System 8000™
Hardware Reference Manual

Zilog

03-3237-04

December 1982

The information in this publication is subject to change without notice.

Zilog assumes no responsibility for the use of any circuitry other than circuitry embodied in a Zilog product. No other circuit patent licenses are implied.

NOTICE TO OWNER

FEDERAL COMMUNICATIONS COMMISSION RADIO FREQUENCY INTERFERENCE STATEMENT

Warning: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. As temporarily permitted by regulation it has not been tested for compliance with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

SYSTEM 8000 HARDWARE REFERENCE MANUAL

03-3237-04

Preface

This manual contains the information necessary to install, operate, and maintain Zilog's System 8000 Model 21, Model 31 microcomputer. This manual addresses field engineers (FE), service technicians, and all others who require knowledge of the hardware aspects of the system.

This manual and the related manuals listed below provide the technical documentation of the System 8000.

Title	Zilog Number
ZEUS System Administrator Manual	03-3246
ZEUS Utilities Manual	03-3196
ZEUS Reference Manual	03-3195
CPU Hardware Reference Manual	03-3200

System 8000 is a registered trademark of Zilog, Inc.

Table of Contents

List of Illustrations

List of Tables

SECTION 1
OVERVIEW

1.1. System Description

The System 8000 is a multiuser free-standing unit, based on Zilog's 16-bit Z8001A microprocessor and running the ZEUS Operating System at 5.5 megahertz. The System 8000 product line consists of several different models. The Model 21 and Model 31 are both similar in appearance having four or more separate stacked modules. The Model 21 contains a 32 MB Winchester Disk Drive within the Disk/Tape module. The Model 31 is equipped with an 84 MB Storage Module Drive (SMD). Both the Model 21 and Model 31 have a 17 MB Cartridge Tape Drive and will accomodate additional Disk or Disk/Tape modules. Figure 1-1 shows the modules that make up the system.

The top Processor Module controls the system and contains the the CPU and various system controller boards within its ten-slot card cage. The system communication is over the 32-bit Z-Bus Backplane Interconect (ZBI). Two I/O panels located at the rear of the processor module interconnect with the lower modules I/O panels and provide disk, tape and terminal communication.

The Disk/Tape module beneath the top module has either a 32-megabyte Winchester disk drive or an 84-megabyte SMD and a 17-megabyte cartridge tape drive. Disk and Tape I/O connectors are provided on the rear I/O panel.

The remaining two modules can be additional Disk/Tape or Accessory modules that are interconnected to the above Disk and/or Disk/Tape Modules. Terminal distribution panels for serial terminal and parallel printer connection can be added to these modules.

Both the Model 21 and 31 system can be expanded from 8 to 24 users by installing two Secondary Serial Boards (SSB) and terminal distribution panels.

The basic system contains five or six printed circuit boards in the ten-slot card cage. These boards (Figures 1-2 through 1-8) are:

the CPU
the Winchester Disk Controller (WDC)
or Storage Module Device Controller Boards A and B
the Cartridge Tape Controller
and the Memory Subsystem
including the ECC Controller
and one or more Memory Arrays.

The lower three slots of the card cage are normally dedicated to the memory subsystem; with the addition of more memory boards, the basic system can have a physical memory of 2 megabytes, not including the small bootstrap memory on the CPU board. An optional memory configuration uses the lower five slots of the card cage for a maximum physical memory of 4 megabytes.

An expansion chassis module will be available that can be added to the system increasing the number of cards attached to the bus from 10 to 18. This allows still more options.

1.2. Functional Relationships

The diagram in Figure 1-9 shows the functional relationships of the boards that make up the basic system. These elements communicate with one another over Zilog's Z-Bus Backplane Interconnect (ZBI), a high-speed, 32-bit, semisynchronous bus. The following paragraphs briefly describe the functions of each element on the bus.

1.2.1. CPU Board (CPU): The CPU is the host of the System 8000; it controls the ZBI and terminal communications into and out of the system. The CPU supports eight serial I/O ports and a parallel I/O port (Figure 1-9). The serial ports are compatible with RS232-C. The parallel port, with the appropriate jumpers inserted, is compatible with the line printer standards of either Centronics or Data Products.

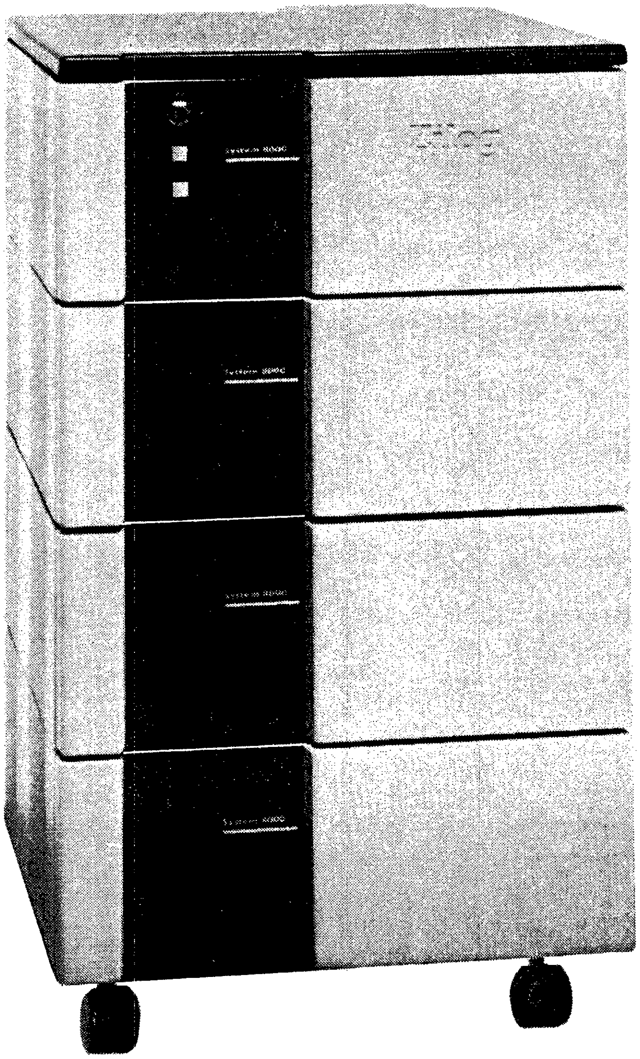


Figure 1-1 System 8000

1.2.2. Winchester Disk Controller (WDC): The disk controller enables the CPU to communicate with up to four 24 or 32 MB Winchester disk drives. An intelligent bus module with an on-board Z80B microprocessor, the controller can be polled or interrupt-driven by the CPU. The appropriate jumper arrangement determines the controller's mode of operation. The disk is organized into 512-byte sectors. A single command can cause the transfer of up to 128 512-byte sectors. In addition, the disk controller uses a full-track buffer which permits the transfer of 24 512-byte sectors in one disk revolution.

1.2.3. Storage Module Device Controller (SMDC): The SMDC is a high performance controller (1.8 MB/second max. data rate) linking the system ZBI backplane to the industry standard Storage Module Device (SMD) interface. The SMDC consists of two cards, SMDC A and SMDC B. Card A connects to the ZBI at the 96-pin board connector P1. It also connects to the SMD A daisy-chain cable at plug P2. Card B connects to four SMD B radial cables. All drive connections are through the P2 backplane connector.

Bit slice processor and sequencer logic control all the operations of the SMDC. The SMDC can control up to four eight inch drives that have SMD interface. SMDC features include packet control, overlapped seek, automatic error recovery, data buffering, flagged sectors, long writes and reads, and self-test on power-up or system initialization. The SMDC can be interrupt driven or operate in the polled mode to the host when a packet is complete. It gives a complete status report of the SMD and the controller.

1.2.4. Tape Cartridge Controller (TCC): The tape controller is the intelligent interface between the CPU and up to four cartridge tape drives. A Zilog Z80B microprocessor controls the operation of the controller. The controller uses direct-memory access (DMA) to transfer data between the cartridge tape drive and the CPU. When the CPU wants to initiate an operation, it sends a command to the controller. The controller completes the specified operation and then interrupts the CPU to notify it that the operation is complete.

1.2.5. Memory Subsystem: The Memory Subsystem includes an Error Checking and Correcting (ECC) Controller board and one or more Memory Array boards.

The ECC Controller can control the operation of up to 4 megabytes of dynamic read/write memory. The data can be transferred as bytes (eight bits), words (16 bits), or long words (32 bits); the controller translates the width of the data and places the data in the proper locations. The controller transparently corrects all single-bit errors (soft) and detects all double-bit errors (hard). The controller uses a soft-error logging system that counts soft errors in each 64K-byte block of memory.

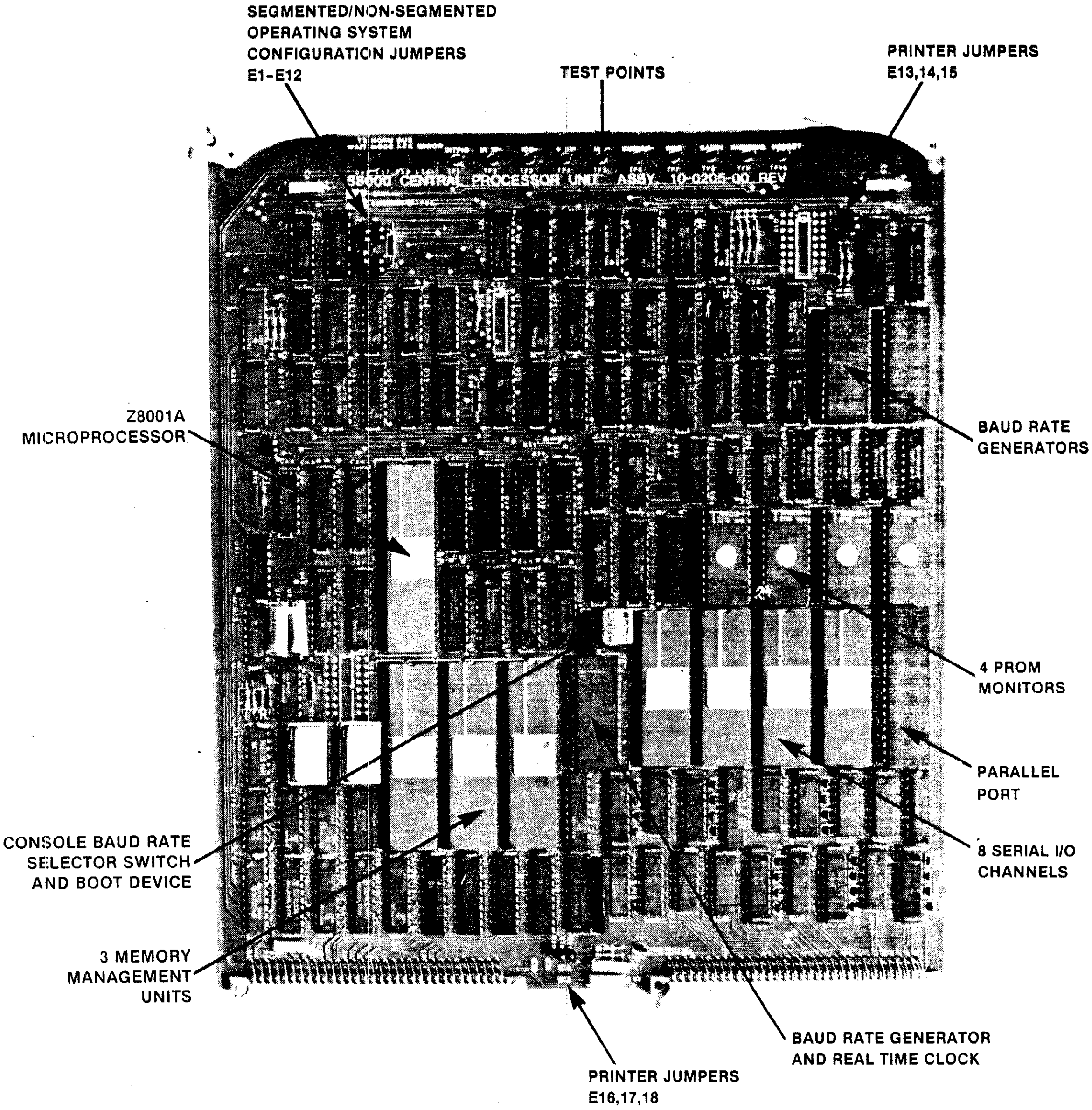


Figure 1-2 System 8000 CPU Board (CPU)

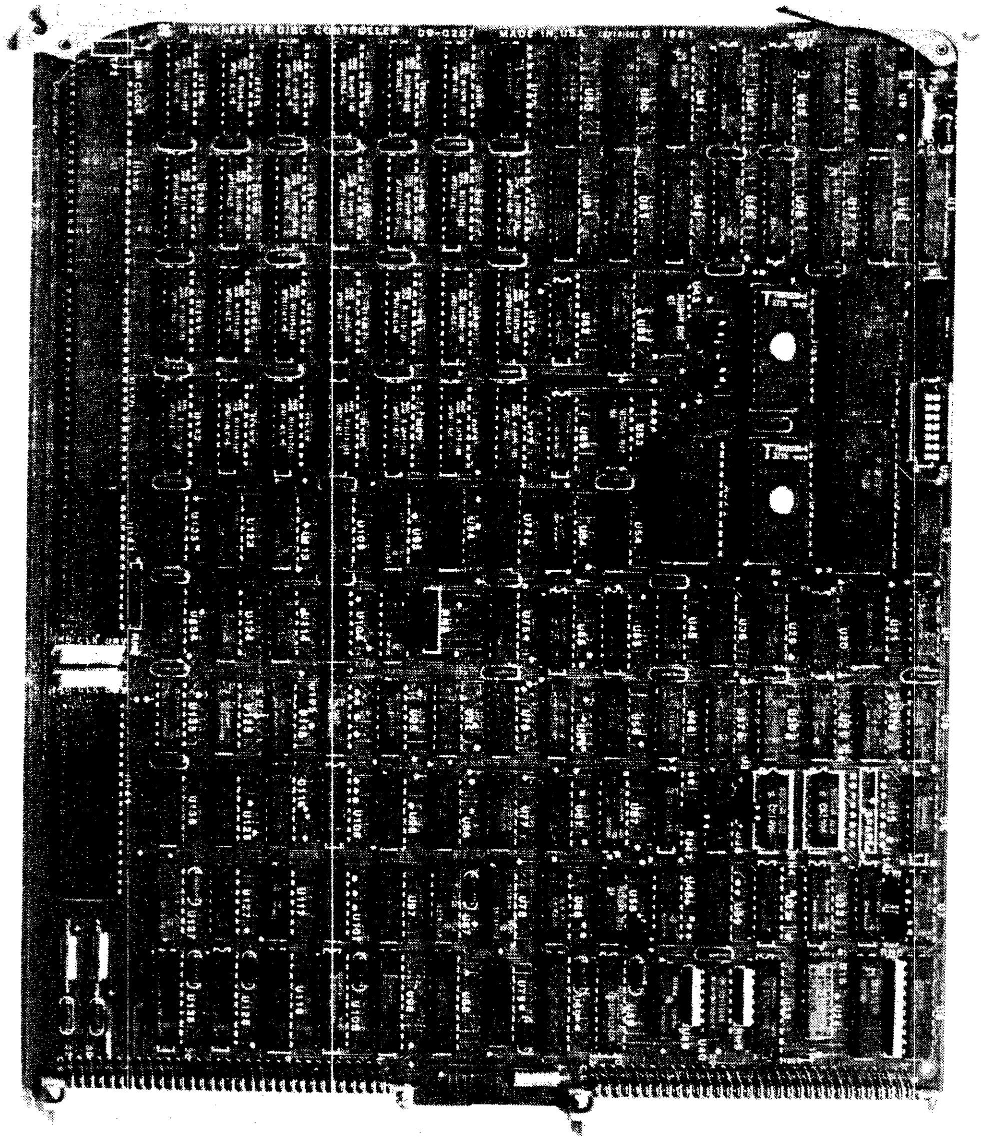


Figure 1-3 System 8000 Winchester Disk Controller (WDC)

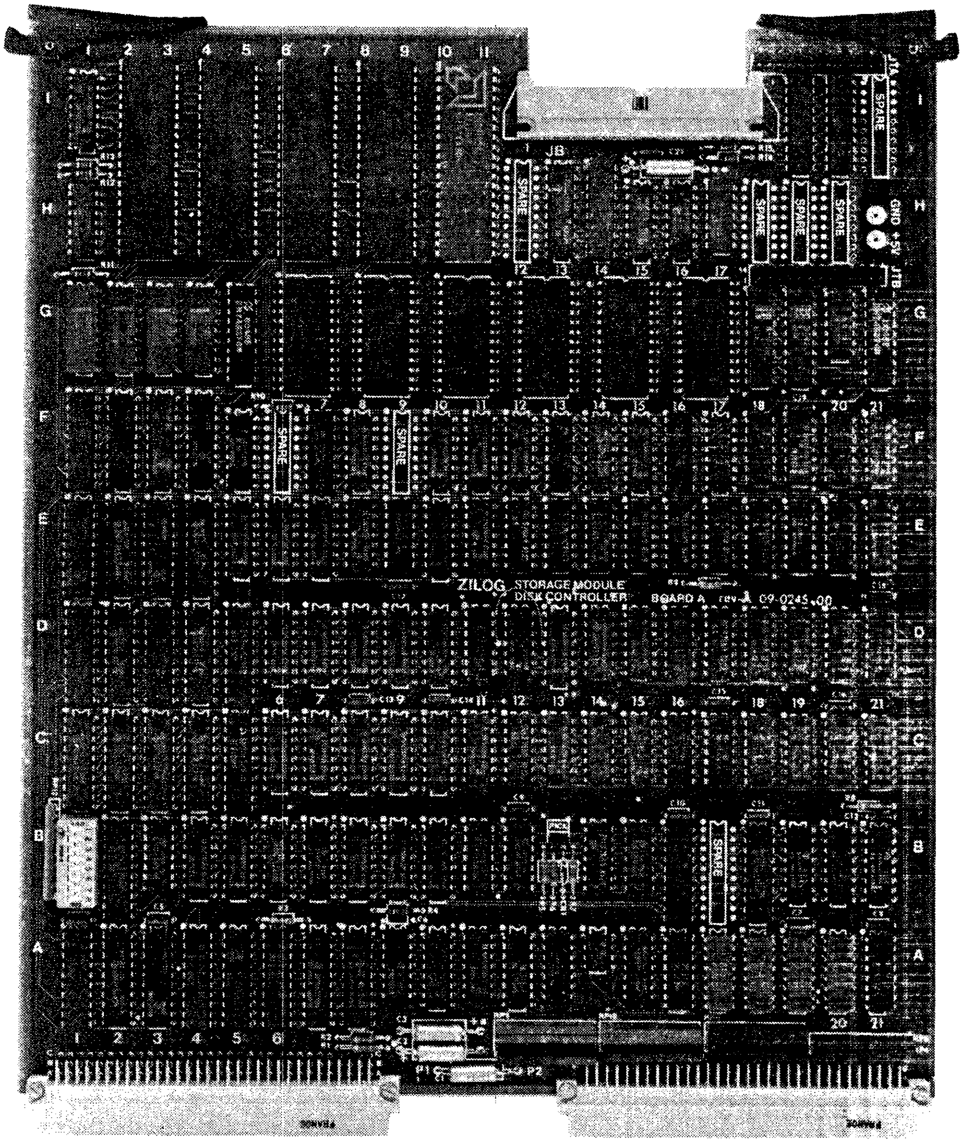


Figure 1-4 System 8000 Storage Module Device Controller (SMDC A)

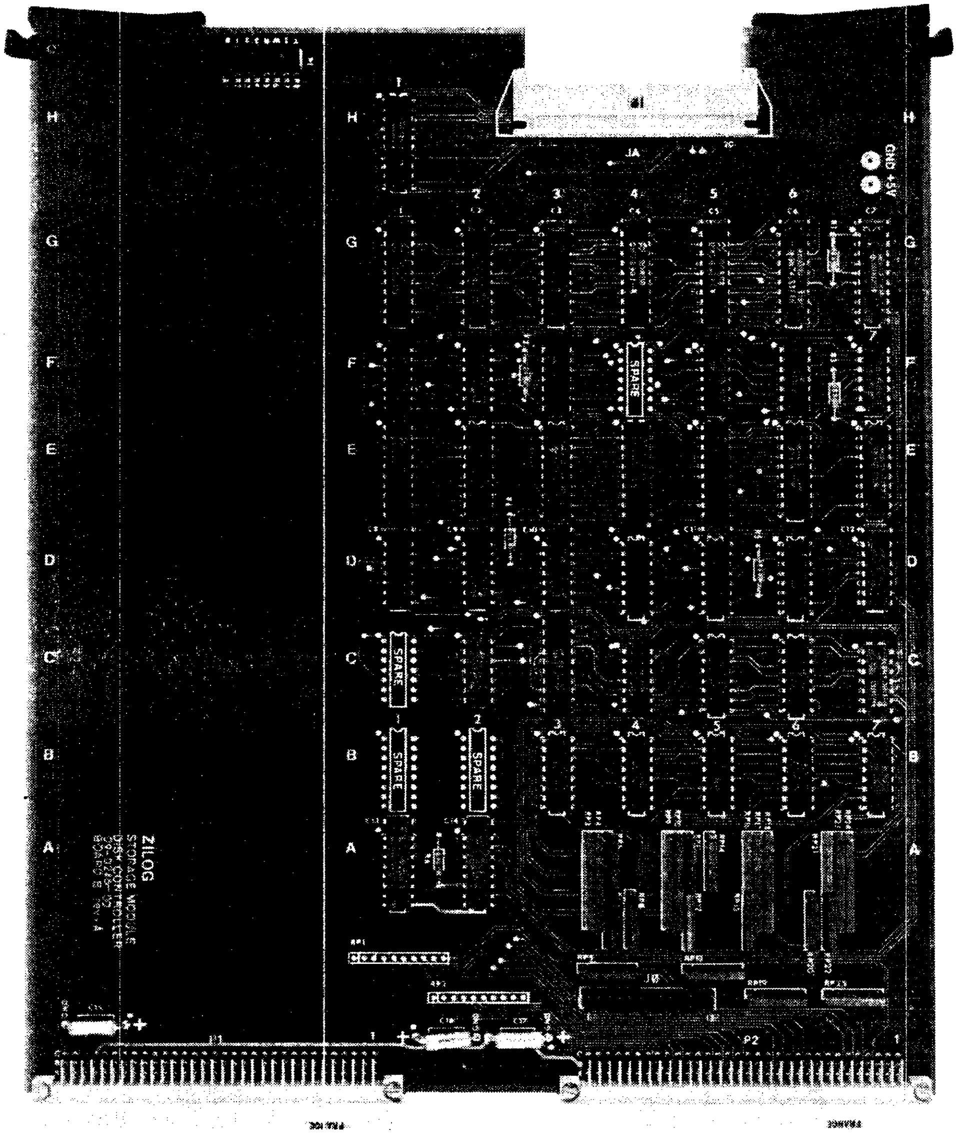


Figure 1-5 System 8000 Storage Module
Device Controller (SMDC B)

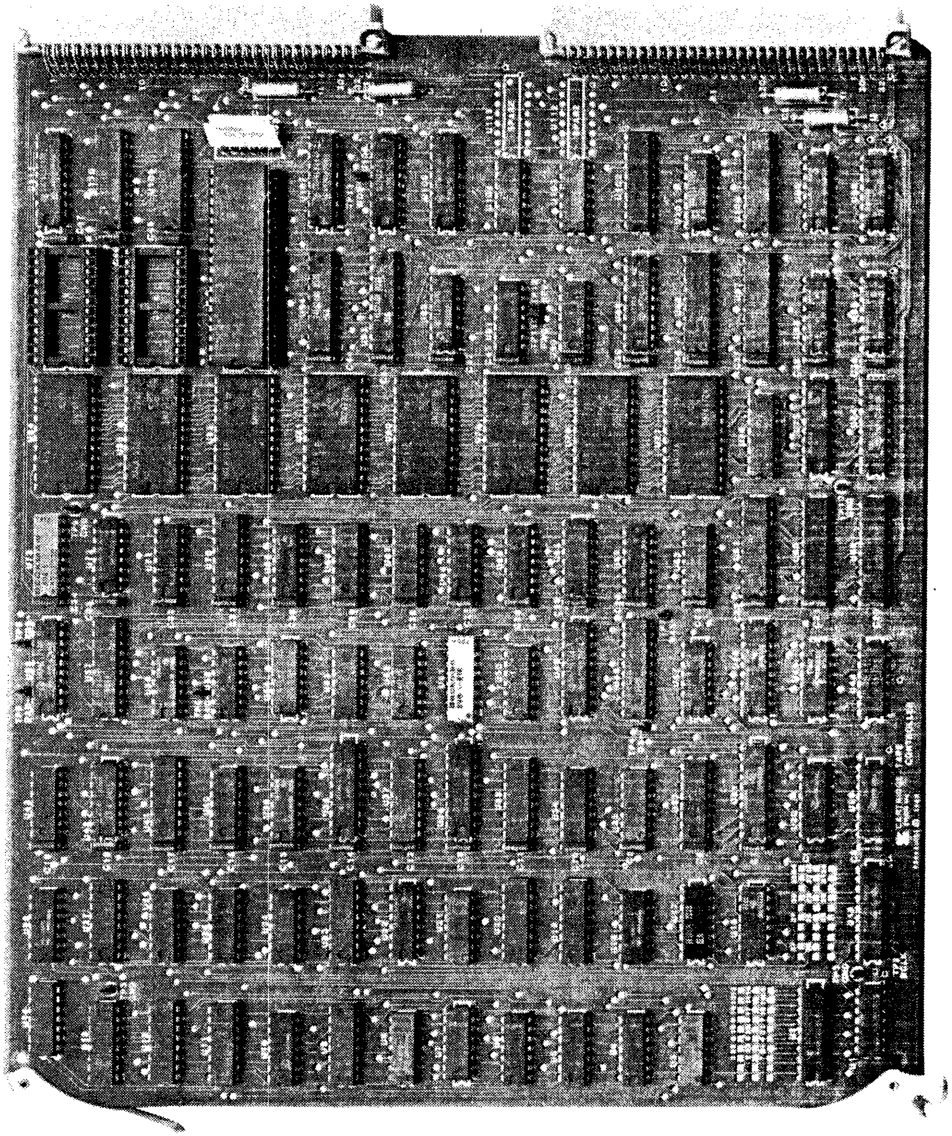


Figure 1-6 System 8000 Tape Cartridge Controller (TCC)

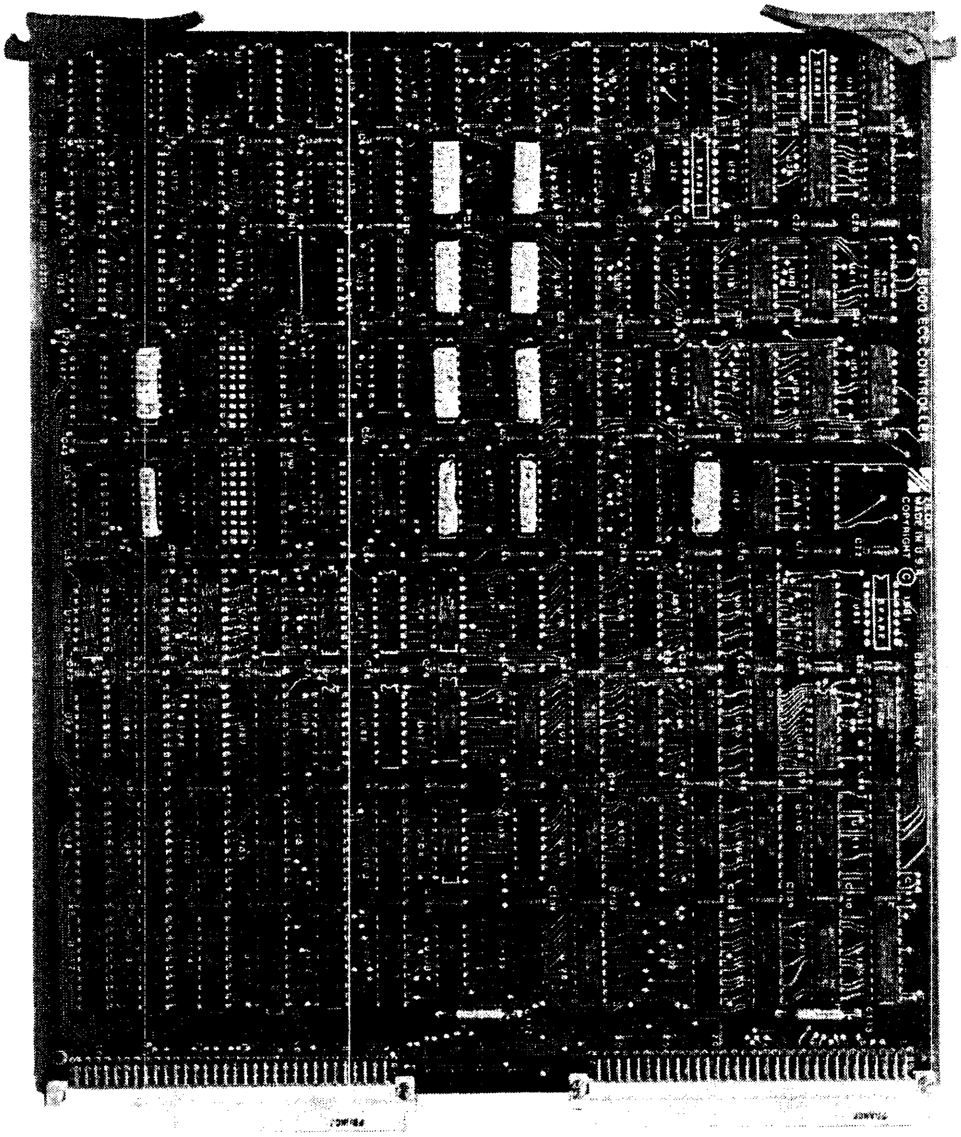


Figure 1-7 System 8000 ECC Controller

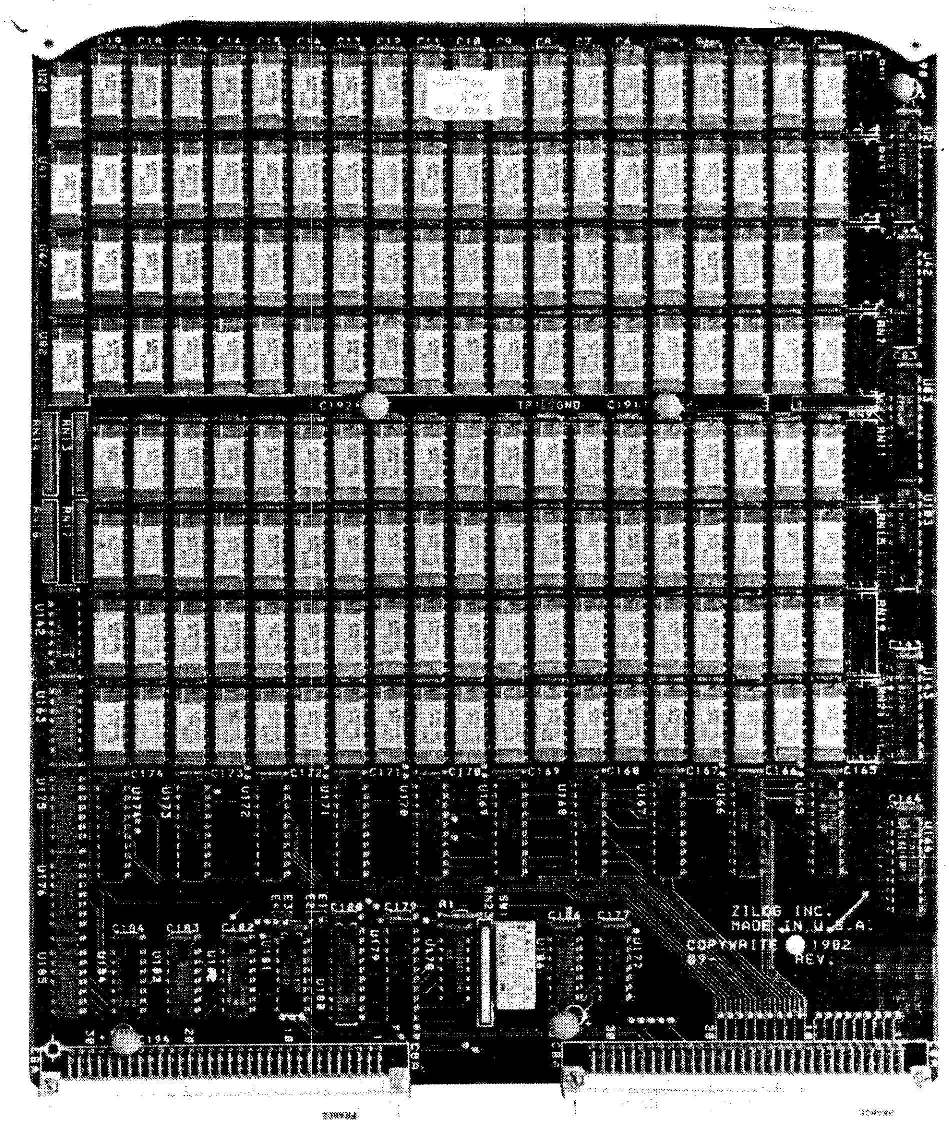


Figure 1-8 1M Byte Memory Array (MEM)

In addition to data storage space, the memory arrays have storage for the check bits used by the error checking and correcting logic. During memory transactions, all data passes through the memory controller; the memory has no direct connection to the system bus (ZBI) (Figure 1-9). Instead, a high-speed, 32-bit bus connects the Memory Array board to the ECC Controller board.

1.2.6. **Secondary Serial Boards (SSB):** Up to two are permitted in certain optional configurations. Each SSB provides for eight additional input/output (I/O) asynchronous full duplex serial channels for terminal distribution to TTYs 8 - 15 and 16 - 23. Each SSB also provides a Centronics/Data Products parallel printer port. The SSBs are connected to the system backplane and supplement the serial and parallel I/O on the CPU board. The Z8001A microprocessor on the CPU board acts as the host and controls the SSB via the system Z-Bus Backplane Interconnect (ZBI).

1.3. System Expansion

The System 8000 can be expanded by adding:

- Disk drives up to a total of four,
- Cartridge tape drives up to a total of four,
- Memory Arrays up to a total of 4 Mbytes,
- SSBs up to a total of 24 users and two parallel printer ports.

Consult your sales representative for specific details concerning optional configurations.

