it is an output control signal used by the DMA Controller to access data from a peripheral during a DMA Write transfer.
$\overline{\text{IOW}}$ I/O WRITE, INPUT/OUTPUT	2	I/O Write is a bidirectional, active low, three-state line. In the idle cycle, it is an input control signal used by the CPU to load information into the DMA Controller. In the Active cycle, it is an output control signal used by the DMA Controller to load data to the peripheral during a DMA Read transfer.

Table 4-9 (Cont.). DMA Controller Signal List

SIGNAL	PIN	FUNCTION
$\overline{\text{EOP}}$ END OF PROCESS, INPUT/OUTPUT	36	<p>$\overline{\text{EOP}}$ is an active low, bidirectional signal. Information concerning the completion of DMA services is available at the bi-directional End of Process ($\overline{\text{EOP}}$) pin. The DMA Controller allows an external signal to terminate an active DMA service. This is accomplished by pulling the $\overline{\text{EOP}}$ input low with an external $\overline{\text{EOP}}$ signal. The DMA Controller also generates a pulse when the terminal count (TC) for any channel is reached. This generates an $\overline{\text{EOP}}$ signal which is output through the $\overline{\text{EOP}}$ line. The reception of $\overline{\text{EOP}}$, either internal or external, will cause the DMA Controller to terminate the service, reset the request; and if Autoinitialize is enabled, to write the base registers to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by $\overline{\text{EOP}}$ unless the channel is programmed for Autoinitialize. In that case, the mask bit remains clear. During memory-to-memory transfers, $\overline{\text{EOP}}$ will be output when the TC for channel 1 occurs.</p>

Table 4-9 (Cont.). DMA Controller Signal List

SIGNAL	PIN	FUNCTION
A0-A3 ADDRESS, INPUT/OUTPUT	32, 33, 34, 35	The four least significant address lines are bidirectional, three-state signals. In the Idle cycle, they are inputs and are used by the DMA Controller to address the control register to be loaded or read. In the Active cycle, they are outputs and provide the lower 4 bits of the output address.
A4-A7 ADDRESS, OUTPUT	37, 38, 39, 40	The four most significant address lines are three-state outputs and provide four bits of address. These lines are enabled only during the DMA service.
HREQ HOLD REQUEST, OUTPUT	10	This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes the DMA controller to issue the HREQ.
DACK0-DACK3 DMA ACKNOWLEDGE, OUTPUT	25, 24, 17, 16	DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The sense of these lines is programmable. Reset initializes them to active low.

Table 4-9 (Cont.). DMA Controller Signal List

SIGNAL	PIN	FUNCTION
AEN ADDRESS ENABLE, OUTPUT	9	The Address Enable is an active high level used to enable the output of the external latch which holds the upper byte of address, and to disable the system bus during the DMA cycle. Note that during DMA transfers, HACK and AEN should be used to deselect all other I/O peripherals which may erroneously be accessed as programmed I/O during the DMA operation. The DMA Controller automatically de-selects itself during DMA transfer.
ADSTB ADDRESS STROBE, OUTPUT	8	The active high Address Strobe is used to strobe the upper address byte into an external latch.
$\overline{\text{MEMR}}$ MEMORY READ, OUTPUT	3	The Memory Read signal is an active low, three-state output used to access data from the selected memory location during a DMA Read or a memory-to-memory transfer.
$\overline{\text{MEMW}}$ MEMORY WRITE, OUTPUT	4	The Memory Write signal is an active low, three-state output used to write data to the selected memory location during a DMA Write or a memory-to-memory transfer.

4.2.2.1 DMA CONTROLLER FUNCTIONAL DESCRIPTION.

The DMA Controller block diagram shown in Figure 4-6 includes the major logic blocks and all of the internal registers. The data interconnection paths are also shown. Not shown are the various control signals between the blocks. The DMA Controller contains 344 bits of internal memory in the form of registers. Figure 4-10 lists these registers by name and shows the size of each. A detailed description of the registers and their functions can be found under Register Description.

Table 4-10. DMA Controller Internal Registers

Name	Size	Number
Base Address Registers	16 bits	4
Base Word Count Registers	16 bits	4
Current Address Registers	16 bits	4
Current Word Count Registers	16 bits	4
Temporary Address Register	16 bits	1
Temporary Word Count Register	16 bits	1
Status Register	8 bits	1
Command Register	8 bits	1
Temporary Register	8 bits	1
Mode Registers	6 bits	4
Mask Register	4 bits	1
Request Register	4 bits	1

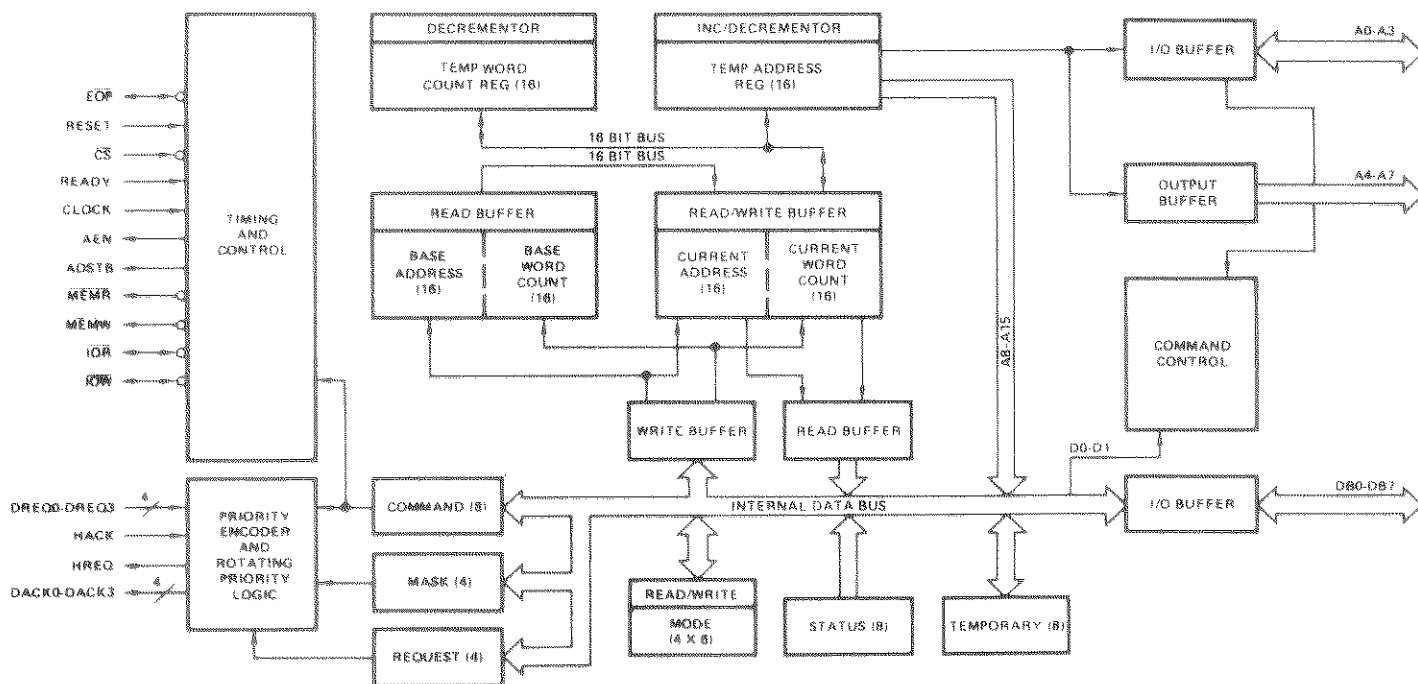


Figure 4-6. DMA Controller Block Diagram

The DMA Controller contains three basic blocks of control logic. The Timing Control block generates internal timing and external control signals for the DMA Controller. The Program Command Control block decodes the various commands given to the DMA Controller by the microprocessor prior to servicing a DMA Request. It also decodes the Mode Control word used to select the type of DMA during the servicing. The Priority Encoder block resolves priority contention between DMA channels requesting service simultaneously.

The Timing Control block derives internal timing from the clock input.

4.2.2.2 DMA OPERATION

The DMA Controller operates in two major cycles. These are called Idle and Active cycles. Each device cycle is made up of a number of states. The DMA Controller can assume seven separate states, each composed of one full clock period. State 1 (S1) is the inactive state. It is entered when the DMA Controller has no valid DMA requests pending. While in S1, the DMA controller is inactive but may be in the Program Condition, being programmed by the processor. State 0 (S0) is the first state of a DMA service. The DMA Controller has requested a hold, but the processor has not yet returned an acknowledge. An acknowledge from the CPU signals that transfers may begin. S1, S2, S3 and S4 are working states of the DMA service. If more time is needed to complete a transfer than is available with normal timing, wait states (SW) can be inserted between S2 or S3 and S4 by the use of the Ready line on the DMA Controller.

Memory-to-memory transfers require a read-from and a write-to-memory to complete each transfer. The states, which resemble the normal working states, use two digit numbers for identification. Eight states are required for a single transfer. The first four

states (S11, S12, S13, S14) are used for the read-from-memory half and the last four states (S21, S22, S23 and S24) for the write-to-memory half of the transfer.

4.2.2.3 IDLE CYCLE.

When no channel is requesting service, the DMA Controller will enter the idle cycle and perform "SI" states. In this cycle, the DMA Controller samples the DREQ lines every clock cycle to determine if any channel is requesting a DMA service. The device also samples \overline{CS} , looking for an attempt by the microprocessor to write or read the internal registers of the DMA Controller. When \overline{CS} is low and HREQ is low, the DMA Controller enters the Program Condition. The CPU can now establish, change or inspect the internal definition of the part by reading from or writing to the internal registers. Address lines A0-A3 are inputs to the device and select which registers are to be read or written. The \overline{IOR} and \overline{IOW} lines are used to select and time reads or writes. Due to the number and size of the internal registers, an internal flip/flop is used to generate an additional bit of address. This bit is used to determine the upper or lower byte of the 16-bit Address and Word Count registers. The flip/flop is reset by Master Clear or Reset. A separate software command can also reset this flip/flop.

Special software commands can be executed by the DMA Controller in the Program Condition. These commands are decoded as sets of addresses with \overline{CS} and \overline{IOW} . The commands do not make use of the data bus. Instructions include Clear First/Last Flip/Flop and Master Clear.

4.2.2.4 ACTIVE CYCLE

When the DMA Controller is in the Idle cycle and a channel requests a DMA service, the device outputs a HREQ to the microprocessor and enters the Active cycle. It is in this cycle that the DMA service takes place in one of three modes:

Single Transfer Mode. In Single Transfer mode, the device is programmed to make one transfer only. The word count is decremented and the address decremented or incremented following each transfer. When the word count goes to zero, a Terminal Count (TC) causes an Autoinitialize if the channel has been programmed to do so.

DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the single transfer, HREQ goes inactive and releases the bus to the system. It will again go active and upon receipt of a new HACK, another single transfer is performed. In 8080A/9080A systems, this will ensure one full machine cycle execution between DMA transfers.

Block Transfer Mode. In Block Transfer mode, the device is activated by DREQ to continue making transfers during the service until a TC, caused by the word count going to zero, or an external End of Process (EOP) is encountered. DREQ need only be held active until DACK becomes active. Again, an Autoinitialize occurs at the end of the service if the channel has been programmed for it.

Demand Transfer Mode. In Demand Transfer mode, the device is programmed to continue making transfers until a TC or external $\overline{\text{EOP}}$ is encountered or until DREQ goes inactive. Thus transfers may continue until the I/O device has exhausted its data capacity. After the I/O device has had a chance to catch up, the DMA service is re-established by means of a DREQ. During the time between services when the microprocessor is allowed to operate, the intermediate values of address and word count are stored in the DMA Controller Current Address and Current Word Count registers. Only an $\overline{\text{EOP}}$ can cause an Autoinitialize at the end of the service. $\overline{\text{EOP}}$ is generated either by TC or by an external signal.

4.2.2.5 TRANSFER TYPES.

Each of the three active transfer modes can perform three different types of transfers. These are Read, Write and Verify. Write transfers move data from an I/O device to the memory by activating $\overline{\text{MEMW}}$ and $\overline{\text{IOR}}$. Read transfers move data from memory to an I/O device by activating MEMR and IOW . Verify transfers are pseudo transfers. The DMA Controller operates as in Read or Write transfers generating addresses, and responding to $\overline{\text{EOP}}$, etc. However, the memory and I/O control lines all remain inactive.

Memory-to-Memory. To perform block moves of data from one memory address space to another with a minimum of program effort and time, the DMA Controller includes memory-to-memory transfer feature. Programming a bit in the Command register selects channels 0 and 1 to operate as memory-to-memory transfer channels. The transfer is initiated by setting the software DREQ for channel 0. The DMA Controller requests a DMA service in the normal manner. After HACK is true, the device, using eight-state transfers in Block Transfer mode, reads data from the memory. The channel 0 Current Address register is the source for the address used and is decremented or incremented in the normal manner. The data byte read from the memory is stored in the DMA Controller internal Temporary register. Channel 1 then writes the data from the Temporary register to memory using the address in its Current Address register and incrementing or decrementing it in the normal manner. The channel 1 Current Word Count is decremented. When the word count of channel 1 goes to zero, a TC is generated causing an $\overline{\text{EOP}}$ output terminating the service.

Channel 0 may be programmed to retain the same address for all transfers. This allows a single word to be written to a block of memory.

The DMA Controller responds to external $\overline{\text{EOP}}$ signals during memory-to-memory transfers. Data comparators in block search schemes

may use this input to terminate the service when a match is found.

Autoinitialize. By programming a bit in the Mode register, a channel may be set up as an Autoinitialize channel. During Auto-initialization, the original values of the Current Address and Current Word Count registers are automatically restored from the Base Address and Base Word Count registers of that channel following \overline{EOP} . The base registers are loaded simultaneously with the current registers by the microprocessor and remain unchanged throughout the DMA service. The mask bit is not set when the channel is in Autoinitialize. Following Autoinitialize, the channel is ready to perform another service without CPU intervention.

Priority. The DMA Controller has two types of priority encoding available as software selectable options. The first is Fixed Priority which fixes the channels in priority order based upon the descending value of their number. The channel with the lowest priority is 3 followed by 2, 1 and the highest priority channel, 0. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is complete.

The second scheme is Rotating Priority. The last channel to get service becomes the lowest priority channel with the others rotating accordingly.



With Rotating Priority in a single chip DMA system, any device requesting service is guaranteed to be recognized after no more than three higher priority services have occurred. This prevents any one channel from monopolizing the system.

Compressed Timing. In order to achieve even greater throughput where system characteristics permit, the DMA Controller can compress the transfer time to two clock cycles. State S3 is used to extend the access time of the read pulse. By removing state S3, the read pulse width is made equal to the write pulse width; and a transfer consists only of state S2 to change the address and state S4 to perform the read/write. S1 states still occur when A8-A15 need updating (see Address Generation).

Address Generation. In order to reduce pin count, the DMA Controller multiplexes the eight higher order address bits on the data lines. State S1 is used to output the higher order address bits to an external latch from which they may be placed on the address bus. The falling edge of Address Strobe (ADSTB) is used to load these bits from the data lines to the latch. Address Enable (AEN) is used to enable the bits onto the address bus through a three-state enable. The lower order address bits are output directly. Lines A0-A7 should be connected to the address bus.