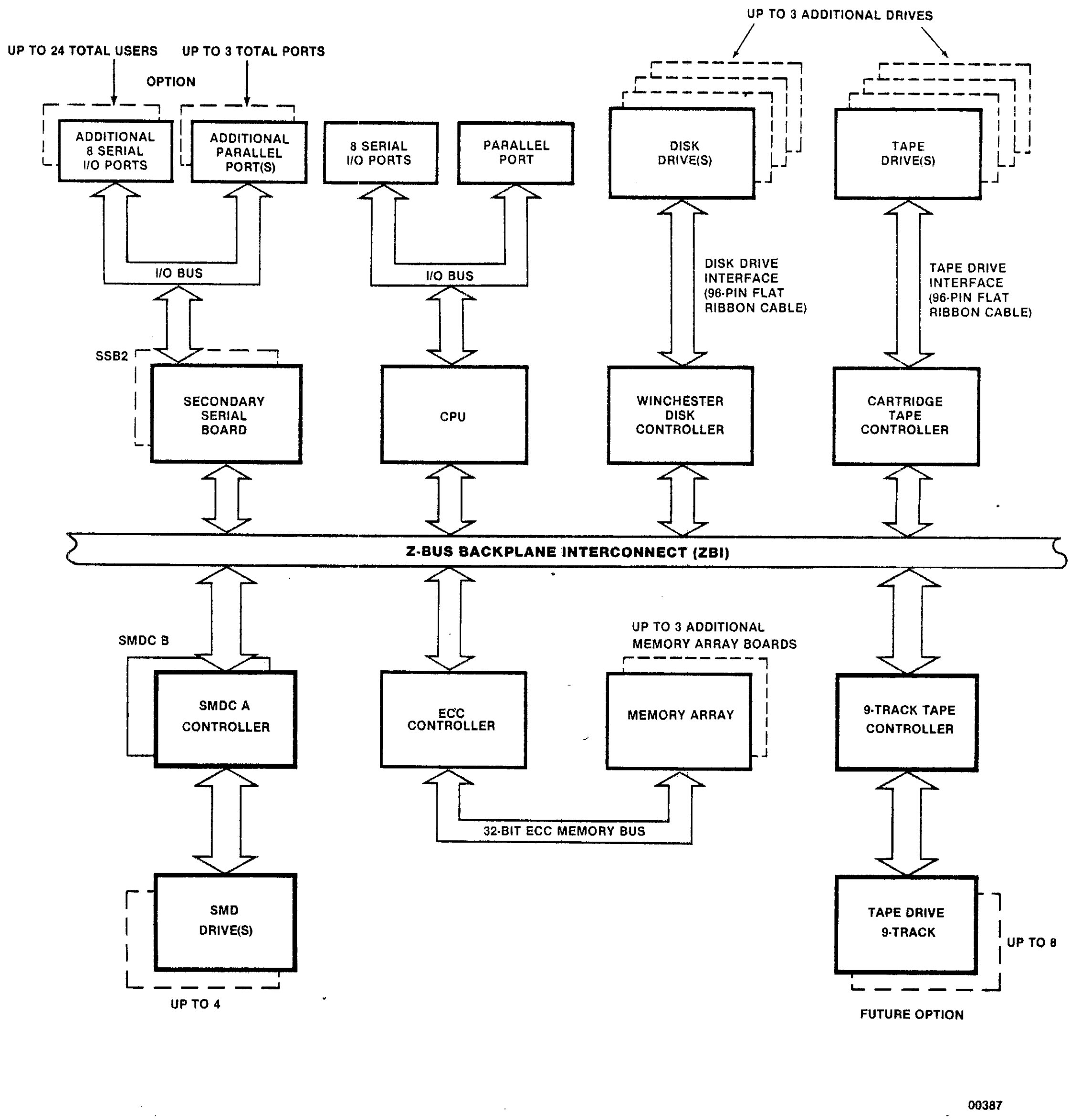


Figure 1-9 System 8000, Functional Relationships

SECTION 2
SYSTEM SPECIFICATIONS

2.1. Introduction

This section contains information on the electrical, physical, and environmental specifications for a System 8000 having a single Disk/Tape module and Processor module. The ZBI backplane pin assignments are included for the printed wiring assemblies mounted in the processor card cage.

2.2. Electrical Specifications

Table 2-1 lists the electrical specifications for the System 8000. All voltages in the following table are single phase, and 47 to 63 hertz.

Table 2-1. Electrical Specifications

	JAPAN	U.S.A.	EUROPE	UNITED KINGDOM
PARAMETER	CHARACTERISTICS			
NOMINAL LINE VOLTAGE	100 Vac +/- 10%	117 Vac +/- 10%	220 Vac +/- 10%	240 Vac +/- 10%
CURRENT (SUSTAINED)	5A, max.	5A, max.	3A, max.	3A, max.
CURRENT (SURGE)	8A, max.	8A, max.	4A, max.	4A, max.
FUSE MODULE SYSTEM	3A 10A	3A 10A	1.5A 5A	1.5A 5A

NOTE: Current requirements may change with variation of system configuration.

2.3. Performance Specifications

Tables 2-2, 2-3, 2-4, and 2-5 define the performance characteristics of the system, Winchester disk, SMD, and cartridge tape drives.

Table 2-2. System Performance Characteristics

ITEM	CHARACTERISTIC
Processor:	Segmented 48-pin Z8001A CPU
CPU Clock Frequency:	5.5 MHz
I/O:	Up to 24 RS-232C serial I/O ports and three parallel printer ports (factory set for Centronics Interface)
Baud Rate:	From 110 to 19,200 baud (set by software)
Front Panel:	Cutouts for keylock ON/OFF switch, RESET switch, and START switch. Translucent plastic for three indicator lamps: POWER (+5V DC), USER (CPU is in normal state), and DMA (CPU is giving up the bus for Direct Memory Access devices). Refer to Figure 2-1.
Rear I/O Panels:	Up to seven 96-pin I/O Panel connectors interface the CPU Module to other Disk/Tape Module I/O Panels and terminal distribution panels. Each terminal distribution panel supplies eight 25-pin (TTY0-TTY23) user terminal connections and one 25-pin printer port. TTY1 is labeled console for System Administrator.
CPU Module Power Supply:	35A Max, +5Vdc +/-0.4% 4A Max, +12Vdc +/-0.2% 2A Max, -5Vdc +/- 0.2%
DISK/TAPE Module Power Supply:	7.6A Max, +5Vdc +/-0.2% 3.5A Max, -5Vdc +/-0.2% 0.6A Max, +12Vdc +/-1.0% 4.0A Max, -12Vdc +/-0.2% 5.0A Max, +24Vdc +/-0.2% 2.0A Max, -24Vdc +/-0.2%

Environmental:	Operating temperature: 50 degrees F (10 degrees C) minimum 104 degrees F (40 degrees C) maximum Relative humidity: 80% noncondensing
Cabinet Size:	(Free Standing) Height: 33 inches (84 cm) Width 19 inches (48 cm) Depth: 24 inches (61 cm)
Total Weight:	Approximately 250 pounds (114 kg)
Rack Mount:	
Overall Size:	Height: 5 ft. 8-1/2 inches (173.6 cm) Width: 22-1/4 inches (56.6 cm) Depth: 31-1/2 inches (70.9 cm)
Total Weight:	Approximately 400 pounds (182 kg)

Table 2-3. 32 MB Winchester Disk Drive Performance Characteristics

ITEM	CHARACTERISTIC
Data Capacity (Unformatted)	
Bytes per Track	13,440 bytes
Bytes per Surface	8,010,240 bytes
Total per Disk Drive	32,040,960 bytes
Track Format	
Variable	User Defined (refer to Zeus Adm. Man. for details 03-3246)
Number of Byte Clocks Per Revolution	13,440
Recording Mode	
Interface	NRZ
Disks	MFM
Data Transfer Rate	6.45 Mbit/s (806 kByte/s)
Data Interface	NRZ DATA + CLOCK
Rotational Speed	3600 r/min +1.5%
Average Latency	8.33 ms
Tracks Per Surface	600
Step Pulse Rate	50 kHz +20% (20 microseconds between Step pulses)
Single Track Seek Time	10 ms Max.
Average Seek Time (Step Pulse Rate of 50 kHz +20%)	50 ms
Maximum Seek Time (605 Tracks) (Step Pulse Rate of 50 kHz +20%)	100 ms
Allowable Read Error Rates:	To be supplied
Hard:	<1 in 10^{12} bits
Soft:	<1 in 10^{10} bits
Allowable Seek Error Rate:	<1 in. 10^{6} seeks

Table 2-4. SMD Performance Characteristics

ITEM	CHARACTERISTIC
Storage Capacity	84,439,040 Bytes
Number of Cylinders	589
Tracks per Cylinder	7
Cylinder Capacity	143,360 Bytes
Track Capacity	20,480 Bytes
Average Rotational Latency	8.3 ms
Positioning Time	
Track to Track	5 ms
Average	20 ms
Maximum	40 ms
Rotational Speed	3600 RPM +-1%
Transfer Rate	1.229MByte/sec
Encoding Method	MFM
Interface Data	NRZ
Recording Density	9550 BPI
Track Density	720 TPI
Start/Stop Time	<20/ <40 sec
Interface	SMD
Number of Sectors	128 (maximum) (refer to Zeus Adm. Man. for details 03-3246)
Allowable Read Error Rates:	
Hard:	10 per 10^{14} (Max)
Soft:	10 per 10^{11} bits (Max)
Allowable Seek Error Rates:	10 per 10^{8} seeks (correctable)

Table 2-5. Tape Cartridge Drive Performance Characteristics

ITEM	CHARACTERISTIC
Storage Capacity (Unformatted)	17.2 Mbytes max.
Read/Write speed	30 inches per second (ips)
Rewind/Search speed	90 inches per second (ips)
Tracks	4
Recording density	6400 BPI
Data Transfer Rate	192,000 Bits/Sec
Error Rates	<1 Error in 10^8 Bits

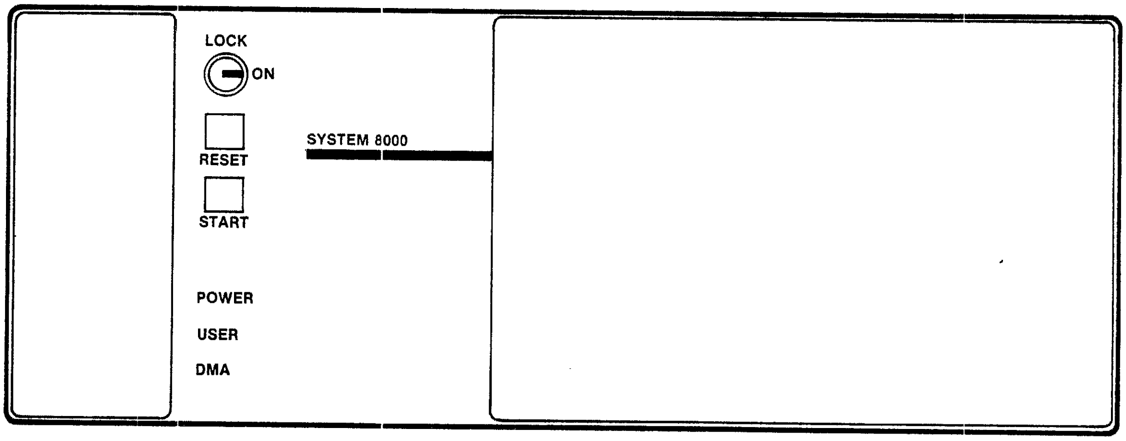


Figure 2-1 Processor Module Controls and Indicators

2.4. Modules

The modules, with their side panels removed, can be mounted in standard 19-inch racks. In stand-alone configurations, the modules are stacked. The dimensions of the module with its side panel removed are given in Figure 2-2.

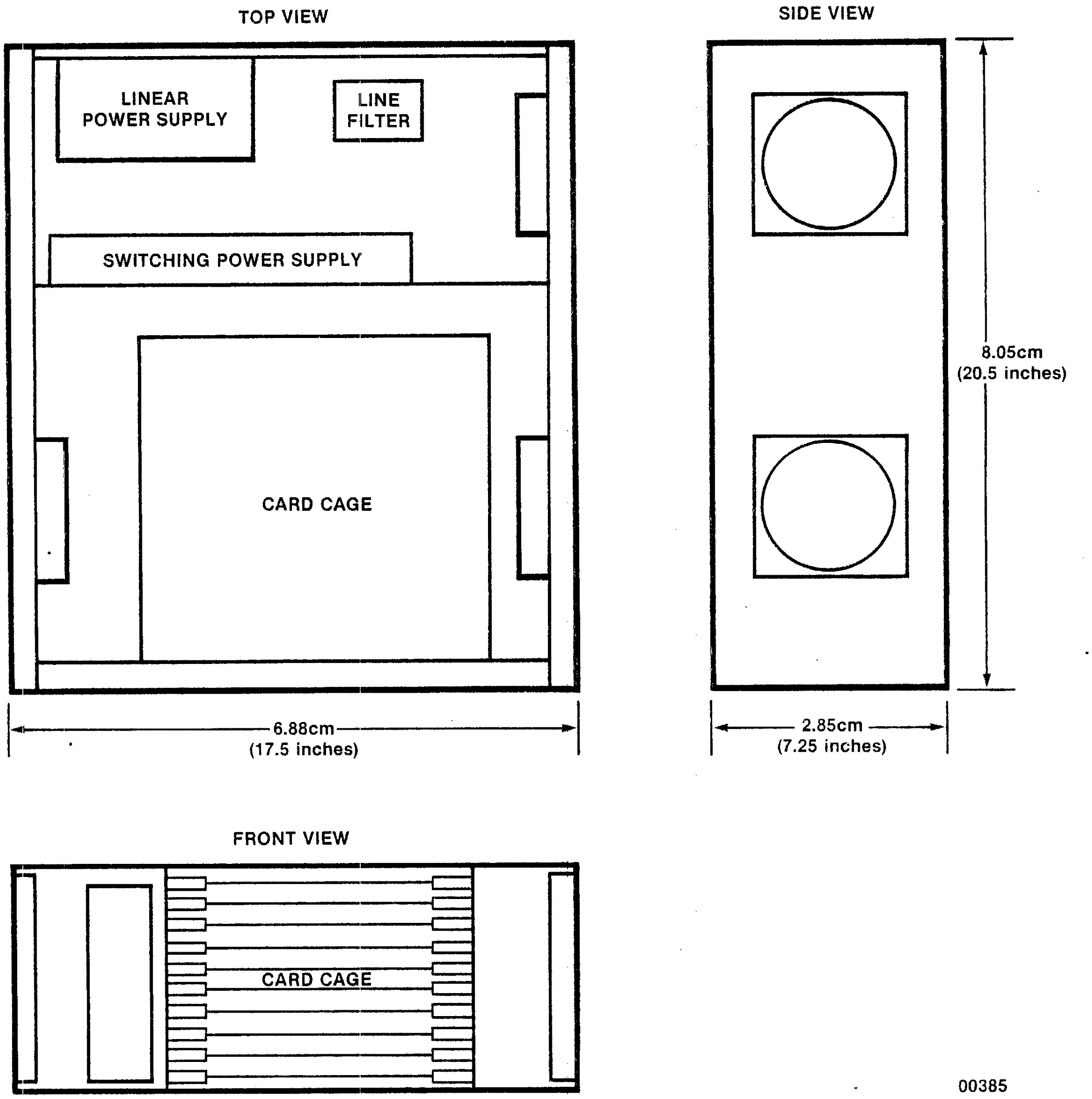


Figure 2-2 Module Dimensions (Without Side Panels)

2.4.1. I/O Connectors: Figures 2-3 and 2-4 identify the 96-pin terminal, disk, tape, SMD, and optional I/O connectors on the two Processor Module I/O connector panels for Models 21 and 31. The two panels are cabled to the I/O and terminal distribution panels on the Disk/Tape and Accessory Modules of the system configuration. Table 2-6 lists the required mating connector cable and sources. Table 2-7 lists the pin assignments of the serial (TTY) I/O connectors. Table 2-8 lists the pin assignments of the parallel printer connectors.

The terminal distribution panel is normally located on the rear of the Disk/Tape Module. It can be located on other modules depending on the system configuration (Figures 3-3,3-5, Section 3).

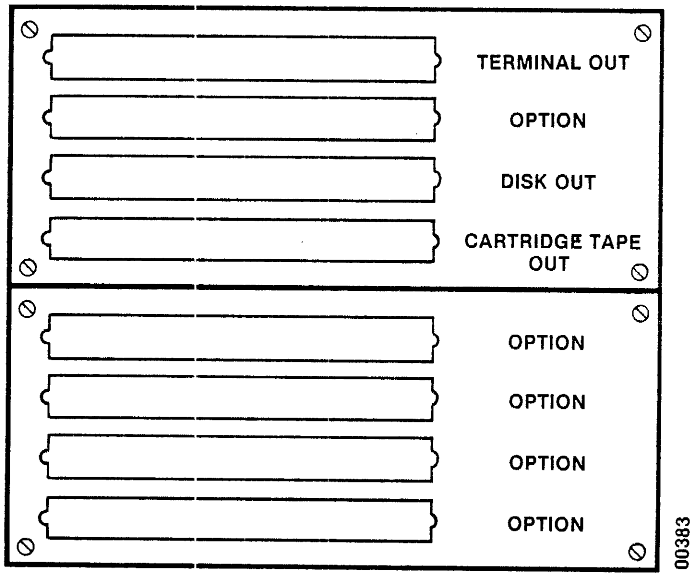


Figure 2-3 Model 21 Processor Module I/O Connector Panel

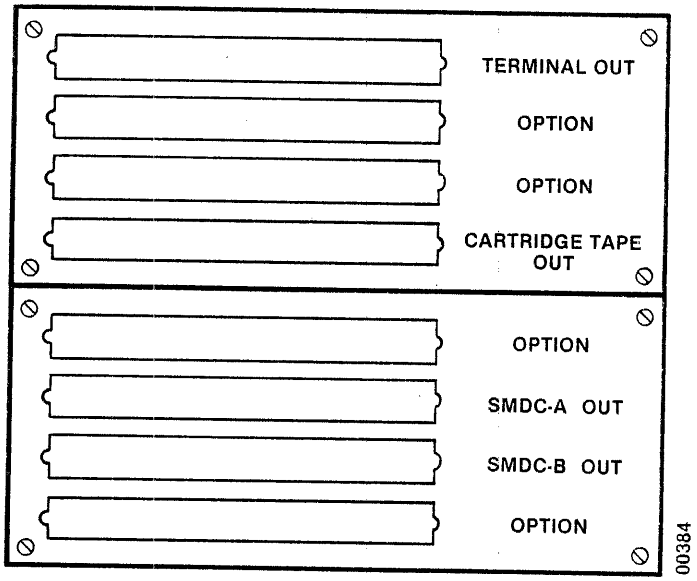


Figure 2-4 Model 31 Processor Module I/O Connector Panel

Table 2-6. I/O Connectors

DESIGNATION	DESCRIPTION	VENDOR/PART NUMBER
Printer	25-pin D Connector	ITT Cannon DBUE25SBB
TTY0-TTY7	25-pin D Connector	TRW Cinch DBUE25SBF
Terminal Expansion Cable	96-pin Din Connector (both ends)	Zilog Cable P/N 59-0217
Disk Drive Cable	96-pin Din Connector (both ends)	Zilog Cable P/N 59-0217
Tape Drive Cable	96-pin Din Connector (both ends)	Zilog Cable P/N 59-0217

Table 2-7. TTY Connector, Pin Assignments

SIGNAL NAME	PIN
Ground	7
DTR	20
RTS	4
TXD	2
DSR	6
CTS	5
RXD	3

Table 2-8. Printer Connector, Pin Assignments

Data Products Interface			
Signal Name	P2 Backplane	Printer Port Connector Pins	Printer Connector Pins
DATA 0	P2-1C	1	B
DATA 1	P2-2C	2	F
DATA 2	P2-3C	3	L
DATA 3	P2-5C	4	R
DATA 4	P2-6C	5	V
DATA 5	P2-8C	6	Z
DATA 6	P2-9C	7	n
DATA 7	Not Used	Not Used	Not Used
DATA STROBE	P2-12C	9	j
INPUT PRIME	Not Used	Not Used	Not Used
DATA DEMAND	P2-16C	11	E
FAULT	P2-20C	12	C
READY	P2-17C	23	cc
ONLINE	P2-21C	24	y
SIGNAL GROUND	P2-19C	22	X

Centronics Interface

Signal Name	P2 Backplane	Printer Port Connector Pins	Printer Connector Pins
DATA 0	P2-1C	1	2
DATA 1	P2-2C	2	3
DATA 2	P2-3C	3	4
DATA 3	P2-5C	4	5
DATA 4	P2-6C	5	6
DATA 5	P2-8C	6	7
DATA 6	P2-9C	7	8
DATA 7	Not Used	Not Used	Not Used
DATA STROBE	P2-12C	9	1
INPUT PRIME	Not Used	Not Used	Not Used
ACKNOWLEDGE	P2-16C	11	10
FAULT	P2-20C	12	32
GROUND	P2-32C	18	24
GROUND	P2-32B	19	25
GROUND	P2-32A	20	26
BUSY	P2-17C	23	11
SELECT	P2-21C	24	13

2.5. Backplane (ZBI) Pin Assignments

Figure 2-5 and 2-6 show the backplane slot assignments for Processor Module Printed Wiring Asssemblies (PWA's). The connectors designated J11 through J20 (on the right) connect to the ZBI system bus. The connectors designated J21 through J30 (on the left) are auxiliary connectors. The P2 connectors J22, J25, J26, and J27 are labeled optional and allow for future board configuration. The pin assignments of all ZBI connectors are the same. Table 2-9 lists the ZBI pin assignments.

Table 2-9. ZBI Backplane Connector Pin Assignments (J11 through J20)

PIN	Row A SIGNAL	Row B SIGNAL	Row C SIGNAL
1	RESET\	WAIT\	CAVAIL
2	CAI\	CAO\	CPUREQ\
3	BAI\	BAO\	BUSREQ\
4	MMAI\	MMAO\	GND
5	IEI3	IEO3	MMREQ\
6	IEI2	IEO2	-
7	IEI1	IEO1	GND
8	INT1\	INT2\	INT3\
9	R/W\	B/W\	W/LW\
10	S2	S3	S4
11	SO	S1	GND
12	ME\	AS\	DS\
13	-	STOP\	N/S\
14	-	-	-
15	AD31	-	GND
16	AD28	AD29	AD30
17	AD25	AD26	AD27
18	AD22	AD23	AD24
19	AD20	AD21	GND
20	AD17	AD18	AD19
21	AD14	AD15	AD16
22	AD11	AD12	AD13
23	AD9	AD10	GND
24	AD6	AD7	AD8
25	AD3	AD4	AD5
26	ADO	AD1	AD2
27	PWRBAD\	MCLK	BLCK
28	+5v	+5v	+5v
29	-5v	-5v	-5v
30	+12v**	+12v**	+12v**
31	-12v*	-12v*	-12v*
32	GND	GND	GND

* -12v is allocated space on the ZBI backplane, but is not used nor generated by the System 8000 .

** +12V is allocated space on the ZBI backplane, but is not used.

The pin assignments for the PWA'S installed in backplane connectors P2/J21 through P2/J30 are described in Tables 2-10 through 2-16. The following list indicates the tables and their present respective board assignments:

1. Table 2-10:
   Connector J21, CPU Board

2. Table 2-11:
   Connector J22, Secondary Serial Board (Option)

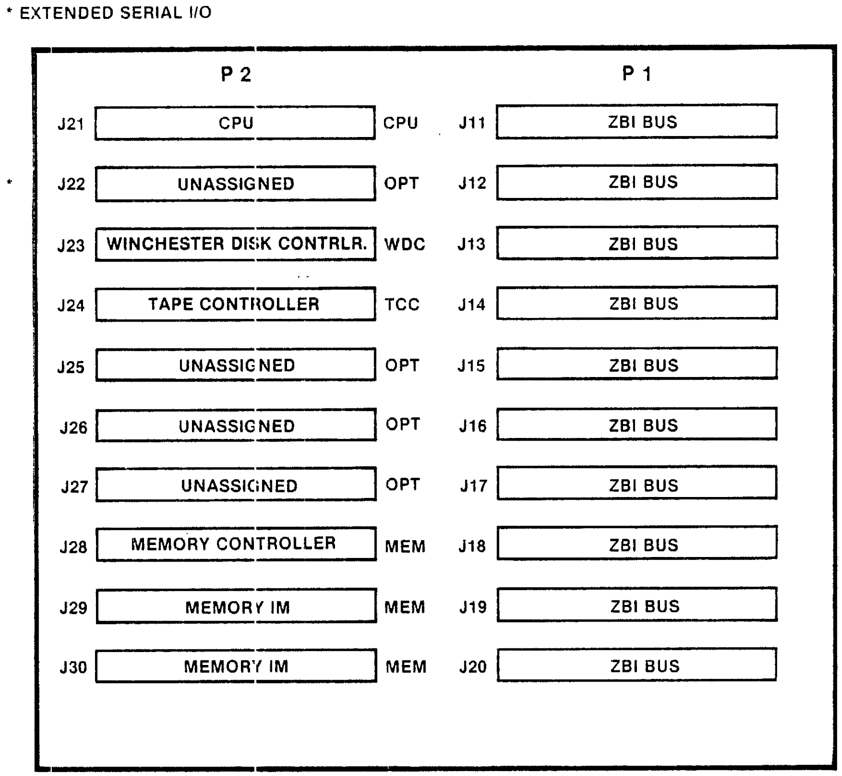
3. Table 2-12:
   Connector J23, Winchester Disk Controller Board (Model 21 only)

4. Table 2-13:
   Connector J24, Tape Cartridge Controller Board

5. Table 2-14:
   Connector J26, Storage Module Device Controller A (Model 31 only)

6. Table 2-15:
   Connector J27, Storage Module Device Controller B (Model 31 only)

7. Table 2-16:
   Connectors J28 through J30, are dedicated for the memory bus (Memory Subsystem Controller and Memory boards).



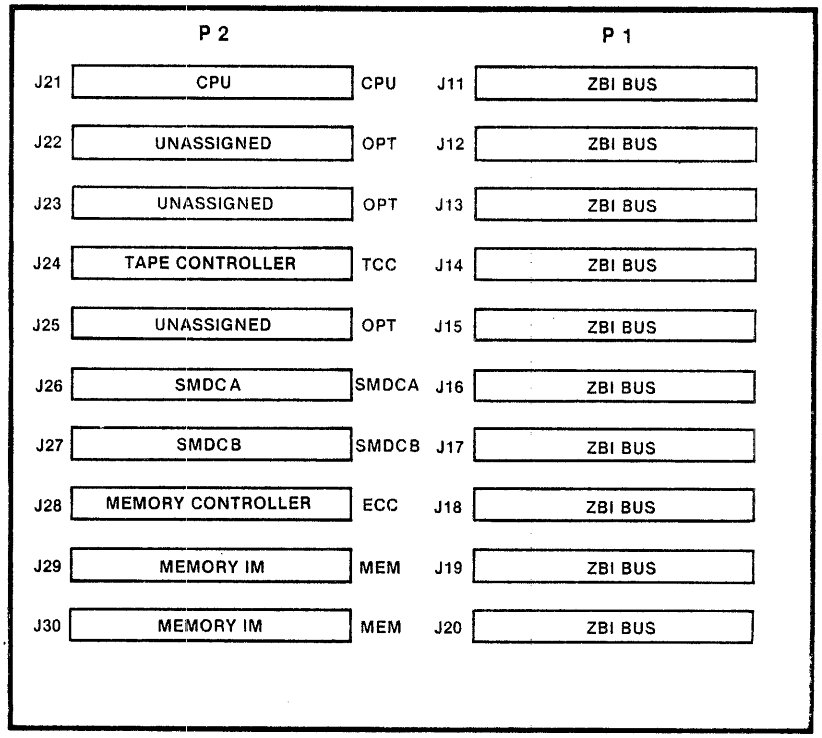
* EXTENDED SERIAL I/O

	P 2			P 1
J21	CPU	CPU	J11	ZBI BUS
* J22	UNASSIGNED	OPT	J12	ZBI BUS
J23	WINCHESTER DISK CONTRLR.	WDC	J13	ZBI BUS
J24	TAPE CONTROLLER	TCC	J14	ZBI BUS
J25	UNASSIGNED	OPT	J15	ZBI BUS
J26	UNASSIGNED	OPT	J16	ZBI BUS
J27	UNASSIGNED	OPT	J17	ZBI BUS
J28	MEMORY CONTROLLER	MEM	J18	ZBI BUS
J29	MEMORY IM	MEM	J19	ZBI BUS
J30	MEMORY IM	MEM	J20	ZBI BUS

00163

Figure 2-5 Model 21 Backplane Slot Assignments for Processor Module PWA's

* EXTENDED SERIAL I/O



	P 2			P 1
J21	CPU	CPU	J11	ZBI BUS
* J22	UNASSIGNED	OPT	J12	ZBI BUS
J23	UNASSIGNED	OPT	J13	ZBI BUS
J24	TAPE CONTROLLER	TCC	J14	ZBI BUS
J25	UNASSIGNED	OPT	J15	ZBI BUS
J26	SMDCA	SMDCA	J16	ZBI BUS
J27	SMDCB	SMDCB	J17	ZBI BUS
J28	MEMORY CONTROLLER	ECC	J18	ZBI BUS
J29	MEMORY IM	MEM	J19	ZBI BUS
J30	MEMORY IM	MEM	J20	ZBI BUS

00382

Figure 2-6 Model 31 Backplane Slot Assignments
for Processor Module PWA's

Table 2-10. CPU Board, Connector P2/J21, Slot 1

PIN	Row A SIGNAL	Row B SIGNAL	Row C SIGNAL
1	TXRTN0	RXD0	DATA0
2	TXD0	CTS0	DATA1
3	RTS0	DTR0	DATA2
4	DSR0	RXD1	GND
5	TXD1	CTS1	DATA3
6	RTS1	DTR1	DATA4
7	DSR1	RXD2	GND
8	TXD2	CTS2	DATA5
9	RTS2	DTR2	DATA6
10	DSR2	RXD3	DATA7
11	TXD3	CTS3	GND
12	RTS3	DTR3	DATA STROBE/DATA STROBE\
13	DSR3	RXD4	N.U./INPUT PRIME
14	TXD4	CTS4	TXRTN1
15	RTS4	DTR4	GND
16	DSR4	RXD5	D.D./ACKNOWLEDGE\
17	TXD5	CTS5	BUSY\
18	RTS5	DTR5	TXRTN2
19	DSR5	RXD6	GND
20	TXD6	CTS6	IFVALID/FAULT\
21	RTS6	DTR6	ON-LINE/SELECT
22	DSR6	RXD7	F.P BUSACK INDICATOR
23	TXD7	CTS7	F.P. POWER-ON INDICATOR (GND)
24	RTS7	DTR7	F.P. NORMAL INDICATOR
25	DSR7	TXRTN5	NMI SWITCH (NORMALLY CLOSED)
26	TXRTN3	TXRTN6	NMI SWITCH (NORMALLY OPEN)
27	TXRTN4	TXRTN7	SW RESET
28	+5v	+5v	F.P. INDICATOR V+ (+5v)
29	-5v	-5v	-5v
30	+12v	+12v	+12v
31	-12v	-12v	-12v
32	GND	GND	GND

Table 2-11. Secondary Serial Board Connector P2/J22, Slots 2,5 (Optional)

PIN	Row A SIGNAL	Row B SIGNAL	Row C SIGNAL
1	TXRTN0	RXD0	DATA0
2	TXD0	CTS0	DATA1
3	RTS0	DTR0	DATA2
4	DSR0	RXD1	GND
5	TXD1	CTS1	DATA3
6	RTS1	DTR1	DATA4
7	DSR1	RXD2	GND
8	TXD2	CTS2	DATA5
9	RTS2	DTR2	DATA6
10	DSR2	RXD3	DATA7
11	TXD3	CTS3	GND
12	RTS3	DTR3	DATA STROBE/DATA STROBE\
13	DSR3	RXD4	N.U/INPUT PRIME
14	TXD4	CTS4	TXRTN1
15	RTS4	DTR4	GND
16	DSR4	RXD5	D.D./ACKNOWLEDGE\
17	TXD5	CTS5	BUSY\
18	RTS5	DTR5	TXRTN2
19	DSR5	RXD6	GND
20	TXD6	CTS6	IFVALID/FAULT\
21	RTS6	DTR6	ON-LINE/SELECT
22	DSR6	RXD7	
23	TXD7	CTS7	GND
24	RTS7	DTR7	N.V./LP. CONT
25	DSR7	TXRTN5	
26	TXRTN3	TXRTN6	
27	TXRTN4	TXRTN7	
28	+5v	+5v	+5v
29	-5v	-5v	-5v
30	+12v	+12v	+12v
31	-12v	-12v	-12v
32	GND	GND	GND

Table 2-12. Winchester Disk Controller Connector P2/J23, Slot 3 (Model 21 Only)

	Row A	Row B	Row C
PIN	SIGNAL	SIGNAL	SIGNAL
1	CB0		
2	CB2	CB1	
3	CB3		
4	CB5	CB4	GND
5	CB6		
6	DIRECTION\	CB7	
7	CWD0\		GND
8		CWD1\	
9	CYACK\	ATTN\	
10	INDEX\		
11		SECTOR\	GND
12	WRITE CLK\	SEEKEND\	
13	DRIVEFAULT\	WRITE-CLK	
14	SYSTEMCLK	ATTACK	
15	UNITACK0\	SYSTEMCLK\	GND
16	BI-DATA\	UNITACK1\	
17	UNITACK2\	BI-DATA	
18	READ-ENABLE\	UNITACK3\	
19	MASTER RESET\	READ ENABLE	GND
20	WRITE ENABLE	WRITE ENABLE\	
21			
22			
23			GND
24			
25			
26			
27			
28	+5v	+5v	+5v
29	-5v	-5v	-5v
30	+12v	+12v	+12v
31	-12v	-12v	-12v
32	GND	GND	GND

Table 2-13. Tape Cartridge Controller, Connector P2/J24, Slot 4

PIN	Row A SIGNAL	Row B SIGNAL	Row C SIGNAL
1		SLD\	GND
2		RDY\	GND
3		WND\	GND
4		FLG\	GND
5		LPS\	GND
6		FUP\	GND
7		BSY\	GND
8		EWS\	GND
9		RWD\	GND
10		REV\	GND
11		FWD\	GND
12		HSP\	GND
13		WEN\	GND
14		SL1\	GND
15		SL2\	GND
16		SL4\	GND
17		SLG\	GND
18		RNZ\	GND
19		RDS\	GND
20		DAD\	GND
21		WDE\	GND
22		WNZ\	GND
23		TR2\	GND
24		WDS\	GND
25		TR1\	GND
26			
27			
28	+5v	+5v	+5v
29	-5v	-5v	-5v
30	+12v	+12v	+12v
31	-12v	-12v	-12v
32	GND	GND	GND

Table 2-14. Storage Module Device Controller Board A Connector P2/J26, Slot 6 (Model 31)

PIN	Row A SIGNAL	Row B SIGNAL	Row C SIGNAL
1	CYLTAG+	SELTAG-	SELTAG+
2	CYLTAG-	HEADTAG+	HEADTAG-
3		CONTAG-	CONTAG+
4	SEL2-	SEL2+	GND
5		SEL8-	SEL8+
6		SEL4+	SEL4-
7	SEL1+	SEL1-	GND
8			
9		BIT2+	BIT1-
10		BIT0-	BIT1+
11	BIT2-	BIT0+	GND
12	BIT3-	BIT3+	BIT7-
13	BIT9+	SPARE-	BIT7+
14	SPARE+		BIT9-
15	BIT4-	BIT4+	GND
16		BIT5+	BIT5-
17		BIT8-	BIT8+
18		BIT6-	BIT6+
19	INDEX+	INDEX-	GND
20			
21		HOLD/PICK+	HOLD/PICK-
22		FAULT+	FAULT-
23	OPENCABLE-	OPENCABLE+	GND
24	SECTOR-	ONCYL-	ONCYL+
25	SKERR-	SECTOR+	SKERR+
26	READY-	WPROT+	WPROT-
27	READY+	BUSY-	BUSY+
28	+5Vdc	+5Vdc	+5Vdc
29	-5Vdc	-5Vdc	-5Vdc
30	+12Vdc	+12Vdc	+12Vdc
31	-12Vdc	-12Vdc	-12Vdc
32	GND	GND	GND

Table 2-15. Storage Module Device Controller B Connector P2/J27, Slot 7 (Model 31)

PIN	Row A SIGNAL	Row B SIGNAL	Row C SIGNAL
1			
2		3.SERVOCLK-	3.SERVOCLK+
3		3.READDATA-	3.READDATA+
4	3.WRITECLK-	3.READCLK-	GND
5	3.WRITECLK+	3.READCLK+	3.SELECTED-
6	3.WRITEDATA+	3.SELECTED+	3.SEEKEND-
7	3.WRITEDATA-	3.SEEKEND+	GND
8			
9			2.SERVOCLK-
10		2.SERVOCLK+	2.READDATA-
11	2.WRITECLK+	2.READDATA+	GND
12	2.WRITECLK-	2.READCLK-	2.READCLK+
13	2.WRITEDATA-	2.SELECTED-	2.SELECTED+
14	2.WRITEDATA+	2.SEEKEND-	2.SEEKEND+
15		1.SERVOCLK-	GND
16		1.SERVOCLK+	1.READDATA-
17	1.WRITECLK-	1.READDATA+	1.READCLK-
18	1.WRITECLK+	1.READCLK+	1.SELECTED-
19	1.WRITEDATA+	1.SELECTED+	GND
20	1.WRITEDATA-	1.SEEKEND-	1.SEEKEND+
21		0.SEEKEND-	0.SEEKEND+
22		0.SELECTED+	0.SELECTED-
23		0.WRITEDATA+	GND
24		0.WRITEDATA-	0.WRITECLK-
25		0.WRITECLK+	0.READCLK-
26		0.READCLK+	0.READDATA-
27	0.SERVOCLK+	0.READDATA+	0.SERVOCLK-
28	+5Vdc	+5Vdc	+5Vdc
29	-5Vdc	-5Vdc	-5Vdc
30	+12Vdc	+12Vdc	+12Vdc
31	-12Vdc	-12Vdc	-12Vdc
32	GND	GND	GND

Table 2-16. System 8000 Memory Bus, Connector P2/J28 to P2/30, Slots 8 through 10

PIN	Row A SIGNAL	Row B SIGNAL	Row C SIGNAL
1	MD38\	MD37\	
2	MD36\	MD35\	
3	MD34\	MD33\	
4	MD32\	MD31\	GND
5	MD30\	MD29\	
6	MD28\	MD27\	
7	MD26\	MD25\	GND
8	MD24\	MD23\	
9	MD22\	MD21\	
10	MD20\	MD19\	
11	MD18\	MD17\	GND
12	MD16\	MD15\	
13	MD14\	MD13\	
14	MD12\	MD11\	
15	MD10\	MD09\	GND
16	MD08\	MD07\	
17	MD06\	MD05\	
18	MD04\	MD03\	
19	MD02\	MD01\	GND
20	MD00\	MA16	RC7\
21	MA17	MA18	BD SELECT\
22	MA19	MA20	READ\
23	MA21	MA22	GND
24	MA23	RC0\	RAS\
25	RC1\	RC2\	REF\a
26	RC3\	RC4\	CAS\
27	RC5\	RC6\	WRITE\
28	+5v	+5v	+5v
29	-5v	-5v	-5v
30	+12v	+12v	+12v
31	-12v	-12v	-12v
32	GND	GND	GND

SECTION 3
INSTALLATION

3.1. Introduction

This section contains detailed information on the following:

- Receiving, Unpacking, and Inspecting the System 8000
- Installation Procedures
- Winchester Disk Drive and SMD Configuration and Checkout Procedures
- Parallel Line Printer Installation Procedures
- System Power-Up Diagnostics (SPUD)
- System Expansion

3.2. Unpacking, Inspection, and Reshipment Procedures

The following paragraphs explain how the System 8000 is shipped to the customer and how to properly unpack and inspect the system. Also included is the method for repacking a system that is being returned to the factory.