During Block and Demand Transfer mode services which include multiple transfers, the addresses generated are sequential. For many transfers, the data held in the external address latch remains the same. This data need only change when a carry or borrow from A7 to A8 takes place in the normal sequence of addresses. To save time and speed transfers, the DMA Controller executes S1 states only when updating of A8-A15 in the latch is necessary. This means for long services, S1 states may occur only once every 256 transfers, a savings of 255 clock cycles for each 256 transfers.

4.2.2.6 REGISTER DESCRIPTION.

Current Address Register. Each channel has a 16-bit Current Address register. This register holds the value of the address used during DMA transfers. The address is automatically incre-

mented or decremented after each transfer, and the intermediate values of the address are stored in the Current Address register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. It may also be reinitialized by an Autoinitialize back to its original value. Autoinitialization takes place only after an $\overline{\text{EOP}}$.

Current Word Count Register. Each channel has a 16-bit Current Word Count register. This register holds the number of transfers to be performed. The word count is decremented after each transfer. The intermediate value of the word count is stored in the register during the transfer. When the value in the register goes to zero, a TC is generated. This register is loaded or read in successive 8-bit bytes by the microprocessor in the Program Condition. Following the end of a DMA service, it may also be reinitialized by an Autoinitialize back to its original value. Autoinitialize can occur only when an $\overline{\text{EOP}}$ occurs.

Base Address and Base Word Count Registers. Each channel has a pair of Base Address and Base Word Count registers. These 16-bit registers store the original values of their associated current registers. During Autoinitialize, these values are used to restore the current registers to their original values. The base registers are written simultaneously with their corresponding current register in 8-bit bytes in the Program Condition by the microprocessor. These registers cannot be read by the microprocessor.

Command Register. This 8-bit register controls the operation of the DMA Controller. It is programmed by the microprocessor in the Program Condition and is cleared by Reset. See Figure 4-7 for the function of the command bits and Table 4-11 for address coding.

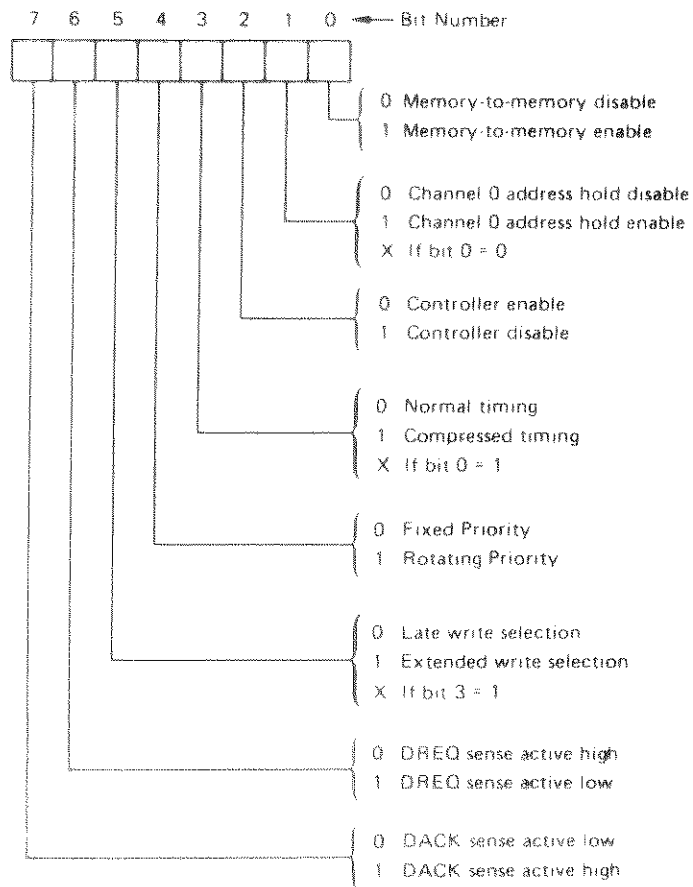


Figure 4-7. DMA Controller Command Bits

Table 4-11. DMA Controller Word Count
and Address Register Command Codes

Channel	Register	Operation	Signals							Internal Flip/Flop	Data Bus DB0—DB7
			CS	IOR	IOW	A3	A2	A1	A0		
0	Base & Current Address	Write	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	0	0	1	0	W0-W7
			0	1	0	0	0	0	1	1	W8-W15
1	Base & Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	0	1	1	0	W0-W7
			0	1	0	0	0	1	1	1	W8-W15
2	Base & Current Address	Write	0	1	0	0	1	0	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	1	0	1	0	W0-W7
			0	1	0	0	1	0	1	1	W8-W15
3	Base & Current Address	Write	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
4	Base & Current Address	Write	0	1	0	0	1	1	1	0	A0-A7
			0	1	0	0	1	1	1	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	1	0	A0-A7
			0	0	1	0	1	1	1	1	A8-A15
	Base & Current Word Count	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15

Mode Register. Each channel has a 6-bit Mode register associated with it. When the register is being written to by the micro-processor in the Program Condition, bits 0 and 1 determine which channel Mode register is to be written. See Figure 4-8 for the mode register bit configuration.

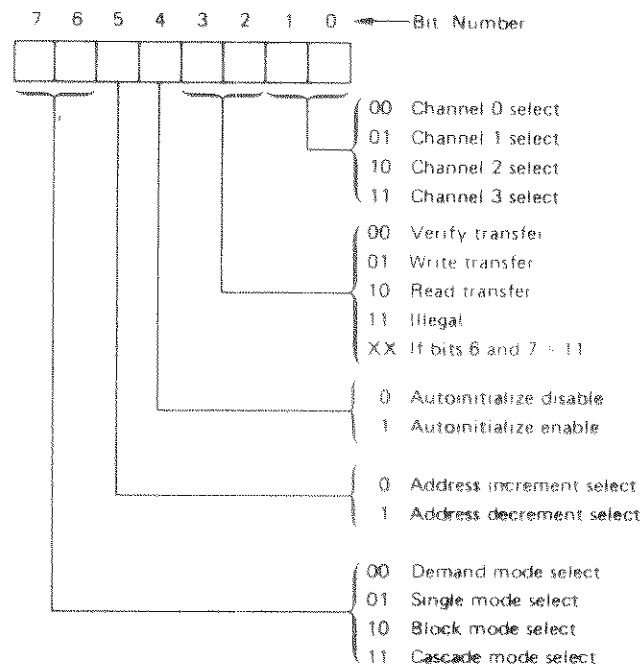


Figure 4-8. DMA Controller Mode Register Bits

Request Register. The DMA Controller can respond to requests for DMA service which are initiated by software as well as by a DREQ. Each channel has a request bit associated with it in the four bit Request register. These are nonmaskable and subject to prioritization by the Priority Encoder network. Each register bit is set or reset separately under software control or is cleared upon generation of a TC or external \overline{EOP} . The entire register is cleared by a Reset. To set or reset a bit, the software loads the proper form of the data word. See Figure 4-9 for the request register bit configuration and Table 4-11 for address coding.

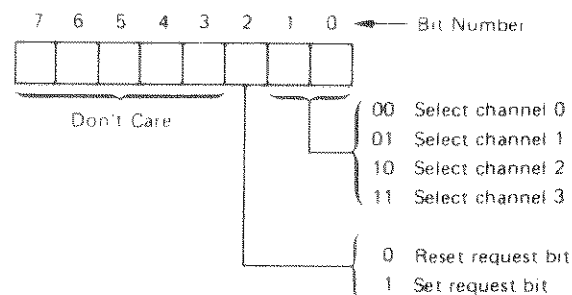
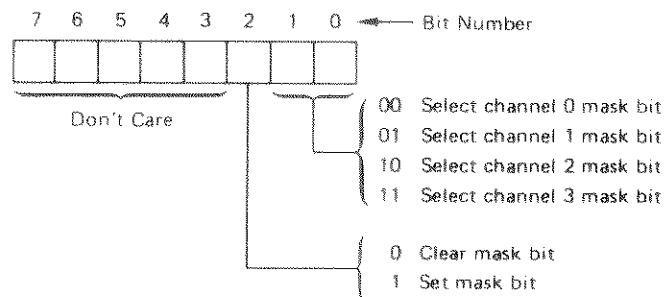


Figure 4-9. DMA Controller Request Register Bits

Software requests will be serviced only if the channel is in Block mode. When initiating a memory-to-memory transfer, the software request for channel 0 should be set.

Mask Register. Each channel has associated with it a mask bit which can be set to disable the incoming DREQ. Each mask bit is set when its associated channel produces an \overline{EOP} if the channel is not programmed for Autoinitialize. Each bit of the four bit Mask register may also be set or cleared separately under software control. The entire register is also set by a Reset. This disables all DMA requests until a clear Mask register instruction allows them to occur. The instruction to separately set or clear the mask bits is similar in form to that used with the Request register. See Figure 4-10 for the mask register bit configuration and Table 4-12 for Register instruction codes.



All four bits of the Mask register may also be written with a single command.

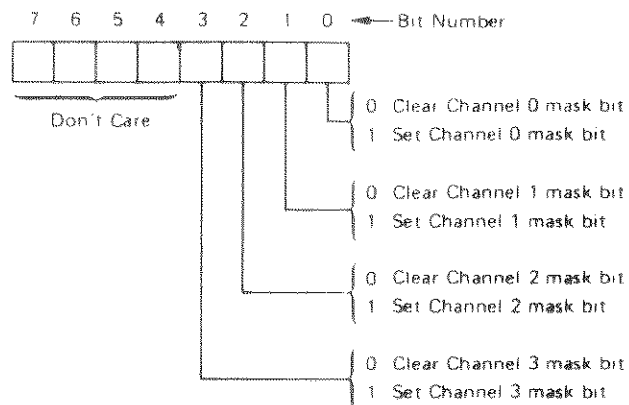


Figure 4-10. DMA Controller Mask Register Bits

Table 4-12. DMA Controller Register Codes

Register	Operation	Signals						
		\overline{CS}	\overline{IOR}	\overline{IOW}	A3	A2	A1	A0
Command	Write	0	1	0	1	0	0	0
Mode	Write	0	1	0	1	0	1	1
Request	Write	0	1	0	1	0	0	1
Mask	Set/Reset	0	1	0	1	0	1	0
Mask	Write	0	1	0	1	1	1	1
Temporary	Read	0	0	1	1	1	0	1
Status	Read	0	0	1	1	0	0	0

Status Register. The Status register is available to be read out of the DMA Controller by the microprocessor. It contains information about the status of the device at that point. This information includes which channels have reached a terminal count and which channels have pending DMA requests. Bits 0-3 are set every time a TC is reached by that channel. These bits are cleared upon Reset and on each Status Read. Bits 4-7 are set whenever their corresponding channel is requesting service. See Figure 4-11 for status register bit configuration.

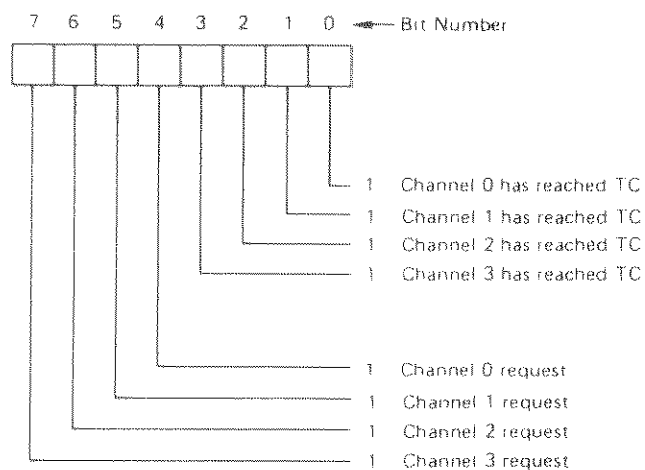


Figure 4-11. DMA Controller Status Register Bits

Temporary Register. The Temporary register is used to hold data during memory-to-memory transfers. Following the completion of the transfers, the last word moved can be read by the microprocessor in the Program Condition. The Temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a Reset.

Software Commands. These are additional special software commands which can be executed in the Program Condition. They do not depend on any specific bit pattern on the data bus. The two software commands are:

Clear First/Last Flip/Flop: This command is executed prior to writing or reading new address or word count information to the DMA Controller. This initializes the flip/flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and Internal First/Last Flip/Flop registers are cleared and the Mask register is set. The DMA Controller enters the Idle cycle. Table 4-13 lists the address codes for the software commands.

Table 4-13. DMA Controller Software
Command Codes

Operation	Registers Affected	Signals							
		CS	IOR	IOW	A3	A2	A1	A0	
Clear FF	Internal First/Last Flip/Flop	0	1	0	1	1	0	0	
Master Clear	Clear: Command Status Request Temporary Internal First/Last Flip/Flop Set: Mask	0	1	0	1	1	0	1	

4.2.3 FIRST-IN FIRST-OUT (FIFO) BUFFER MEMORY (U7, U8).

This device is an expandable fall-through type, high-speed First-In First-Out (FIFO) buffer memory optimized for high speed disc or tape controllers and communications buffers. It is organized as 16 words by four bits and may be expanded to any number of words or any number of bits (in multiples of 4). Data may be entered or extracted asynchronously in serial or parallel. The device has tri-state outputs. See Figure 4-12 for the FIFO Buffer connections.

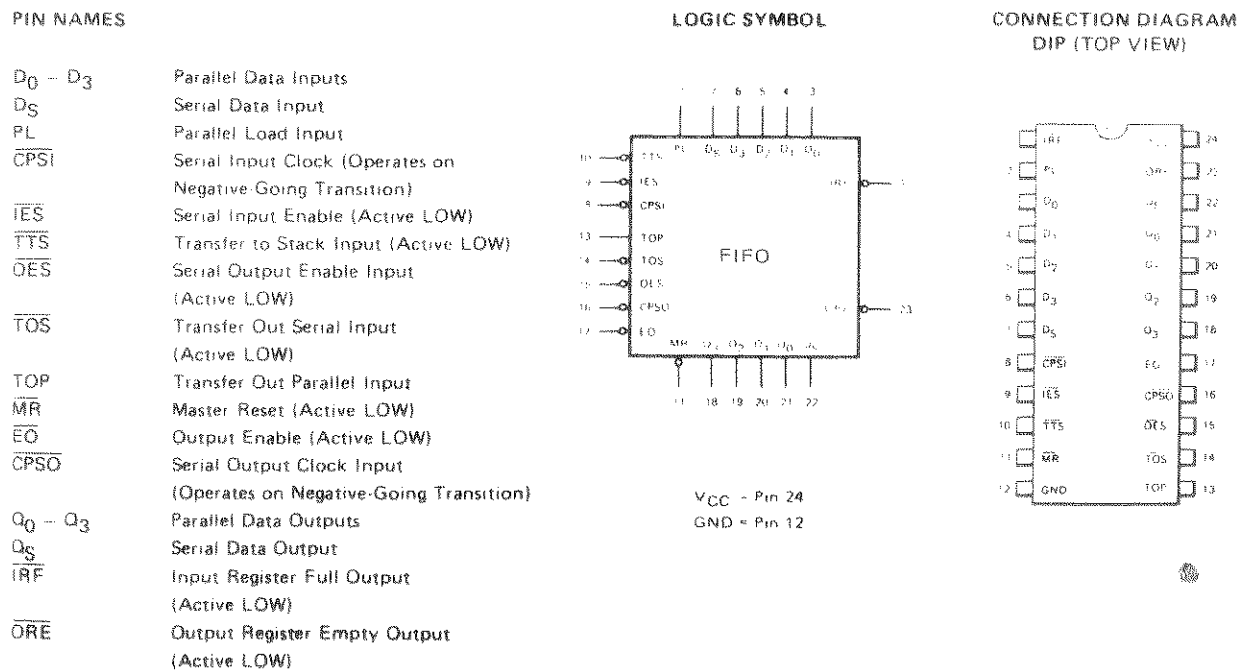


Figure 4-12. FIFO Buffer Connections

4.2.3.1 FUNCTIONAL DESCRIPTION.

As shown in the Block Diagram in Figure 4-13, the device consists of three parts:

1. An input Register with Parallel and Serial Data Inputs as well as control inputs and outputs for input hand-shaking and expansion.
2. A 4-bit wide, 14-word deep Fall-Through Stack with self-contained control logic.
3. An Output Register with Parallel and Serial Data Outputs as well as control inputs and outputs for output hand-shaking and expansion.