3.2.1. Shipping Container: The System 8000 is shipped in the fully loaded configuration; all Printed Wiring Assemblies (PWAs) are installed. Before shipment: the side and front panels are installed on each module. The system is protected by foam inserts. Except for the casters, the complete system is covered by a cardboard box. The box is secured to a pallet by straps passing under the pallet and over the box in two directions.

In addition to the system shipping container, another container will be shipped containing the system documentation, operating system and diagnostic tapes.

3.2.2. Unpacking and Inspection Procedures: The System 8000 is shipped in a reusable container. Save all packing materials, such as boxes, foam inserts and supports, in case the system is reshipped.

The following unpacking and inspection procedure should be performed:

(1) Inspect the shipping container for damage. Request the carrier's agent to remain until the inspection and inventory have been completed.

(2) Remove the cardboard box from the system by lifting it straight up.

(3) Remove the foam inserts and spacers.

(4) Inspect the system cabinet and panels for damage (e.g. marks and dents).

(5) Before accepting delivery of a System 8000, note on the waybill any obvious external damage found upon inspection, and request the delivery agent to sign the waybill. Then immediately notify the transfer agent and submit a damage report to the company and to Zilog Inc.

3.2.3. Internal Inspection: Unusual shipping and handling damage that is not obvious from external visual inspections, may have occurred. The following internal inspection procedures should be performed on the Processor and Disk/Tape Modules:

PROCESSOR MODULE

(1) Remove the front panel and foam insert.

(2) Loosen the fasteners that secure the top cover; slide it back from the guideposts, and remove from system.

(3) Using Figure 2-4 (Card Cage Backplane, Slot Assignments), check each printed wiring assembly by slot, and type of board installed.

(4) Examine each board to ensure it is properly seated in its appropriate backplane connector.

(5) Remove the sheet metal covers for the card cage and power supply.

(6) Examine all cable harnesses for signs of stress (broken terminals, loose or broken wires, broken cable straps).

(7) Examine the Z-Bus Backplane Interconnect (ZBI), on the rear of the card cage, for signs of stress (cracks, broken or lifted traces, damaged connectors).

(8) Check the I/O connector panel at the rear of the module for the same stresses described in step 7.

(9) Verify the AC line voltage set on the power supply with the AC voltage specified on the module ID plate. If there is a difference, notify Zilog Field Service.

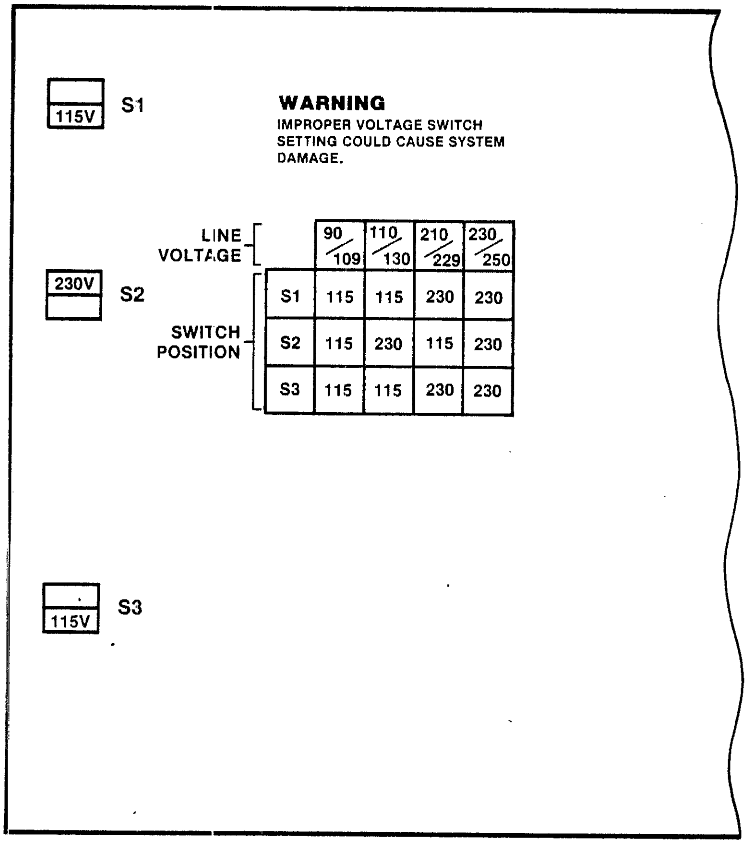
NOTE

AC line voltages are switch selectable. A SWITCH POSITION/LINE VOLTAGE selection matrix is silkscreened on the power supply cover; shown are the available line voltages. Refer to Figure 3-1.

WARNING

The line voltage setting must only be performed by Zilog Field Service personnel. Improper switch settings could damage the system.

This illustration shows switches S1, S2 and S3 set for a line voltage of 110/130.



WARNING
IMPROPER VOLTAGE SWITCH SETTING COULD CAUSE SYSTEM DAMAGE.

LINE VOLTAGE		90/109	110/130	210/229	230/250
SWITCH POSITION	S1	115	115	230	230
	S2	115	230	115	230
	S3	115	115	230	230

00404

Figure 3-1 AC Line Voltages, Processor Module

After the internal inspection of the Processor Module is complete, proceed to the Disk/Tape Module.

DISK/TAPE MODULE

The Processor Module must be removed from the top of the Disk/Tape Module to access the Disk/Tape module components.

(1) Remove the front panel from the Processor Module.

(2) Remove the Processor Module as follows:

a. Unfasten two (2) intermodule captive fasteners on the rear of the module.

b. Slide the Processor Module back and disengage from the guide posts on the Disk/Tape Module.

c. Remove the Processor Module and set aside.

(3) Examine all Disk/Tape Module cable harnesses for signs of stress (broken terminals, loose or broken wires, broken cable straps).

(4) Make sure all harness connectors are securely fastened to their mating connectors.

(5) Model 21: Check that the rotary-arm shipping lock on the FINCH Disk Drive (located atop the environmental enclosure) is in the SHIP position (heads locked). Model 31: For SMD drives, check that the locking lever at the bottom front of the drive is in the SHIPPING (locked) position.

(6) Verify the AC line voltage set on the power supply with the AC voltage specified on the module ID plate. If there is a difference notify Zilog Field Service.

(7) Reassemble and interconnect the system by reversing steps 2 and 1.

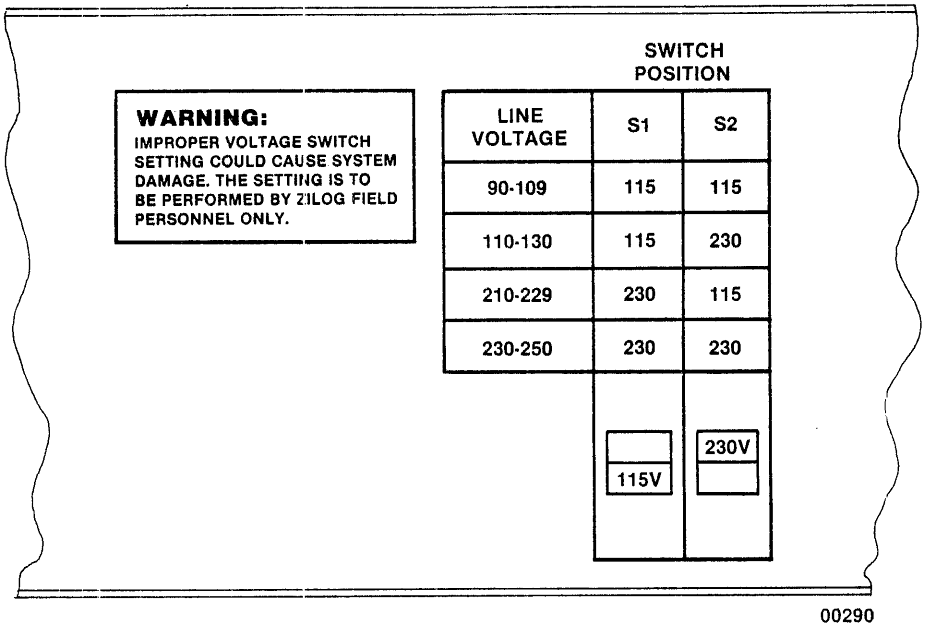
NOTE

AC line voltages are switch selectable. A SWITCH POSITION/LINE VOLTAGE selection matrix is silkscreened on the power supply cover; shown are the available line voltages. Refer to Figure 3-2.

WARNING

The line voltage setting must only be performed by Zilog Field Service personnel. Improper switch settings could damage the system.

This illustration shows switches S1 and S2 set for a line voltage of 110/130.



WARNING:
IMPROPER VOLTAGE SWITCH SETTING COULD CAUSE SYSTEM DAMAGE. THE SETTING IS TO BE PERFORMED BY ZILOG FIELD PERSONNEL ONLY.

	SWITCH POSITION	
LINE VOLTAGE	S1	S2
90-109	115	115
110-130	115	230
210-229	230	115
230-250	230	230

Figure 3-2 AC Line Voltages Disk/Tape Module

NOTE

If the system is configured with a Disk Module, repeat the Disk/Tape Module internal inspection procedure.

3.2.4. Reshipment Procedures: The following procedures should be followed if the System 8000 must be reshipped:

(1) Unplug the AC input cable from the facility power outlet and on the rear of the Processor Module.

(2) Unplug and remove the AC intermodule jumper cables.

(3) Unfasten and open the cable ducts on the rear of the system.

(4) Unplug and remove the terminal distribution, disk and tape intermodule jumper cables.

(5) Place the cables removed in steps 1 through 4 in the Accessory Module.

(6) Place the operating system tape and the SADIE diagnostic tape in the Accessory Module.

(7) Before reshipping the Model 21 system, the Winchester Disk Drive(s) read/write heads must be locked.

a. Model 21: Using the flat end of the lock/unlock tool, or a screwdriver, lift and disengage the rotary-arm shipping lock. Push the rotary-arm shipping lock toward the rear of the drive until it locks into the SHIP position. Refer to paragraph 3.4.3.

b. Model 31: The SMD drives lock the read/write heads by moving the locking lever to the SHIP position but the lever is located at the bottom front of the drives.

(8) Place the panel key in an envelope and tape the envelope securely to the top of the system.

(9) Tape or strap foam spacers into position to prevent the cardboard shipping container from making contact with the system.

(10) Slip the cardboard shipping container over the system.

3.3. Installation Procedures

The following paragraphs cover moving the system to the site, system interconnection, disk drive configuration checks, power-up procedures, and System Power-Up Diagnostics (SPUD).

3.3.1. Site Preparation: After completing the unpacking and inspection of the System 8000 enclosure and peripherals for possible damage, the equipment must be moved to the site.

Ensure that the site has proper ventilation, good lighting, and that the required voltage is present at the receptacle that will power the system. When positioning the system, make sure that it can be easily rolled away from walls or other equipment for maintenance purposes.

3.3.2. System Interconnection: A typical system as shipped from the factory consists of four stacked modules and the four castor base of the enclosure. CRT terminals and high-speed printers interface to the system by eight RS232C serial ports and a parallel printer port on each of the rear terminal distribution panels. They are normally located on the rear of the Disk/Tape Module. Identification of the I/O Panel and terminal distribution panel connectors is shown in Figures 3-3 and 3-4 for Models 21 and 31.

The stackable module packaging concept allows additional peripherals, Disk or DisK/Tape Modules, to be integrated into the system as dictated by application requirements.

Interconnect Procedure

(1) Model 21: Install the tape and disk interface cables between the CARTRIDGE TAPE OUT and DISK OUT connectors on the Processor I/O panel and CARTRIIDGE TAPE IN and DISK IN I/O panel connectors of the Disk/Tape module. For a second Disk/Tape module, I/O panel outputs from the top Disk/Tape module connect to the disk and tape inputs on the lower module. Refer to Figure 3-5 for Model 21 cable orientation.

Model 31: Install the tape and SMD interface cables between CARTRIDGE TAPE OUT, SMDC A OUT, and SMDC B OUT on the Processor I/O panels and CARTRIDGE TAPE IN, SMDC A IN, and SMDC B IN on the Disk/Tape Module. For a second Disk Module, an OUT connector and separate DRIVE 1,2,and 3 connectors on the SMDC I/O panel are connected to the IN and appropriate DRIVE connections on the second Disk Module SMDC I/O panel. Refer to Figure 3-6 for Model 31 cable orientation.

(2) Install the terminal interface cable between the Processor Module TERMINAL OUT and TERMINAL IN on the first terminal distribution panel.

(3) Install short AC power cables between modules from top to bottom module in daisy-chain configuration. The main power cord plugs into the Processor Module.

(4) Position the peripheral components (terminals and printers) for easy access to the terminal distribution panel I/O ports and to the facility AC power.

(5) The terminal used for system bootstrap should be connected to the terminal I/O port labeled CONSOLE.

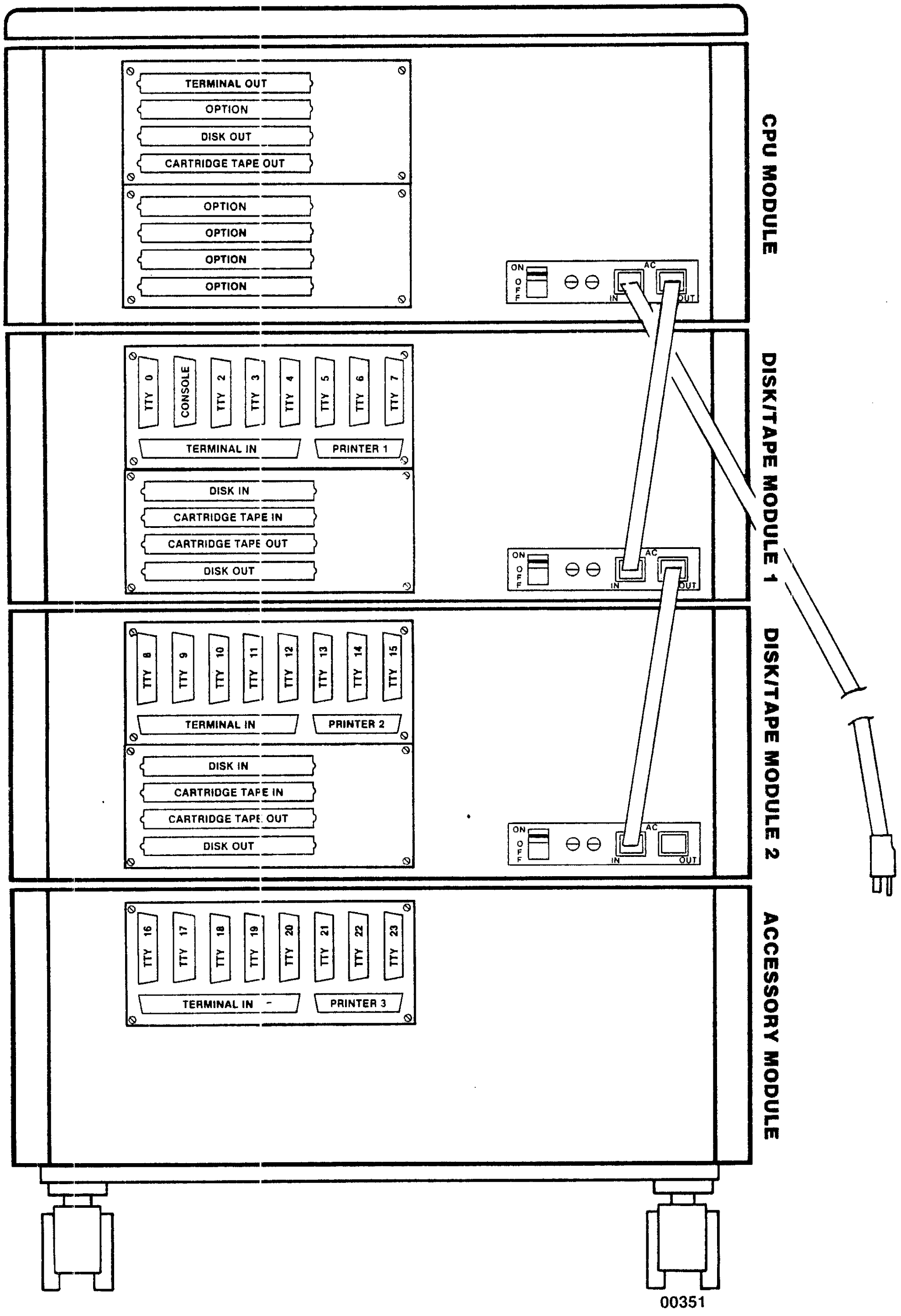
(6) Connect the user terminals, as required, to the terminal I/O ports labeled TTY0, or TTY2 through TTY7.

(7) If the system is configured for the 16 user option, cable the Processor Module I/O panel OPTION connector to the second terminal distribution panel TERMINAL IN connector. Connect the next eight (8) terminals to I/O ports TTY8 through TTY15, on the second terminal distribution panel.

(8) Connect the line printer interface cable to the first terminal distribution panel parallel port labeled PRINTER 1.

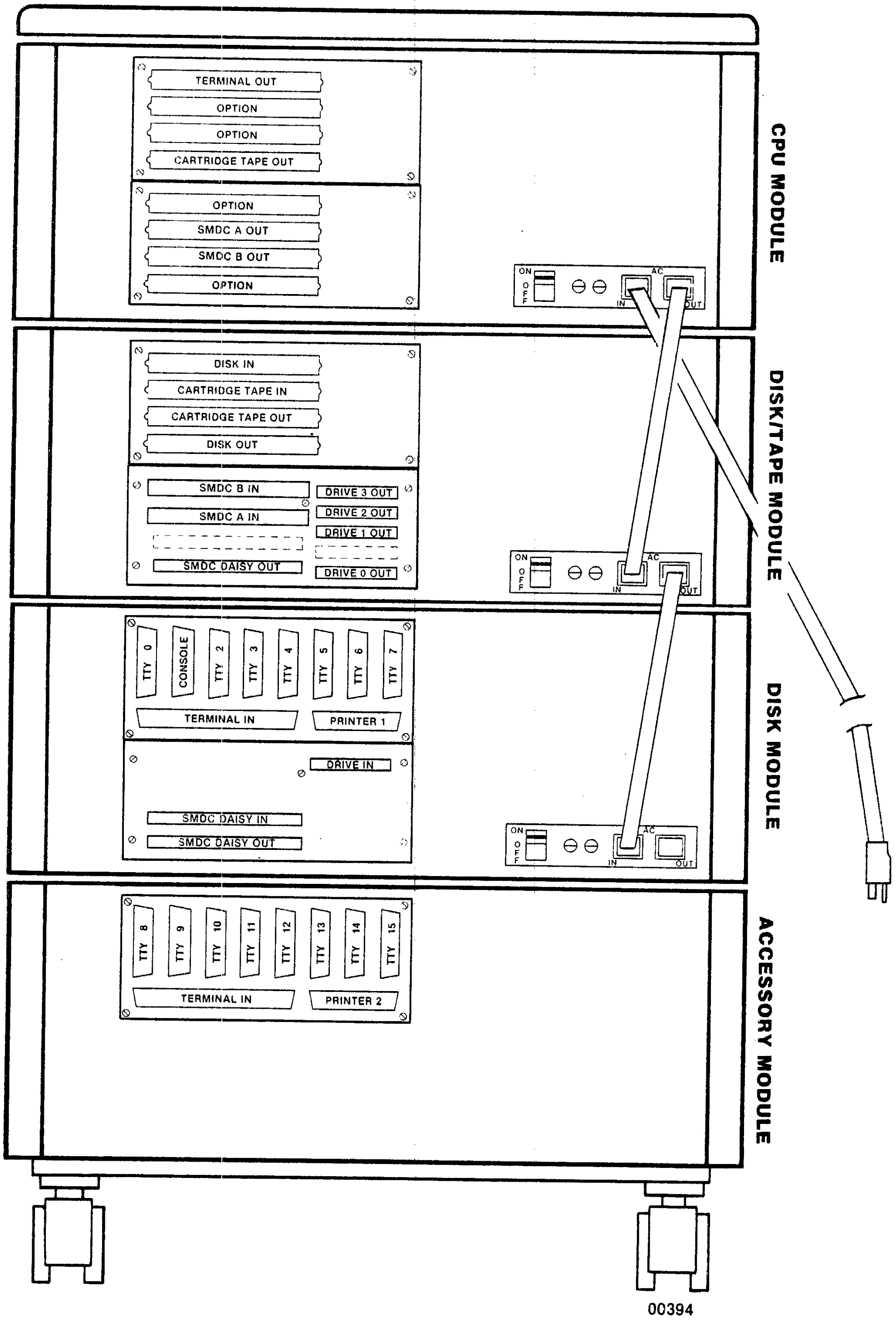
(9) If an additional terminal distribution panel is supplied with the system, connect the printer interface cable to the I/O ports labelled PRINTER 2.

(10) Connect the system, all terminals, and the line printer(s) to facility AC power.



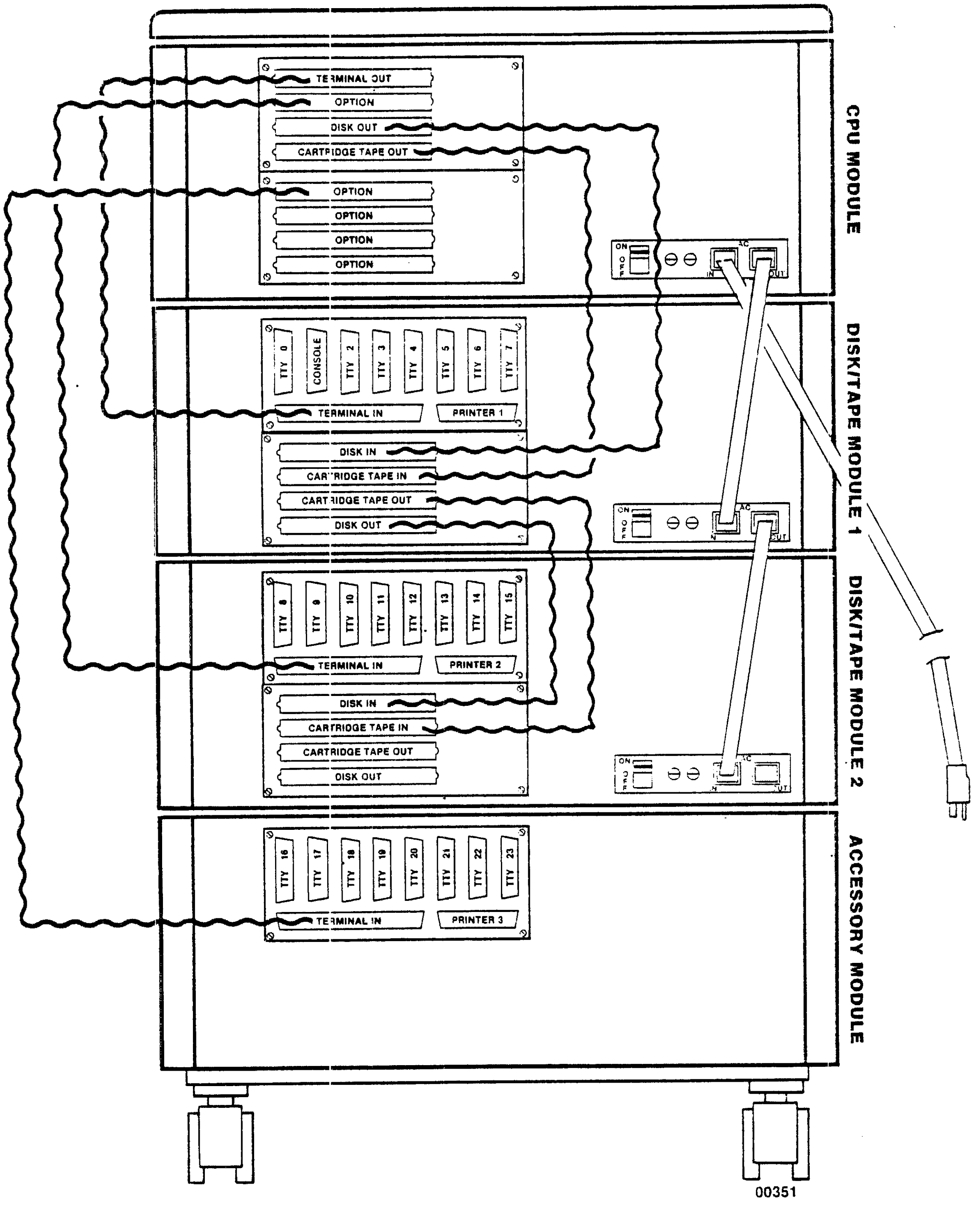
MODEL 21

Figure 3-3 Model 21 Connector Identification -- Sample Configuration



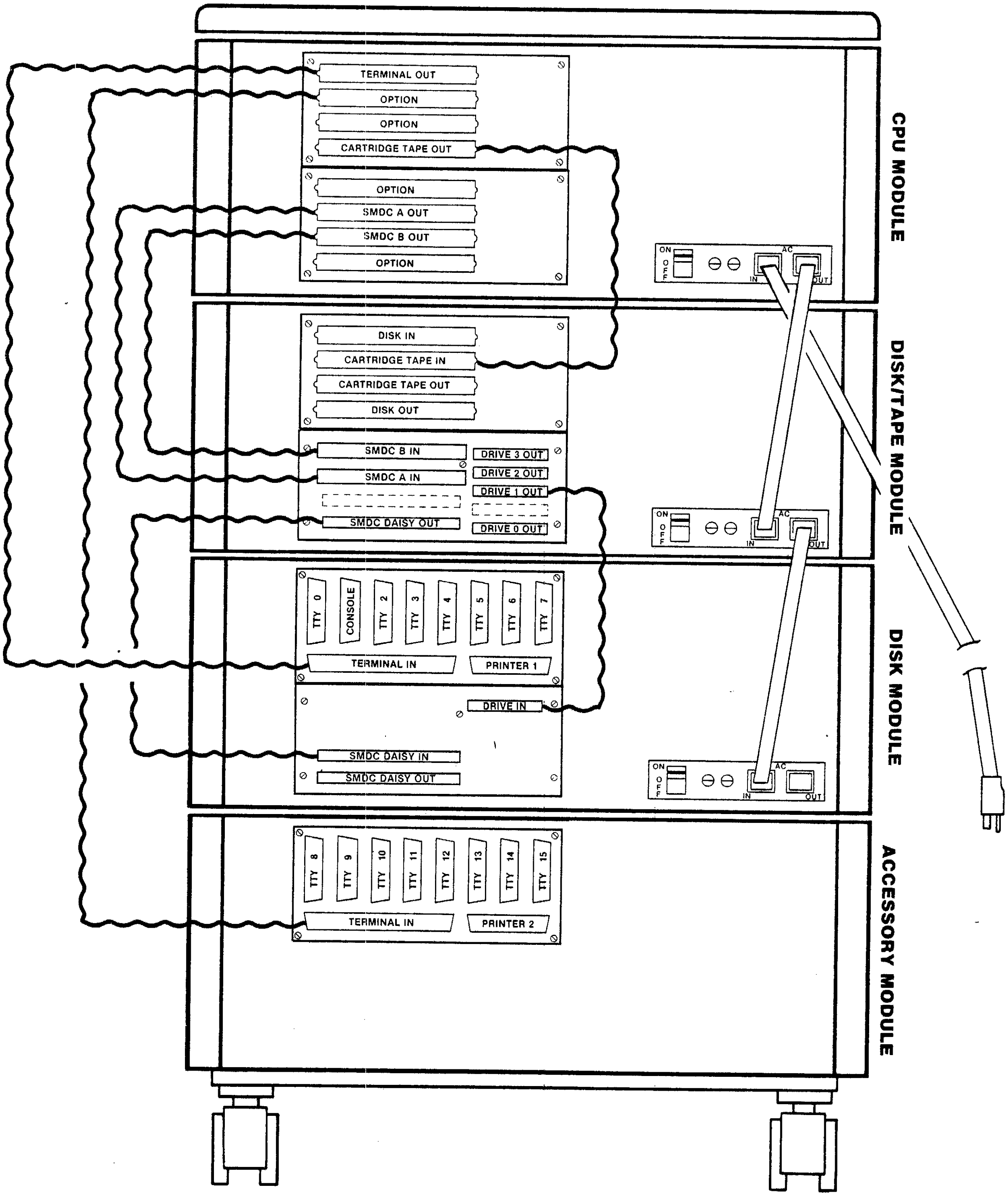
MODEL 31

Figure 3-4 Model 31 Connector Identification -- Sample Configuration



MODEL 21

Figure 3-5 Model 21 System Intermodule Cabling
-- Sample Configuration



MODEL 31

Figure 3-6 Model 31 System Intermodule Cabling -- Sample Configuration

3.4. 32 MB Winchester Disk Drive Configurations

3.4.1. Drive Cabling: The required cable connections to the drive are power and signal cables.· All input/output signals and power exit at the FINCH Adapter Board. The adapter board attaches to the drive at the command, DC power, and data connectors. Refer to Figure 3-7 for power and signal connector orientation.

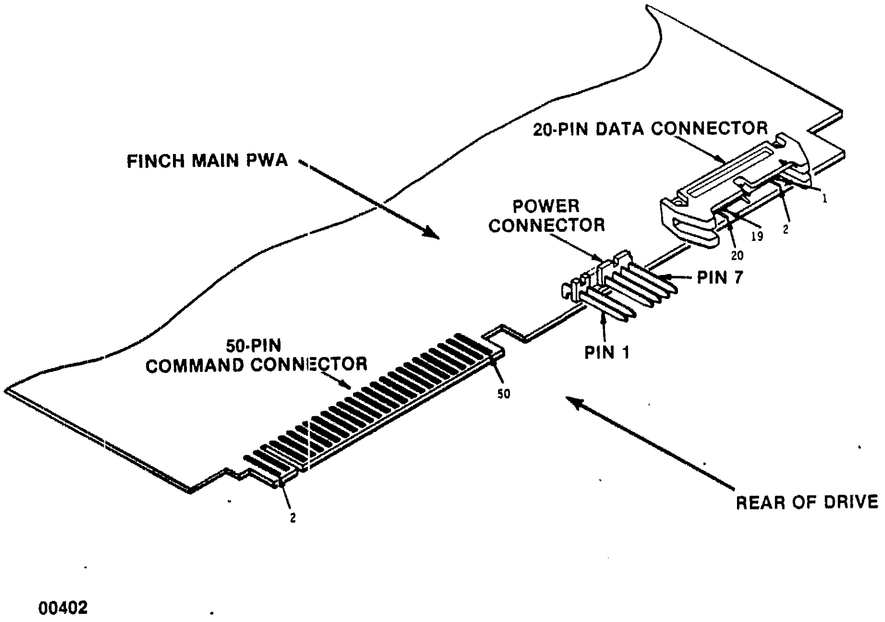


Figure 3-7 FINCH Power and Signal Connector Orientation

Terminator resistor packs are included on each FINCH Adapter Board. The terminators consist of a DIP resistor module which is plugged into a DIP socket. Only the last FINCH in a daisy-chain configuration requires a terminator resistor pack for the Command/Data cable; the others must be removed. A daisy-chain configuration incorporates parallel interfacing of the disk drives on a common Command/Data cable. A maximum of four drives may be daisy-chained on the cable.

3.4.2. Drive Configuration Procedures: Using the following procedure, verify the configuration of the FINCH drive:

(1) Disconnect the 40-pin Command/Data cable connector, and the power connector, at the top of the FINCH Adapter Board.

(2) Using a small Phillips head screwdriver, remove two (2) 4-40 x 1/4 -inch screws securing the Adapter Board to the mounting brackets.

(3) Unseat the Adapter Board, and remove.

(4) Verify that a terminator resistor pack is installed on the Adapter Board, as shown in Figure 3-8.

(5) On the Unit Selection dip switch, verify that switch position 1 is ON (refer to Figure 3-8). The other switch positions must be OFF. The ZEUS software and SADIE diagnostic firmware will recognize the FINCH as Drive 0.

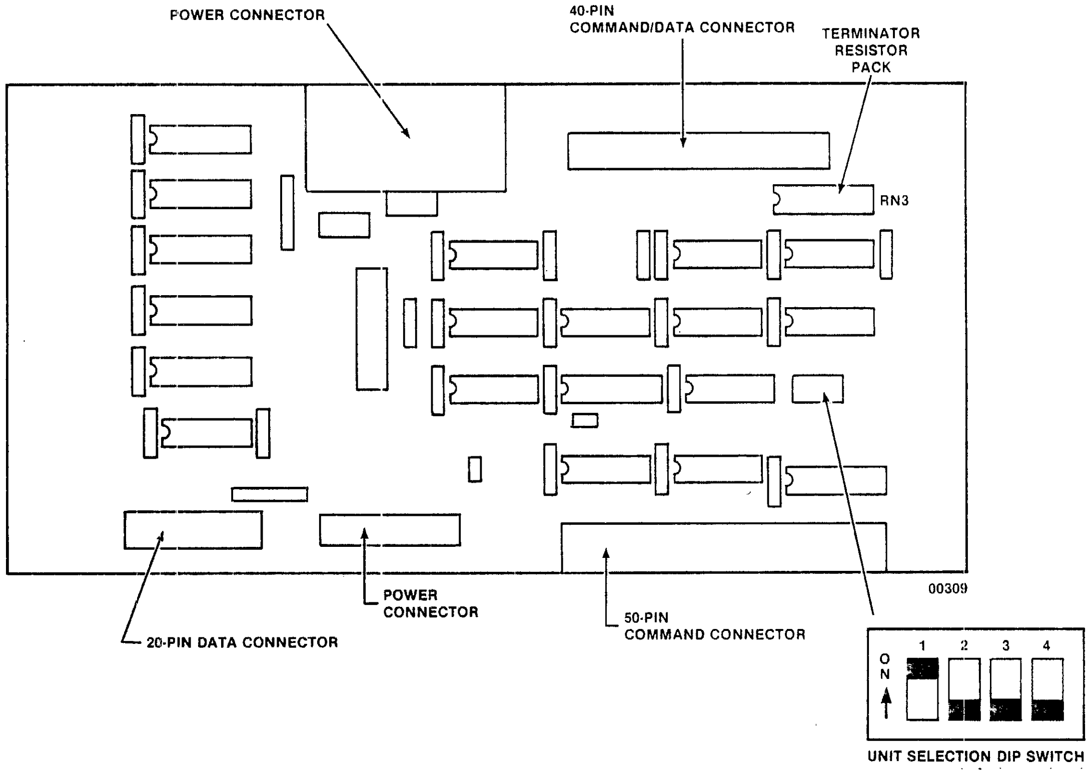


Figure 3-8 FINCH Adapter Board

(6) On the FINCH Main PWA, verify that a keyed Unit Select jumper plug is installed in position 0 on header J3. The plug can be installed in one of four possible orientations and will display the unit selected number to the rear of the drive.