Since these three sections operate asynchronously and almost independently, they will be described separately in the following paragraphs.

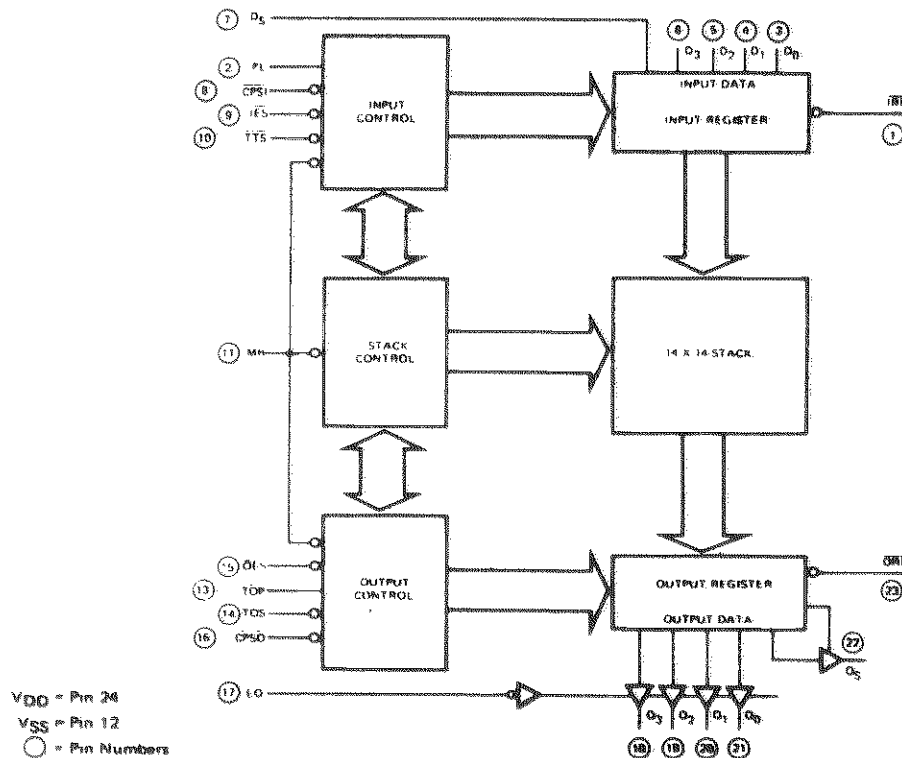


Figure 4-13. FIFO Block Diagram

4.2.3.2 INPUT REGISTER (DATA ENTRY).

The Input Register can receive data in either bit-serial or in 4-bit parallel form, store it until it is sent to the Fall-Through Stack and generate and accept the necessary status and control signals.

Figure 4-14 is a conceptual logic diagram of the input section. As described later, this 5-bit register is initialized by setting the F3 Flip-Flop and resetting the other flip-flops. The \overline{Q} Output of the last Flip-Flop (FC) is brought out as the "Input Register Full" output (\overline{IRF}). After initialization this output is HIGH.

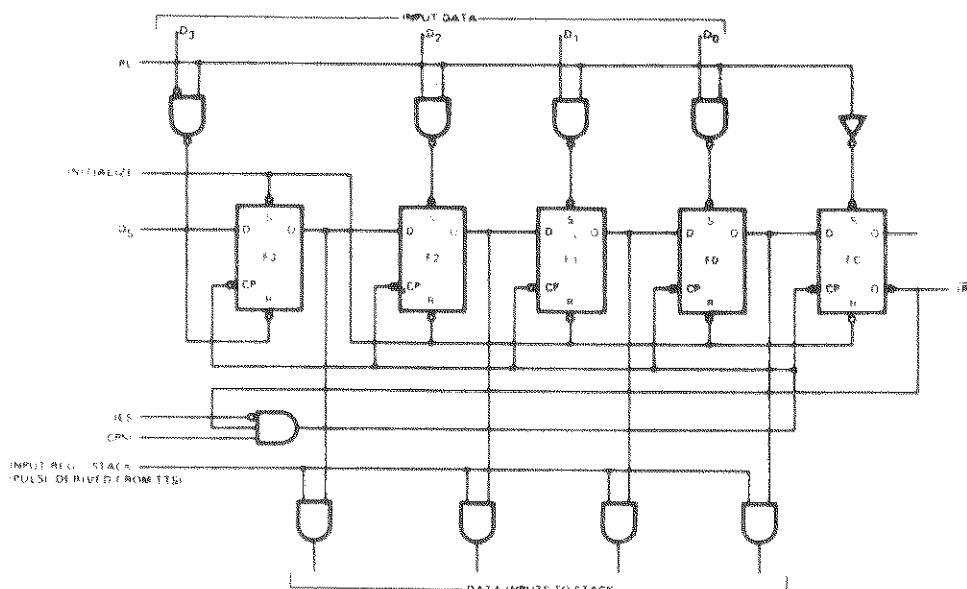


Figure 4-14. FIFO Conceptual Input Section

Parallel Entry. A HIGH level on the PL Input loads the D0-D3 Data Inputs into the F0-F3 Flip-Flops and sets the FC Flip-Flop, which forces \overline{IRF} LOW, indicating "Input Register Full." The D inputs must be stable while PL is HIGH. During parallel entry, the \overline{IES} Input should be LOW; the \overline{CPST} Input may be either HIGH or LOW.

After the fourth clock transition, the four serial data bits are aligned in the four data flip-flops and the FC Flip-Flop is set, forcing $\overline{\text{IRF}}$ LOW (Input Register full) and internally inhibiting further $\overline{\text{CPST}}$ clock pulses. Figure 4-15 illustrates the final positions resulting from a 64-bit serial bit train. B0 is the first bit, B63 the last bit.

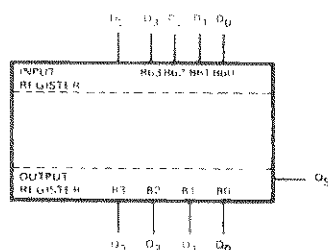


Figure 4-15. Final Positions in a FIFO
Resulting from a 64-Bit Serial Train

4.2.3.3 TRANSFER TO THE FALL-THROUGH STACK.

The outputs of Flip-Flops F0 - F3 feed the Stack. A LOW level on the $\overline{\text{TTS}}$ Input attempts to initiate a "fall-through" action. If the top location of the Stack is empty, data is loaded into the Stack and the input register is re-initialized. Note that this initialization is postponed until PL is LOW again. Thus, automatic FIFO action is achieved by connecting the $\overline{\text{IRF}}$ output to the $\overline{\text{TTS}}$ input.

Data falls through the Stack automatically, pausing only when it is necessary to wait for the next empty location. The $\overline{\text{MR}}$ input only initializes the Stack control section and does not clear the data.

Serial Entry. Data on the DS Input is serially entered into the F3, F2, F1, F0, FC Shift Register on each HIGH-to-LOW transition of the $\overline{\text{CPST}}$ Clock Input, provided $\overline{\text{IES}}$ and PL are LOW.

4.2.3.4 OUTPUT REGISTER (DATA EXTRACTION).

The Output Register receives 4-bit data words from the bottom Stack location, stores it and outputs data on a 3-state 4-bit parallel data bus or on a 3-state serial data bus. The output section generates and receives the necessary status and control signals. Figure 4-16 is a conceptual logic diagram of the output section.

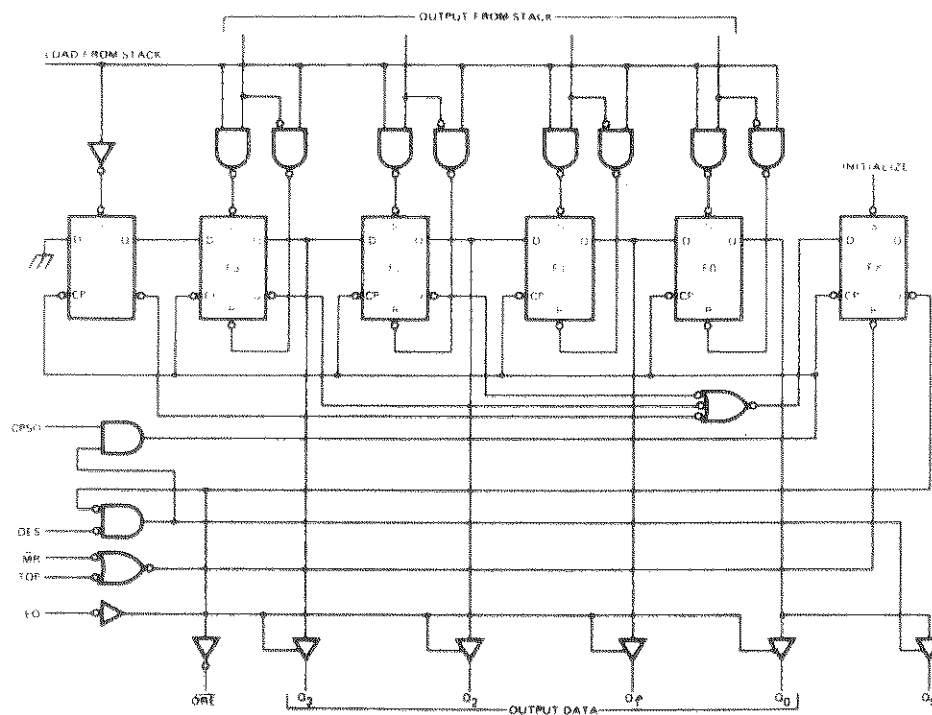


Figure 4-16. FIFO Conceptual Output Section

Parallel Data Extraction. When the FIFO is empty after a LOW pulse is applied to \overline{MR} , the Output Register Empty (\overline{ORE}) Output is LOW. After data has been entered into the FIFO and has fallen through to the bottom Stack location, it is transferred into the output register, provided the "Transfer Out Parallel" (TOP) Input is HIGH, and the \overline{OES} Input is LOW. As a result of the data transfer, \overline{ORE} goes HIGH indicating valid data on the data outputs (provided the 3-state buffer is enabled). TOP can now be used to clock out the next word. When TOP goes LOW, \overline{ORE} goes LOW indicating that the output data has been extracted; but the data itself remains on the output bus until the next LOW-to-HIGH transition of TOP transfers the next word (if available) into the output register as explained above. During parallel data extraction, \overline{TOS} , \overline{CPSO} , and \overline{OES} should be LOW.

Serial Data Extraction. When the FIFO is empty after a LOW pulse is applied to \overline{MR} , the Output Register Empty (\overline{ORE}) output is LOW. After data has been entered into the FIFO and has fallen through to the bottom Stack location, it is transferred into the output shift register provided the "Transfer Out Serial" (\overline{TOS}) is LOW. TOP must be HIGH, and \overline{OES} and \overline{CPSO} must be LOW. As a result of the data transfer, \overline{ORE} goes HIGH indicating valid data in the shift register. The 3-state serial Data Output QS is automatically enabled and puts the first data bit on the output bus. Data is serially shifted out on the HIGH-to-LOW transition of \overline{CPSO} . The fourth transition empties the shift register, forces \overline{ORE} LOW and disables the serial output OS. For serial operation, the \overline{ORE} output may be tied to the \overline{TOS} input, requesting a new word from the Stack as soon as the previous one has been shifted out.

4.2.3.5 HORIZONTAL EXPANSION.

The device may be horizontally expanded to store long words (in multiples of four bits) without external logic. The interconnections necessary to form a 16-word by 12-bit FIFO are shown in Figure 4-17. Using the same technique, any FIFO of 16 words by 4 X n bits can be constructed. When expanding in the horizontal direction, it is usual to connect the $\overline{\text{TRF}}$ and $\overline{\text{ORE}}$ outputs of the right most device (most significant device) to the $\overline{\text{TTS}}$ and $\overline{\text{TOS}}$ inputs respectively of all devices to the left (less significant devices) to guarantee that no operation is initiated before all devices are ready.

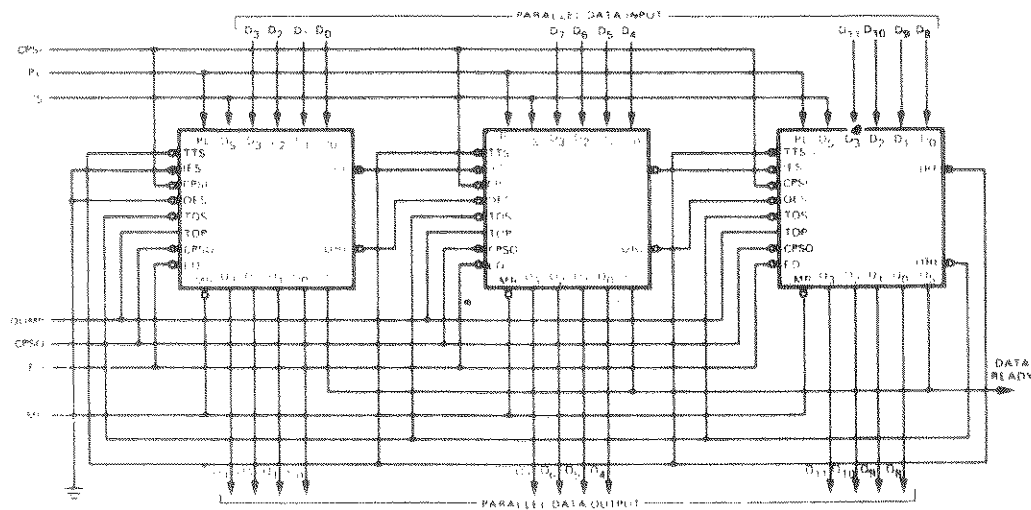
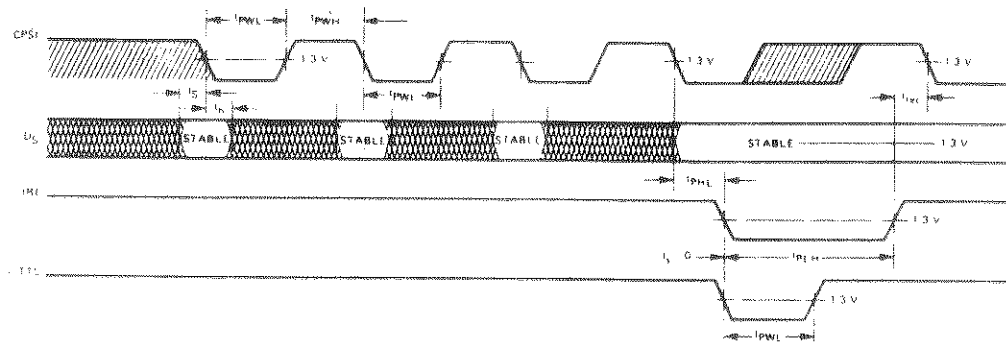


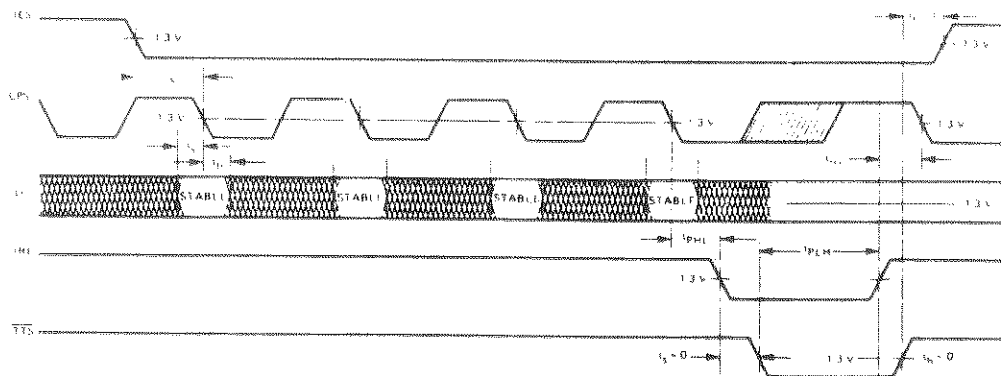
Figure 4-17. FIFO Horizontal Expansion Scheme

FIFO Buffer Timing of various configurations and conditions is shown in Figures 4-18 through 4-25.