(7) Reattach the FINCH Adapter Board to the FINCH drive, by reversing the steps of this procedure.

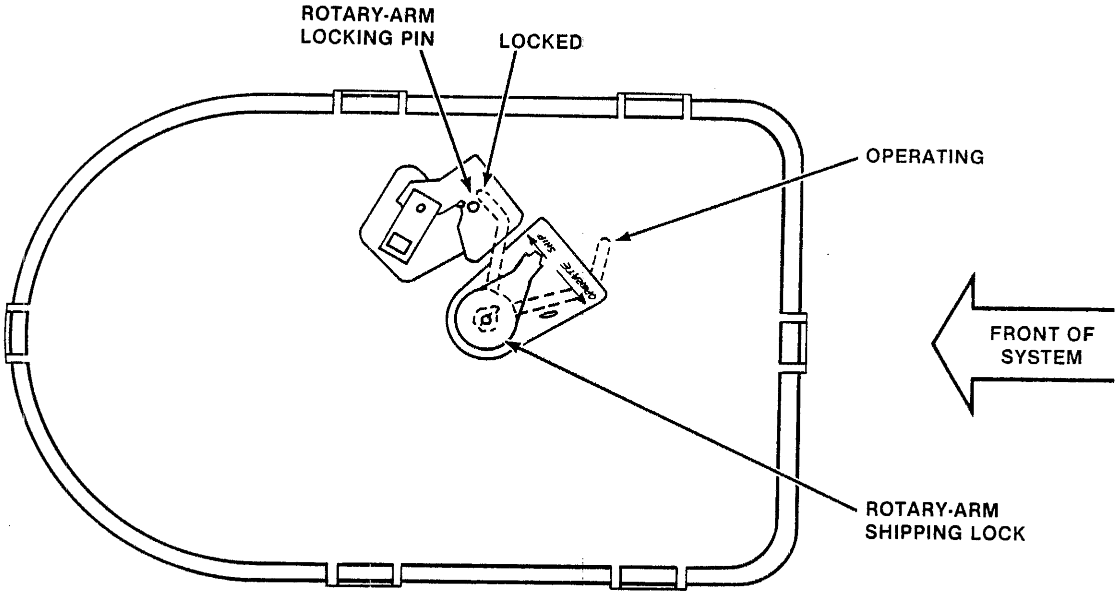
3.4.3. **Initial Checkout and Startup Procedure:** This procedure assumes that drive cabling and configuration procedures of this section have been performed and satisfied.

(1) Remove the front panel from the Disk/Tape Module.

(2) Place the rotary-arm shipping lock to the OPERATE position. Refer to Figure 3-9 for drive-to-system orientation.

CAUTION

Figure 3-9 shows the rotary-arm shipping lock in the locked and operating positions. To prevent damage to the read/write heads or the disk itself, place the rotary-arm shipping lock in the OPERATE position only after installation has been completed.



00405

Figure 3-9 Rotary-Arm Shipping Lock Orientation

(1) To place the rotary-arm shipping lock in either the OPERATE or SHIP position, the use of a head lock/unlock tool for FINCH Disk Drives is required (part number 31-0092-00).

(2) To unlock the read/write heads (OPERATE position):

a. Turn off AC power.

b. Using the crooked end of the head lock/unlock tool, hook and disengage the rotary-arm shipping lock. Pull towards the front of the drive until it locks into the OPERATE position.

(3) To lock the read/write heads (SHIP position):

a. Turn off AC power.

b. Using the flat end of the lock/unlock tool, lift and disengage the rotary-arm shipping lock. Push towards the rear of the drive until it locks into the SHIP position.

(4) After the rotary-arm shipping lock is placed in the OPERATE position, turn on AC power.

(5) Run data non-destructive SADIE diagnostics to ensure the operability of the disk drive.

3.5. SMD Winchester Drive Configurations

When the 84 MB SMD is installed in the system, the Mode Select Switch (SW1) is set according to system requirements. This switch is located on the CNAM PCB assembly (Figure 3-10).

3.5.1. SMD Disk Addressing: The Disk Logical Unit Number 0 to 7 is selected by SW1 to the desired disk address by the three key switches of SW1 (1, 2,and 3) by using the binary code shown in Table 3-1.

* **NOTE:** Dip switches SW2 and SW3 are set at the factory and should not be altered by customer.

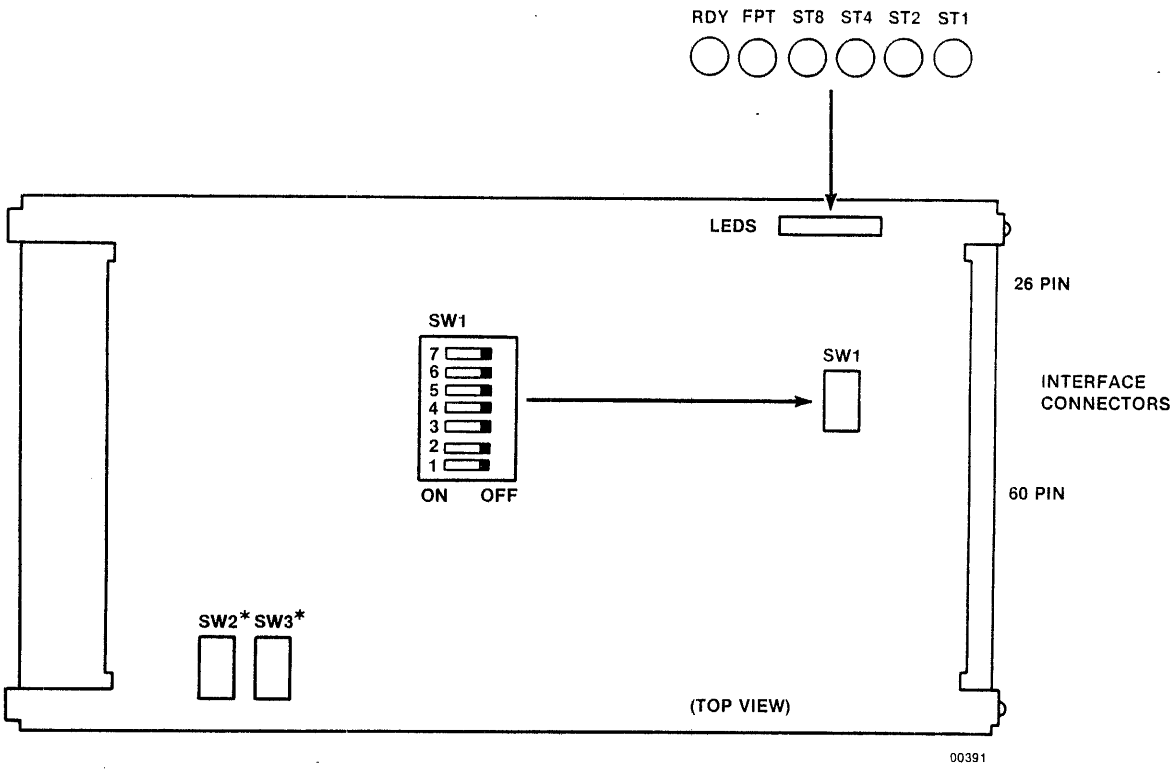


Figure 3-10 Mode Select Switch Location

Table 3-1. SMD Disk Addressing

Disk Unit	Key 1	Key 2	Key 3
0	OFF	OFF	OFF
1	ON	OFF	OFF
2	OFF	ON	OFF
3	ON	ON	OFF
4	OFF	OFF	ON
5	ON	OFF	ON
6	OFF	ON	ON
7	ON	ON	ON

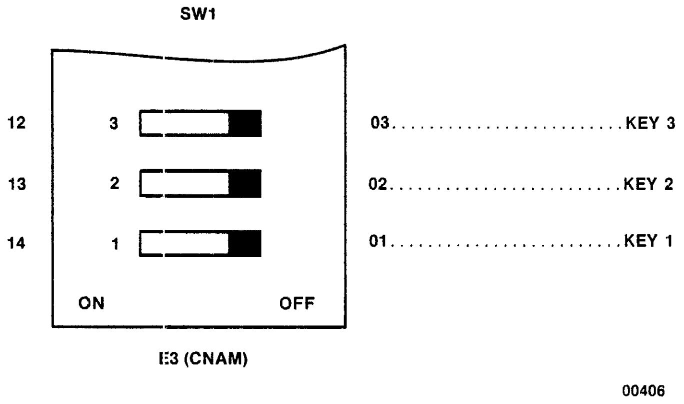
The SMD controller has only one 16 bit I/O port, the address is set by a dip switch located on SMDC A near the P1 (ZBI) connector. The functions of each key are described below:

KEY	FUNCTION	POSITION FOR "7F00"
1	Reserved	ON {ON = 0}
2	Address 01	ON {OFF = 1}
3	Address 02	ON
4	Address 03	ON
5	Address 04	ON
6	Address 05	ON
7	Address 06	ON
8	Address 07	ON

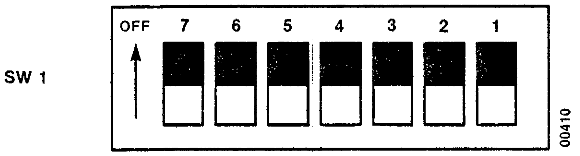
The I/O port address is "7FXY", where XY is specified by the dip switch keys 2 through 8. The I/O address must be even. The first key is reserved and should be in the "ON" position.

A jumper labelled "Pick" (located at SMDC A) may be optionally grounded (with shunt installed) to prevent the drive from cycling down when controller power is lost. (This feature called Pick and Hold only works with drives that have this feature built in). In the case of the Zilog 8" SMD disk drive, leave pick jumper off (shunt not installed).

At the power up sequence, if pick is grounded (with shunt installed) it will enable the first SMD drive to get up to speed, then the pick signal is transferred to the next active SMD drive. The drives spin up sequentially, one by one.


SW1
12
13
14
3
2
1
ON
OFF
03.........................KEY 3
02.........................KEY 2
01.........................KEY 1
E3 (CNAM)
00406

SMD Board Switch



3.5.2. SMD Disk Status Indicators: The drive has six LEDs, four for status (ST1, ST2, ST4, ST8), one for ready (RDY), and one for file protect (FPT). Drive fault indication is given by the four status LEDs (ST1, ST2, ST4, ST8) on the SMD PCB. The indicator LEDs are defined as follows:

FPT (File Protect) switch: SW1-Key 7

This switch inhibits the write operation and should be in the OFF position.

RDY (Ready) indicator: Green

This RDY LED indicates that the initial seek has been performed or indicates the termination of a Seek or RTZ operation.

FPT (File Protect) indicator: Red

This LED indicates that writing is inhibited and is controlled by SW1-Key 7 on the SMD disk.

ST1 to ST8 (Status 1 to 8) LEDs: Red

Fifteen fault statuses are visible by binary code as shown in Table 3-2.

Table 3-2. Fault Indicator

Status Bit					Fault Status	
ST8	ST4	ST2	ST1	Code (Hex)	Fault	Description
0	0	0	1	1	DC motor failure (DMFL)	indicates spindle motor failure.
0	0	1	0	2	VCM over heat (VCMHT)	indicates VCM over-heating.
0	0	1	1	3	Initial seek time out (INTMOT)	indicates initial seek has terminated with time-out.
0	1	0	0	4	Control check 1 (CTCK1)	indicates that a Read/Write command was issued during busy status.
0	1	0	1	5	Control check 2 (CTCK2)	indicates that write gate was issued during a fault condition.
0	1	1	0	6	Read/write check 1 (RWCK1)	indicates that write gate was issued during off-track.
0	1	1	1	7	Read/write check 2 (RWCK2)	indicates that write current did not flow to the head during a Write operation.
1	0	0	0	8	Read/write check 3 (RWCK3)	indicates that write gate was issued during File-Protected status.
1	0	0	1	9	Read/write check 4 (RWCK4)	indicates that write gate was issued during a multihead-selected status.
1	0	1	0	A	Time-out (TMOT)	indicates that seek or RTZ sequence was not terminated within 500 ms.

Status Bit					Fault Status	
ST8	ST4	ST2	ST1	Code (Hex)	Fault	Description
1	0	1	1	B	Seek guard band (SEKGB)	indicates that a guard band was detected during a direct seek operation.
1	1	0	0	C	Linear mode guard band (LNMGB)	indicates that a guard band was detected during a linear mode.
1	1	0	1	D	RTZ outer guard band (RTOGB)	indicates that an outer guard band was detected during an RTZ operation.
1	1	1	0	E	Over-shoot check (OVSHT)	indicates that the head overshot the new cylinder address during settling time.
1	1.	1	1	F	Illegal cylinder check (ILCYL)	indicates that an illegal cylinder address (>588) was issued by the controller.

3.5.3. SMD Configuration: Figure 3-11 illustrates the fundamental configuration of the SMD with a cutout of the sealed disk unit. The 84 MB SMD has four platters and seven read/write heads.

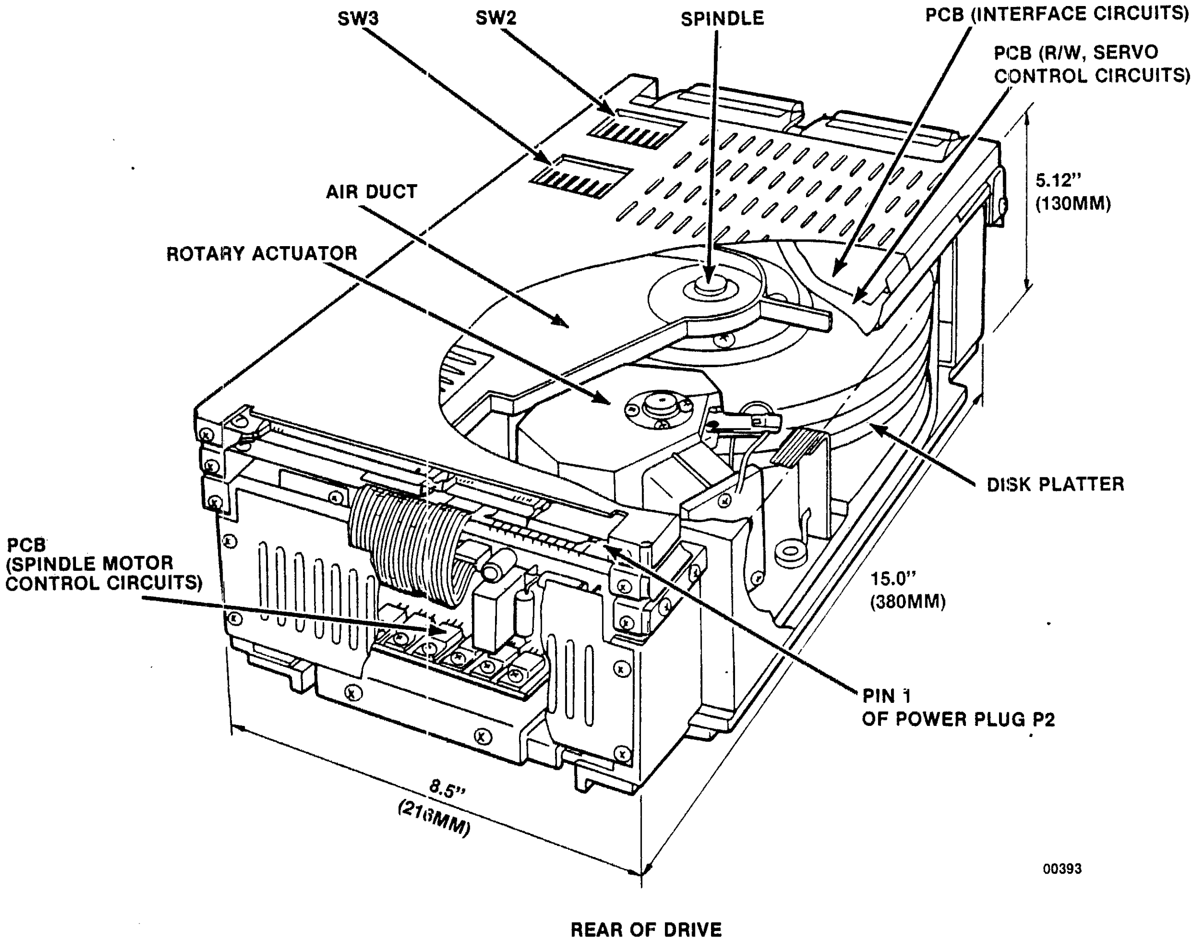


Figure 3-11 84 MB SMD Configuration

3.5.4. SMD Cabling: Figures 3-12 and 3-13 illustrate the Interface cabling (Interface cable A and B) to the 60 pin and 26 pin connectors of the Interface PCB on top and to the rear of the SMD. The POWER cable connector is also shown.

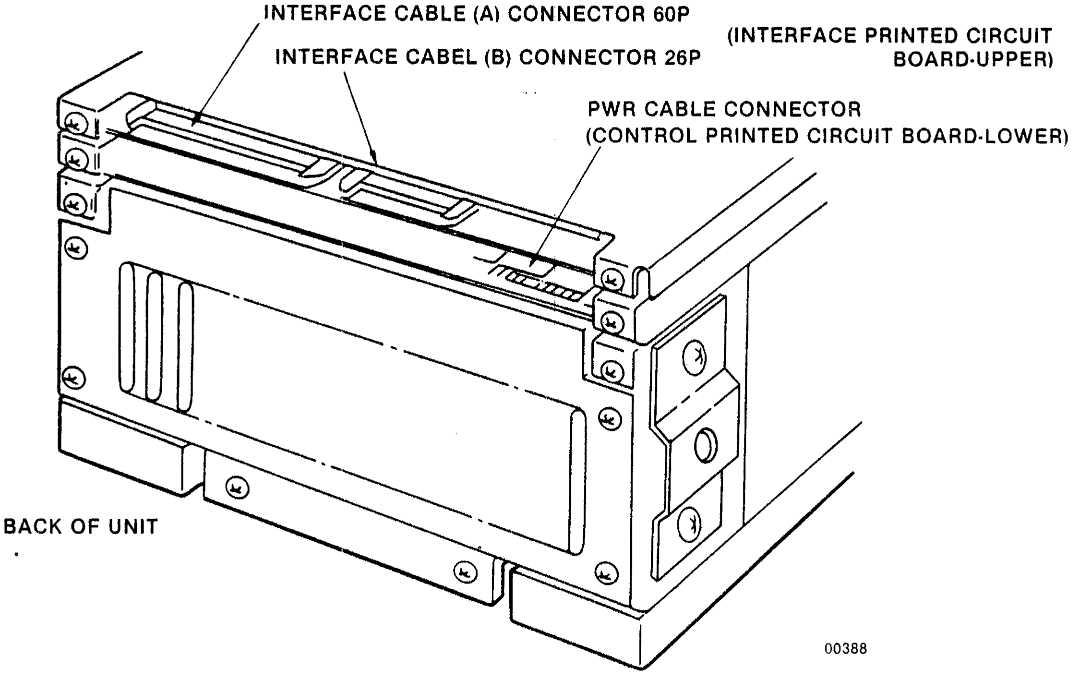

INTERFACE CABLE (A) CONNECTOR 60P
INTERFACE CABEL (B) CONNECTOR 26P
(INTERFACE PRINTED CIRCUIT BOARD-UPPER)
PWR CABLE CONNECTOR
(CONTROL PRINTED CIRCUIT BOARD-LOWER)
BACK OF UNIT
00388

Figure 3-12 Mounting Positions of SMD Connectors

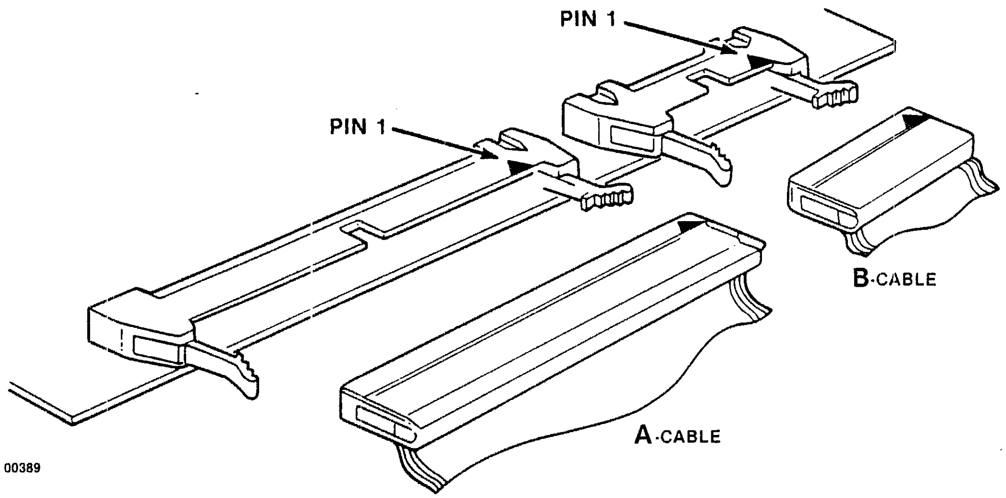


Figure 3-13 SMD Interface Cabling

3.5.5. Head Lock Actuator: The SMD mounting bracket has a head locking and unlocking actuator lever located at the front of the SMD (Figure 3-14).

1. To unlock the read/write heads (OPERATE position):

   a. Turn off AC power.

   b. Disengage from shipping lock and position actuator lever and lock in the OPERATE position.

   c. Turn on AC power.

   d. Run SADIE Diagnostics called SMDCRC (Appendix A) to ensure operability of the disk drive.

CAUTION

Certain SADIE diagnostics are DATA-DESTRUCTIVE and could result in overwriting the disk media.

2. To lock the read/write heads (SHIP position):

   a. Turn off AC power.

   b. Position actuator lever until it locks heads in SHIP position.

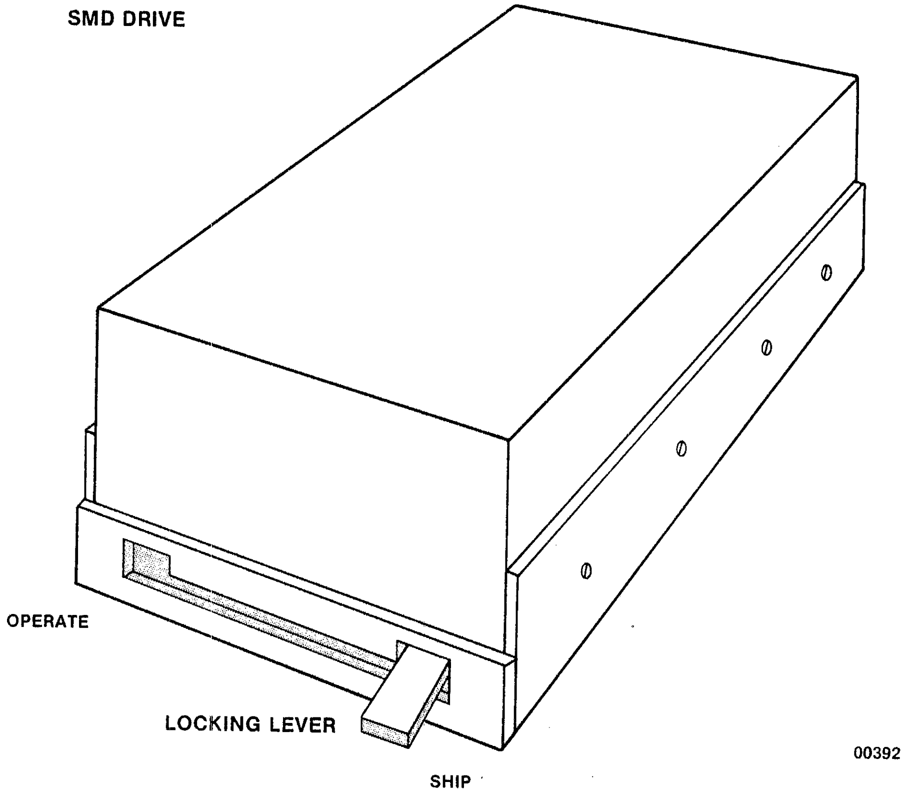


Figure 3-14 SMD Mounting Bracket with Head Locking Actuators

3.5.6. Cable Termination: When the SMD disk drives are added in a daisy-chain configuration, the A-cable signals are terminated by four resistor networks (16 pin DIP) on the interface board of the last disk drive (Figure 3-15). These resistor DIPs are removed from all disk drives except for the last in the system configuration.

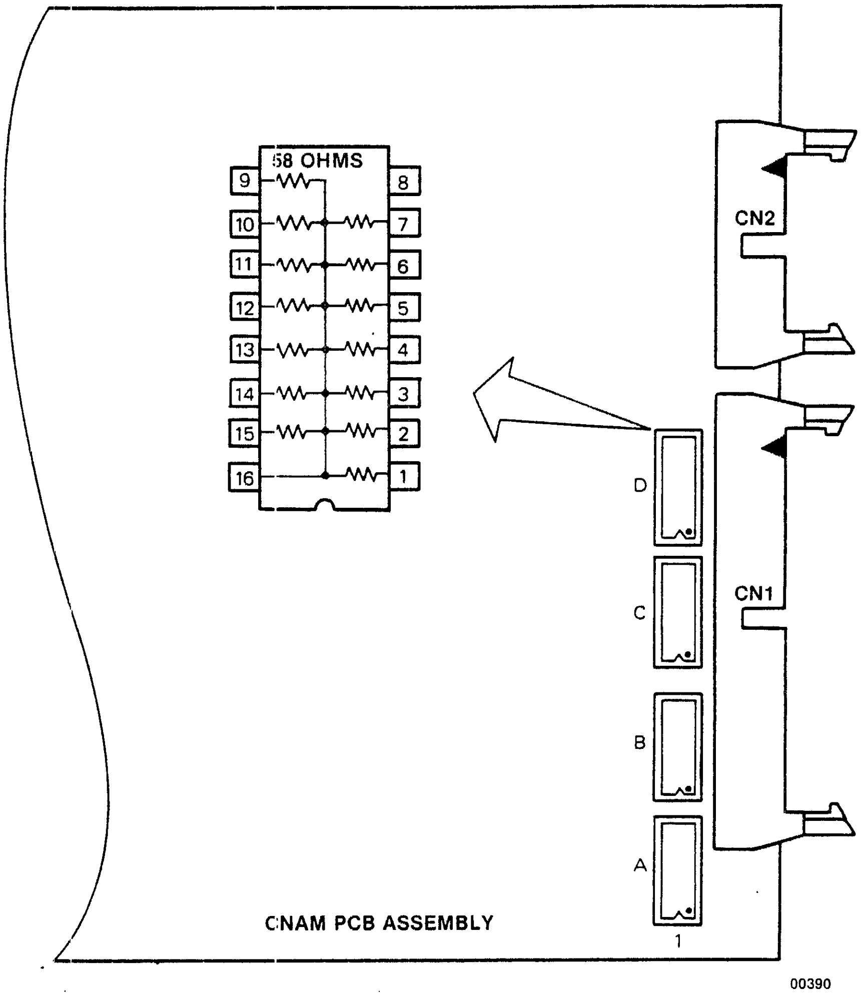


Figure 3-15 SMD Cable Terminators

3.6. Line Printer Installation Procedures

This procedure provides information for the hardware interface between the printer and the host system. The ZEUS Operating System can support two (2) printers with either a Centronics or Data Products interface. The line printer driver has been tested with the following printers:

Zilog PRZ 2/10 -- Centronics interface

Zilog PRZ 3/30 -- Centronics interface

Zilog PRZ 3/60 -- Centronics interface

The installation procedure is as follows:

(1) Disengage the thumbscrews, and open the cable covers on all modules.

(2) Attach the line printer interface cable to the connector labeled PRINTER 1, on the terminal distribution panel. Refer to Figures 3-3 and 3-4 for connector identification.

If a second line printer is used with the system (requiring an SSB option), attach that interface cable to the connector labeled PRINTER 2, on the second terminal distribution panel (refer to Figures 3-3 and 3-4).

(3) Dress the line printer interface cable(s) flush to the system, and close the cable covers on all modules.

(4) Remove the Processor Module front panel.

(5) Unseat and remove the CPU Board from slot 1 of the card cage.

NOTE

The printer port on the first terminal distribution panel is controlled by jumpers on the CPU Board.

(1) Verify that printer configuration jumpers are installed on the CPU Board as follows:

PRINTER INTERFACE	JUMPER GROUP
Centronics	E13 to E14 E17 to E18
Data Products	E14 to E15 E16 to E17

(1) Reseat the CPU Board into slot 1.

(2) If the system is configured for two (2) line printers, unseat and remove the Secondary Serial Board (SSB) from slot 2 of the card cage.

NOTE

The printer port on the second terminal distribution panel is controlled by jumpers on the first of two possible optional SSB boards.

(1) Verify that printer configuration jumpers are installed on the SSB as follows:

PRINTER INTERFACE	JUMPER GROUP
Centronics	E2 to E3 E4 to E5
Data Products	E1 to E2 E5 to E6

(1) Reseat the SSB into slot 2.

(2) For line printer software consideration refer to paragraph 7.1, Line Printer Information, in the ZEUS System Administrator Manual (03-3246).

3.7. System Power-Up Diagnostics (SPUD)

The System Power-Up Diagnostics (SPUD) reside in a Read-Only-Memory (ROM) on the CPU board. These diagnostics automatically execute in response to pressing RESET and START during system turn-on. They can also be initiated from the CPU Monitor by pressing RESET and entering T <CARRIAGE RETURN>.

SPUD tests the primary functions of the CPU and peripheral components. The diagnostics verify the system's ability to execute a limited number of instructions, and to communicate with Winchester Disk and Tape Cartridge Controllers. The specific functions of SPUD are:

(1) System 8000 Instruction Test - Specified system instructions are tested.

(2) MMU Test - All accessible internal registers and the segment trap functions are tested.

(3) Memory Test - All memory locations are tested for read and write functions.

(4) ECC Test - Tests the system's error detection capabilities.

(5) Peripheral Equipment Test - Does a cursory check of the Winchester Disk and Tape Cartridge Controllers.

If SPUD detects a problem with the system, an error message is displayed on the console of the system administrator (ZEUS super-user). Table 3-3 lists the error messages and descriptions.

A possible solution to a power-up error condition is to check that all cable connectors are properly mated, and that all system boards are seated in their backplane connectors. If problems still exist, run the SADIE diagnostics described in Section 5, Maintenance.

Table 3-3 SPUD Diagnostics Error List

ERROR #	P1	P2	P3	P4	CHRS * PRINTED	DESCRIPTION
0000	—	—	—	—	P	No External Memory**
0001	SEG #	ADDR	RD	—	O	Seg. Addr Fault **
0100	SEG #	ADDR	TD	RD	W	Mem. Addr Fault
0101	SEG #	ADDR	TD	RD		Data Line Fault
0102	SEG #	ADDR	TD	RD		'As' Data Fault
0103	SEG #	ADDR	TD	RD		'5s' Data Fault
0104	—	—	—	—		No Good Segments Above Zero**
0100	SEG #	ADDR	TD	RD	E	Segment Zero Memory Test (Descriptions As Above)
0101	SEG #	ADDR	TD	RD		
0102	SEG #	ADDR	TD	RD		
0103	SEG #	ADDR	TD	RD		
0200					R (sp)	ECC Single-bit Correction Failure
0201					U	ECC two-bit trap failure
0202					P (sp)	ECC two-bit error not reported
0203	SEG #	ADDR				ECC Check Byte RAM error
0300	MMU PORT #	SDR FIELD #	TD	RD	D	MMU's Not Individually Addressable
0301	MMU PORT #	SDR FIELD #	TD	RD		SAR or DSCR Indexing Fault
0302	MMU PORT #	SDR FIELD #	TD	RD		SDR 'As' or '5s' Data Fault
0303	MMU CMD #	TD	RD	—		MMU Control Register 'As' or '5s' Fault
0304	REG #	TD	RD	—		System/Normal Break Register 'As' or '5s' Fault
0305	MMU ID #	SDR #	VDAT	—	I	Stack MMU Did Not Trap On Limit Test
0305					A	Unexpected Trap
0305					G	Unexpected Trap
0305					N	Data MMU Did Not Trap On Limit Test
0305					O	Stack MMU Did Not Trap On Read-Only Test
0305					S	Data MMU Did Not Trap On Read-Only Test
0306	MMU Port #	SDR #	TD	RD	T	Translation Fault On Data MMU
0307	MMU PORT #	SDR #	VDAT	—		Unexpected Trap
0308	MMU PORT #	SDR #	TD	RD	I	Translation Fault On Stack MMU
0309	MMU PORT #	SDR #	VDAT	—		Unexpected Trap
0310	MMU PORT #	SDR #	TD	RD	C	Translation Fault On Code MMU
0311	MMU PORT #	SDR #	VDAT	—		Unexpected Trap
0312	MMU PORT #	SDR #	—	—	S (sp)	No Trap On Code MMU Limit Test

Table 3-3 **SPUD Diagnostics Error List (Continued)**

ERROR #	P1	P2	P3	P4	CHRS * PRINTED	DESCRIPTION
1000	—	—	—	—		No WDC Board In System
1001	DS1	DS2	DS3	DS4		WDC Self Test Error
1002	—	—	—	—		WDC Drive 0 Error
2000	—	—	—	—		No TCC Board In System
2001	—	—	—	—		Busy Bit Always Set***
2002	REG #	TD	RD	—		'5s' Data Fault
2003	REG #	TD	RD	—		'As' Data Fault
2004	IV REG	STATO REG	MIC REG	—		TCC Self-Test Error***
2005	IV REG	STATO REG	MIC REG	—		TCC Hardware Error***
3000						MDC Not Responding
3001	ADDR	—	—	—		RAM Error (P1 holds location)
3002						PROM Checksum Error
3003						Time Out Condition
3004						Read ABORT Error
3005						Wait ABORT Error
3006						Parity Error
3007						Not Used But Reserved
3008						Seek Not Complete Error
3009						Cylinder Not Found
3010						Drive Not Selected
3011						Head/Secter Not Found
3012						Invalid Command
3013						No Track 0 Found
3014						Drive Not Ready
3015						Bad Interrupt
3016						Bad MAP
3017						Illegal Cylinder Selected
3018						BEP Error
4000						SMC Not Responding
4001						SMC Initialization Error
4002						SMC RAM Error
4003	STATUS	—	—	—		SMC Self Test Timed Out Host Waiting (P1 holds SMC status register)
4004						Drive 0 Not Selected
4005						Drive 0 Not Ready
4006						Drive 0 Not On Cylinder
4007						Drive 0 Read Only
4008						Drive 0 Drive Fault
4009						Drive 0 Seek Error
4010						Drive 0 Not Formatted (Can't Size Disk)
COMPLETE						Last Characters of SPUD Message

Table 3-3 SPUD Diagnostics Error List (Continued)

* — Characters of SPUD message printed before entering test

** — Fatal error preventing further memory-related tests from being run

*** — The TCU test may take up to two minutes if the drive is busy or if the 'busy' status bit is stuck. The last two TCU error messages dump out the contents of the status registers for troubleshooting.