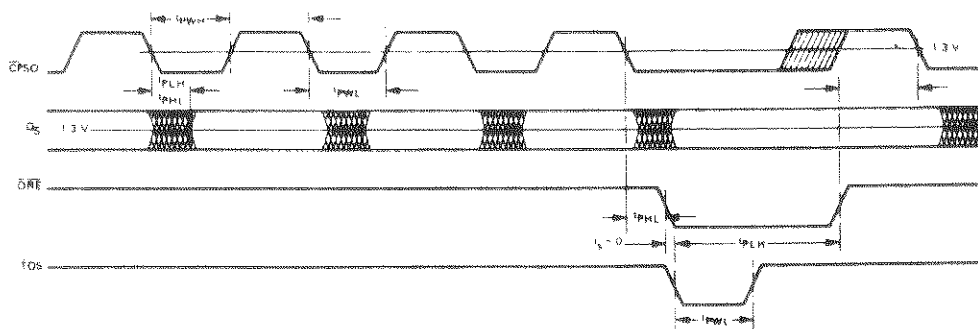
Conditions: Stack not full, \overline{IES} , PL LOW

Figure 4-18. FIFO Timing -
Serial Input, Unexpanded or Master Operation



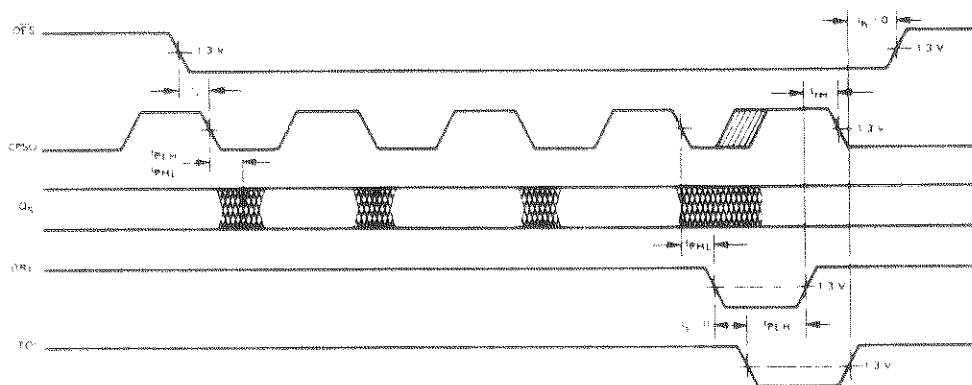
Conditions: Stack not full, \overline{IES} HIGH when initialized, PL LOW

Figure 4-19. FIFO Timing -
Serial Input, Expanded Slave Operation



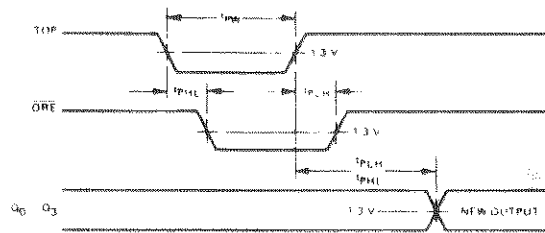
Conditions: Data in stack, TOP HIGH,
 \overline{IES} LOW when initialized, \overline{OES} LOW

Figure 4-20. FIFO Timing -
 Serial Output, Unexpanded or Master Operation



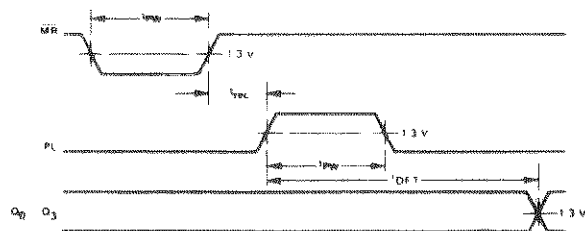
Conditions: Data in stack, TOP HIGH
 \overline{IES} HIGH when initialized

Figure 4-21. FIFO Timing -
 Serial Output, Slave Operation



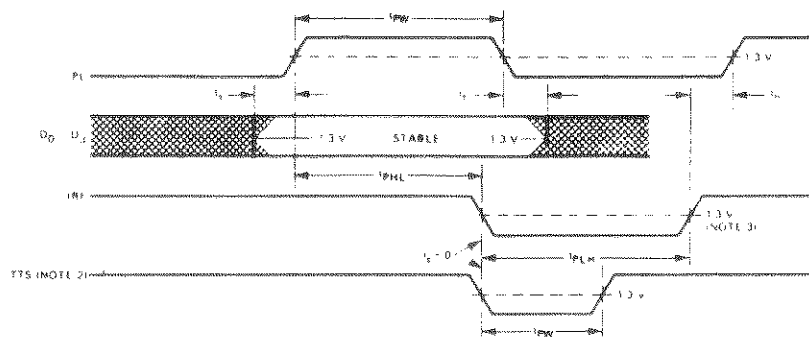
Conditions: $\overline{\text{IES}}$ LOW when initialized,
 $\overline{\text{EO}}$, $\overline{\text{CPSO}}$ LOW. Data available in stack

Figure 4-22. FIFO Timing - Parallel Output,
 4-Bit Word or Master in Parallel Expansion



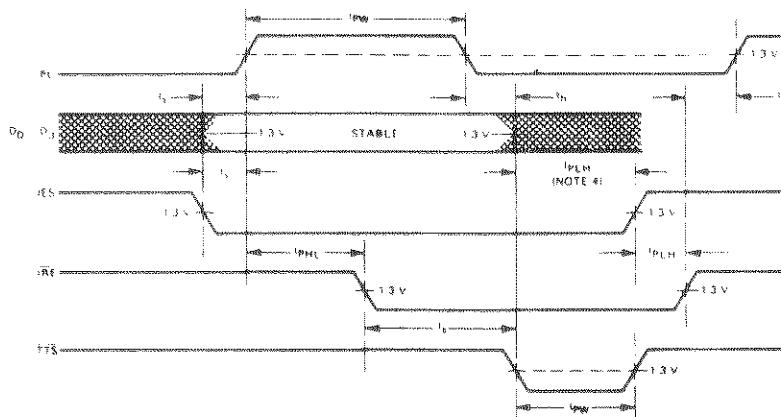
Conditions: $\overline{\text{TTS}}$ connected to $\overline{\text{IRF}}$, $\overline{\text{TOS}}$ connected
 to $\overline{\text{ORE}}$, $\overline{\text{IES}}$, $\overline{\text{OES}}$, $\overline{\text{EO}}$, $\overline{\text{CPSO}}$ LOW, TOP HIGH

Figure 4-23. FIFO Fall Through Time



Conditions: Stack not full, \overline{IES} LOW when initialized

Figure 4-24. FIFO Timing - Parallel Load Mode, 4-Bit Word (Unexpanded) or Master in Parallel Expansion



Conditions: Stack not full, device initialized
(Note 1) with \overline{IES} HIGH

Figure 4-25. FIFO Timing - Parallel Load, Slave Mode

4.2.4 CYCLIC REDUNDANCY CHECK (CRC) GENERATOR/CHECKER (U23).
Cyclic Redundancy Check (CRC) Generator/Checker is a 16-bit programmable device which operates on serial data streams and provides a means of detecting transmission errors. Cyclic encoding and decoding schemes for error detection are based on polynomial manipulation in modulo arithmetic. For encoding, the data stream (message polynomial) is divided by a selected polynomial. This division results in a remainder which is appended to the message as check bits. For error checking, the bit stream containing both data and check bits is divided by the same selected polynomial. If there are no detectable errors, this division results in a zero remainder. Although it is possible to choose many generating polynomials of a given degree, standards exist that specify a small number of useful polynomials. This device implements the polynomials listed in Table 4-14 by applying the appropriate logic levels to the select pins S0, S1 and S2. The CRC connections are shown in Figure 4-26.

The device consists of a 16-bit register, a Read Only Memory (ROM) and associated control circuitry as shown in the Block Diagram. The polynomial control code presented at inputs S0, S1 and S2 is decoded by the ROM, selecting the desired polynomial by establishing shift mode operation on the register with Exclusive OR gates at appropriate inputs. To generate the check bits, the data stream is entered via the Data Inputs (D), using the HIGH to LOW transition of the Clock Input (\overline{CP}). This data is gated with the most significant Output (Q) of the register, and controls the Exclusive OR gates (Figure 4-27). The Check Word Enable (CWE) must be held HIGH while the data is being entered. After the last data bit is entered, the CWE is brought LOW and the check bits are shifted out of the register and appended to the data bits using external gating (Figure 4-28).

To check an incoming message for errors, both the data and check bits are entered through the D Input with the CWE Input held high. The device is not in the data path, but only monitors the message. The Error Output becomes valid after the last check bit has been entered into the device by a HIGH to LOW transition of \overline{CP} . If no detectable errors have occurred during the data transmission, the resultant internal register bits are all LOW and the Error Output (ER) is LOW. If a detectable error has occurred, ER is HIGH; ER remains valid until the next HIGH to LOW transition of \overline{CP} or until the device has been preset or reset.

A HIGH on the Master Reset Input (MR) asynchronously clears the register. A LOW on the Preset Input (\overline{P}) asynchronously sets the entire register if the control code inputs specify a 16-bit polynomial; in the case of 12 or 8-bit check polynomials, only the most significant 12 or 8 register bits are set, and the remaining bits are cleared.

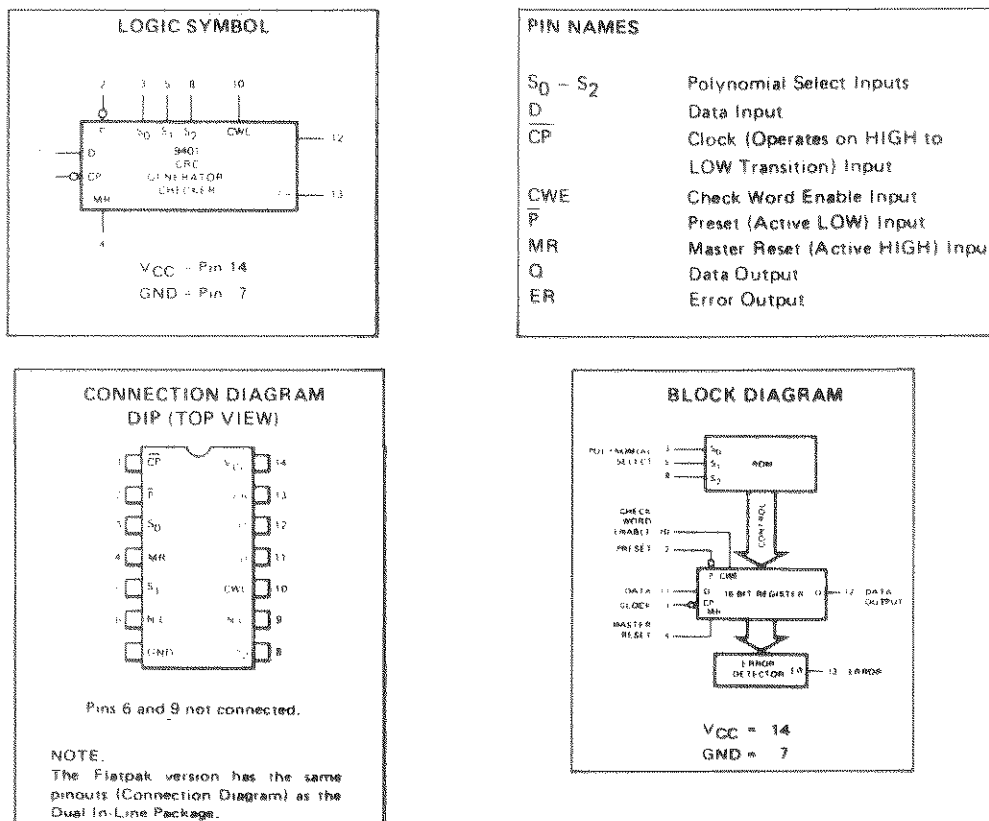


Figure 4-26. CRC Checker Connections

Table 4-14. CRC Polynomial Select Codes

SELECT CODE			POLYNOMIAL	REMARKS
S ₂	S ₁	S ₀		
L	L	L	$x^{16} + x^{15} + x^2 + 1$	CRC-16
L	L	H	$x^{16} + x^{14} + x + 1$	CRC-16 REVERSE
L	H	L	$x^{16} + x^{15} + x^{13} + x^7 + x^4 + x^2 + x + 1$	CRC-12
L	H	H	$x^{12} + x^{11} + x^3 + x^2 + x + 1$	
H	L	L	$x^8 + x^7 + x^5 + x^4 + x + 1$	LRC-8
H	L	H	$x^8 + 1$	
H	H	L	$x^{16} + x^{12} + x^5 + 1$	CRC-CCITT
H	H	H	$x^{16} + x^{11} + x^4 + 1$	CRC-CCITT REVERSE

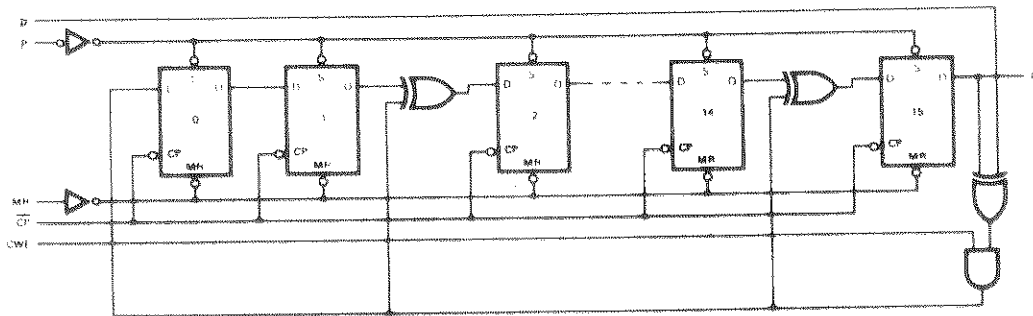
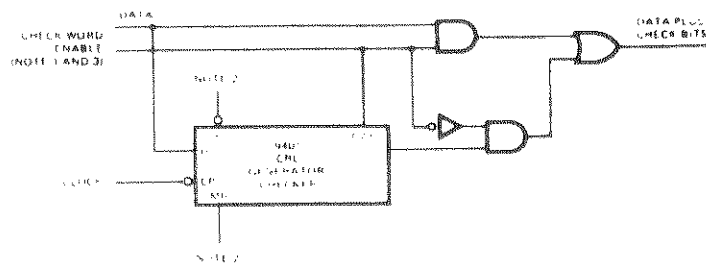


Figure 4-27. CRC Equivalent Circuit For $x^{16} + x^{15} + x^2 + 1$



NOTES:

1. Check word Enable is HIGH while data is being clocked, LOW during transmission of check bits.
2. 9401 must be reset or preset before each computation.
3. CRC check bits are generated and appended to data bits.

Figure 4-28. CRC Check Word Generation

4.2.5 16K (2K x 8) UV ERASABLE PROM (U21).

This device is a 16,384-bit ultraviolet erasable and electrically programmable read-only memory (EPROM).

Pin Configuration

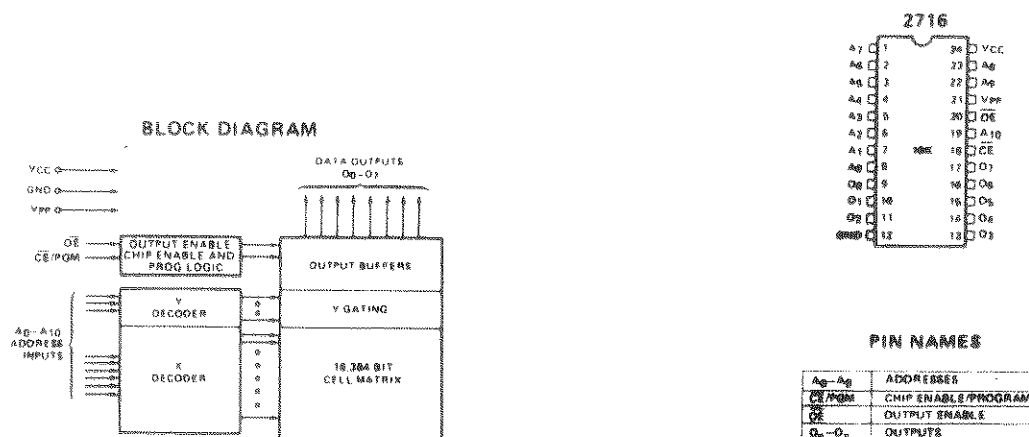


Figure 4-29. PROM Connections

4.2.5.1 ERASURE CHARACTERISTICS.

The erasure characteristics of this device are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical device in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the device is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available which should be placed over the window to prevent unintentional erasure.

4.2.5.2 DEVICE OPERATION.

The five modes of operation of the device are listed in Table 4-15. It should be noted that all inputs for the five modes are at TTL levels. The power supplies required are a +5V VCC and a Vpp. The Vpp power supply must be at 25V during the three programming modes, and must be at 5V in the other two modes.

Table 4-15. PROM Mode Selection

MODE \ PINS	CE/PROM (18)	OE (28)	Vpp (29)	Vcc (28)	OUTPUTS (10, 11, 12, 17)
Read	V _{IL}	V _{IL}	+5	+5	Output
Standby	V _{HH}	Don't Care	+5	+5	High Z
Program	Pulsed V _{IL} to V _{HH}	V _{HH}	+25	+5	Output
Program Verify	V _{IL}	V _{IL}	+25	+5	Output
Program Inhibit	V _{IL}	V _{HH}	+25	+5	High Z

Read Mode. The device has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable (\overline{CE}) is the power control and should be used for device selection. Output Enable (\overline{OE}) is the output control and should be used to gate data to the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from \overline{CE} to output (t_{CE}). Data is available at the outputs 120 ns (t_{OE}) after the falling edge of \overline{OE} , assuming that \overline{CE} has been low and addresses have been stable for at least $t_{ACC} - t_{OE}$.

Standby Mode. The device has a standby mode which reduces the active power dissipation by 75%, from 525 mW to 132 mW. The device is placed in the standby mode by applying a TTL high signal to the \overline{CE} input. When in standby mode, the outputs are in a high impedance state, independent of the \overline{OE} input.

Output Deselection. The outputs of two or more devices may be OR-tied together on the same data bus. Only one should have its output selected (\overline{OE} low) to prevent data bus contention

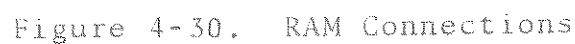
between devices in this configuration. The outputs of the others should be deselected by raising the OE input to a TTL high level.

Programming. Initially, and after each erasure, all bits of the device are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The device is in the programming mode when the Vpp power supply is at 25V and \overline{OE} is at VIH. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 msec, active high, TTL program pulse is applied to the \overline{CE}/PGM input. A program pulse must be applied at each address location to be programmed. One can program any location at any time -- either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The device must not be programmed with a DC signal applied to the \overline{CE}/PGM input.

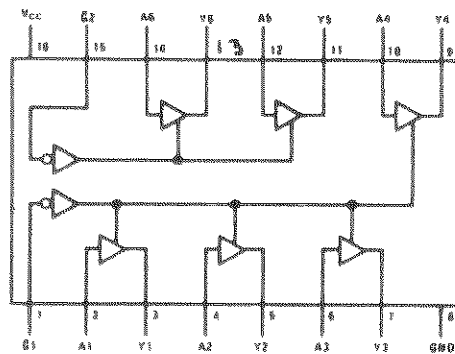
This device is a 4096-bit static Random Access Memory organized as 1024 words by 4 bits and requires no clocks or refreshing to operate. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided and a separate chip select (\overline{CS}) lead allows easy selection of an individual package.



4.2.7 TRI-STATE BUFFERS (U39).

These devices provide six, two-input buffers in each package. One of the two inputs to each buffer is used as a control line to gate the output onto the high-impedance state, while the other input passes the data through the buffer. The outputs are placed in the tri-state condition by applying a high logic level to the control pins.

Logic and Connection Diagram



Truth Table

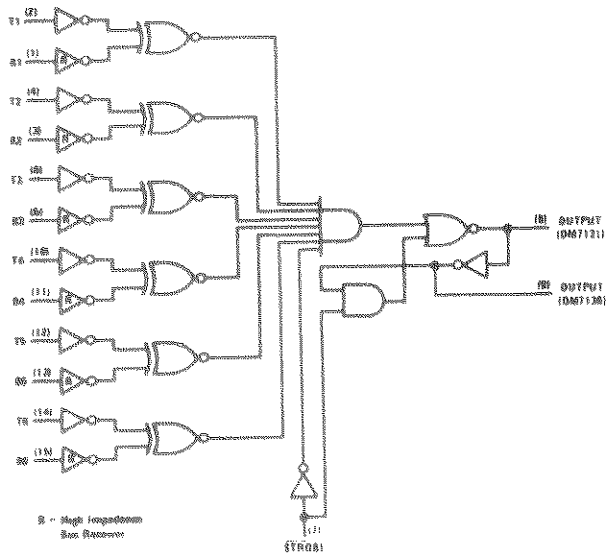
INPUTS		OUTPUT
\bar{G}	A	Y
H	X	H, Z
L	H	H
L	L	L

Figure 4-31. Tri-State Buffer Connections

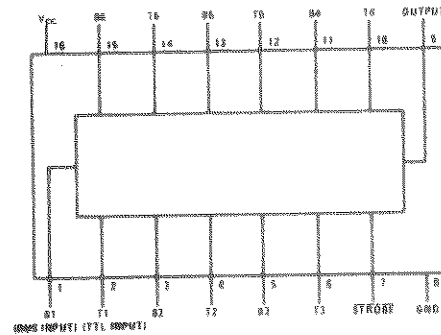
4.2.8 BUS COMPARATOR (U37).

This device compares two binary words of two to six bits in length and indicates matching (bit-for-bit) of the two words. Inputs for one word are TTL inputs, whereas inputs of the second word are high impedance receivers driven by a terminated data bus. The output has a latch that is strobe controlled. The transfer of information to the output occurs when the $\overline{\text{STROBE}}$ input goes from a logical 1 to a logical 0 state. Inputs may be changed while the $\overline{\text{STROBE}}$ is at the logical 1 level, without affecting the state of the output.

Logic Diagram



Connection Diagram *



Truth Table

CONDITION	$\overline{\text{STROBE}}$	OUTPUT	
		DM71/8131	DM71/8138
$T = B, T \neq B$	H	Q_{M-1}^*	Q_{M-1}^*
$T = B$	L	L	H
$T \neq B$	L	H	L

* Latched in previous state

Figure 4-32. Bus Comparator Connections

4.2.9 SCHMITT TRIGGER (U27).

	<u>Min</u>	<u>Typ</u>	<u>Max</u>
Positive-going Threshold Voltage	1.4	1.6	1.9
Negative-going Threshold Voltage	0.5	0.8	1.0

$$Y = \overline{A}$$

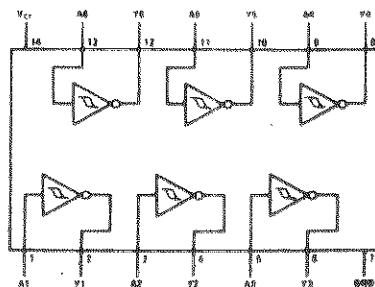


Figure 4-33. Schmitt Trigger Connections

4.2.10 D POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR (U2, U16, U25, U26, U36).

TRUTH TABLE

INPUTS				OUTPUTS	
PR	CLR	CLK	D	Q	\bar{Q}
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H [*]	H [*]
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q0	$\bar{Q}0$

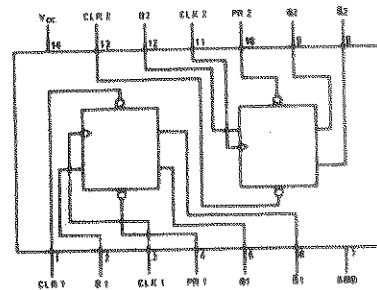
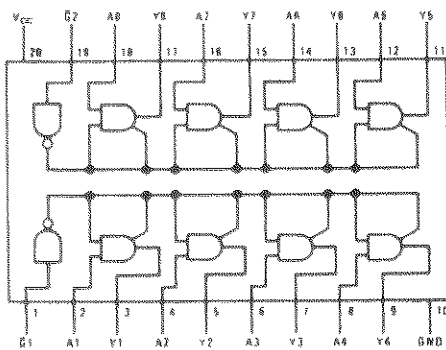


Figure 4-34. D Flip-Flop Connections

4.2.11 TRI-STATE BUFFERS (U14, U38).

This device provides eight, two-input buffers in each package that employ low power Schottky TTL technology. One of the two inputs to each buffer is used as a control line to gate the output into the high impedance state, while the other input passes the data through the buffer. The outputs are placed in the tri-state condition by applying a high logic level to the enable pins.

Logic and Connection Diagram



Truth Table

INPUTS		OUTPUT
G	A	Y
H	X	Z
L	H	H
L	L	L

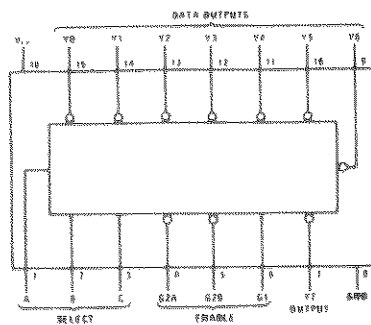
Figure 4-35. Tri-State Buffer Connections

4.2.12 DECODER (U10, U17).

These are Schottky-clamped circuits designed for memory-decoding or data-routing applications requiring very short propagation delay times. This DIP decodes one of eight lines, based on the conditions at the three binary select inputs and the three enable inputs.

Connection and Logic Diagrams

Truth Table



INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H	H
H	L	H	H	L	H	H	H	H	H	L	H	H
H	L	H	H	H	H	H	H	H	H	H	L	H

$$G2 = G2A \div G2B$$

H = High level, L = low level, X = don't care

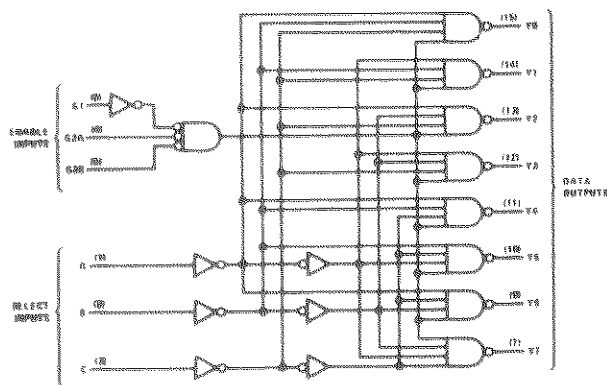
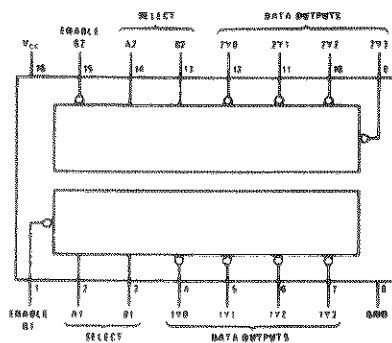


Figure 4-36. Decoder Connections

4.2.13 DECODER/DEMULTIPLEXER (U22).

These Schottky-clamped circuits are designed to be used in high performance memory decoding or data routing applications requiring very short propagation delay times. This device contains two separate two-line to four-line decoders in one package. The active-low enable input can be used as a data line in demultiplexing applications. The device features fully buffered inputs, presenting only one normalized load to its driving circuit. All inputs are clamped with high performance Schottky diodes to suppress line-ringing.