Pn — Test parameters of error printed (in hexadecimal):

SEG #	—	segment number
ADDR	—	address offset
TD	—	test data
RD	—	returned data
MMU PORT #	—	full work port number of MMU under test
MMU CMD #	—	MMU port number with command 'ored' in
SDR FIELD #	—	indicates a particular SDR in the range 0-255
MMU ID #	—	ID of MMU(s) returned from a segment trap
	—	1 = code MMU
	—	2 = data MMU
	—	4 = stack MMU
SDR #	—	logical segment number or set of SDR's (0-63)
VDAT	—	violation data from a single MMU trapping
(HB)	—	bus cycle status register data
(LB)	—	violation type register data
DS1	—	WDC detailed status - disk ready register
DS2	—	- disk status register
DS3	—	- operation error status
DS4	—	- self-test error status
REG #	—	register port number of unit under test
—	—	no parameter printed
STATUS	—	SMC status port contents

When the diagnostics are complete, the maximum available segment number will be displayed as follows (xx in hexadecimal):

```
POWER UP DIAGNOSTICS               POWER UP DIAGNOSTICS
ACTIVE PERIPHERALS:                ACTIVE PERIPHERALS:
 WDC                                 SMC (MODEL 31)
  TCC (MODEL 21)                     TCC (STD CONFIG)
  ECC (STD CONFIG)                   ECC
COMPLETE                           COMPLETE
MAXSEG=<xx>                        MAXSEG=<xx>
```

3.8. System Expansion

This section explains the installation of additional hardware to support the expansion or upgrading of an existing system.

3.8.1. Installing Additional Terminal Ports: The following instructions allow systems to be upgraded from 8 to 24 users. This capability requires the addition of the following items:

NOTE

> **The assumption is made that any of these system expansion options (listed in sub-sections 3.8.1 through 3.8.4) are being installed in a standard configured system. If the system has already been configured for another option, consult your local Zilog field representative and any notes sent with the upgrade kits for integration details.**

PART NUMBER	DESCRIPTION
05-8002-XX	Top Assembly, 16 User Upgrade
05-0177-XX	Top Assembly, 24 User Upgrade

To install the 16 user upgrade, proceed as follows:

1. Follow the necessary procedures for an orderly shutdown of the system (See "Down(M)" and "Halt(M)" commands in the Zeus System Administrator - Part # 03-3246).

2. Power down the system.

3. Remove the front panel and top cover from the Processor Module.

4. Remove the topmost blank plate from the topmost System 8000 module. Save the mounting hardware for assembly of the terminal distribution panel for secondary I/O.

5. Install the second terminal distribution panel for TTY8 to TTY15 (part number 08-0165-XX). Use the mounting hardware removed in step 3. Refer to the Installation Note, 03-0215, contained in the upgrade kit 05-8002-XX.

6. Attach the terminal expansion cable assembly (part number 59-0217-00) from I/O port connector OPTION, on the Processor Module, to I/O port connector TERMINAL IN, on the expansion terminal distribution panel. Refer to Figures 3-5 and 3-6.

7. Configure the Secondary Serial Board for use with the following printer interfaces:

PRINTER INTERFACE	JUMPER GROUP
Data Products	E1 to E2 E5 to E6
Centronics	E2 to E3 E4 to E5

8. Plug the Secondary Serial Board into slot 2 of the card cage.

9. Unseat and remove the CPU Board from slot 1. Replace the PROM at location U74 with the 16 user PROM (part number 34-0719-00.

10. Verify the printer configuration jumpers on the CPU Board as follows:

PRINTER INTERFACE	JUMPER GROUP
Data Products	E14 to E15 E16 to E17
Centronics	E13 to E14 E17 to E18

11. Verify CPU switch settings (U70) with Table 4-11 and reseat the CPU Board into slot 1.

12. Replace the Processor Module front panel.

13. Remove two (2) backplane jumpers at E12-B and E12-D for 16 users.

14. Replace the top cover.

15. Power up the system and run the appropriate SADIE diagnostics test.

16. On completion of the diagnostic test, perform the software modification procedure for adding terminals. This procedure is contained in the ZEUS System Administrator Manual (03-3246) paragraph 7.4, Adding Additional Terminals.

For the 24 user upgrade, refer to the Installation Note (03-0215). The upgrade is to be done only be Zilog Field Service.

3.8.2. Adding A 32MB Disk Module: The following instructions allow Model 21 systems to be upgraded by adding a Disk Module. This requires the addition of the following items:

PART NUMBER	DESCRIPTION	QUANTITY
05-0096-00	Front Panel, Disk Module	1
08-0169-00	Subassembly, Disk Module	1
98-8017-00	Stud, Ball, Threaded	4

Use the following procedure to install the Disk Module, and to modify the configuration of the FINCH Disk Drive in the present Disk/Tape Module. Install the added Disk Module between the Disk/Tape Module and the Accessory Module. Figure 3-16 shows the configurations for an existing and an upgraded system.

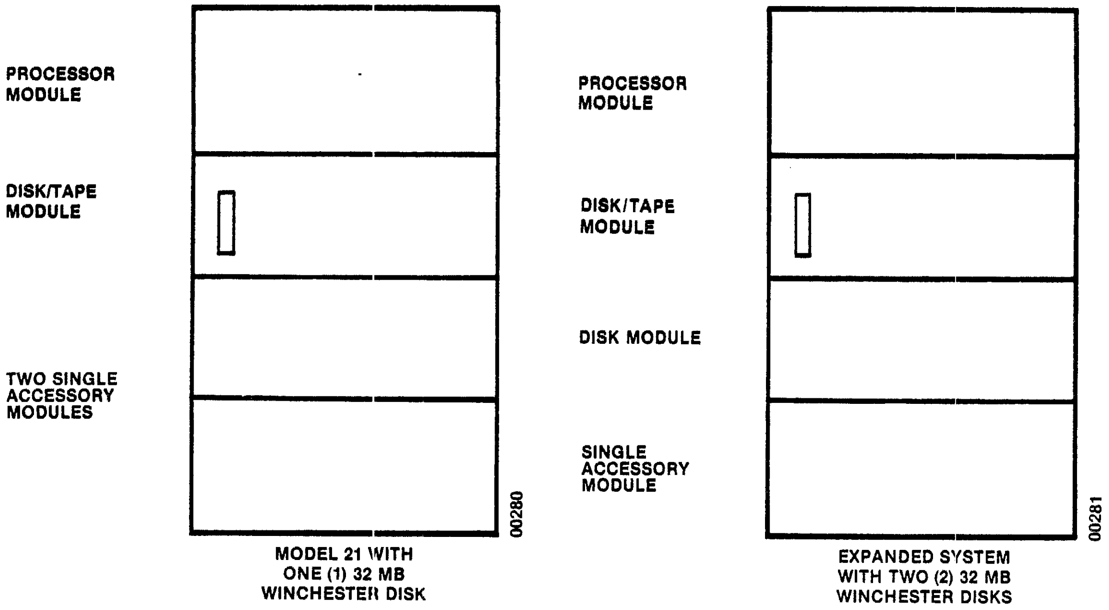

PROCESSOR MODULE
DISK/TAPE MODULE
TWO SINGLE ACCESSORY MODULES
00280
MODEL 21 WITH ONE (1) 32 MB WINCHESTER DISK
PROCESSOR MODULE
DISK/TAPE MODULE
DISK MODULE
SINGLE ACCESSORY MODULE
00281
EXPANDED SYSTEM WITH TWO (2) 32 MB WINCHESTER DISKS

Figure 3-16 Existing Model 21 and Expanded System Configurations

To install the Disk Module, proceed as follows:

CAUTION

Before the Disk/Tape Module can be removed from the present system, the FINCH Disk Drive read/write heads must be locked.

(1) To lock the FINCH Disk Drive read/write heads:

a. Follow the necessary procedures for an orderly shutdown of the system. Refer to Down(M) and Halt(M) commands inthe ZEUS System Administrator (03-3246).

b. Turn off AC power switch and remove AC power cord from outlet.

c. Using the flat end of the lock/unlock tool, lift and disengage the rotary-arm shipping lock. Push toward the rear of the drive until it locks into the SHIP position.

(2) Disassemble the system by removing the Processor and Disk/Tape Modules from the Accessory Modules. Proceed as follows:

a. Disengage the thumbscrews and open the cable covers on all modules.

b. Remove the disk and tape intermodule cables. Refer to Figure 3-4.

c. Remove terminal and printer cables from the terminal distribution I/O ports.

d. Loosen the captive fasteners on the rear of the Processor and Disk/Tape Modules.

e. Disengage the Processor Module from the guide posts on the front of the Disk/Tape Module, and remove.

f. Disengage the Disk/Tape Module from the guide posts on the front of the Accessory Module, and remove.

(3) Configure the FINCH drive in the new Disk Module as follows:

a. Remove the FINCH Adapter Board, mounted on the rear of the drive.

b. Verify that a terminator resistor pack is installed on the Adapter board in location RN3 (refer to Figure 3-6). The FINCH drive in the new Disk Module will be configured as the last drive in the Data/Command cable daisy chain.

c. The Unit Selection Dip Switch on the Adapter board must be set with switch position 2, ON. All other switch positions must be set to OFF. This dip switch specifies the unit address which ZEUS software and SADIE firmware will recognize.

d. On the FINCH Main PWA, verify that a Keyed Unit Select jumper plug is installed in position 1 on header J3. The plug establishes the logical unit designation; and can be installed in one of four possible orientations displaying the unit selected number to the rear of the drive.

e. Reinstall the Adapter board.

(4) Reconfigure the FINCH drive in the original Disk/Tape Module as follows:

a. Remove the FINCH Adapter Board.

b. Remove the terminator resistor pack in location RN3 (refer to Figure 3-8).

c. Verify that position 1 of the Unit Selection Dip Switch is set to ON. All other switch positions must be set to OFF.

d. On the FINCH Main PWA, verify that a keyed Unit Select jumper plug is installed in position 0 on the header J3.

e. Reinstall the Adapter board.

(5) Assemble the Disk Module, Disk/Tape Module, and Processor Module in the order shown in Figure 3-16 (Expanded System).

(6) Unlock the drives read/write heads. Using the crooked end of the head lock/unlock tool disengage the rotary-arm shipping lock. Pull towards the front of the drive until the arm locks into the OPERATE position.

(7) Attach all intermodule power and signal cables (see Figure 3-5 and 3-6).

(8) Run data non-destructive SADIE diagnostic WDCCRC on the FINCH drive in the Disk/Tape Module to ensure operability.

(9) Run SADIE diagnostic tests WDCTST3, WDCTST7, and WDCMEDIA on the FINCH drive in the Disk Module.

The addition of disk drives to an existing system requires that each new drive be configured to allow access by the ZEUS software. Refer to the "Adding Additional Disks" procedures specified in the ZEUS System Administrator Manual (03-3246).

3.8.3. Adding SMD Disk Module #1 to Model 31: Use the following procedures to add additional SMD Disk Modules to the System 8000.

1. Follow the necessary procedures for an orderly shutdown of the system. Refer to Down(M) and Halt(M) commands in the ZEUS System Administrator Manual (03-3246).

2. Power down the system, turn off AC power switch and remove AC input power cord from outlet.

3. Remove front panels from CPU and existing Disk Module(s) and lock heads by positioning Head Locking Actuator Lever in SHIP position. Ensure lever is locked in SHIP position.

4. Disassemble the system as follows:

   a. Disengage the thumbscrews, open cable covers and remove intermodule cables from system.

   b. Remove terminal and printer cables from I/O panel ports.

   c. Loosen captive fasteners on the rear of the processor and Disk/Tape Modules.

   d. Disengage the Processor Module from the guide posts on the front of the Disk/Tape Module, and remove.

5. On new Disk Module, locate SMD Mode Select Swtich (SW1) exposed through top cover of SMD drive (Figure 3-10). Set drive unit number using first three switch positons (SW1) (Table 3-1). SW2 and SW3 are factory set and are not to be changed.

6. On original Disk or Disk/Tape Modules (last one in system), remove the SMD metal cover exposing switches and terminators by loosening two screws at rear of SMD. Remove four cable terminators (Figure 3-14). These terminators are used in last drive only.

7. Reassemble system starting with new Disk Module which should be assmembled as the the bottom most Disk Module in the system (Figure 3-4).

8. Position SMD read/write heads in unlocked or OPERATE position.

9. Attach all intermodule power and signal cables (Figure 3-5, 3-6). Reconnect terminal and printer cables to terminal distribution panels.

10. Connect system to AC power outlet.

11. Power-up the system (Refer to ZEUS System Administrator (03-3246)).

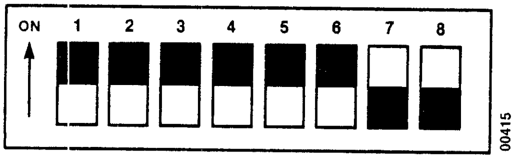
12. Run non-destructive SADIE diagnostics SMDCRC command on SADIE tape.

The addition of SMD disk drives to an existing system requires that each new drive be configured to allow access by the ZEUS software. Refer to the "Adding Additional Disks" procedures in the ZEUS System Administrator Manual (03-3246).

3.8.4. 1 Mbyte Memory Array Board Segment Setting: Each Memory Array has a DIP switch (S1) located near the edge connector at the center of the board. This switch has eight individual switches normally set at the factory. However, if a board is replaced or additional Memory Arrays added, these switches must be set for the correct segment addresses. Table 3-4 gives the option to be selected for 1 MByte to 4 MByte memory. Switches 1-4 are used to set the mega-segment offset on the memory board.

Table 3-4. Segment Address Settings on 1 MByte Memory

MEMORY	SEGMENTS	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
1MB	0-F	ON	ON	ON	ON	ON	ON	OFF	OFF
2MB	10-1F	OFF	ON	ON	ON	ON	ON	OFF	OFF
3MB	20-2F	ON	OFF	ON	ON	ON	ON	OFF	OFF
4MB	30-3F	OFF	OFF	ON	ON	ON	ON	OFF	OFF



Memory Array Switch S1

SECTION 4
THEORY OF OPERATION

4.1. General

This section discusses the theory of operation of the System 8000 based on block diagrams of various levels of complexity. The discussion begins with the basic building blocks of the system, system bus, the board modules on bus, bus conventions, and, its input/output. The text goes more deeply into the operation of the system: interrupts, addressing, 32 MB Winchester and 84 MB SMD disk and cartridge tape operations and interface, and peripheral connection. For the most part, this discussion does not cover the internal operation of the system boards.

4.2. System Bus

The block diagram in Figure 4-1 shows all the major elements of the System 8000 connected to the 32-bit Z-Bus Backplane Interconnect (ZBI). The ZBI is the system bus over which all communication between the elements on the bus take place. On the backplane of card cage, the ZBI is connected to connectors J11 through J20. On the logic diagrams and assembly drawings for the printed circuit boards, the signal lines of the ZBI are connected to the P1 connectors.

The CPU is the bus controller, and any other element on the bus that needs to gain control of the bus must request control from the CPU.

4.3. Bus Conventions

Signals on the bus may be active in either a high or a low state. All signals that use the bus have names or mnemonics that identify them. These names also indicate the active state of the signal; for example, the signal AS\ (address strobe) is an active low signal because it has a back slash appended to it. If the signal appears as AS with no backslash, it is active high. The back-slash is the same notation as the over bar that indicates the logical complement of a signal. Sometimes, active-low signals have a minus sign appended to them as in AS-. The over-bar, the backslash, and the minus sign all mean the same thing. This text uses the back-slash; some of the drawings use either the over-bar or the minus sign (-).

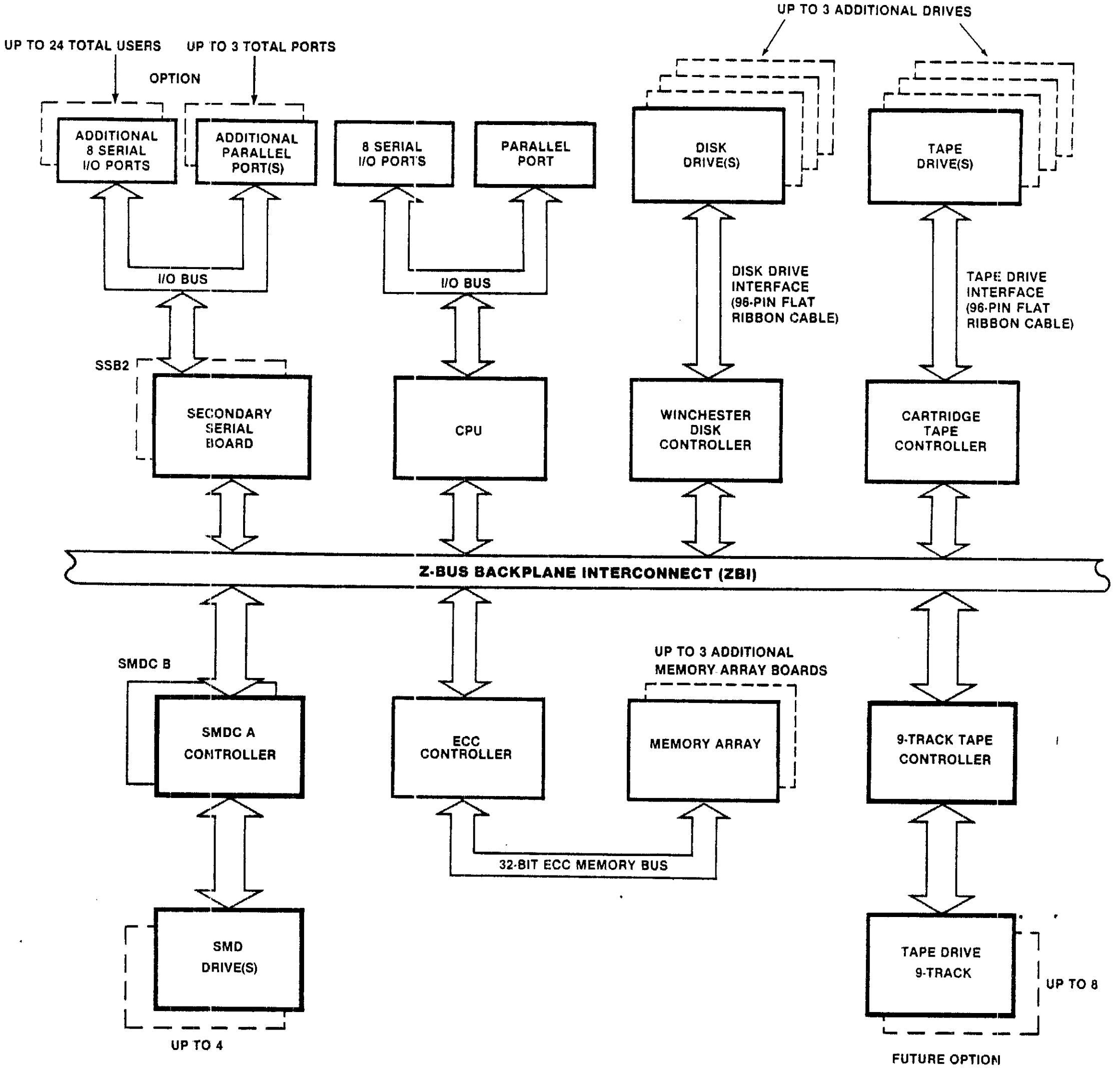


Figure 4-1 System 8000, Functional Relationships

4.4. Bus Signals

Table 4-1 lists all the ZBI signal lines and their definitions. One signal name can designate more than one signal line; in this case, the signal name is followed by numbers in angles brackets (<>) which indicate the quantity of lines and their designations. For example, the signal designated AD<31:0> is the name for the 32 address and data lines (AD0 through AD31) that are part of the bus. Within the angled brackets, the 31 indicates the most significant line.

The status lines, ST<4:0>, and the data width lines, B/W\ and W/LW\, form specific codes that cause a number of discrete operations to occur. Tables 4-2 and 4-3 list the various codes on the status and data width lines respectively, and the operations that the codes initiate.

Table 4-1 Bus Lines

SIGNAL NAME\ MNEMONIC	NUMBER OF LINES	FUNCTION
AD<31:0>	32	Multiplexed Address/Data lines: These lines are driven by the bus master. The address strobe (AS\) and data strobe (DS\) determine when the information on these lines is valid.
ME\	1	Memory Error: During a memory access, if the memory controller detects an uncorrectable error, the controller sends the ME\ signal to the bus master.
ST<4:0>	5	Status Lines: These active-high lines indicate the type of transaction currently occurring on the bus. (See Table 4-2 for the various codings and their associated transactions.)
R/W\	1	Read-Write: This is a dual-purpose line. When this line is high, it indicates that the current operation is a read operation; when the line is low, the operation is a write operation.
N/S\	1	Normal-System: Indicates the mode of operation of the master that is currently controlling the bus.

SIGNAL NAME\ MNEMONIC	NUMBER OF LINES	FUNCTION
B/W\	1	Byte-Word Select: This signal is used with signal W/LW\ (listed below) to define the data access width. (See Table 4-3 for coding.)
W/LW\	1	Word/Long-Word Select: This signal is used with signal B/W\ (listed above) to define the data access width. (See Table 4-3 for coding.)
AS\	1	Address Strobe: The bus master drives this line low to initiate a bus transaction. The rising trailing edge indicates that the current address and status are valid.
DS\	1	Data Strobe: The bus master uses this signal to time the movement of data to and from itself along the data bus.
WAIT\	1	Wait: By forcing this line low, a bus slave causes the bus master to suspend operation while the slave completes its activity.
STOP\	1	Stop Line: This line, when driven low by an EPU device causes a Z8000 CPU to generate null transactions. During a null transaction, the data strobe (DS\) remains high.
BAI\	1	Bus Acknowledge In: This signal along with BAO\ forms the bus priority chain.
BAO\	1	Bus Acknowledge Out: This signal along with signal BAI\ forms the bus priority chain.

SIGNAL NAME\ MNEMONIC	NUMBER OF LINES	FUNCTION
BUSREQ\	1	Bus Request: A module uses the BUSREQ\ signal to gain access to the bus. This signal is part of the priority scheme that is set up by the connection of signals BAI\ and BAO\.
CAI\	1	CPU Acknowledge In: Not currently implemented--the CAI, CAO, CPUREQ, CAVAIL signals are designed to allow multiple CPUs to share a single bus. They may be used on a future S8000 product and should be considered reserved.
CAO\	1	CPU Acknowledge Out: Not currently implemented--reserved
CPUREQ\	1	CPU Request: Not currently implemented--reserved
CAVAIL	1	CPU Available: Not currently implemented--reserved
INT1\	1	Level-1 Interrupt: This interrupt line has the highest priority in the system. This line when driven by a slave generates a non-maskable interrupt (NMI).
INT2\	1	Level-2 Interrupt: This interrupt line has the second to the highest priority in the system. This line, when driven by a slave, generates a vectored interrupt.

SIGNAL NAME\ MNEMONIC	NUMBER OF LINES	FUNCTION
INT3\	1	Level-3 Interrupt: This interrupt line has the lowest priority in the system. This line, when driven by a slave, generates a non-vectored interrupt.
IEI1	1	Level-1 Interrupt Enable In: This signal works with Level-1 interrupt enable out to form the NMI acknowledge daisy chain.
IEO1	1	Level-1 Interrupt Enable Out:
IEI2	1	Level-2 Interrupt Enable In: This signal works with Level-2 interrupt enable out to form the VI acknowledge daisy chain.
IEO2	1	Level-2 Interrupt Enable Out:
IEI3	1	Level-3 Interrupt Enable In: This signal works with Level-3 interrupt enable out to form the NVI acknowledge daisy chain.
IEO3	1	Level-3 Interrupt Enable Out:
MMREQ\	1	Multimicro Request: When this signal is active a module can request the use of a common resource. The MMREQ\ signal works with signals MMAI\ and MMAO\.
MMAI\	1	Multimicro Acknowledge In: This signal works with signal MMAO\ to form the resource-request daisy chain.

SIGNAL NAME\ MNEMONIC	NUMBER OF LINES	FUNCTION
MMAO\	1	Multimicro Acknowledge Out: This signal works with signal MMAI\ to form the resource-request daisy chain.
PWRBAD\	1	Power Bad: The processor power supply generates this as an early warning to the system that the DC power will soon disappear.
MCLK	1	Master Clock: This signal is the system clock and is the foundation for all timing in the system. The frequency of the MCLK signal is four times (4X) that of the bus clock (BCLK).
BCLK	1	Bus Clock: The system derives this clock from the master clock (MCLK). The BCLK is one fourth the frequency of the master clock and synchronizes the operation of the elements in the system that require synchronization. All bus transfers are synchronized to this clock. The system CPU board is the generator of this clock and MCLK above.
RESET\	1	Reset: This is the master reset signal for the entire system. This signal is generated by the front panel master reset switch or upon power-up by the power-up reset circuit. When it is forced low, it initializes the entire system.

Table 4-2 ZBI Status Lines, Transaction Coding

S4	S3	S2	S1	S0	TRANSACTION
0	0	0	0	0	Internal operation
0	0	0	0	1	Memory refresh
0	0	0	1	0	I/O reference
0	0	0	1	1	Special I/O reference
0	0	1	0	0	Segment trap acknowledge
0	0	1	0	1	INT 1 Interrupt acknowledge
0	0	1	1	0	INT 3 Interrupt acknowledge
0	0	1	1	1	INT 2 Interrupt acknowledge
0	1	0	0	0	Data memory request
0	1	0	0	1	Stack memory request
0	1	0	1	0	Transfer between data memory and an EPU
0	1	0	1	1	Transfer between stack memory and an EPU
0	1	1	0	0	Program reference, nth cycle
0	1	1	0	1	Program reference, 1st cycle
0	1	1	1	0	Transfer between CPU and EPU
0	1	1	1	1	Reserved
1	X	X	X	X	Reserved

Table 4-3 Data Width Codes: Byte, Word, and Long Word

B/W\	W/LW	DATA WIDTH
1	1	Sets the data width to byte width, 8 bits, data on lines AD<7:0>
0	1	Sets data width to word size, 16 bits, data on lines AD<15:0>
1	0	Sets data width to double-word size, 32 bits, data on lines AD<31:0>
0	0	Reserved

4.5. Bus Modules

The bus modules are the major blocks that communicate directly with the CPU over the bus. The one exception is the ECC Memory Array module whose communications path to the bus is through the Memory Subsystem Controller. The following paragraphs deal more closely with the individual modules.

4.5.1. CPU Module: The CPU module is the bus controller, sometimes called the host, which initiates and controls transactions on the bus. Also as shown in Figure 4-2, the CPU connects directly to and controls the I/O bus. All transactions with the outside world pass through either the parallel port or one of the eight serial I/O ports. The I/O lines from the CPU module pass through mating connectors P2 and J21, located on the CPU module and backplane, respectively. Table 4-4 lists the lines on the CPU I/O bus and their definitions.

Table 4-4 CPU I/O Bus, Signal Definitions

SIGNAL NAME	DEFINITION
TXD7 to TXD0	Transmit Data, 8 bits
RXD7 to RXD0	Receive Data, 8 bits
CTS7 to CTS0	Clear to Send
DTR7 to DTS0	Data Terminal Ready
RTS7 to RTS0	Request to Send
DSR7 to DSR0	Data Set Ready
TXRTN7 to TXRTN0	Transmit Return
DATA7 to DATA0	PIO Data
DATA STROBE	Data Products Data Strobe
DATA STROBE\	Centronics Data Strobe, Active low

SIGNAL NAME	DEFINITION
DATA DEMAND/ ACKNOWLEDGE\	Demand (Data Products) when high Acknowledge (Centronics) when low
BUSY\	Printer Busy
IFVALID	Interface valid (Data Products)
FAULT\	Paper empty indication (Centronics)
ON-LINE/SELECT	ON-LINE (Data Products), SELECT (Centronics)
F.P. BUSACK INDICATOR (Front Panel)	DMA in process Disk or tape controller in control of bus
F.P. POWER-ON INDICATOR (Gnd) (Front Panel)	Ground for power-on indicator
F.P. POWER-ON INDICATOR V+ (Front Panel)	Indicates system is on
F.P. NORMAL INDICATOR (Front Panel)	CPU running user process
SWITCH N.C. START (Front Panel)	Auto boot
SWITCH N.O. START (Front Panel)	Auto boot
SWITCH RESET (Front Panel)	Resets system

Hardware jumpers on the CPU board can be set for either a Centronics or Data Products printer interface. Refer to Table 4-5 for possible jumper configurations.

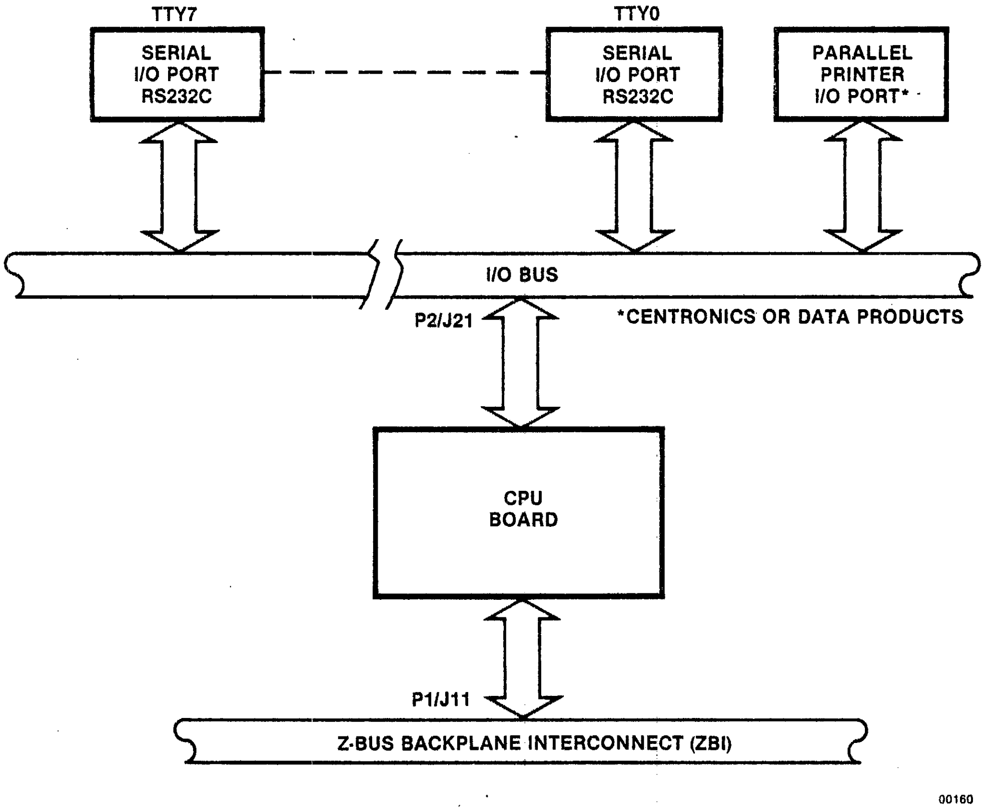

TTY7
SERIAL I/O PORT RS232C
TTY0
SERIAL I/O PORT RS232C
PARALLEL PRINTER I/O PORT*
I/O BUS
P2/J21
*CENTRONICS OR DATA PRODUCTS
CPU BOARD
P1/J11
Z-BUS BACKPLANE INTERCONNECT (ZBI)
00160

Figure 4-2 · CPU Board, Functional Relationships

The System 8000 CPU board is currently configured for a non-segmented operating system. A segmented mode configuration is shown for future consideration only. Jumpers E1 through E12 (refer to Figure 1-2) determine the operating mode. Support of both segmented and nonsegmented users is provided and is not determined by these jumpers.

Jumpers are factory set for nonsegmented operation as follows:

Table 4-5 CPU Board Jumper Selection

NON-SEGMENTED	SEGMENTED (FUTURE)
E2 to E3	E1 to E2
E4 to E5	E6 to E5
E7 to E8	E8 to E9
E11 to E12	E10 to E12

JUMPERS E1 THROUGH E12 ARE PROPERLY SET BY FACTORY AND ARE NOT TO BE CHANGED BY USER.

For either a Data Products or Centronics interface, connect Jumpers E13 through E18 as follows:

CENTRONICS INTERFACE	DATA PRODUCTS INTERFACE
E13 to E14	E14 to E15
E17 to E18	E16 to E17

4.5.1.1 I/O Bus: The CPU board I/O bus connects the System 8000 with the outside world. All devices that control both the serial and parallel I/O are on the CPU board; functionally, these devices form eight serial I/O channels and two parallel I/O channels. The serial channels support the RS-232C standard and the parallel ports can be configured for either the Data Products or Centronics standard. One parallel port, Port B of a Z80-PIO, is the data-out port; port A handles status and control information. Table 4-6 lists the control signals for the parallel printers. Table 4-7 lists the status signals for the printers, and Table 4-8 lists the data output of port B.

Table 4-6 Parallel Printer Output Control Signals, Port A

BIT NUMBER	CENTRONICS	DATA PRODUCTS
0	Data strobe\	Data strobe
1	not used	not used
2	not used	not used
3	not used	not used

Table 4-7 Parallel Printer Input Status Signals, Port A

BIT NUMBER	CENTRONICS	DATA PRODUCTS
4	Busy\	Busy\
5	Select	Online
6	Fault\	Interface Valid
7	Acknowledge\	Data Demand

Table 4-8 Parallel Printer Data, Port B

BIT NUMBER	CENTRONICS	DATA PRODUCTS
Bit 0	Data Bit 0	Data Bit 0
Bit 1	Data Bit 1	Data Bit 1
Bit 2	Data Bit 2	Data Bit 2
Bit 3	Data Bit 3	Data Bit 3
Bit 4	Data Bit 4	Data Bit 4
Bit 5	Data Bit 5	Data Bit 5
Bit 6	Data Bit 6	Data Bit 6

4.5.1.2 Serial I/O: The serial I/O comprises 4 Z80B-SIO/2 devices. Each device has two channels; Table 4-9 lists the devices and their assigned channels.

Table 4-9 Serial I/O Devices and Channel Assignments

DEVICE NUMBER	CHANNEL ASSIGNMENT
SIO 0	Channels 0 and 1
SIO 1	Channels 2 and 3
SIO 2	Channels 4 and 5
SIO 3	Channels 6 and 7

Each channel of the serial I/O connects to its own baud-rate generator. These generators are channels in Z80B-CTC devices. The SIO channels and their corresponding baud-rate generators are listed in Table 4-10. The baud rate clock comes from an independent baud-rate oscillator. The frequency of the baud-rate clock is 1.2288 megahertz.

Table 4-10 Serial Channels and Baud-rate Generators

SIO	SIO CHANNELS	BAUD NO.	CTC NO.	CTC CHANNEL
0	0	0	0	0
0	1	1	0	1
1	2	2	0	2
1	3	3	1	0
2	4	4	1	1
2	5	5	1	2
3	6	6	2	0
3	7	7	2	1

Serial channel 1 (console), with two exception, is identical to every other serial channel.

The first exception is that the on-board monitor on the CPU board uses channel 1 to communicate with the system operator when the system is turned on. The initial baud rate for channel 1 is factory set for 9600 by switch U70. It can be set to one of the four values listed in Table 4-11. These settings permit the use of a variety of terminals as the system console. After the system has been booted, the console baud rate can also be changed under software control. The 4-pole dip swtich on the CPU board selects the primary boot device setting the baud rate for the monitor console. The primary boot device will always be listed and tested first by the SPUD Diagnostics. System error messages are sent to the console. This is the second exception.

Table 4-11 Baud Rate and Primary Boot Device Switch Setting (SW U70)

SWITCH (ON = 0)	BAUD RATE	SWITCH	PRIMARY BOOT DEVICE
1, 4		2, 3	
0 0	300 Baud	1 1	8" Disk
1 0	1200 Baud	0 1	5.25" Disk
0 1	9600 Baud	1 0	SMDC
1 1	19200 Baud	0 0	Reserved

System software can access the I/O channels using standard Z8000 I/O instructions. Table 4-12 lists the I/O addresses of the I/O channels.

Table 4-12 I/O Channels and Their Addresses

I/O ADDRESS	I/O DEVICE AND CHANNEL
FF81	SIO 0, channel 0, data
FF83	SIO 0, channel 1, data
FF85	SIO 0, channel 0, control
FF87	SIO 0, channel 1, control
FF89	SIO 1, channel 2, data
FF8B	SIO 1, channel 3, data
FF8D	SIO 1, channel 2, control
FF8F	SIO 1, channel 3, control
FF91	SIO 2, channel 4, data
FF93	SIO 2, channel 5, data
FF95	SIO 2, channel 4, control
FF97	SIO 2, channel 5, control
FF99	SIO 3, channel 6, data
FF9B	SIO 3, channel 7, data
FF9D	SIO 3, channel 6, control
FF9F	SIO 3, channel 7, control
FFA1	CTC 0, channel 0 (baud 0 for SIO 0, channel 0)
FFA3	CTC 0, channel 1 (baud 1 for SIO 1, channel 1)
FFA5	CTC 0, channel 2 (baud 2 for SIO 1, channel 2)
FFA7	CTC 0, channel 3
FFA9	CTC 1, channel 0 (baud 3 for SIO 1, channel 3)
FFAB	CTC 1, channel 1 (baud 4 for SIO 2, channel 4)
FFAD	CTC 1, channel 2 (baud 5 for SIO 2, channel 5)
FFAF	CTC 1, channel 3
FFB1	CTC 2, channel 0 (baud 6 for SIO 3, channel 6)
FFB3	CTC 2, channel 1 (baud 7 for SIO 3, channel 7)
FFB5	CTC 2, channel 2
FFB7	CTC 2, channel 3
FFB9	PIO 0, channel A, data
FFBD	PIO 0, channel A, control
FFBB	PIO 0, channel B, data
FFBF	PIO 0, channel B, control

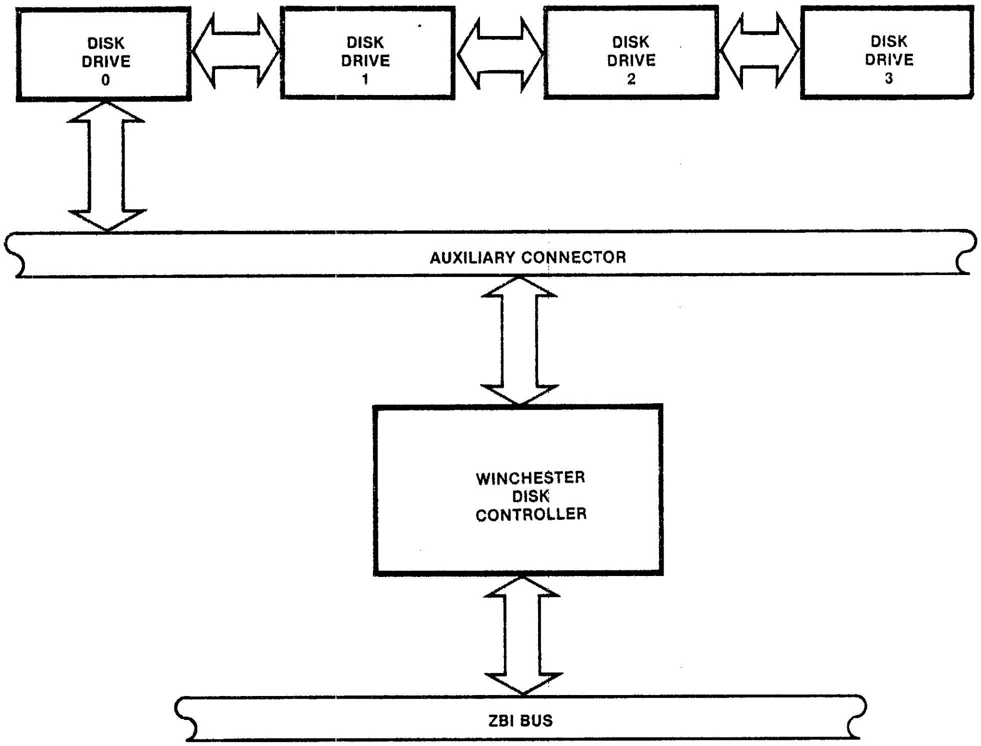
4.5.2. Winchester Disk Controller: The Disk Controller controls the fully buffered transfer of data between the CPU (host) and a selected disk drive. The block diagram in Figure 4-3 shows the relationship between the controller and both the ZBI and the disk drives. All transactions between the controller and the host pass through connectors P1 and J13 and over the ZBI. Transactions between the controller and a selected disk drive pass through connectors P2 and J23, the drive bus. The signals on connector P1 are the standard ZBI signals; the signals on P2 are common to only slot three on the back plane. Table 4-13 lists the interface signals between the disk controller and the disk drives.

4.5.2.1 CPU Interface: The CPU communicates with the controller through 16 8-bit command registers and an 8-bit command-status (C/S) register (Figure 4-4). Each register has a specific command assignment and a specific address. The CPU writes commands into command registers xx00 through xx0F; the controller reads these registers and performs the specified commands. The controller places the results or status of the specified command in the C/S register which the CPU reads. Table 4-14 lists the command and C/S registers.

The command and C/S registers reside in the CPU's I/O space on any 256-word boundary. Within this 256-word block, the command registers are at relative addresses xx00 to xx0F hexadecimal and the C/S register at address xx10. The two most significant hex values of the address, xx, can be set by jumpers on the controller board. Table 4-15 lists the jumpers and the bits (15 through 8) that the jumpers control. This scheme allows more than one controller within the same I/O space.

Figure 4-5 shows a segment of I/O space, containing three 256-word blocks. The two most significant hexadecimal nibbles (AA, BB, and CC) of the addresses can be set by using the jumpers listed in Table 4-15. For example, in the address AA00, if AA is to equal FF hex, then no jumper is connected in any of the jumper groups, and all the lines are high. When a jumper is connected in any jumper group, the jumper shorts its associated line to ground, a low level.

The controller can accommodate either 2716 or 2732 EPROMs. Jumpers on the controller board permit the selection of both the type of EPROM and any necessary wait states. Table 4-16 lists the jumpers for memory selections.



00161

Figure 4-3 Winchester Disk Controller, Functional Relationships

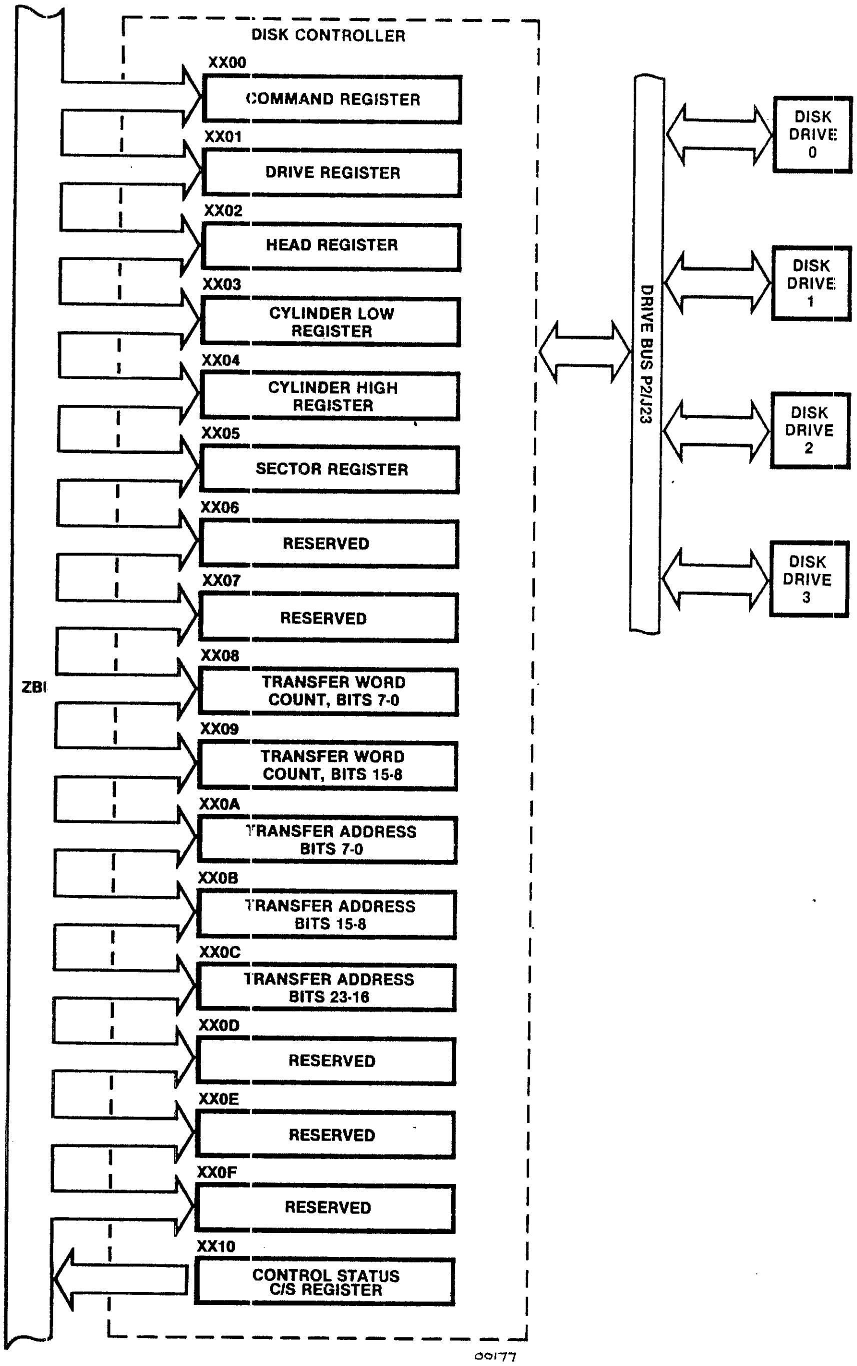


Figure 4-4 Disk Controller Command and Command Status (C/S) Registers

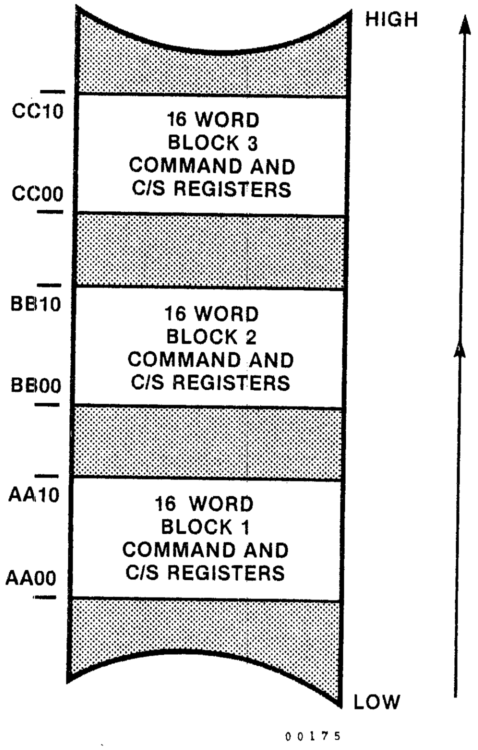


Figure 4-5 Disk Controller, I/O Space

4.5.2.2 Controller/Drive Interface: Communications between the disk controller and a selected disk drive travel over the P2/J23 mating connectors. Table 4-13 lists and defines the signals on these lines. The disk controller can control either a CDC or Finch drives. All drives attached to any controller must be of one type. Table 4-17 lists the codes for the command and status words on the control bus between the disk controller and the disk drive.

Table 4-13 Disk Controller and Disk Drive, Interface Signals, Connector P2/J23, Slot 3

SIGNAL NAME	FUNCTION
Control Bus CB7 to CB0	These eight signal lines are bidirectional. Control signals from the Controller to the selected drive are transmitted on these lines. The DIRECTION\ signal controls the direction transfer.
DIRECTION\	This signal determines the direction of transfer over the control bus (CB7 through CB0). When DIRECTION\ is high, the controller reads status information from one of the four status registers. When this line is low the controller is sending commands to the selected drive.
Control Words 0 and 1, CWD0 and CWD1	These signals identify one of four bytes that can be on the control bus. The DIRECTION\ signal identifies the current byte as a command or status byte.
Attention, ATTN\	The controller sends this signal to all drives to initiate handshaking.
Cycle Acknowledge CYAK\	Each drive generates the CYAK\ signal in response to the ATTACK signal.
Attention Acknowledge ATTACK	The selected drive generates this active-high signal in response to the ATTN\ signal from the controller. ATTACK then causes the selected drive to generate the CYACK signal.
INDEX\	The selected drive generates an index pulse for each revolution of the disk. Each cylinder produces a pulse which is 2.5 microseconds long and which occurs every 16.67 milliseconds. Sector 00 immediately follows the pulse.
SECTOR\	The selected drive generates this 2.5 microseconds signal.
SEEK END\	The selected drive generates this signal at the end of a seek operation.
DRIVE FAULT\	The selected drive generates this signal to indicate an error condition within the drive itself.

SIGNAL NAME	FUNCTION
Unit Acknowledge 0 to 3 UNITACK0\ to UNITACK3\	The selected drive places its binary address on these lines.
READ ENABLE\	The controller generates this signal to read the selected drive at the current cylinder. Before activating this signal the controller must send a control word 2 and then wait until the selected drive returns the CYACK signal.
WRITE ENABLE\	The controller generates this signal to write to the selected drive at the current cylinder. If the heads are write-protected, a fault results. The controller must send a control word 2 and then wait until the selected drive retunrs the CYACK signal.
MASTER RESET\	A high-to-low transistion on this line resets all internal latches and output ports.
BI-DATA\	These lines form a bidirectional, differential pair that transmits NRZI read data from the currently selected drive to the controller whenever READ ENABLE\ is active.
SYSTEM CLK\	A pre-recorded pattern on the surface of one of the disks on the drive generates this signal. The selected drive sends this signal to the controller: the controller derives the write clock (WRITE CLK\) from system CLLX. The controller sends the WRITE CLK\ to the selected drive.
WRITE CLK\	This is a differential signal which the controller derives from the SYSTEM CLK\.

Table 4-14 Command and Command-Status Registers

REGISTER ADDRESS	NAME	DEFINITION
xx00	Command Register	The CPU sends specific commands to the controller through this register. This must be the last register the CPU writes to., for it clears the status byte to enable handshaking between the CPU and the controller and begins the operation.
xx01	Unit Register	The CPU uses this register to specify which unit (drive) it wants to participate in I/O activity. One of four drives (0, 1, 2, or 3) can be specified.
xx02	Head Register	The CPU uses this register to specify the read-write head it wants for the current operation.
xx03	Cylinder Address Low Register	The CPU stores the low-order 8 bits (7 to 0) of the 16-bit cylinder address in this register.
xx04	Cylinder Address High Register	The CPU stores the high-order 8 bits (15 to 8) of the cylinder address in this register. With the current disk drives, only bits 8 and 9 are used.
xx05	Sector Register	The CPU stores in this register the number of the sector it wants to read from or write to.
xx06	Reserved	For expansion
xx07	Reserved	For expansion
xx08	Transfer Word Count Bits 7 to 0	This register contains the loworder byte of the transfer word count.
xx09	Transfer Word Count Bits 15 to 8	This register contains the highorder byte of the transfer word count.

REGISTER ADDRESS	NAME	DEFINITION
xx0A	Transfer Address Bits 7 to 0	This register contains the loworder byte of the 3-byte transfer address. The transfer address is the location of the first word of a block of memory allocated for the transfer. Data can be leaving memory or coming to it. The command from the CPU determines the direction of transfer. The Read Sector command moves data from disk to main memory. The Write Sector command moves data from main memory to disk.
xx0B	Transfer Address, Bits 15 to 8	The CPU stores the intermediate byte of the transfer address in this register.
xx0C	Transfer Address	The CPU stores the high-order byte of the transfer address in this register.
xx0D	Reserved	For transfer address bits 32 to 24.
xx0E	Reserved	
xx0F	Reserved	
xx10	Command-Status (C/S) Register	

Table 4-15 Jumper Settings for Address of Command and C/S Registers

JUMPERS	PURPOSE	CONNECTION AND RESULT
E10,E18	Causes bit 15 of address	E10 to E18: bit 15 low to be either a high or low level
E11,E19	Causes bit 14 of address	E11 to E19: bit 14 low to be either a high or low level
E12,E20	Causes bit 13 of address	E12 to E20: bit 13 low to be either a high or low level
E13,E21	Causes bit 12 of address	E13 to E21: bit 12 low to be either a high or low level
E14,E22	Causes bit 11 of address	E14 to E22: bit 11 low to be either a high or low level
E16;E23	Causes bit 10 of address	E15 to E23: bit 10 low to be either a high or low level
E15,E24	Causes bit 09 of address	E16 to E24: bit 09 low to be either a high or low level
E17,E25	Causes bit 08 of address	E17 to E25: bit 08 low to be either a high or low level

Table 4-16 Disk Controller Jumper Settings for Memory

JUMPER GROUP (E)	PURPOSE	CONNECTION AND RESULT
E1,E2,E3 E41,E42,E43	Permit inserting one wait state whenever any on-board memory is accessed or whenever only on-board EPROMs are accessed.	E2 to E3 and E42 to E43: Inserts one wait state during access of any on-board memory. E1 and E3 and E42 to E43: Inserts one wait state during access of EPROM only. E41 to E43: No wait states for controller on-board RAM or EPROM.
E4,E5,E6 E7,E8,E9	Permit selecting either 2716 or 2732 EPROMs.	E4 to E6 and E7 to E9: Use 2716 EPROMs. E5 to E6 and E8 to E9: Use 2732 EPROMs.

4.5.2.3 Command and Status Words: The disk drive control bus contains a command word when the direction line is activated, otherwise it contains a status word. The number of the command or status words is determined by the coding of CWD0 and CWD1.

Command Word 0--Command Word 0 is used to select one of fifteen drives by a four bit unit address (uA).

Command Word 1--Command Word 1 is used, together with the two low-order bits of Command Word 0, to establish the binary address of the desired cylinder specified by the Cylinder Address Register (CAR).

Command Word 2--Command Word 2 has three basic purposes: (1) select forward or reverse offset, (2) select early or late data strobe, and (3) select one of three possible head positions by the HAR0 and HAR1 bit positions.

Command Word 3--Command Word 3 enters a Diagnostic mode, Customer Engineering (CE) mode, Rezero (Return the heads to cylinder zero), clears a fault, or establishes which heads are to be write protected.

Status Words--The control bus contains a status word when the direction line is deactivated for four possible standard status words. Status Word 0--Status Word 0 communicates eight specific error conditions to the Controller-Formatter when a fault condition exists.

Status Word 1--Status Word 1 signals seven different error/exception conditions to the Controller-Formatter.

Status Words 2 and 3--Together contain the contents of the Position Address Register (PAR) for current locations of the heads.

Table 4-17. Disk Command and Status Words

		DIR	CWD1	CWD0	(MSB) Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	(LSB) Bit 0
COMMAND	WORD 0	1	0	0	UNIT ADDR3	UNIT ADDR2	UNIT ADDR1	UNIT ADDR0	SPARE 0	SPARE 0	CAR9	CAR8
	WORD 1	1	0	1	CAR7	CAR6	CAR5	CAR4	CAR3	CAR2	CAR1	CAR0
	WORD 2	1	1	0	SERVO OFFSET REV	SERVO OFFSET FWD	STROBE LATE	STROBE EARLY	SPARE 0	SPARE 0	HAR1	HAR0
	WORD 3	1	1	1	DIAG MODE	CE MODE	REZERO	FAULT CLEAR	SPARE	SPARE	EXT PROT1	EXT PROT0
STATUS	WORD 0	0	0	0	NOT READY	SERVO ERROR	R/W FAULT	SPEED ERROR	PWR LOSS	WRITE PROTD	SEEKING RZRNG	NOT ON CYL
	WORD 1	0	0	1	GUARD BAND	PLO ERROR	UNSAFE	INVAL CMND	TIME OUT	POR/ MR	SPARE	ILL ADDR
	WORD 2	0	1	0	SPARE	SPARE	SPARE	SPARE	SPARE	SPARE	PAR9	PAR8
	WORD 3	0	1	1	PAR7	PAR6	PAR5	PAR4	PAR3	PAR2	PAR1	PAR0

4.5.3. Storage Module Device Controller: The storage Module Device Controller (SMDC) is a high performance controller that links the ZBI system bus to each storage Module Drive (SMD) interface board. The SMDC consists of two boards (SMDCA and SMDCB). Board SMDC A is the ZBI control signal connection to the I/O panel SMDCA OUT connector. External cabling to SMDC A IN on the Disk/Tape Module connects the ZBI signals through a 60-pin SMD A cable I/O connector to the SMD Interface board. The SMDC B board is the data I/O connection through I/O panel SMDC B OUT on the processor and SMDC B IN on the Disk/Tape Modules. I/O Data to and from the Processor Module is provided at the SMDC B panel connectors that connect up to four 26-Pin SMD B cables directly to the interface of each SMD drive. Local interface signals between the two boards are cabled on a flat 40 pin cable to the front of both boards. All drive connections are made by cabling backplane connectors to the SMD A and SMD B I/O connectors on the processor SMDC I/O panel and from the I/O panel to the SMDC A and SMDC B inputs on the Disk/Tape Module. Figure 4-6 shows the SMDC A and SMDC B Functional Relationships. The SMDC A ZBI connection is through P1/J16. All data transfers and input/output addresses are 16 bits. All data addresses are 24 bits. The interrupt vector is programmable by the host CPU.

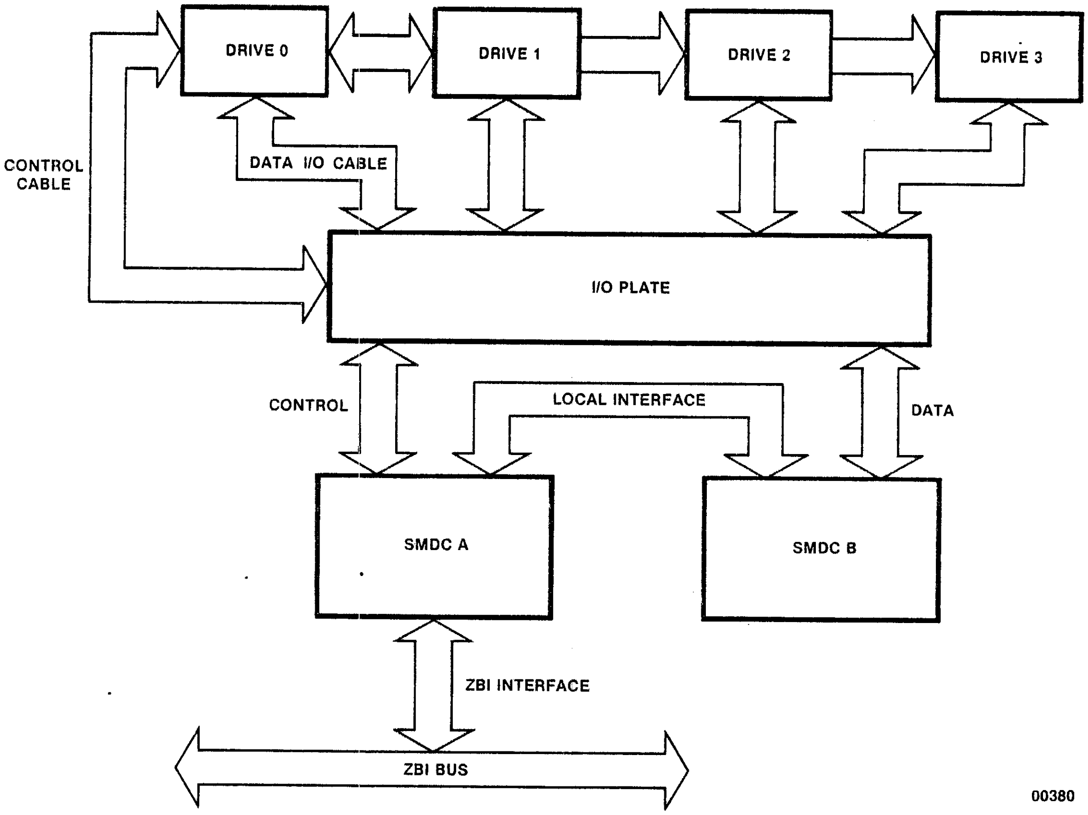


Figure 4-6 Storage Module Device Controller Functional Relationship

4.5.3.1 Command Packet Control: Commands are sent to the controller in the form of 32-byte packets that describe the operation to be performed. At any given time, the controller may be executing one command packet for each SMD. Upon completion of the command, the packet is written back into main memory and the appropriate status registers are updated. The controller maintains a separate packet address for each SMD. Most commands are sent to the SMDC in a packet of 32 bytes. However, certain information, such as packet address and interrupt controls, is communicated by one 16-bit write-only command and one 16-bit read-only status register that share a common address.

Command Register

```
   15 ... 8     7      6     5     4     3    2 1 0
-----------------------------------------------
    DTA     ¦ INIT   ¦ RI  ¦ DI  ¦ EI  ¦ WK  ¦ CMD    ¦
-----------------------------------------------
```

NAME	BITS	FUNCTION
CR:CMD	0-2	COMMAND 0 - nop 1 - read packet addresses from DT 2 - reserved 3 - reserved 4 - dispatch table address byte 0 (lsb) 5 - dispatch table address byte 1 6 - dispatch table address byte 2 (msb) 7 - interrupt vector
CR:WK	3	wakeup
CR:EI	4	enable interrupts (reset by IUS)
CR:DI	5	disable interrupts
CR:RI	6	reset IP and IUS
CR:INIT	7	initialize controller
CR:DTA	8-15	dispatch table address byte or interrupt vector

Status Register

```
 15 14 13 12 11 10  9  8   7    6    5    4    3    2    1     0
-------------------------------------------------------------------
| DRV |ES: ending status| NDT | 0 | 0 | 0 | 0 | IP | IUS | BZ
-------------------------------------------------------------------
```

NAME	BIT	MEANING
SR:BZ	0	controller busy from CR:CMD
SR:IUS	1	interrupt under service
SR:IP	2	interrupt pending
-	3-6	reserved
SR:NDT	7	no dispatch table/interrupt vector
SR:ES	8-13	packet ST or selftest code
SR:DRV	14-15	drive number 0-3

Self-test

On power-up or after a controller initialization (CR:INIT), the controller will be busy (SR:BZ) until the selftest routine and initialization are complete. If the selftest fails, the SMDC will remain busy. The status register (SR:ES) may contain one of the following error codes:

8:	2910 sequencer error
9:	2901 ALU error
A:	internal memory error

The status register should be examined only after it has been determined that the busy bit was not reset. The selftest takes less than a second when successful.

Power-Up

The SMDC's SR:NDT status bit will be on after power-up or controller initialization (CR:INIT). This indicates that the dispatch table address and interrupt vector have not yet been sent by the host. The four bytes are sent with the CR command. After each byte is sent, the controller will briefly be busy (SR:BZ) while the byte is absorbed. The host waits until SR:BZ goes to zero after sending each byte. The read-packet-addresses command is then given. When all five commands have been given, SR:NDT is reset.

When a packet command is complete and the SMDC command register CR:EI bit is set, the SMC interrupts the host. The interrupt acknowledge vector is:

Interrupt Acknowledge Vector

```
 15 14 13 12 11 10  9  8  7  ... 0
-----------------------------------
| DRV |  ending status  | vector   |
-----------------------------------
```

NAME	BITS	CONTENTS
IV:VEC	0-7	vector from CR
IV:ES	8-13	packet command ending status
IV:DRV	14-15	drive number 0-3

Dispatch Table

The dispatch table provides the address and status of each of four packets. If fewer than four drives are present, the dispatch table entries corresponding to nonexistant drives should be present but set to zero. The dispatch table may not cross a 64 kilobyte boundry. Before the dispatch table address is sent to the controller, all packet status entries should be initialized to IDLE(0). All packets should also be initialized to 0.

DISPATCH TABLE

		15 14 13 12 11 10 9 8 6 5 4 3 2 1 0
00	PS0	packet status - drive 0
02	PS1	packet status - drive 1
04	PS2	packet status - drive 2
06	PS3	packet status - drive 3
08	PH0	packet address msh - drive 0
0A	PL0	packet address lsh - drive 0
0C	PH1	packet address msh - drive 1
0E	PL1	packet address lsh - drive 1
10	PH2	packet address msh - drive 2
12	PL2	packet address lsh - drive 2
14	PH3	packet address msh - drive 3
16	PL3	packet address lsh - drive 3

DT:PS VALUES: 0 = IDLE (set by host)
1 = GO (set by host)
2 = BUSY (set by SMDC)
4 = DONE (set by SMDC)

When an operation is to be initiated, the control information must be loaded into the appropriate packet and the corresponding dispatch table status word (DT:PS) set to GO. A command word is then issued with CR:WK (wakeup) and optionally CR:EI (enable interrupt). When the controller is idle, it interrogates the wakeup bit, sees it turned on, turns it off, and reads the dispatch table. Any dispatch table entries with packet status = GO cause the corresponding packets to be read into the SMDC's internal packet tables and DT:PS to be set to BUSY.

Internally, seeks are initiated on any drive with an active packet requiring a seek. When a seek is complete, or if no seek is required (e.g. select command), the command is performed and the packet in host memory updated with status, the dispatch table status set to DONE and IP posted. The host may be scanning SR:IP or waiting for an interrupt with CR:EI set.

Once IP is turned on or the interrupt acknowledged, the host reads SR and then issues CR:RI to reset IP and IUS regardless of whether the interrupts were enabled. Once IP is reset, the controller may interrupt again. The interrupt enable flag in the controller is reset by CR:DI, which may be issued with CR:RI if desired.

Ending status and interrupting drive number are made available in the high order byte of the interrupt vector returned by the controller during an interrupt acknowledge transaction. The drive number is also available in SR:DRV until CR:RI is issued.

Packet

```
          15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
          ------------------------------------------------
00 CM     ¦  ¦  ¦  ¦  ¦  ¦NR¦NE¦NO¦         CMD          ¦
          ------------------------------------------------
02 ST     ¦           0           ¦ 0 ¦0 ¦ ending status ¦
          ------------------------------------------------
04 SB     ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦RZ¦RT¦EC¦
          ------------------------------------------------
06 DS     ¦  SKE      ¦    SEL    ¦SM¦XM¦BZ¦RO¦FT¦SE¦OC¦RY¦
          ------------------------------------------------
08 CT     ¦           byte by sector count               ¦
          ------------------------------------------------
0A AH     ¦           dma address 23-16                  ¦
          ------------------------------------------------
0C AL     ¦           dma address 15-0                   ¦
          ------------------------------------------------
0E UN     ¦           unit                               ¦
          ------------------------------------------------
10 CY     ¦           cylinder                           ¦
          ------------------------------------------------
12 HD     ¦           head                               ¦
          ------------------------------------------------
14 VS     ¦FS¦NW¦          ¦ head bias & volume select   ¦
          ------------------------------------------------
16 SC     ¦           sector                             ¦
          ------------------------------------------------
18 OF     ¦  ¦  ¦  ¦  ¦  ¦  ¦  ¦SL¦SE¦  ¦  ¦  ¦O-¦O+¦  ¦  ¦
          ------------------------------------------------
1A-1E     ¦           reserved                           ¦
          ------------------------------------------------
          15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
```

CM defines the current operation by the CMD code and several command modifier flags.

CM:NR:	no retries; use OF
CM:NE:	no error correction
CM:NO:	no offsets during retries
CM:CMD:	0: nop 1: write RAM 2: read RAM 3: select drive 4: priority select 5: release 6: reset fault 7: position (seek/rezero) 8: write format 9: write long A: write B: reserved C: read format D: read long E: read F: size disk

COMMANDS

NOP is provided for diagnostic purposes; IP is posted immediately. The microcode revision is returned in CT.

WRITE RAM copies data from the specified host memory address to the data buffer space of the controller from location 0 up to local variable and packet storage. Provided for diagnostic purposes.

READ RAM copies data from controller memory, starting at location 0, to host memory. Provided for diagnostic purposes.

SELECT causes the specified drive to be selected and its status returned in DS.

PRIORITY SELECT is provided for dual access support.

RELEASE is provided for dual-access support.

RESET FAULT is provided for diagnostic and error recovery. A RESET FAULT command is issued to the specified drive.

POSITION - For diagnostic purposes, a seek is performed to the specified cylinder. If CY is -1 a rezero and reset-fault rather than seek occurs.

WRITE FORMAT initializes the ID and data fields of one or more sectors in a track. The host prepares the IDs contiguously in a buffer, four words per sector. CT is set to the number of sectors to be formatted; SC specifies the starting sector. Sector numbers may be arranged as desired. One or more sectors may be flagged as bad or as spare.

The sector IDs buffer prepared by the host contains the following 4 words for each sector to be formatted:

```
 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
-------------------------------------------------
¦ET¦EC¦EP¦                                      ¦
-------------------------------------------------
¦                  CYLINDER                     ¦
-------------------------------------------------
¦                  HEAD                         ¦
-------------------------------------------------
¦FL¦SP¦            SECTOR                       ¦
-------------------------------------------------
```

ID:ET flags last sector on a track

ID:EC Flags last sector on a cylinder

ID:EP flags last sector on a pack

ID:FL flags a sector as bad

ID:SP flags a sector as a spare

WRITE causes data to be written to a specific location on a specified drive. If a partial sector is written with WRITE or WRITE LONG, the contents of the remainder of the sector are undefined.

WRITE LONG is similar to WRITE except that the data ECC field is written from the four bytes following the data instead of being internally computed. This command is provided for diagnostic purposes.

READ FORMAT is the inverse of WRITE FORMAT, except that the ID ECC is read along with each ID. Six words are returned for each sector specified. Sectors are read in physical order starting with the sector specified by SC. ID contents are passed directly without error checking.

READ LONG causes sectors to be read along with their ECCs, which are appended to the end of each data block. Provided for diagnostic purposes to test error correction, the host writes a sector with a normal WRITE, reads it back with READ LONG, induces an error by switching one or more bits, writes the erroneous buffer with WRITE LONG, and then reads the sector with a normal READ command.

READ causes the specified sector(s) to be read and transferred to host memory. If necessary and not suppressed, error correction and retry procedures are invoked.

SIZE DISK causes the controller to examine the specified drive and return in the packet cylinder, head and sector words, the number of cylinders, heads, and sectors on the drive. The controller finds the size by scanning the IDs and looking for end-of-track, cylinder, and pack ID flags. This command takes several seconds. If the controller scans past cylinder 4096 without finding an end-of-pack bit, a pack overflow error is declared.

SECTOR ID FORMAT

```
 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
-------------------------------------------------
|ET|EC|EP|                                      |
-------------------------------------------------
|                   CYLINDER                    |
-------------------------------------------------
|                   HEAD                        |
-------------------------------------------------
|FL|SP|            SECTOR                       |
-------------------------------------------------
|                   ECC MSH                     |
-------------------------------------------------
|                   ECC LSH                     |
-------------------------------------------------
```

ID:ET flags the last sector on each track

ID:EC flags the last sector on each cylinder

ID:EP flags the last sector on the pack

ID:FL flagged (bad) sector

ID:SP spare (unused) sector

SECTOR FORMAT

Each sector contains an ID field and a data field separated by gaps. The gap sizes depend on the drive type , as does the number of sectors per track. One sector on each track is reserved to serve as a spare sector. If a bad sector is found, it is flagged and the spare is substituted. The SMDC finds sectors by searching for a matching ID (cylinder, head and sector). The formatter program can implement sector reassignment and sector interleaving when it formats a track by properly setting up the ID fields. The gap figures given are for SMD and may vary for other drives.

ITEM	BYTES	DESCRIPTION
SECTOR MARK	~1	marks beginning of each sector
GAP 1	8+16	head scatter + PLO lock time
ID SYNC	2	x'00F0'
ID	8	flags, cylinder, head, sector
ID ECC	4	ID error checking
GAP 2	1+16	write splice + PLO lock time
DATA SYNC	2	x'00F0'
DATA	512	
DATA ECC	4	data error detection and correction
GAP 3	1+8	pad + end of track

ST is returned by the SMDC when the operation is complete. It contains an error identification code. For some errors, it is necessary to refer to SB and DS below for details.

0: no error
1: initialization error (no DT or IV)
CR:CMD 4-7 and 1 must be given
2: sector overrun error
read or write gate on at end of sector
3: DMA memory error (ME) returned in SR:ES*

*** NOTE: When the SMDC detects a parity error, it does NOT rewrite the packet, nor does it update the packet status in the dispatch table to DONE. A ME code in SR:ES is the only indication of a DMA parity error.**

4: select error
no drive or multiple drives selected
5: CT (count) invalid
6: SMD dual access busy
7: multiple rezero error
rezero didn't correct SMD fault
8: SMD status error (see DS)
9: odd DMA address
10: pack overflow
multiple-sector read/write past end of pack
11: power failure detected during read/write
12: undefined operation (CM:CMD)
13: unrecovered data error
14: sector not found
15: write protect violation
timeout errors:
32: idle loop
33: waiting for IP and IUS to clear
34: waiting for DMA to complete
35: waiting for SMD on-cylinder
36: waiting for SMD servo clock
37: waiting for SMD data clock
38: waiting for sector/index mark
39: waiting for ID sync
40: waiting for data sync

SB contains status bits:

NAME	BIT	MEANING
SB:EC	0	error correction attempted
SB:RT	1	retry attempted
SB:RZ	2	rezero required to clear fault

DS contains status bits returned from the selected drive. The bit assignments for DS:SKE and DS:SEL correspond to ports 0-3, not drives 0-3. In order for the drives and ports to correspond, drive 0 is plugged into port 0, drive 1 into port 1, etc.