Connection Diagram



Logic Diagram

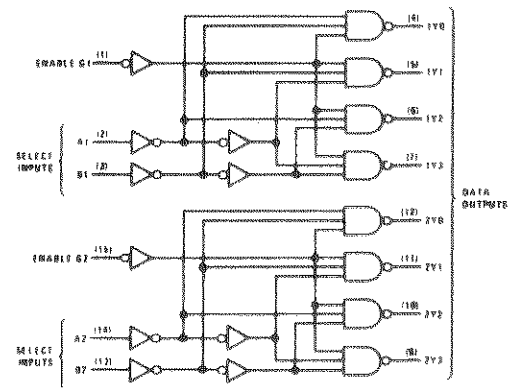
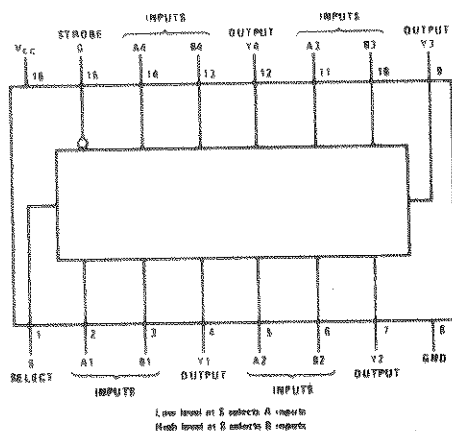


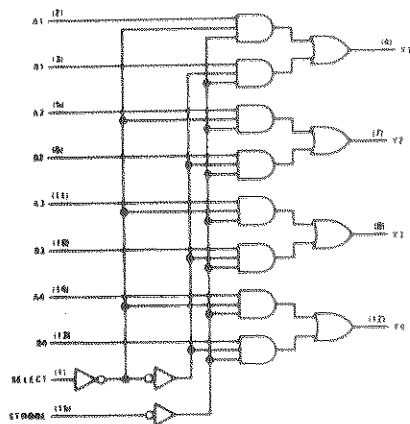
Figure 4-37. Decoder/Demultiplexer Connections

4.2.14 QUAD 2-LINE TO 1-LINE DATA SELECTOR/MULTIPLEXER (U9).
 This data selector/multiplexer contains inverters and drivers to supply full on-chip data selection to the four output gates. A separate strobe input is provided. A four-bit word is selected from one of two sources and is routed to the four outputs.

Connection Diagram



Logic Diagram



Truth Table

INPUTS				OUTPUT Y	
STROBE	SELECT	A	B	157 L 157A LS157, S157	LS158 S158
H	X	X	X	L	H
L	L	L	X	L	H
L	L	H	X	H	L
L	H	X	L	L	H
L	H	X	H	H	L

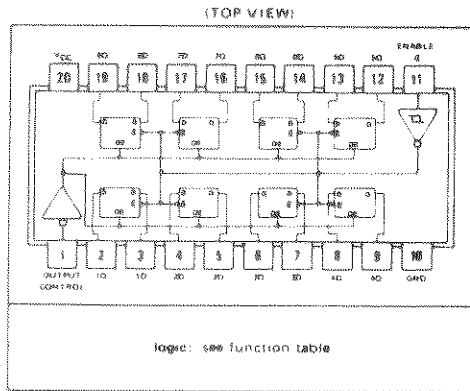
H = High Level, L = Low Level, X = Don't Care

Figure 4-38. Quad 2-Line to 1-Line
 Data Selector/Multiplexer Connections.

4.2.15 D-TYPE TRANSPARENT LATCHES (U42).

These 8-bit registers feature totem-pole three state outputs designed specifically for driving highly capacitive or relatively low impedance loads. The high impedance third state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving bus lines. Transparent operation means that while the enable (G) is high, the Q outputs follow the data (D) inputs. When the enable is low, the output is latched at the level of the data that was set up. The output control does not affect the internal operation of the latches. That is, the old data can be retained or new data can be entered even while the outputs are off.

Connection Diagram



Truth Table

OUTPUT CONTROL	ENABLE G	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q ₀
H	X	X	Z

Logic Diagram

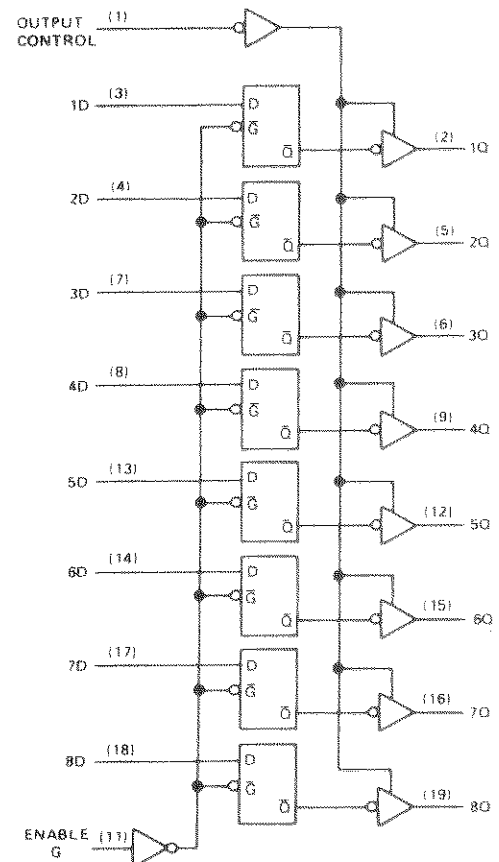
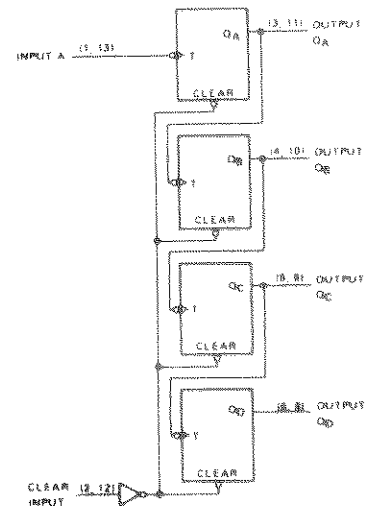


Figure 4-39. D-Type Transparent Latch Connections

4.2.16 DUAL 4-BIT BINARY COUNTER (815).
These devices contain eight master-slave flip-flops and additional gating to implement two individual four bit counters in a single package, each with a clear and clock input. Parallel outputs are available from each counter so that any submultiple of the input count frequency is available.

Logic Diagram



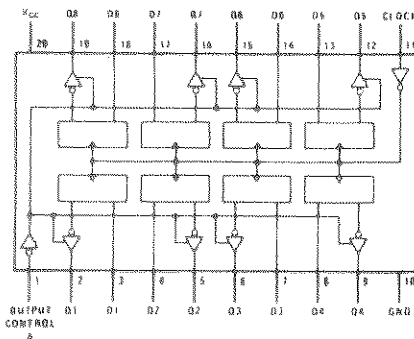
Truth Table

COUNT SEQUENCE (EACH COUNTER)				
COUNT	OUTPUT			
	Q _D	Q _C	Q _B	Q _A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H

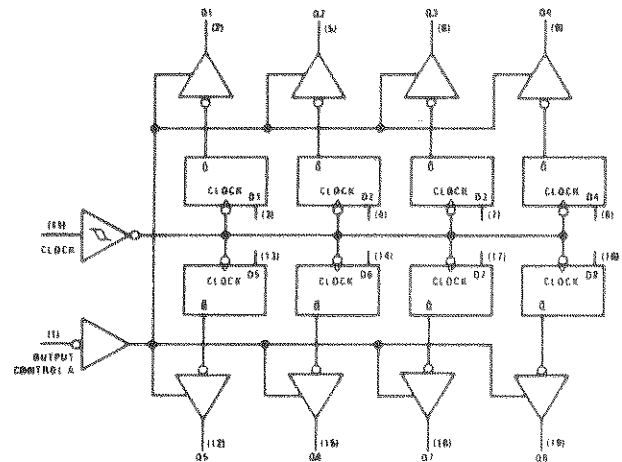
Figure 4-40. 4-Bit Binary Counter Connections

4.2.17 TRI-STATE D FLIP-FLOPS (U4, U11, U12, U13, U30, U31, U41). These 8-bit registers contain D-Type flip-flops with totem-pole tri-state outputs capable of driving highly capacitive or low impedance loads. When the output control is taken to a high logic level, the outputs go into the high impedance state. When a low logic level is applied to the output control, data at the D inputs are loaded into their respective flip-flops on the next positive-going transition of the clock.

Connection Diagram



Logic Diagram



Truth Table

OUTPUT CONTROL	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q0
H	X	X	Z

Figure 4-41. Tri-State D Flip-Flop Connections

SECTION 5

MAINTENANCE AND TROUBLESHOOTING

5.0 INTRODUCTION.

The AM-500 circuit board performs to full capability with a minimum of maintenance. This Section describes maintenance and troubleshooting procedures and procedures for handling warranty returns.

5.1 CIRCUIT BOARD CHECKOUT.

The AM-500 circuit board was fully tested before it left Alpha Microsystems and will operate satisfactorily in the system if the hardware and software requirements of Sections Two and Three of this manual are met. Should a problem arise after the circuit card has been in operation, use the following procedures to identify and locate the fault.

1. Check all cabling for proper seating of connectors.
2. Check the circuit board for proper seating in the slot.
3. Check all power connections for correct voltages.
4. Check jumper options to ensure correctness of application.
5. Verify that the fault is in the AM-500 and not in the system or in the peripherals. This can best be accomplished with substitution of a known good circuit board.
6. Perform the diagnostic tests included with the circuit board as described in paragraph 5.3.

5.2 WARRANTY PROCEDURES.

This circuit board is covered by warranty issued by Alpha Microsystems Inc., Irvine, California. Complete details of the warranty are included with the circuit board. Should a problem arise with this circuit board, call your dealer or the Alpha Micro International Support Services Group for information.

5.3 TROUBLESHOOTING PROCEDURES.

Diagnostic testing software is included with the AM-500 circuit board and should be used for troubleshooting and to verify proper operation. Test description and operating procedures are contained in the following paragraphs.

5.3.1 TEST DESCRIPTION - EXERCISE MODE.

1. Test-A. This test formats all specified tracks of the disk under test and then Cyclic Redundancy Check (CRC) verifies that all sector addresses and codes are written correctly.
2. Test-B, Test-C, Test-D, Test-E. These tests write to all specified tracks a data word supplied by the user. They then read all sectors written to verify correct data. Sectors are written and read sequentially.
3. Test-F. This test writes random data to all specified tracks. Addressing is also random. Each sector will be written to only once during this test. After all appropriate sectors have been written, a verify operation is performed using the same random addressing scheme.
4. Test-G. This test writes to all specified tracks using an incrementing data pattern. Addressing

alternates between two counters; one counting down from the maximum specified sector address, and the other counting up from the minimum specified sector address. Only odd numbered sectors are written to when addressing from the counter counting down, and only even numbered sectors are written to when addressing from the counter counting up. When all appropriate sectors have been written, a read operation begins using the same data and addressing scheme.

5. Test-H. This test forces error conditions which should be detected by the drive. First, it tries to access an illegal sector address. Secondly, it tries to seek an illegal cylinder address. Both cases should cause detectable errors.

Test-H error reporting. An error is recorded when the proper error code is not returned by either the Hawk or the AM-500. For example: During the "illegal cylinder" portion of the test, a cylinder address of 500 is sent to the Hawk. This address is illegal and should cause the Hawk to send the appropriate error status back to the program. If the actual status does not indicate that an illegal cylinder address was received by the Hawk, the program records an error.

NOTE

Before attempting this test, one switch setting must be changed on the Hawk. The switch to be changed is located on the I/O board, which is the PC board directly below the 50 pin ribbon cable coming from the AM-500. The switch itself (S4) is located in the lower left hand corner (see Figure 1). When the switch has been located, set position 3 to the "ON" position as described

5.3.2 TEST DESCRIPTION - ADJUSTMENT MODE.

1. Test-A. Performs one "RESTORE" operation on the disk under test. This causes the disk to reset its internal logic and seek cylinder 0.
2. Test-B. Performs an alternate seek between any two user specified cylinders without performing a read operation.
3. Test-C. Seeks any legal cylinder head and enables the drive select and read gate.

5.3.3 TEST OPERATION.

5.3.3.1 RUNNING THE PROGRAM.

Load HWKTST.PRG into memory and run by typing "HWKTST" followed by a carriage return.

5.3.3.2 OPERATOR INPUTS.

The operator must specify a disk number (0-7) to indicate which disk will be subjected to tests.

The operator must specify the test mode (exercisor adjustment).

If the adjustment mode was selected, the user must select a test from the "menu" presented by the program. If either test B or C is chosen, the user will be prompted to enter additional data regarding the cylinders and head to be used.

If the exercise mode was selected, the user may define a sequence of tests to be performed. Up to 20 test characters may be entered followed by a carriage return. A letter may be repeated, which will cause the test to be repeated. The standard default is all tests (A,B,C,D,E,F,G & H) and is selected by entering a carriage return only.

Some tests (A,B,C,D,E & F) require additional operator inputs to fully define them. If any of these tests are selected, the operator will be prompted to enter the appropriate data. Default values will be used if the operator enters only a carriage return in response to these prompts. Below is a list of parameters and their standard default values:

<u>PARAMETER/TEST</u>	<u>DATA</u>
Maximum Track	807
Minimum Track	0
Test-A	0
Test-B	0
Test-C	FFFF
Test-D	0333
Test-E	39CE
Test-F	FFFF

5.3.3.3 USER OPTIONS.

The user may suppress all error outputs by answering yes (Y) to the first user option.

If errors are not to be suppressed, the user may include a dump of the AM-500 data buffer after each data miscompare.

The user may elect to dump an error summary after each test by answering yes (Y) to the appropriate prompt.

The user may elect to pause after each test. To continue, the user types a carriage return on the controlling terminal.

The user may elect to continuously loop through the entire test sequence, or only loop X times as specified by the user.

5.3.3.4 ERROR HANDLING.

All disk errors will be printed as they occur unless the user has selected the option suppressing error printouts. The user may also select an option which will cause the AM-500 to dump its data buffer after every data miscompare. Additionally, error summaries may be output after each test or just a final error summary after all tests are complete.

5.3.3.5 ERROR TYPES.

Errors come under two types. The first type would be disk errors, such as seek, sector-not-found, and sentinel. These types of errors will cause up to four re-tries to occur before giving up. If the error persists for three or more re-tries, it is considered a hard error and is marked as such in the error summary. If the error corrects itself in less than three re-tries, it is marked as a soft error in the error summary.

The second type would be data errors. These errors are due to data miscompares during read operations. These errors are always marked as "hard" errors in the error summary.

5.3.3.6 TEST TERMINATION.

The test program may be aborted at any time by typing "CTRL-C."

5.3.3.7 BUFFER PRESETS.

In order to make disk read transfers apparent, tests A,B,C,D and E fill the AM-500 buffer with complement data prior to each read operation. The buffer is not preset in other tests since data will never repeat between any two consecutive reads.

5.3.4 WORST CASE DATA PATTERNS.

The following data patterns generate worst case data patterns for the disk:

<u>DATA</u>	<u>EFFECT</u>
0	Produces highest amplitude, lowest frequency. This is the default pattern for tests A and B.
FFFF	Produces lowest amplitude, highest frequency. This is the default pattern for test C.
333	Worst case for peak shift. This is the default pattern for test D.
39CE	Worst case for one-bit amplitude. This is the default pattern for test E.

5.3.5 DISK ERROR CODES.

<u>ERROR NUMBER</u>	<u>DESCRIPTION</u>
5	Disk not ready or fault exists.
11	Write protected.
65	Seek error.
67	Sector not found.
68	CRC error.
69	Sentinel field error.

NOTE

All other error codes are illegal at this time.
If any other codes appear, the AM-500 board may
have problems.