NAME	BIT	MEANING
DS:RY	0	selected drive ready
DS:OC	1	selected drive on cylinder
DS:SE	2	selected drive seek error
DS:FT	3	selected drive fault
DS:RO	4	selected drive read only
DS:BZ	5	selected drive busy (dual access)
DS:XM	6	index mark (internal use only)
DS:SM	7	sector mark (internal use only)
DS:SEL	8-11	ports 0-3 selected
DS:SKE	12-15	ports 0-3 seek end

CT is the number of bytes to be read or written in read/write operations or the number of sectors to be read or written in read/write format operations. If a CT byte count is odd, the command is rejected. A NOP command returns the microcode revision in CT.

AH and AL form the DMA starting address for operations involving a data transfer. If AL is odd the command is rejected.

UN is the SMD unit number 0-15 to be used for the operation.

CY is the starting cylinder number 0-n.

HD is the starting head number 0-n.

VS contains control information for multi-volume drives. For SMD it is set for 0. VS:HB is ORed into the head ID field during a format write.

VS:FS	force seek for volume select (CMD)
VS:NW	no wait for on-cylinder (MMD fixed heads)
VS:HB	head bias (volume select for CMD)

SC is the starting sector number 0-n.

OF specifies four bits of head offset and strobe timing data to be used if CM:NR (no retry) is set.

OF:0+	servo offset plus
OF:0-	servo offset minus
OF:SE	data strobe early
OF:SL	data strobe late

4.5.3.2 SMD Interface: The SMDC interface signals to the SMD drives consist of the address and control functions that are transferred on the Bit 0 through Bit 9 signals on the SMDC A. The information on these lines is indicated by the the individual CYLTAG,SELTAG, HEADTAG, and CONTAG tag lines when true. CYLTAG- When true low, the ten bus lines carry the cylinder address to the SMD. Since it is a direct addressing device, the SMDC need only place the new address on the lines and strobe the lines with CYLTAG-. The SMD must be On Cylinder before CYLTAG- is sent. The bus lines should be stable throughout the tag time.

HEADTAG- When true low, this signal is the head address selected by the bits on the bus line.

CONTAG- This signal enables the following individual control bits and must be true for the entire control operation.

1. Write Gate (Bit 0) — The Write Gate line enables the write driver.

2. Read Gate (Bit 1) — Enables the digital read data on the transmission lines.

3. Servo Offset + (Bit 2) — When true, the actuator is offset from the nominal On Cylinder position toward the spindle.

4. Servo Offset - (Bit 3) — When true, the actuator is offset from the nominal On Cylinder position away from the spindle.

5. Fault Clear (Bit 4) — A pulse is sent to the device to clear the fault if condition no longer exists.

6. AM Enable (Bit 5) — The Address Mark (AM) Enable together with Write Gate or Read Gate allows the writing or recovering of address marks.

7. RTZ (Bit 6) — A Return to Zero (RTZ) pulse when sent to the device will cause the actuator to seek track 0, reset the head register, and clear the Seek Error flip-flop.

8. Data Strobe Early (Bit 7) — When true, the device PLO Data Separator will strobe data at a time earlier than normal.

9. Data Strobe Late (Bit 8) — When true, the device PLO Data Separator will strobe the data at a time later than normal.

10. Bit 9 — Reserved

SECTOR - The sector mark is derived from the servo track. Timing integrity is maintained throughout seek operations. The number of sectors per revolution is switch selectable and determined by dibits/sector clocks.

FAULT - When true, a fault condition exists in the SMD. The following types of faults are detected: DC Voltage, Head Select, Write, Write or Read while Off Cylinder, and Write Gate during a Read operation. The line may be cleared by Control Select or Fault Clear with the CONTAG enable.

SKERR - When this line is true, a Seek Error has occurred. This line indicates the unit was unable to complete a move within 500 ms or that the carriage has moved to a position outside the recording field or received an illegal track address. A Return to Zero (RTZ) clears the Seek Error and returns the heads to cylinder zero while enabling the ONCYL signal to the controller.

ONCYL - The On Cylinder status indicates the servo has positioned the heads over a track. The status is cleared with any seek instruction causing carriage movement or zero-track seek.

INDEX - The signal occcurs once per revolution and its leading edge is considered the leading edge of sector zero. Timing integrity is retained throughout seek operations for all SMDs.

READY - When true, with the device selected, this line indicates: the SMD is up to speed, the heads are positioned over the recording tracks, and no fault condition exists within the SMD.

OPENCABLE - The open cable detector circuit disables the interface when the SMD A interface cable is disconnected or controller power is lost.

SELTAG - Using SEL1 through SEL8, the Unit Select Tag gates the desired binary logic number of the unit selected into the logic number compare circuit.

SEL1,SEL2,SEL4,SEL8 - These four lines are binary coded to select the logical number of one of sixteen devices. The individual unit number (0 - 15) is selectable by a logic plug on the units operator panel.

SELECTED- (1.- 4.) - When the four (SEL1, SEL2,SEL4,SEL8) select and compare the logical number with the selectable individual unit and the SELTAG is received, the SELECTED- line becomes true and transmitted to the controller on the SMD B cable.

WPROT - Enabling the Write Protect function inhibits the writer under all conditions, illuminates an LED, and sends the WPROT- signal to the SMDC A controller. The Write

Protect function is enabled by a switch on the operator panel.

SEEKEND - Seek End is a combination of On Cylinder or Seek Error indicating that a seek operation has terminated.

HOLD/PICK - Applying ground to the Pick and Hold lines enables the first SMD in a Power-Up sequence. Once up to speed, the PICK- signal is transferred to the next active SMD.

BUSY - If the device is already reserved and/or selected, a BUSY is issued to the controller on the SMD A cable and SELECTED- issued on the SMD B cable.

WRITEDATA (0-3) - This line carries data (NRZ FORM) which is to be recorded on the disk pack.

READDATA (0-3) - This line transmits the recovered data to the controller in the NRZ form. SERVOCLK- (0-3) The Servo Clock is a phased-locked 9.677 MHz clock generated from dibits on the SMD and used to generate write data. This signal is available at all times to the controller.

READCLK (0-3) - The Read Clock defines the beginning of a data cell. It is an internally derived clock signal and is synchronous with the detected data. It is transmitted continuously to the controller.

WRITECLK (0-3) - The Write Clock signal is synchronized to the NRZ data. The Write Clock is the received Servo Clock which is retransmitted by the controller during a write operation.

4.5.4. Cartridge Tape Controller: The tape controller is the intelligent interface between the System 8000 CPU and the tape drives (also called decks). The controller derives its intelligence from its on-board Z80B microprocessor. Figure 4-7 shows the basic relationship between the controller and both the System 8000 bus (ZBI) and the tape decks. Information flows between the controller and the CPU over the ZBI (mating connectors P1/J14). The flow of information between the controller and a selected tape drive is through mating connectors P2/J24 of slot 4 only of the system backplane. Figure 4-8 shows the allocation of the I/O space of the controller.

4.5.4.1 ZBI Interface: The controller and the host communicate through eight 16-bit (word) read/write registers. These registers appear in the controller's I/O space at addresses 40H through 4EH. (The H stands for hexadecimal.) Table 4-18 lists these registers and their assignments. On-board jumpers provide a means of changing the ZBI address of the controller board. These jumpers are listed in Table 4-19. The bit assignments of the upper byte of the interrupt vector are listed in Table 4-20. The commands that the host sends to the controller are listed in Table 4-21, and Table 4-22 defines the bits in the status register. Table 4-23 lists the bits in the Master Interrupt Control (MIC) register. All of the possible error conditions are listed in Appendix C.

4.5.4.2 Drive Interface: The tape controller sends commands to the tape drive to control its operation. These commands set the drive address, track address, and motion controls. Table 4-24 lists the commands that the controller sends to the drive. The drive responds to the controller by sending information back to the controller. Table 4-25 lists the information that the drive sends to the controller.

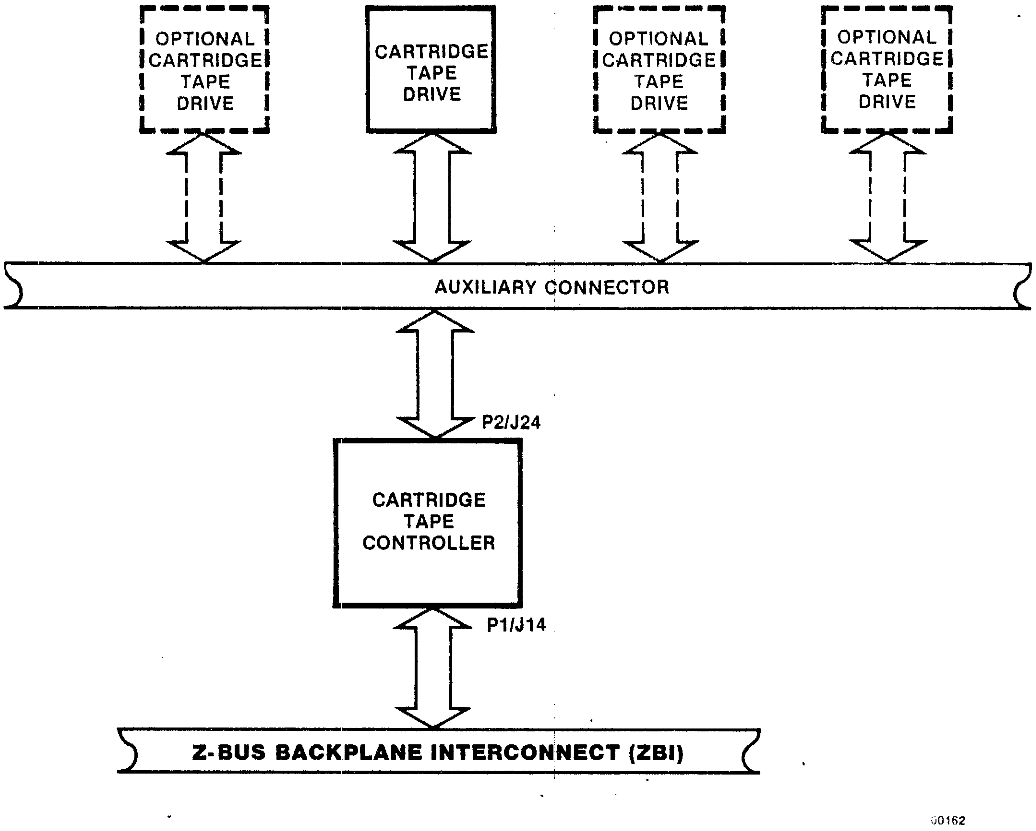

OPTIONAL CARTRIDGE TAPE DRIVE
CARTRIDGE TAPE DRIVE
OPTIONAL CARTRIDGE TAPE DRIVE
OPTIONAL CARTRIDGE TAPE DRIVE
AUXILIARY CONNECTOR
P2/J24
CARTRIDGE TAPE CONTROLLER
P1/J14
Z-BUS BACKPLANE INTERCONNECT (ZBI)
00162

Figure 4-7 Cartridge Tape Controller Functional Relationships

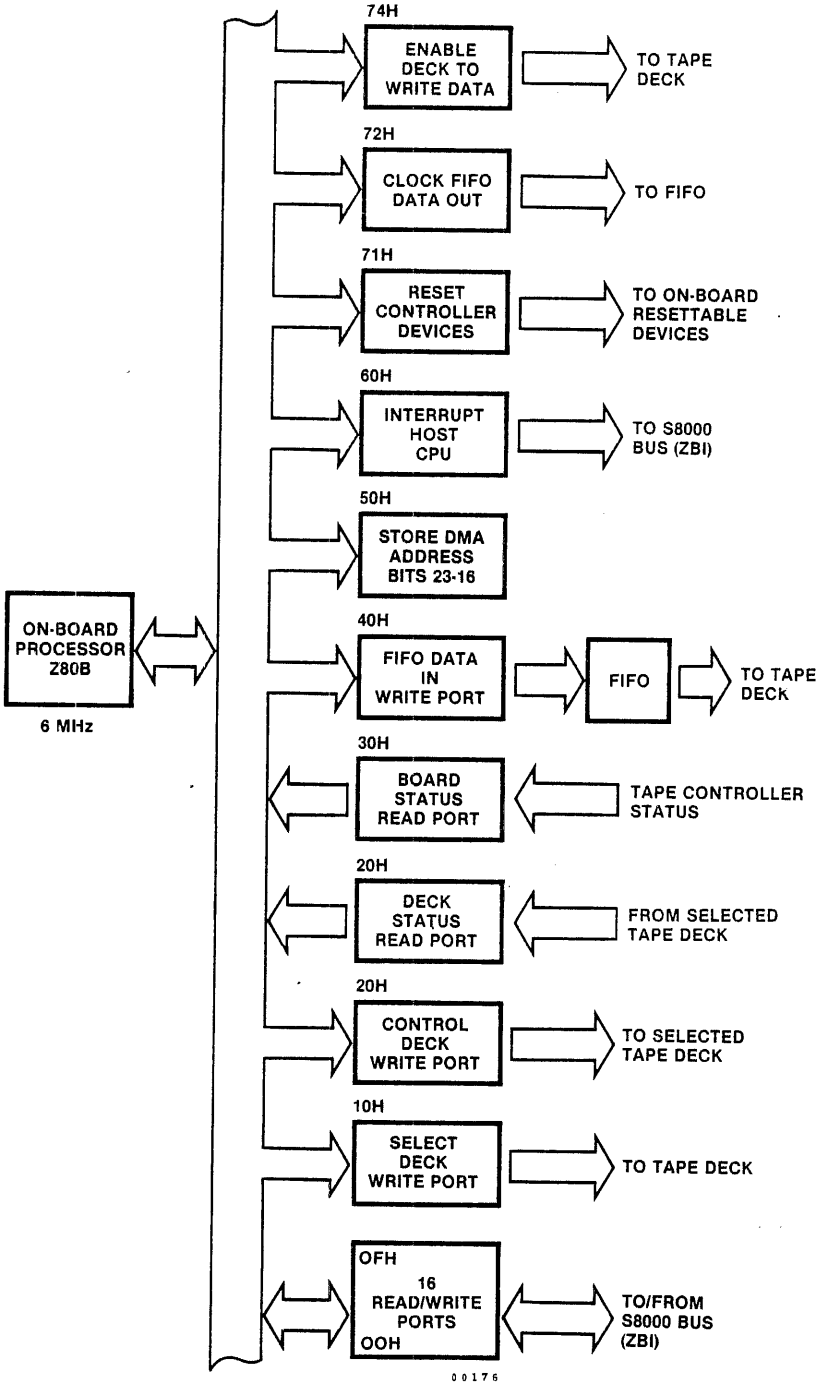


Figure 4-8 Cartridge Tape Controller I/O Address Space

Table 4-18 ZBI Tape Controller Interface Registers

ADDRESS	REGISTER	DESCRIPTION
40H	Interrupt Vector	The low-order byte contains the interrupt vector that host CPU writes to the controller. The high-order byte contains status information that the controller sends to the host.
42H	Command	The host sends commands to this register. The controller accepts only valid commands.
44H	Low DMA Start Address	The host sends the low word of the DMA starting address in this register. Bit 0 of this byte must be a 0 so that the address starts on a word boundary.
46H	High DMA Start Address	This register contains the high-order byte (bits 16 to 23) of the DMA start address.
48H	DMA Length	This register contains the length of the DMA transfer. This value must be less than 32 kilobytes (1k=1024).
4AH	Status	The controller stores information about the tape drive and controller in this register. The host reads this information. Another table defines the bits.

ADDRESS	REGISTER	DESCRIPTION
4CH	Status 1	Bits 0 to 3 define the number of retries for a read or write command. Bits 8 to 15 define the number of blocks or files that have been skipped during a skip command.
4EH	Interrupt Control	This is the master interrupt control register. Another table lists the bit definitions.

Table 4-19 Tape Controller Jumper Selection for Base Address

JUMPER GROUP (E)	PURPOSE	CONNECTION AND RESULT
E1, E2, E3	Set to expect either a low or high bit SAD15	E1 to E2 (normal): bit SAD15 low E2 to E3: bit SAD15 high
E4, E5, E6	Set to expect either a low or high bit SAD14	E4 to E5 (normal): bit SAD14 low E5 to E6: bit SAD14 high
E7, E8, E9	Set to expect either a low or high bit SAD13	E7 to E8 (normal): bit SAD13 low E8 to E9: bit SAD13 high
E10,E11,E12	Set to expect either a low or high bit SAD12	E10 to E11 (normal): bit SAD12 low E11 to E12: bit SAD12 high
E13,E14,E15	Set to expect either a low or high bit SAD11	E13 to E14 (normal): bit SAD11 low E14 to E15: bit SAD11 high

JUMPER GROUP (E)	PURPOSE	CONNECTION AND RESULT
E16,E17,E18	Set to expect either a low or high bit SAD10	E16 to E17 (normal): bit SAD10 low E17 to E18: bit SAD10 high
E19,E20,E21	Set to expect either a low or high bit SAD09	E19 to E20 (normal): bit 09 low E20 to E21: bit 09 high
E22,E23,E24	Set to expect either a low or high bit SAD08	E22 to E23 (normal): bit 08 low E23 to E24: bit 08 high
E25,E26,E27	Set to expect either a low or high bit SAD07	E25 to E26 (normal): bit 07 low E26 to E27: bit 07 high
E28,E29,E30	Set to expect either a low or high bit SAD06	E28 to E29: bit 06 low E29 to E30 (normal): bit 06 h
E31,E32,E33	Set to expect either a low or high bit SAD05	E31 to E32 (normal): bit 05 low E32 to E33: bit 05 high
E34,E35,E36	Set to expect either a low or high bit SAD04	E34 to E35 (normal): bit SAD04 low E35 to E36: bit SAD04 high
E37,E38,E39	Set to enable or disable the Controller board	E37 to E38 (normal): enables board to receive address from ZBI bus E38 to E39: disenable board

NOTE

The normal connection is wired on the board. Changing the normal connection to a new base address requires the cutting of traces and adding jumpers.

Table 4-20 Tape Interrupt Vector, Bit Definitions

BIT	NAME	MEANING
Bit 8	INTV	The current operation requires invention.
Bit 9	BUSY	The controller is busy executing the last command.
Bit 10	CMDREJ	The controller rejects the current command.
Bit 11	DATERR	An uncorrectable data error has occurred.
Bit 12	SKNDNE	The current skip operation has not been completed.
Bit 13	OVERFL	A buffer overflow has occurred.
Bit 14	FFERR	A FIFO error has occurred.
Bit 15	Not Used	

Table 4-21 Host-Tape Controller Commands

CODE (HEX)	NAME	DEFINITION
0000	NOP	The controller loops while waiting for a command from the host.
0001	READ	The controller reads one block. If necessary, controller backspaces and retries.
0002	WRITE	The controller writes one block. If necessary the controller back-spaces, erases three inches of tape and retries.
0003	REWIND	Controller rewinds tape to it logical beginning (6 inches past the load point).
nn04	SKBF	Controller skips nn blocks forward (nn is any value from 0 to 255).
nn05	SKBR	Controller skips nn blocks in reverse (nn=0 to 255).

CODE (HEX)	NAME	DEFINITION
nn06	SKFF	Controller skips nn files forward (nn=0 to 255). A file is a group of blocks followed by a file mark.
nn07	SKFR	Controller skips nn files in reverse (nn=0 to 255).
0008	WFM	Controller writes file mark on tape.
0009	LOAD	Controller moves the tape from the phyical load point to logical load point: beginning of tape, track 0 selected.
000A	UNLD	Controller moves the tape to the physical load point.
0n0B	SEL	Controller selects a new drive: address is n, a value from 0 to 3.
0n0C	MRTRY	This sets the maximum number of retries the controller is allowed for reads and writes. At power on, the default is 10 retries: n=0 to 15.
0n0E	STRK	Controller rewinds the tape and selects new track: n=0 to 3.
0n0F	MODE	Controller changes to mode n (n=0,1). In mode 1, tape is divided into four separate tracks. The logical beginning of tape is at the beginning of each track. Logical end of tape is at the end of each track. REWIND moves the tape to the start of each track; skips do not carry from track to track. In mode 0, tape is one long track. Logical beginning of tape is at the start of track 0 and the logical end of tape is at the end of track 3. REWIND moves the tape to the start of track 0; skips carry from track to track. At power on, the default is mode 0.
0010	DIAG	The controller executes a diagnostic test, checking the ROM, the FIFO, and the host interface ports.

Table 4-22 Status Register, Bit Definitions

BIT	NAME	DEFINITION
Bit 0	NOTAP	No tape cartridge in drive
Bit 1	FMDET	File mark detected during read or or skip blocks
Bit 2	HWERR	Hardware error
Bit 3	INVAL	Invalid command
Bit 4	INAP	Inappropriate command
Bit 5	(Not Used)	
Bit 6	BPARM	Bad DMA parameters
Bit 7	BLKTAP	Blank tape
Bit 8	PROT	Tape cartridge write protected
Bit 9	LBOT	Tape at logical beginning of tape
Bit 10	LEOT	Tape at logical end of tape
Bit 11	RTRYAT	One or more retries attempted
Bit 12	UNIT0	Tape drive address bit 0
Bit 13	UNIT1	Tape drive address bit 1
Bit 14	TRK0	Track address bit 0
Bit 15	TRK1	Track address bit 1

Table 4-23 Master Interrupt Control Register, Bit Definitions

BIT	NAME	DEFINITION
Bit 0	MIE	Master interrupt enabled
Bit 1	IE	Interrupt enabled
Bit 2	DLC	Disable lower chain
Bit 3		Not defined
Bit 4		Not defined
Bit 5		Not defined
Bit 6	IUS	Interrupt under service
Bit 7	IP	Interrupt pending

Table 4-24 Tape Controller to Drive Interface Signals

SIGNAL	DEFINITION
RWD\	Rewinds the tape
REV\	Moves the tape backwards
FWD\	Moves the tape forward
WEN\	Enables writing and erasing on the tape
WDE\	Enables sending of write-data strobes and the writing of data on the tape
WNZ\	Serial data to be written to the tape drive
TR1\,TR2\	Select tracks during read, write, and erase track operation, according to following code:

SIGNAL	DEFINITION			
	Track Number	TR2\	TR1\	
	0	High	Low	
	1	Low	High	
	2	Low	Low	
	3	High	High	
SLG\	Allows selection of tape drive designated by unit select codes: SL4\, SL2\, SL1\.			
SL4\,SL2\,SL1\	These form the unit (drive) select code listed below:			
	Drive Selected	SL4\	SL2\	SL1\
	0	H	H	L
	1	H	L	H
	2	H	L	L
	3	L	H	H

4.5.5. Tape Controller Operation: To start a tape operation, the host CPU reads the high-order byte of the controller's interrupt vector to see if BUSY is set. If the busy bit is set, the controller is still executing the last command. If the controller is not busy, the host initializes the interface registers and then writes a non-zero command in the controller's command register. The flowchart in Figure 4-9 shows the steps taken by the host.

The controller normally loops while it waits for a non-zero command from the host. When the controller receives a command, it resets the interrupt-pending bit (IP) in the master interrupt control register. The controller sets the busy bit in the upper byte of the interrupt vector register to inform the host that it is busy with the current command. However, before the controller processes the command, it checks the validity of the command.

After processing the command, the controller resets the command register and sets the IP bit. Next, the controller sends an interrupt to the host and waits for an acknowledgement, the controller sends its interrupt vector in response to the acknowledgement. An upper byte of zero means that no errors occurred. The controller also sets the interrupt-under-service bit (IUS).

Table 4-25 Tape Drive to Controller Interface Signals

SIGNAL	DEFINITION
SLD\	Selected drive informs controller that the drive has received its unit address.
RDY\	Tape cartridge is installed.
WND\	Selected drive has received a write enable signal.
FLG\	Rewind completed.
LPS\	Load point sensed.
FUP\	Installed tape cartridge is unprotected.
BSY\	The drive is doing one of the following: 1) Automatic rewind after cartridge is installed. 2) Executing rewind, forward, or reverse command.
EWS\	The upper early warning hole in forward direction has been reached.
WDS\	The drive is examining the state of WNZ\ signal.

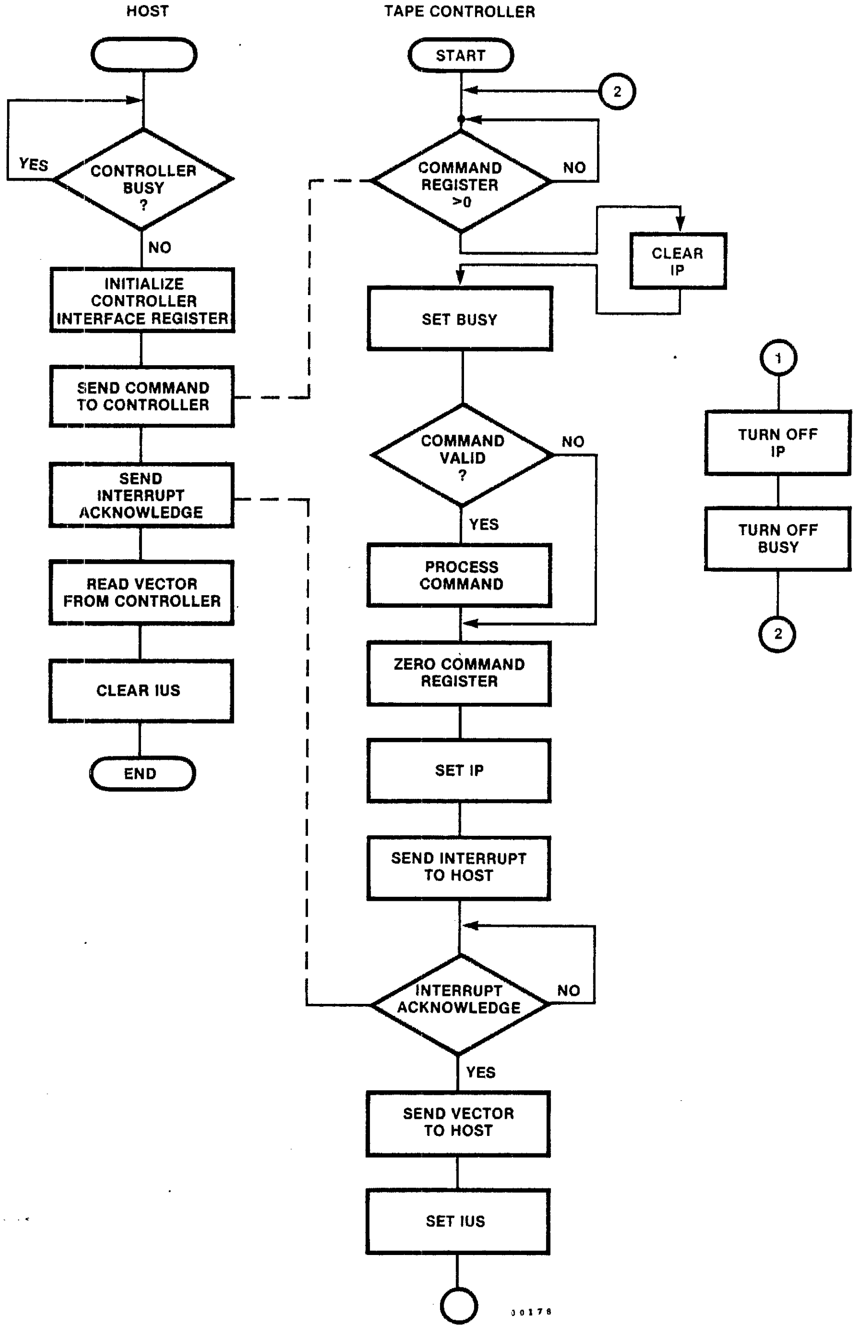


Figure 4-9 Cartridge Tape Controller, Command Processing

The host reads the vector from the controller and clears the controller's IUS bit. This action ends the interrupt subroutine for the host. The controller clears both the IP and the BUSY bits and loops while waiting for a new command.

4.5.6. Memory Subsystem Controller: The Memory Subsystem controller controls up to 4 megabytes of dynamic read-write memory. The controller can perform read-write operations with byte (8-bit), word (16-bit), and long-word (32-bit) quantities. Figure 4-10 shows the functional relationship between the controller and the ZBI and the memory modules that it controls.

The controller stores data as 32-bit long words and adds to this, seven bits of information for use by the error-detection and correction circuits. Figure 4-11 shows the overall organization of memory. During memory transactions, the controller accepts a 24-bit address and the B/W\ and W/LW\ control signals over the ZBI. The two least-significant address bits and the B/W and W/LW\ signals select one of the four bytes at the location to be read or modified.

4.5.5.1 Byte Translation: During transactions involving bytes, the controller receives a data byte from ZBI lines AD0 through AD7. Figure 4-12 shows the flow from a register, through the controller, and to memory. The controller places byte A, the first byte, in location 0 (bits 31 to 24). For this transaction, the bus controller sets both the B/W\ and W/LW\ control lines high to identify the current transfer as a byte transfer. The bus controller also places low levels (0) on ZBI address lines AD00 and AD01 to tell the memory controller to place byte A in bit positions 31 to 24.

Next, the memory controller places byte B, C, and D in the succeeding memory locations to fill up the current double word of memory. The memory controller places the next, the fifth, byte in the first location of the next double word of memory, bits 31 through 24 of byte 4 (not shown).

4.5.5.2 Word Translation: For word (16-bit) translations, the bus controller sets line B/W\ low and W/LW\ high. This code tells the memory controller that the current transfer is a 16-bit transfer. The bus controller places the 24-bit address on the ZBI to point to the double-word location in memory where the memory controller is to place the current transfer. Only the 22 most-significant bits of the address point to the memory location. The memory controller ignores the least-significant address bit (AD00); the state of address bit AD01 tells the memory controller to store the current word (16-bits) in either the upper half or lower half of the double-word space in memory. Figure 4-13 shows the path of two 16-bit words, E and F. The controller stores word E in the word 0 location (bits 31 to 16) and stores word F in the word 1 location (bits 15 to 0)). All word-size transfers must occur only on a word boundary.

4.5.5.3 **Long-Word Translation:** Long words (Figure 4-14 contain four bytes (32 bits) and occupy the entire width of the ZBI and memory controller, and end up in a location in memory. The 22 most-significant address bits point to the location; the controller ignores the two least-significant bits, AD00 and AD01.

4.6. System Reset

A system reset can be generated from a power-up circuit on the CPU board or by the RESET button on the front panel of the System 8000. The power-up reset circuit makes certain that all functions in the system start in an orderly manner. The RESET button on the front panel is disabled by the ON/LOCK keylock switch when in the LOCK position.

When the system is reset, the following actions occur:

1) All eight serial I/O channels are disabled. All SIO control registers must be initialized.

2) All CTC channels stop counting and all interrupt-enable bits are cleared. CTC control registers must be initialized.

3) Parallel data is inhibited.

4.7. Non-Maskable Interrupts (NMI)

Non-maskable interrupts are typically reserved for external events that require immediate attention. They cannot be disabled (masked) by software. The System 8000 provides three sources of NMI: manual NMI, power-fail NMI, and a double-bit non-correctable ECC memory error NMI.