5.3.6 RECOMMENDED TEST PROCEDURE.

The following sequence is recommended as a short but comprehensive test and will take about 7 minutes:

Type HWKTST followed by a RETURN.

Enter DISK NUMBER when prompted by program.

Select the EXERCISE MODE (#1).

When prompted to enter test sequence (A-H), enter a carriage return. This will cause a predetermined sequence of tests which consists of tests A,B,C,D, E,F,G,H in that order.

When asked for "HIGHEST TRACK," enter 220. When asked for "LOWEST TRACK," enter 200.

Enter a return for each of the following:

Format Data Word= Test B Data Word=

Test C Data Word= Test D Data Word=

Test E Data Word=

Enter a "Y" for "Suppress all error messages."

Enter "N" for "Dump data and CRC error tallies after each test."

Enter "N" for "Pause after each test."

Enter "N" for "Loop continuously?"

Enter 2 for number of passes.

This test sequence will use all of the default data values which were described in paragraph 5.3.5.

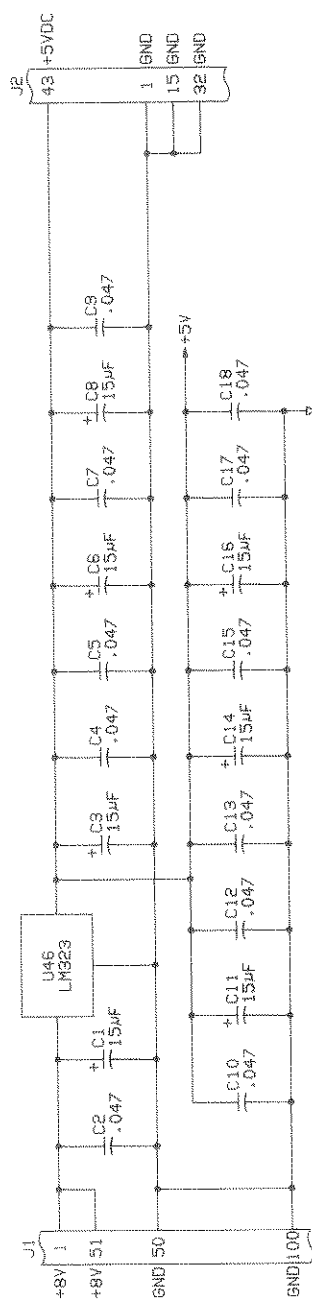
Upon completion of the tests, the program will give an ERROR REPORT. If any ERRORS exist, restart the program and use the same test sequence that was used originally, with one exception: when prompted with "SUPPRESS ALL ERROR MESSAGES?" enter "N." This will cause the program to output an error message each time an error is detected, indicating the test that was being executed when the error was detected.

SECTION 6
SCHEMATIC AND PARTS LIST

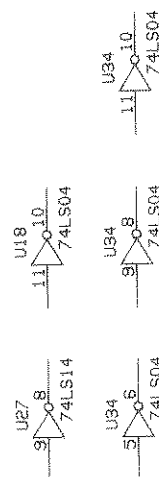
Table 6-1. Component Crossreference List

REF. DESIG.	MFR. TYPE NO.	PAR. NO.*	REF. DESIG.	MFR. TYPE NO.	PAR NO.*
U1	10Mhz Oscillator	-	U22	74LS139	4.2.13
U2	74LS74	4.2.10	U23	9401	4.2.4
U3	74LS08	-	U25	74LS74	4.2.10
U4	74LS374	4.2.17	U26	74LS74	4.2.10
U5	2114	4.2.6	U27	74LS14	4.2.9
U6	2114	4.2.6	U29	74LS32	-
U7	9403	4.2.3	U30	74LS374	4.2.17
U8	9403	4.2.3	U31	74LS374	4.2.17
U9	74LS157	4.2.14	U32	74LS00	-
U10	74LS138	4.2.12	U33	74LS08	-
U11	74LS374	4.2.17	U34	74LS04	-
U12	74LS374	4.2.17	U35	75451	-
U13	74LS374	4.2.17	U36	74LS74	4.2.10
U14	81LS97	4.2.11	U37	8131	4.2.8
U15	220/330 Resistors	-	U38	81LS97	4.2.11
U16	74LS74	4.2.10	U39	8097	4.2.7
U17	74LS138	4.2.12	U40	74LS30	-
U18	74LS04	-	U41	74LS374	4.2.17
U19	280	4.2.1	U42	74LS373	4.2.15
U20	9517	4.2.2	U43	75451	-
U21	2716 C28498	4.2.5 -	U44	74LS32	-
			U45	74LS393	4.2.16

*Paragraph number in Section 4 where a description of integrated circuit logic can be found.



0
1
2
3
4



2. U15 220Ω/330Ω RESISTOR
ARRAY.
1. LM23 TO BE ATTACHED TO
THERMALLOY 5002 HEATSINK.
- S; UNLESS OTHERWISE SPECIFIED

APPROVED	DESIGNED	SHEET NO.	PROJECT NO.
		1	DWL-00500-XX
8/28/78 TITLE A 8-1-78 IRVINE, CA 92714 ALPHA SYSTEMS 41000			
SCHEMATIC AM-500 DISK CONTROLLER			

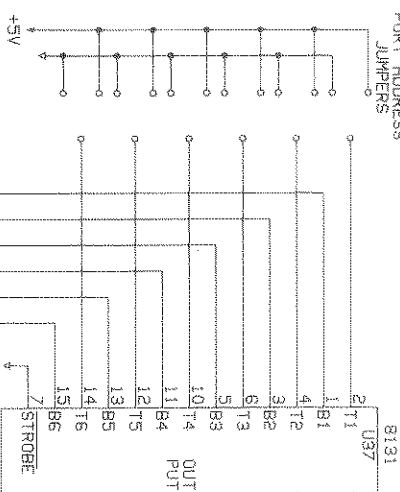
PART OF J1

POBIN 78

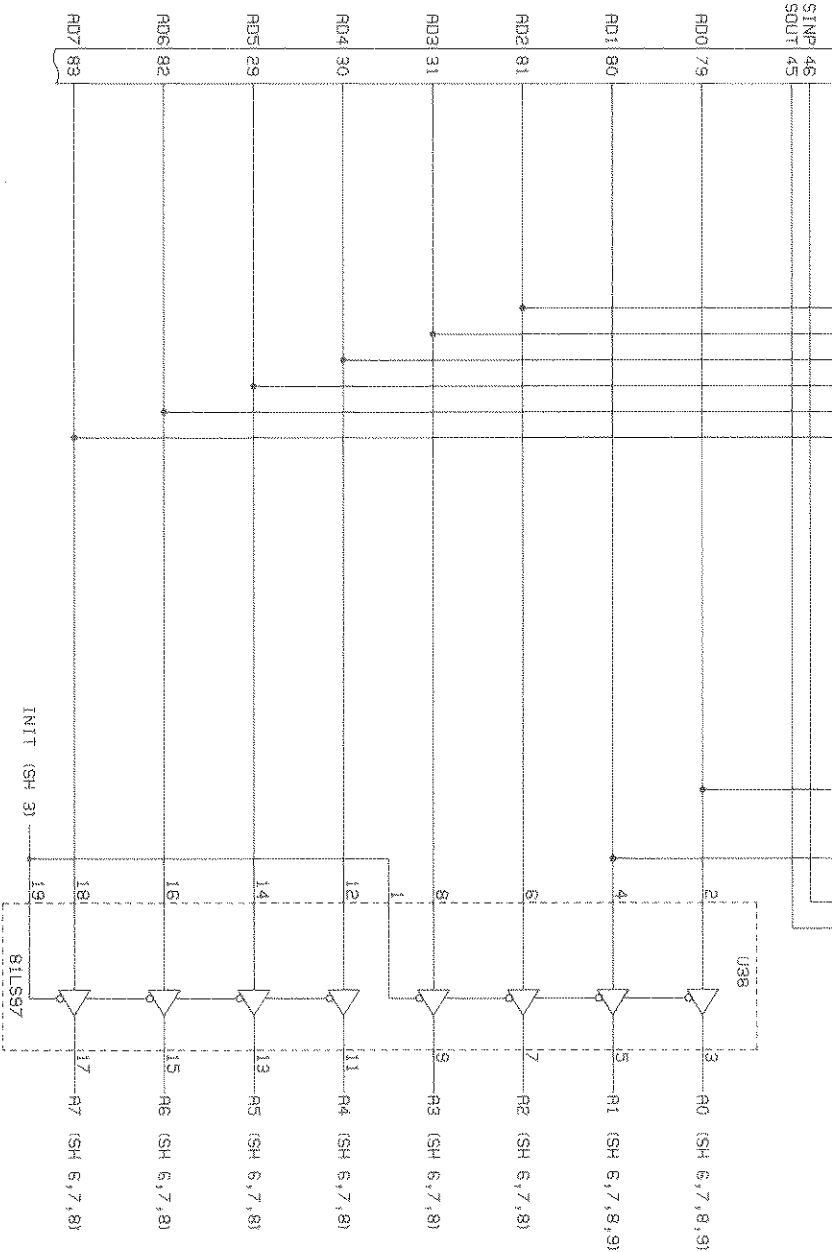
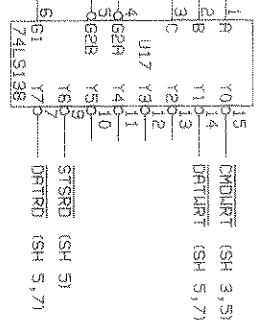
PAR 77

S100 ADDRESS LINES

PORT ADDRESS
JUMPERS



74LS32

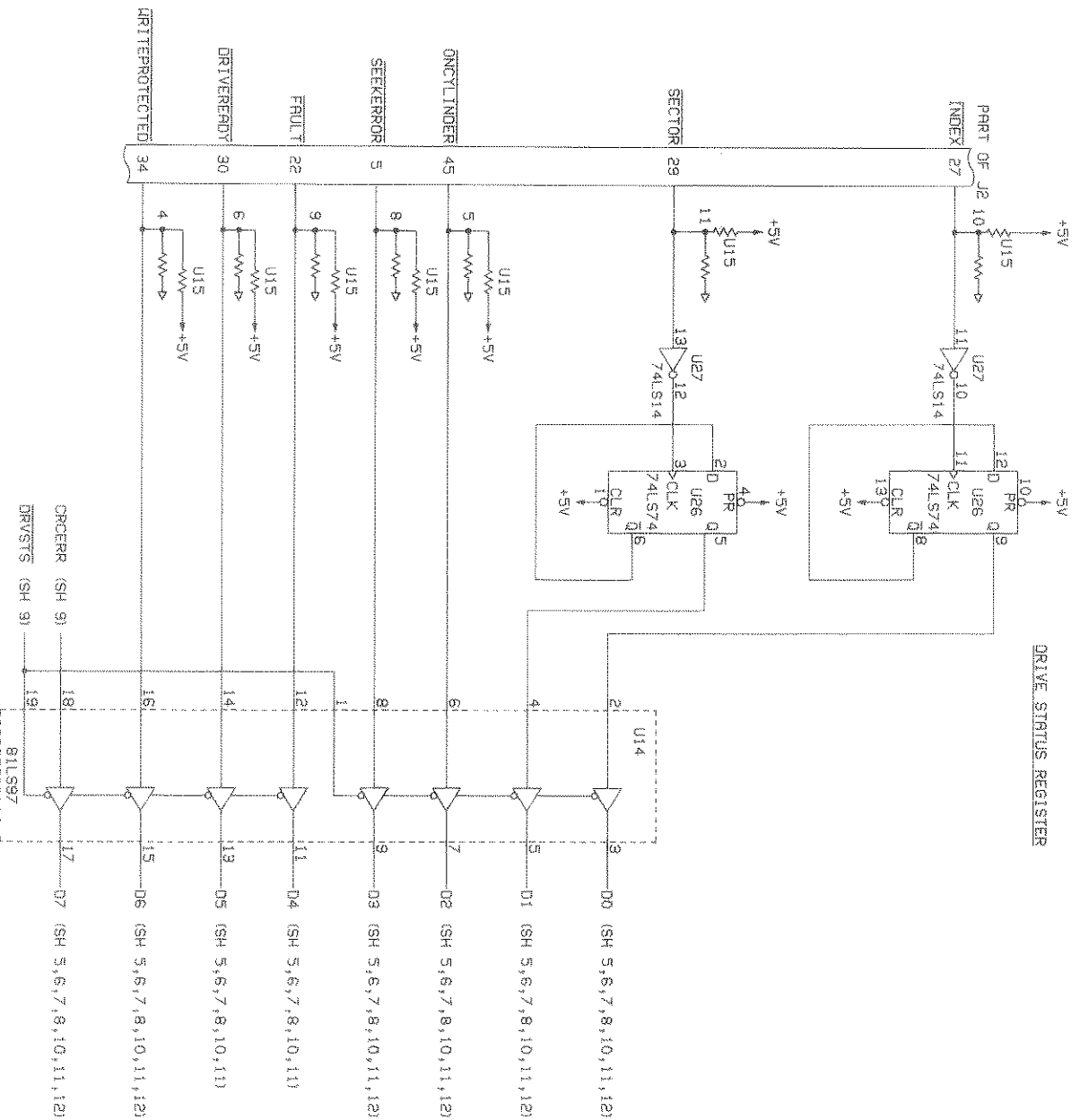


ALPHA SYSTEMS

IRVINE, CA 92714

SCHEMATIC RM-500
DISK CONTROLLER

DESIGNED	DRAWN	CHECKED	APPROVED
SHEET NO. 2			
DML-00500-XX			



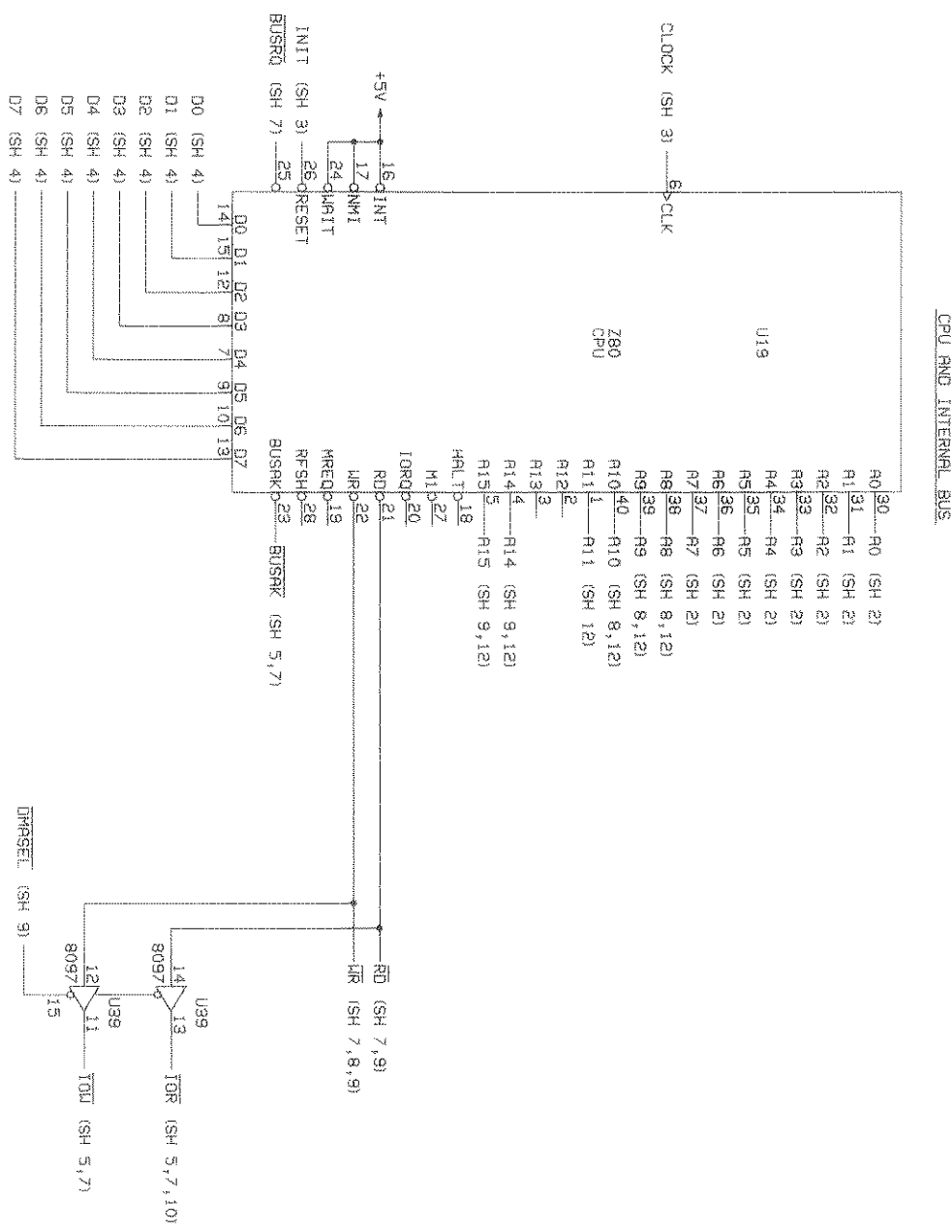
ALPHA
micro SYSTEMS

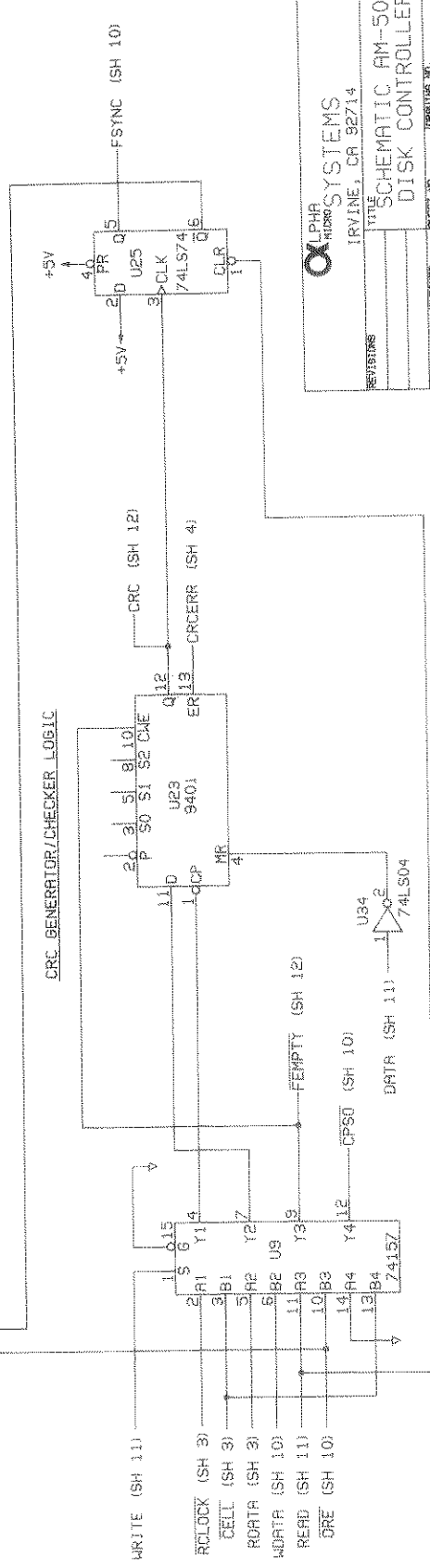
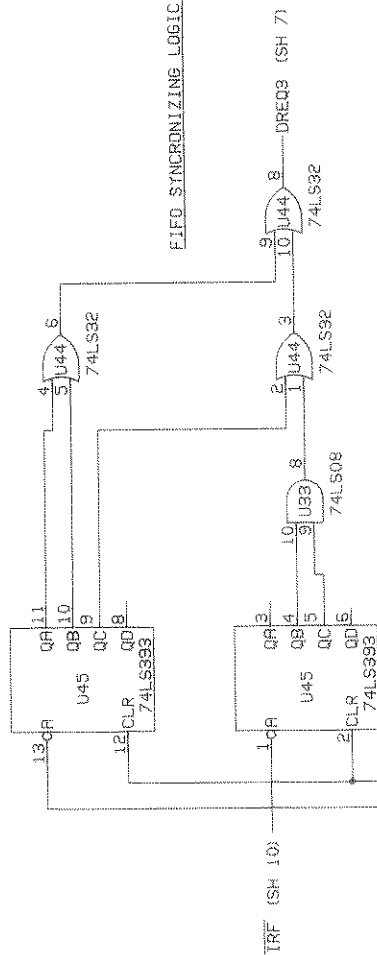
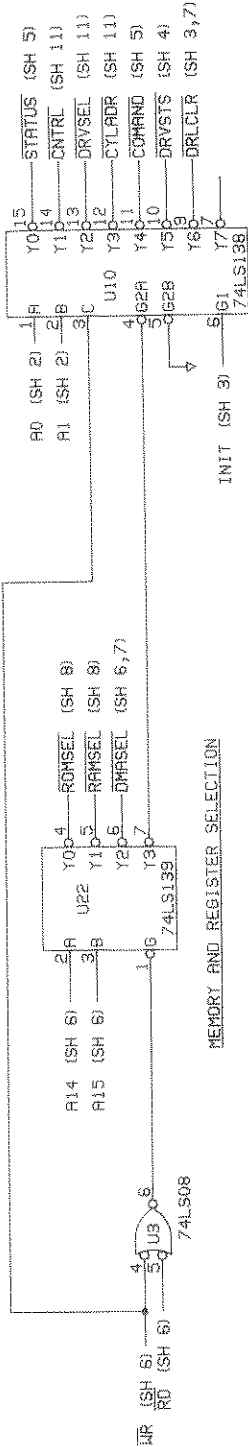
IRVINE, CA 92714

REVISING
TITLE
SCHEMATIC AM-500
DISK CONTROLLER

DESIGN
CHECKED
SHEET NO.
4

APPROVED
DATE
DIAL-00500-XX





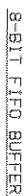
ALPHA SYSTEMS
IRVINE, CA 92714


REVISIONS

NO.	DESCRIPTION	DATE
1	INITIAL DESIGN	10/1/80
2	REVISED FOR MANUFACTURING	10/1/80
3	REVISED FOR MANUFACTURING	10/1/80
4	REVISED FOR MANUFACTURING	10/1/80
5	REVISED FOR MANUFACTURING	10/1/80
6	REVISED FOR MANUFACTURING	10/1/80
7	REVISED FOR MANUFACTURING	10/1/80
8	REVISED FOR MANUFACTURING	10/1/80
9	REVISED FOR MANUFACTURING	10/1/80

DESIGNED BY: []
CHECKED BY: []
APPROVED BY: []

SHEET NO. 9
DWL-00500-XX



 ALPHAGRAPHICS SYSTEMS		IRVINE, CA 92714	
REVISIONS		TITLE	
DISK CONTROLLER		SCHEMATIC AM-500	
DRAWN		CHECKED	
DATE		SHEET NO.	
10		DML-00500-XX	

BILL # DWB-00500-00

RETAIL PRICE \$ 0.00

REV. A03

WHSLE PRICE \$ 0.00

22 MAY 1979

NET PRICE \$ 0.00

DESCRIPTION DISK CONTROLLER AM-500

PART NUMBER	DESCRIPTION	QTY
1. DWF-00500-00	PCB DISK CONTROLLER AM-500	1
2. CNS-00040-00	SOCKET 40 PIN DIP	2
3. CNS-00024-00	SOCKET 24 PIN DIP	3
4. CNS-00020-00	SOCKET 20 PIN DIP	10
5. CNS-00018-00	SOCKET 18 PIN DIP	2
6. CNS-00016-00	SOCKET 16 PIN DIP	6
7. CNS-00014-00	SOCKET 14 PIN DIP	19
8. CNS-00008-00	SOCKET 8 PIN DIP	2
9. IC1-74367-00	IC HEX BUFFER	1
10. CPP-00156-01	CAPACITOR 15 UF 20V	7
11. CPN-00473-01	CAPACITOR .047UF	11
12. HDM-00001-00	HEATSINK 1.500 X 1.810 .500LG	1
13. ICL-00323-00	IC REGULATOR +5V TO-3	1
14. RS2-00101-00	RESISTOR 100 OHM 1/4W 5% CAR	1
15. RS2-00222-00	RESISTOR 2.2 K 1/4W 5% CAR	1
16. IC1-74393-01	IC DUAL 4 BIT BINARY COUNTER	1
17. IC1-75451-00	IC DUAL INTERFACE DRIVER NI OC	2
18. IC1-07430-01	IC 8 INPUT NAND GATE	1
19. CNF-00002-11	CONN HEADER 50PIN .092LG RT ANGLE	1
20. IC1-08197-01	IC BUFFER OCTAL	2
21. IC1-74373-01	IC OCTAL D FLIPFLOP	1
22. IC1-74374-01	IC OCTAL D FLIPFLOP	3
23. IC1-74374-02	IC OCTAL D FLIPFLOP	4
24. IC1-07474-01	IC DUAL D FLIPFLOP	4
25. ICS-09403-00	IC 8 BIT FIFO	2
26. IC1-74138-01	IC DECODER 3 TO 8 LINE	2
27. IC1-07404-01	IC HEX INVERTER	2
28. IC1-07408-01	IC QUAD 2 INPUT AND GATE	2
29. ICM-02114-04	RAM 256 X 4 BIT STATIC	2
30. IC1-74139-01	IC DECODER 2 TO 4 LINE DUAL	1
31. IC1-08131-00	IC COMPARATOR 6 BIT	1
32. IC1-74157-00	IC QUAD 2 TO 1 DATA SELECTOR	1
33. ICS-09401-00	IC GENERATOR CRC	1
34. IC1-07414-01	IC HEX INVERTER W/HYSTERESIS	1
35. IC1-07400-01	IC QUAD 2 INPUT NAND GATE	1
36. IC1-07432-01	IC QUAD 2 INPUT OR GATE	2
37. ICS-00001-01	IC OSCILLATOR 10MHZ	1
38. RSN-00003-00	RESISTOR PACK 14PIN DIP 220/330OHM	1
39. ICS-09517-00	IC CHIP DMA	1
40. ICS-00080-00	IC MICROPROCESSOR Z-80	1
41. HDS-00632-01	SCREW 6-32 X .375	2
42. HDN-00632-01	NUT HEX 6-32 STL SM PATTERN	2
43. HDW-00632-01	WASHER LOCK 6-32	2
44. IC1-07474-02	IC DUAL D FLIPFLOP	1
45. ICM-00501-01	ROM 2K X 8 BIT	1

This is a dump of the Z-80 and WD16 programs on the AM500 Hawk card.
The source is a ROM like 2716 with Rev D. #C28498M
This is an Intel format dump.

```
:1000000031FE43210B80183FCB46C20703C30C03CC
:10001000CB4EC21103C317030624C3AA0200000007B
:10002000EF368D3ECDC33102DD36049AC38902001E
:100030002100C0320D80182F2100C07EB728FC4E51
:1000400036807EB720FCC936443649368A360721C9
:1000500001C0368F2336FF7EFFF72100C077CB79B2
:1000600028F623360F18F1DD210080CB412045CB47
:1000700071280423368F28CB492838CB69200C21DB
:100080000040CB61280F210440180A21FE43CB61B8
:1000900028032AFE43CB592801E97C45160021009C
:1000A00080CB51200314232370772336003640DDA4
:1000B000720AAFC923CB692858D92102405E235662
:1000C000AF676F477BE60F4F7BE6F05F20047AB7A0
:1000D000280E19CB3ACB1BCB3ACB1BE809EB18E420
:1000E0000604CB3CCB1D10FA79FE0C380323D60C4A
:1000F000CB3CCB1D3002CBCC0604CB2710FCB45735
:100100005D2101407ECB1F3002CB02772373237257
:10011000D93A01403C473E08CB2710FC473A034000
:10012000E60FB02F3202C079E677FE032006CDF647
:1001300002C32C02CB662008CDF602CB66CAE701CB
:10014000CB6EC2E701CB6128193A02402F3203C0BF
:10015000368B368FCB56280ACB5E20F83E4108C33B
:10016000D401CB59CAEF01CB76CAEB013E5A320417
:1001700042AF320542CB51203CCB49C22C02EFDFCB
:1001800036CDDDD36090636EDDD360404DD360440B5
:100190000620210580360F3600DD360A02DD3609DD
:1001A000060010F136013600DD360A02DD3609069A
:1001B0002101C0181ACF368D3A0340E60F320340B2
:1001C000E73A0340C610320340E6F0FEC020F13E9D
:1001D0008FD718583A0040B7280A3D320040CDF674
:1001E00002C311010818463E0518423E0B183ECRCB
:1001F000512839AF320442DFDD360604DD360640D1
:10020000DD360707DD360702368E36DEDD360A03B9
:10021000D9210880CB5E28FCD93E4408CB7E20B48F
:10022000368F3E45083A0442EE5A20A8AF2336FFE7
:10023000C9D7060A10FEDD36090636EDDD360402A2
:10024000DD360440DD360501DD360500DD360A0207
:10025000DD360906EF36CDDDD36090636EDDD36042E
:1002600004DD3604400620210580360F3600DD36D9
:100270000A02DD3609060010F136013600DD360AC5
:1002800002DD3609062101C0C9DD360402DD36056E
:100290000FDD360500DD360A02C90000000000004F
:1002A00000000000000000000000080D9ED5B0240214A
:1002B00008800680D9DD770D3E8ED7DD36060CDD51
:1002C000360642DD360701DD36070036DEDD360A4A
:1002D00003D9AF3E8FCB5E200510FAD9180D2A0C3A
:1002E00042ED52D92005CB7E368FC8368F10BB3EEB
:1002F00043D108C3D401368736833687368FCB5631
:100300002804CB5E20F8C9CB4620FCC9CB4628FC8C
:10031000C9CB4E20FC77C9CB4E28FC77C9000000022
:10032000000000000000000000000000000000CD
:10033000000000000000000000000000000000BD
:10034000000000000000000000000000000000AD
:100350000000000000000000000000000000009D
:100360000000000000000000000000000000008D
```


:10072000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD9
:10073000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC9
:10074000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB9
:10075000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA9
:10076000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF99
:10077000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF89
:10078000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF79
:10079000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF69
:1007A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF59
:1007B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF49
:1007C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF39
:1007D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF29
:1007E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF19
:1007F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF09
:00000001FF

EOJ

*