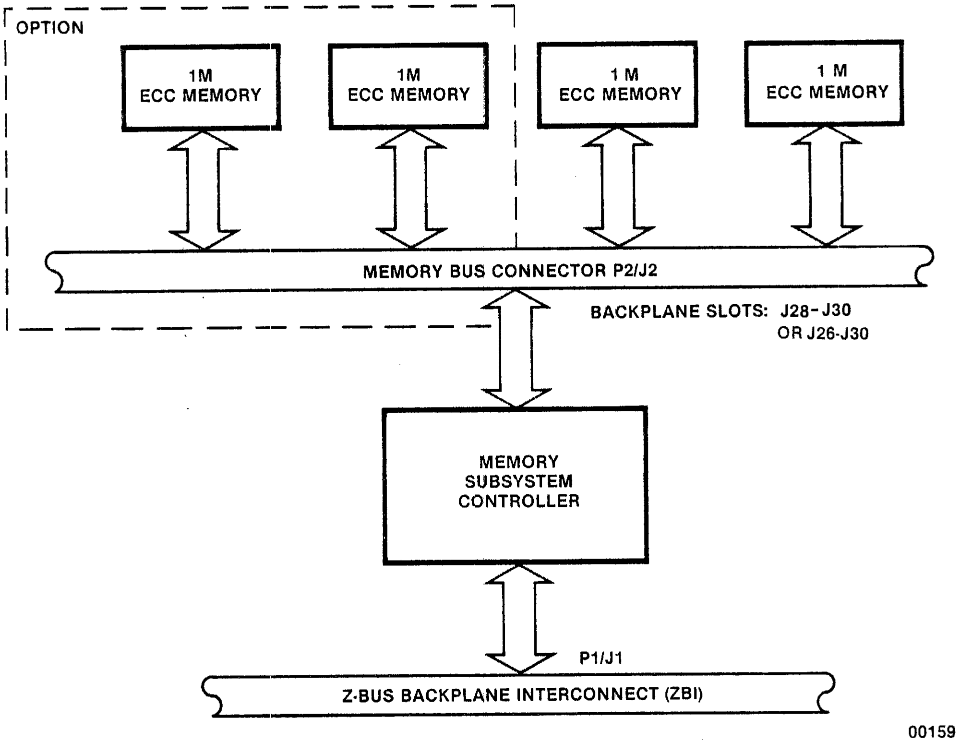

OPTION
1M ECC MEMORY
1M ECC MEMORY
1 M ECC MEMORY
1 M ECC MEMORY
MEMORY BUS CONNECTOR P2/J2
BACKPLANE SLOTS: J28-J30 OR J26-J30
MEMORY SUBSYSTEM CONTROLLER
P1/J1
Z-BUS BACKPLANE INTERCONNECT (ZBI)
00159

Figure 4-10 Memory Subsystem Controller, Functional Relationships

BIT 38 BIT 31 BIT 23 BIT 15 BIT 07 BIT 00

CHECK BITS	BYTE 0	BYTE 1	BYTE 2	BYTE 3
CHECK BITS	BYTE 4	BYTE 5	BYTE 6	BYTE 7
CHECK BITS	BYTE 8	BYTE 9	BYTE 10	BYTE 11
(ADDITIONAL BYTES)				
CHECK BITS	BYTE N-3	BYTE N-2	BYTE N-1	BYTE N

BIT 38 BIT 31 BIT 23 BIT 15 BIT 07 BIT 00

00170

Figure 4-11 Memory Organization

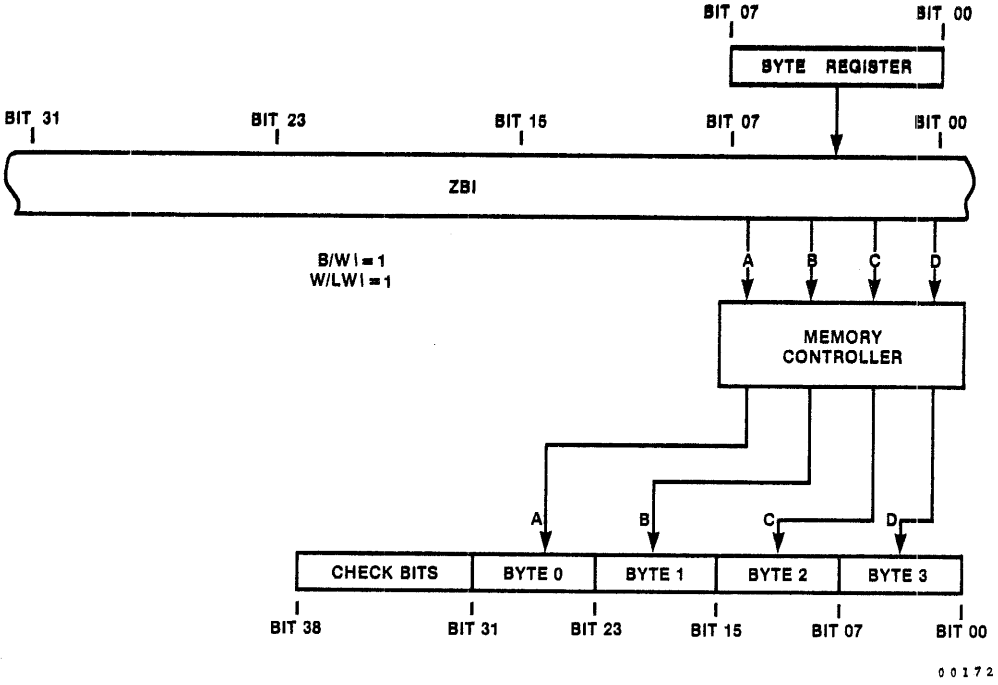


Figure 4-12 Byte Translation

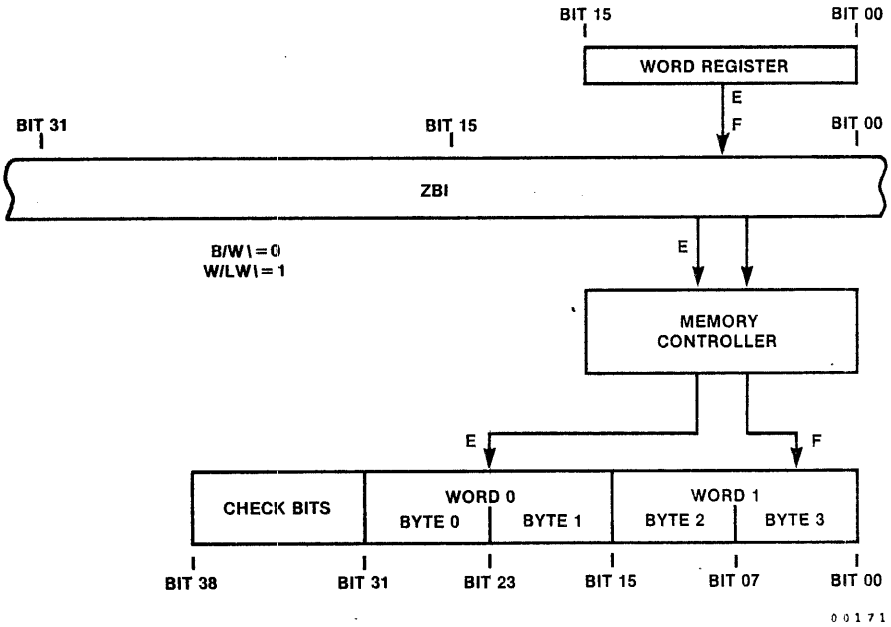


Figure 4-13 Word Translation

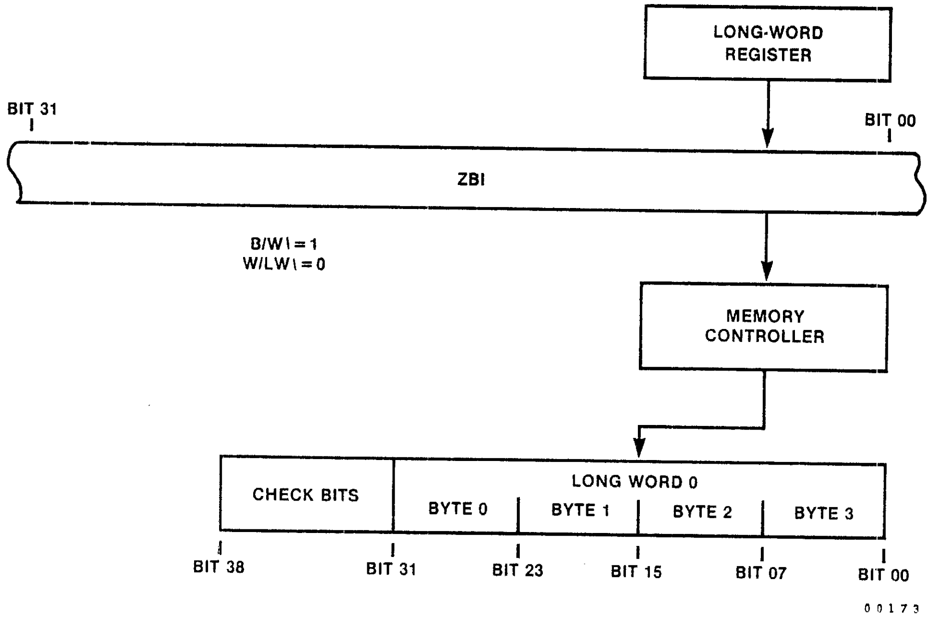


Figure 4-14 Long-Word Translation

4.7.1. **Manual NMI:** A manual NMI can be generated in the System 8000 from the START pushbutton on the front panel. The NMI can be disabled by placing the ON/LOCK keylock switch in the LOCK position. When the START pushbutton is pressed immediately after a manual or power-up reset, the system power-up diagnostics (SPUD) firmware is invoked. At the conclusion of the diagnostic, if no errors have been recorded, the message "POWER UP DIAGNOSTICS COMPLETE" appears on the console screen and the ZEUS Operating System is automatically booted.

4.7.2. **Power-Fail NMI:** A power-fail NMI will be sent to the Z8001 CPU when the system power supply detects a decrease in line voltage signifying a potential power failure. After receiving a power-fail NMI, the system has approximately 2 msec to power-down. When a power-fail NMI identifier is read by the operating system, the operating system generates a software reset which in turn, becomes a hardware system reset. Therefore, the Winchester disk drives are protected from crashing during a power failure.

4.7.3. ECC Memory Error NMI: A non-maskable interrupt will be sent to the CPU when a double-bit non-correctable ECC memory error is flagged by the ECC Controller, and when the CLEAR ECC ERROR-bit of the System Configuration Register (SCR) is set to enable an ECC NMI. The error bit is initially cleared at the SCR during a system reset or power-up.

4.7.4. NMI Identifier: When a non-maskable interrupt is detected by the CPU, the subsequent initial instruction fetch cycle is initiated, but aborted. The program counter (PC) is not updated, but the system stack pointer is decremented. The next CPU machine cycle is the interrupt acknowledge cycle. This cycle acknowledges the interrupt and reads a 16-bit IDENTIFIER word (all 16 bits can represent peripheral device status and ID information) from the device that generated the interrupt (in this case, an NMI source). This identifier word, along with the program status information, is stored on the system stack and new status information is loaded into the PC and FCW (flag and control word register).

When the CPU generates an NMI acknowledge status code (0101), the source of the NMI will be either a manual, power-fail, or ECC memory error NMI. Dedicated logic on the CPU board enables a 4-bit error buffer to place a 4-bit NMI IDENTIFIER on mulitplexed address/data lines AD0 to AD3, as follows:

AD3	AD2	AD1	AD0	SOURCE
0	0	0	1	Manual NMI
0	0	1	0	Power-fail NMI
0	1	0	0	ECC Memory Error NMI

Bits AD4 to AD15 are "don't care" bits in an NMI acknowledge identifier word if the NMI source is any one of the three listed sources of NMI in the System 8000. If the NMI source is external to the CPU and not one of the three listed sources of the NMI, the NMI buffer will remain off and the 16-bit identifier word will be read from the system bus.

4.8. Vectored Interrupts

When the CPU acknowledges an interrupt from a peripheral device, it reads a 16-bit identifer word to identify the source of the interrupt. In vectored interrupts, the identifer is also used by the CPU as a pointer to select a particular interrupt service routine associated with the peripheral that was the source of the interrupt.

The System 8000 CPU can configure all PIO (Parallel I/O Controller), CTC (Counter/Timer Circuit), and SIO (Serial I/O Controller) peripheral devices for vectored interrupt operation. The interrupt vectors associated with these peripherals can be loaded at any time since these devices are initialized with their interrupts disabled (masked).

NOTE

> It is recommended that vectored interrupts be disabled by the CPU until all peripherals on the System 8000 CPU have been properly initialized because of the vectored interrupt daisy-chain.

Since there are several Z80B peripheral devices on the CPU board, an interrupt daisy-chain is used to prioritize the devices and to accelerate their interrupt request time. Each Z80B device contains two lines that function as links in the daisy-chain: IEI, Interrupt Enable In (INPUT, active high) and IEO, Interrupt Enable Out (OUTPUT, active high).

The Z80B peripherals on the CPU board are prioritized in a daisy-chain as indicated in Table 4-26. Figure 4-15 illustrates the daisy-chain configuration.

Table 4-26 Device Priority Scheme

PRIORITY	PERIPHERAL DEVICE	FUNCTION
1	CTC 0	Single Step, also generates BAUD0, BAUD1, BAUD2
2	CTC 1	Generates BAUD3, BAUD4, BAUD5
3	CTC 2	Generates BAUD6, BAUD7, and the Real Time Clock
4	SIO 0	Serial Channels 0, 1
5	SIO 1	Serial Channels 2, 3
6	SIO 2	Serial Channels 4, 5
7	SIO 3	Serial Channels 6, 7
8	PIO 0	Line Printer Interface
9	(OFF CPU BOARD I/O)	(Secondary Serial Board)
10	Winchester Disk Controller or SMD Controller	
11	Tape Controller	

Vectored interrupts from any of the peripherals, 1 to 8 in Table 4-26, automatically disables interrupts from lower priority peripheral devices in the chain. Off-board devices have the lowest priority in the chain.

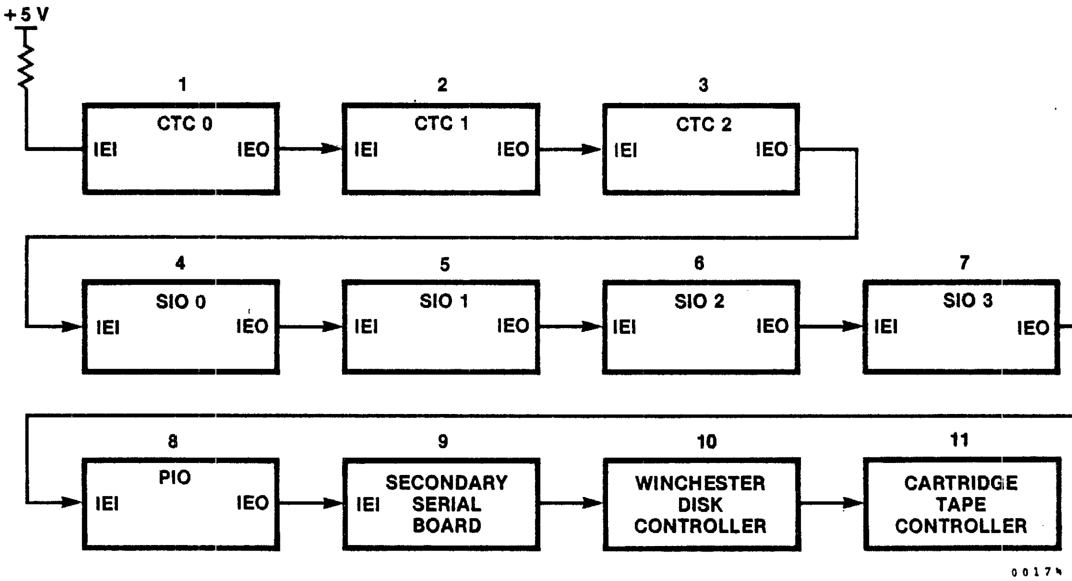


Figure 4-15 Interrupt Priority Connections

4.9. Memory Management Unit (MMU)

The Z8010 Memory Management Unit manages the 16M byte main memory address space of the System 8000 CPU. The MMU also provides the following features:

1. Flexible and efficient allocation of main memory resources during the execution of both operating system and user tasks.

2. Support multiple, independent tasks that share access to common resources.

3. Protection from unauthorized or unintentional access to data or other memory resources.

4. Detection of incorrect use of memory by an executing task.

5. Partitioning of main memory resources to separate user functions from system functions.

4.9.1. MMU Operation (Non-Segmented): A non-segmented operating system runs in segment 0, using Segment Descriptor Register (SDR) number 0 of memory management units M1, M2 and M3 for code, data, and stack areas, respectively. MMU

M1 is used for translating program memory references while M2 and M3 translate all other memory references. The selection between M2 and M3 is based on a comparison between the logical address and the contents of the System Break Register (SBR), a program addressable hardware register. Logical addresses with values lower than the SBR are treated as data addresses and are directed to MMU M2 (data). Logical addresses that are equal to or greater than the SBR, are treated as stack addresses and are directed to MMU M3 (stack).

A non-segmented user program runs in segment 63; however, the hardware will allow any segment between 2 and 63 to be used. As in the non-segmented operating system, Segment Descriptor Register number 63 in M1, M2 and M3 is used to provide separate code, data, and stack areas, respectively. The only difference is that the Normal Break Register (NBR), also a program addressable hardware register, is used to distinguish between code, data, and stack references instead of the System Break Register.

4.9.2. MMU Operation (Segmented): A segmented operating system uses MMU M1 (code) to provide an address space consisting of up to 63 segments, e.g., segments 0 to 62. Segment 63 is used to run non-segmented user programs. Since the attribute flags in the segment descriptor registers of MMU M3 (stack) are used to configure different segments, no separation between code, data or stack references is required.

A segmented user program uses M2 and M3 to provide an address space consisting of 124 or 128 segments, without separating code, data, and stack areas. If the operating system is non-segmented, then segment numbers 0,1 and 64,65 are reserved for the operating system since it requires SDR number 0 of M2 and M3. In a segmented operating system, all 128 segments are usable.

4.9.3. MMU Configurations: The MMU configuration is set by hardware jumpers on the CPU board and by the operating system software. The jumpers are used to configure the MMU select logic for either a segmented or nonsegmented operating system. The operating system software configures the System Configuration Register (SCR) for running segmented or nonsegmented user processes (programs). Refer to Table 4-5 for possible jumper configurations.

4.9.4. Break Registers: Two 8-bit hardware registers, the System Break Register (SBR) and the Normal Break Register (NBR), are accessable as I/O ports on the CPU board. During any memory reference, the 16-bit logical address offset generated by the CPU is compared to the break value given by the contents of either the SBR or the NBR. The SBR is referenced for the break value if the segment number is zero or one, and the NBR if it is otherwise. If the MMU configuration specifies separation of code, data, and stack areas, and the CPU status indicates a nonprogram reference (status 10xx), then the result of this comparison selects between data and stack references. If the logical address offset is less than the break value, the current reference is for data (MMU M2 is enabled); otherwise, it is a stack reference (MMU M3 is enabled). The following paragraphs describe the possible configurations of operating systems (OS) and user programs (USER).

Nonsegmented OS, Nonsegmented USER

This configuration is intended to run operating systems in memory segment 0 and user programs in any segment, 2 to 63. (Segment 63 is recommended for running user programs.)

For operating systems executing in this configuration, MMU M1 (code) is enabled for program references indicated by a CPU status code 11xx, an instruction space access. For memory references other than program references, the logical address offset generated by the CPU is compared against the contents of the SBR, if the segment number is zero or one; however, if the segment number is 2 - 63, the comparison is made against the contents of the NBR. If the result of the comparison is less than zero, the select logic enables MMU M2 (data); otherwise, MMU M3 (stack) is enabled. Additionally, logic on the CPU board detects memory references made to segment 0,1 while the CPU is in normal operation mode. This logic generates a segment trap violation to the Z8001 CPU, disables the three MMUs, and asserts a suppress signal that prohibits main memory references.

If the operating system, or any part thereof, executes in segment mode, the separation of code, data, and stack spaces still applies.

NOTE

> The separation between data and stack spaces is based on the contents of the SBR for segment 0, and the NBR for references to all other segments.

A segmented user program can run in this configuration, although the Nonsegmented OS, Segmented USER configuration is intended for that purpose. Such a user program has a potential address space of 62 code and data segments.

Nonsegmented OS, Segmented USER

This configuration is intended to run exactly as the previous configuration, provided that the CPU is in system mode and the operating system is running in memory segment 0. In addition, code, data, and stack references are directed to M1, M2, and M3 respectively, and the contents of the SBR are used to select between data and stack references. However, if the CPU is in normal mode, MMU M2 is enabled for segment numbers 2 to 63, and MMU M3 is enabled for segment numbers 65 to 127. If a memory reference is made, a segment trap violation is generated and the three MMUs are disabled. Also, the suppress signal is generated by the CPU to protect the system data and stack areas from being accessed by the user program. In system mode, if the segment number of a user segment is generated (segments 1 to 63, or 66 to 127), the address translation is the same as in normal mode. Separation of code, data, and stack spaces are deactivated; MMU M2 is enabled for segments 2 to 63; and MMU M3 is enabled for segments 66 to 127. This allows the operating system to directly access any user segment.

4.9.5. **System Access to User Space:** To access a user segment, the operating system can use a free segment slot and set its Segment Descriptor Register to point at the same memory area as the target user segment's SDR.

A nonsegmented operating system running a nonsegmented user program can directly access the user data and stack areas by switching to system mode and using the user segment number. To access the user code segment, one of the unused segment slots is set to point at the code segment; for example, number 62. The SDRs for this segment slot in M2 and M3 are both set to point at the code segment, negating the contents of the NBR.

A nonsegmented operating system running a segmented user program can directly access any protion of the user space by switching directly into segmented mode.

4.9.6. System Segments and Protection: Logic on the CPU board partitions segments into system segments (logical segments 0, 1, 64 and 65), and user segments (logical segments 2 to 63 and 66 to 127). Any reference to a system segment always enables the System Break Register for comparison with the logical address offset, while any reference to a user segment always enables the Normal Break Register. These comparisons are independent of whether the Z8001 CPU is executing in system or normal mode. The function of the system segment detection logic is to prohibit normal mode programs from accessing system mode segments. Normal mode references to system segments are not valid and cause no MMU to be selected, and a segment violation forced upon the CPU. This violation is maintained until cleared by the segment trap acknowledge status of the CPU.

SECTION 5
MAINTENANCE

5.1. Introduction

This section contains the procedures necessary to maintain the System 8000. The information is presented in the form of preventive and corrective maintenance. All maintenance should be performed by qualified and trained field service personnel.

5.2. Preventive Maintenance

Preventive maintenance consists of routine cleaning procedures and adjustments performed in compliance with schedules provided in paragraphs 5.3 and 5.4. The CPU and Disk Tape Modules have fan filters which should be inspected and cleaned periodically.

5.3. Tape Drive Maintenance

Components of the Tape Cartridge Drive requiring cleaning are shown in Figure 5-1. The Magnetic Tape Mechanism cleaning procedures described in paragraph 5.3.1 through 5.3.4 should be performed in accordance with the schedule in Table 5-1.

Table 5-1. Cleaning Schedule

ITEM	HOURS OF USE
Magnetic Head	8
Tape Cleaner	8
Motor Capstan	8

5.3.1. Magnetic Head Cleaning: The magnetic head should be cleaned daily if the tape drive is in regular use. Dirty heads can cause the loss of data during read and write operations. Use a nonresidue, noncorrosive cleaning agent, such as isopropyl alcohol, and a soft cotton swab to clean the head assembly. Be sure to wipe off any excess cleaning agent and allow the heads to dry prior to operating the drive.

CAUTION

Spray cleaners are not recommended because overspray will contaminate the motor bearings. Also, never clean the head with hard objects. This results in permanent head damage.

5.3.2. **Tape Cleaner Cleaning:** The tape cleaner removes loose tape oxide and other foreign material from the tape before it contacts the head. This foreign material accumulates in and around the tape cleaner and must be removed to ensure that the tape cleaner continues to work effectively. The tape cleaner should be cleaned on the same schedule as the head.

To clean, insert a folded sheet of paper into the bottom of the cleaning slot of the cleaner. Slide the paper up, lifting the foreign material from the cleaner. Compressed air or a soft brush can be used to remove the foreign material from the area around the tape cleaner and head assembly. Alternatively, the tape cleaner can be cleaned using the same materials used to clean the magnetic head.

CAUTION

Do not use hard objects to clean the tape cleaner. If the tape cleaner becomes chipped, it will scratch the tape surface, resulting in lost data and/or permanent tape damage.

5.3.3. **Motor Capstan Cleaning:** The drive capstan is composed of hard polyurethane and must be cleaned after foreign material has built up. Clean, using isopropyl alcohol and a soft cotton swab. The cleaning schedule is the same as for the head.

Do not allow cleaning solvent to contaminate the drive motor bearings.

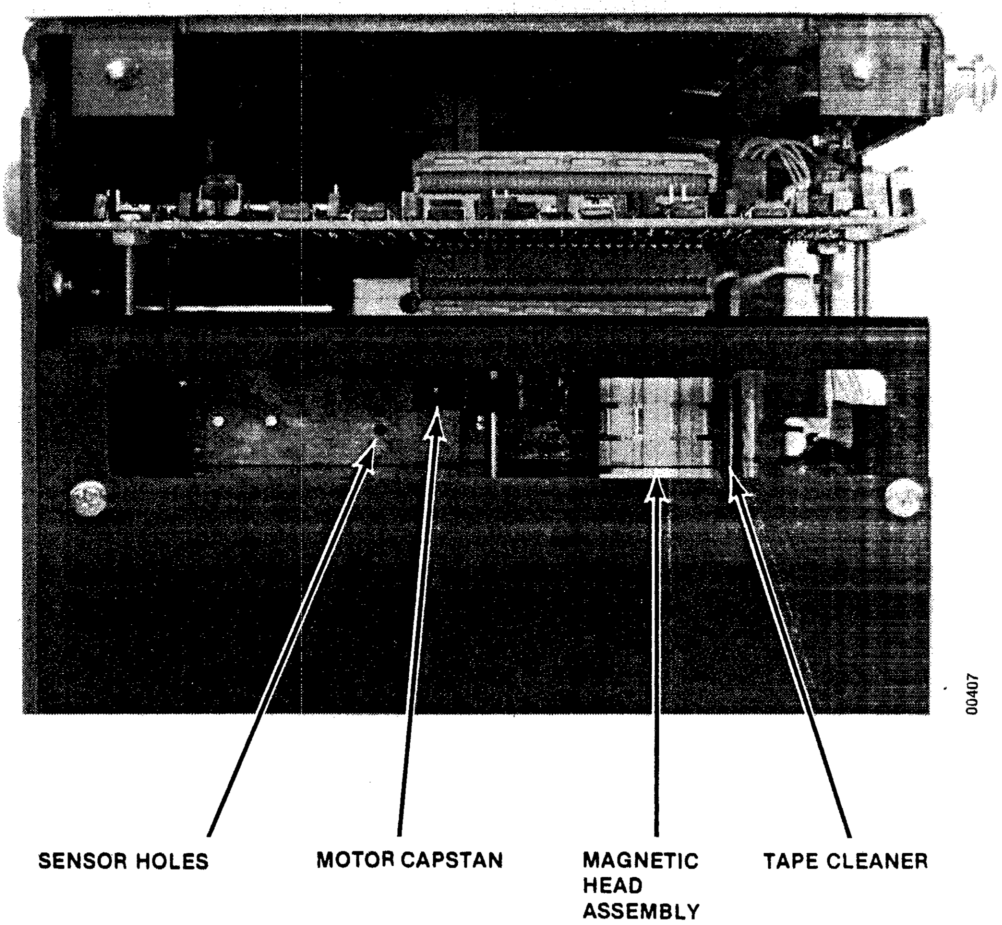


Figure 5-1 Location of Parts Requiring Periodic Cleaning

5.3.4. Heat Sink, Circuit Board, and Sensor Hole Cleaning: To prevent possible overheating, dust and dirt must be removed from the heat sink and drive assembly components. The time period between cleanings varies widely, depending on the operating environment. Use a soft brush and/or low pressure compressed air for cleaning. The sensor holes are cleaned in the same manner.

5.4. Disk Drive Assembly Cleaning

Both the SMD and Finch disk assemblies are sealed units and therefore do not require preventive maintenance procedures. The major components of the FINCH drive are shown in Figure 5-2.

CAUTION

NEVER disassemble either the FINCH or SMD. This exploded view is for information only. Servicing items in the upper sealed environmental enclosure (heads, media, actuator, etc.) requires special facilities. Only the printed-circuit boards, brake and motor external to the sealed area can be replaced without special facilities.

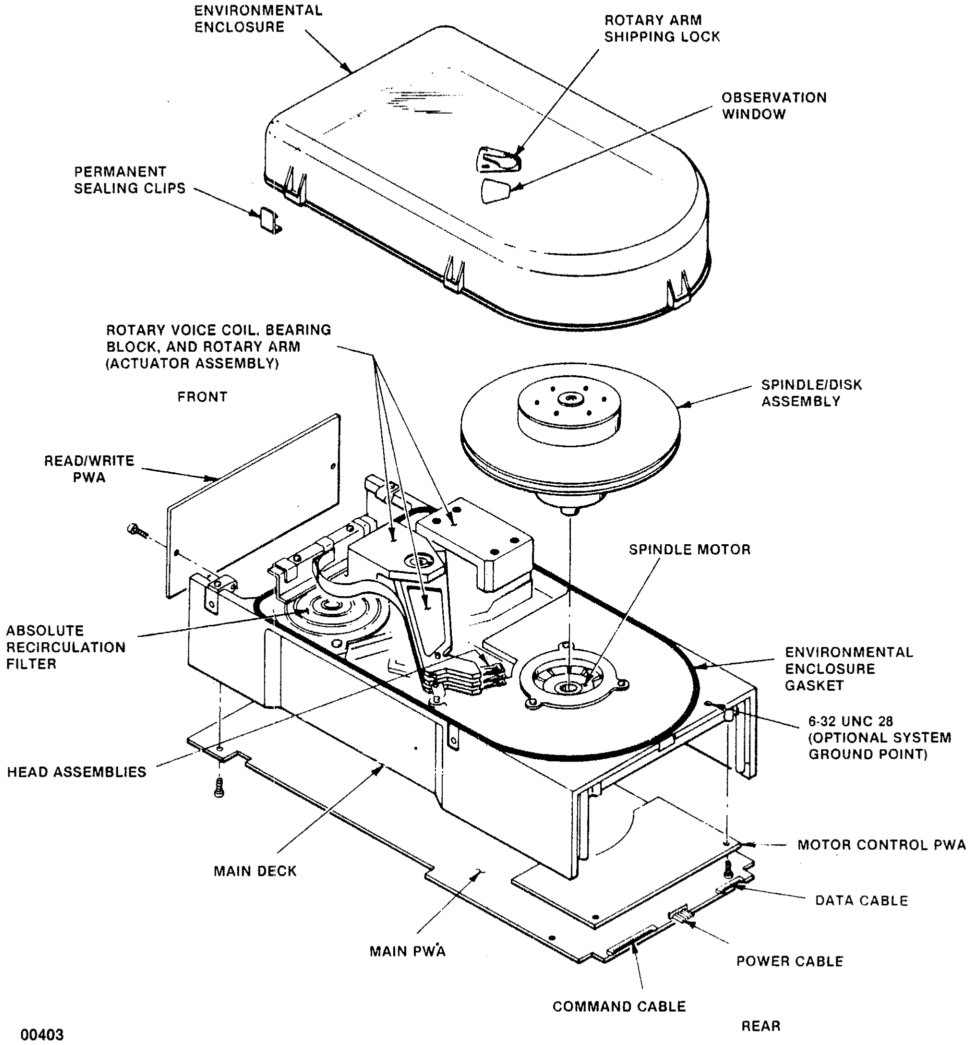


Figure 5-2 FINCH Major Components

5.5. Disk Drive Mounting

Four 8-32 tapped holes are provided in the base of the main deck casting to facilitate disk drive mounting in the horizontal position.

The FINCH or SMD is mounted directly to a bracket, which is then attached to the chassis using 8-32 thread screws. Shock mounts are installed in the chassis. Place the drive in the chassis and secure it with washers and screws with sufficient length to ensure adequate thread engagement (refer to Figures 5-3 and 5-4).

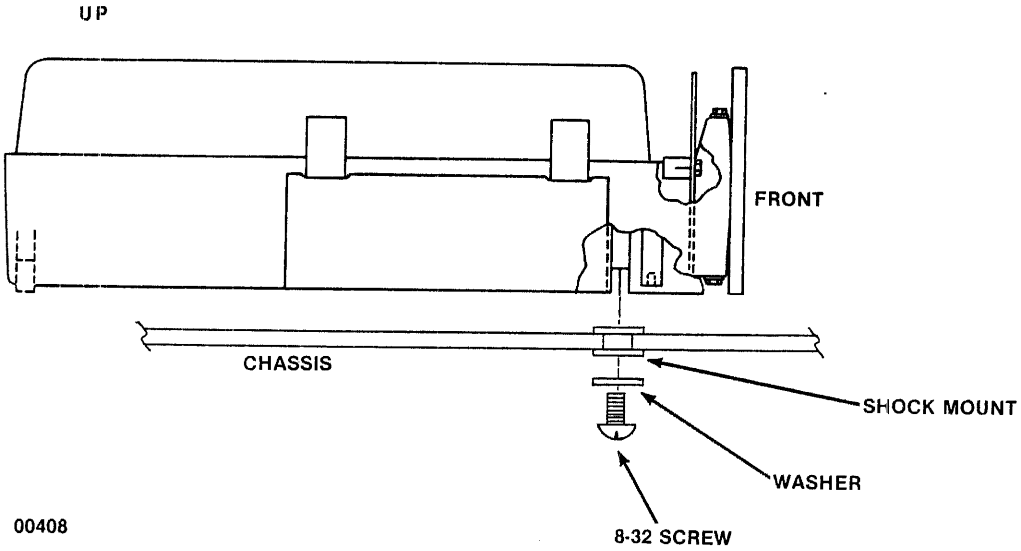


Figure 5-3 Disk Drive Mounting

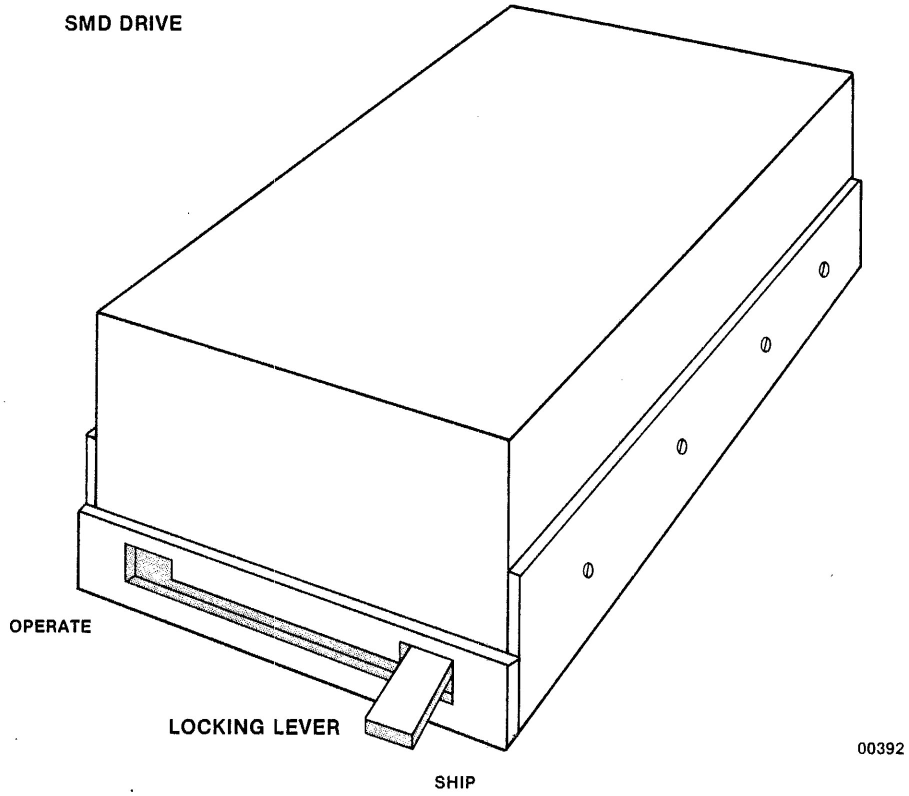


Figure 5-4 SMD Mounting Bracket

5.6. Drive Cabling

The cables that connect either the FINCH or SMD drive to the Disk Bus Interface, to DC power, and to the host system are listed in Table 5-2.

CAUTION

> **The interface connector for the Disk Bus Interface is located very close to W1 (the jumper location for connecting DC ground to frame ground). Use extreme care when inserting or removing the interface cable so the locking tabs for the connector do not break W1.**

All input/output cables exit at the rear of the disk drive. The FINCH Adapter Board attaches to connectors on the drives

Main PWA. The Disk Bus Interface and DC power cables exit from connectors at the top of the Adapter board. For the SMD, two interface cables (SMD A & SMD B) connect directly to the SMD Interface board. Power connection is by a 14 pin connector (P2) cable that plugs into the SMD below SMD B.

Table 5-2. Interface Cables

MODEL 21

CABLE NAME	DESCRIPTION	VENDOR/PART NUMBER
Signal Cable	40-conductor flat ribbon cable - maximum of 30 ft. (9.1 cm)	Zilog Part Number 59-0223-00
	Connector (Open strain relief)	3M 3417-6040
DC Power Cable	10-conductor, 18 gauge wire - maximum of 4 ft. (1.2 m)	Zilog Part Number 59-0131-00
	Connector	AMP 1-640426-D

MODEL 31

CABLE NAME	DESCRIPTION	VENDOR/PART NUMBER
SMDC-B Signal Cable	26-conductor flat ribbon and	59-0235-00
SMDC-A Signal Cable	60-conductor twisted cable	59-0234-00
DC Power Cable	14-conductor 18 grange wire radial	59-0227-00
AC Power Cable SMD	2-wire AC, 120V Fan	N/A

5.7. Disk Power ON Procedure

This section describes the procedure for applying power to the drive. The disk is a nonvolatile storage media; therefore, once data is recorded, it is not lost when power is turned OFF.

The following checks must be performed prior to turning power ON:

1. Inspect the physical mounting to ensure that the drive is secured at all mounting points, and that all ribbon cables are securely connected.

CAUTION

Improper orientation of the power connector can result in serious damage to the drive.

2. Place the rotary-arm shipping lock to the OPERATE position. The shipping lock is located on the bottom support bracket on the front of the SMD drive.

3. Apply power to the drive. Observe that the actuator mechanism performs a first seek operation in less than one minute (the read/write heads move from the landing area into the data area of the disk).

5.8. Power Supply Voltages

Power supply voltages should be checked each time on-premise maintenance is performed. This should only be performed by Zilog Field Service personnel. Ensuring that each supply is within tolerance is essential to maintain normal system operations.

TEST EQUIPMENT REQUIRED: HP3466A Multimeter or equivalent

5.8.1. AC Input Voltage Check: Using the multimeter, check RMS voltage at the facility outlet where the System 8000 will be plugged in. Ensure that the voltage reading is within the range specified by the switch settings on the CPU and Disk/Tape Module power supplies.

5.8.2. CPU Module DC Voltage Test:

1. Remove the module top cover. Unfasten two captive fasteners at the rear of the cover; pull toward the rear of the module, and remove from system.

2. Remove the sheet metal cover on the front portion of the module. Lift to clear the cover guides and pull toward the front of the module.

3. Test points are located in the upper right hand corner of the card cage backplane. Refer to Figure 5-5.

4. Check the following DC output voltages at their respective test points:

   a. +5 Vdc +0.25/-0.1 at TP2
   b. +12 Vdc +-0.1 at TP3
   c. -5 Vdc +-0.1 at TP5

In the CPU Module, voltage adjustments can be made at the following locations:

1. On the switching power supply, voltage adjustments are for +5 Vdc (silkscreened V. ADJ.) and for power fail (P.F.). DO NOT ADJUST POWER FAIL, IT IS SET AT THE FACTORY.

2. Voltage adjustments on the linear power supply are for -5 Vdc (5 ADJ.) and +12 Vdc (12 ADJ.).

5.8.3. Disk/Tape Module (Model 21) DC Voltage Test:

1. Remove the CPU Module from atop the Disk/Tape Module.

2. DC output voltages are checked at the Winchester disk drive power connector. The power connector is located in the FINCH Adapter Board (attached to the rear of the drive). Refer to Figure 5-6 for connector orientation and voltage test points.

3. Voltage adjustments on the power supply are for -5.2 Vdc (-5VDC ADJ), +5 Vdc (+5VDC ADJ), -24 Vdc (-24VDC ADJ), and +24 Vdc (+24VDC ADJ). (Refer to Figure 5-7 for location of adjustment potentiometers).

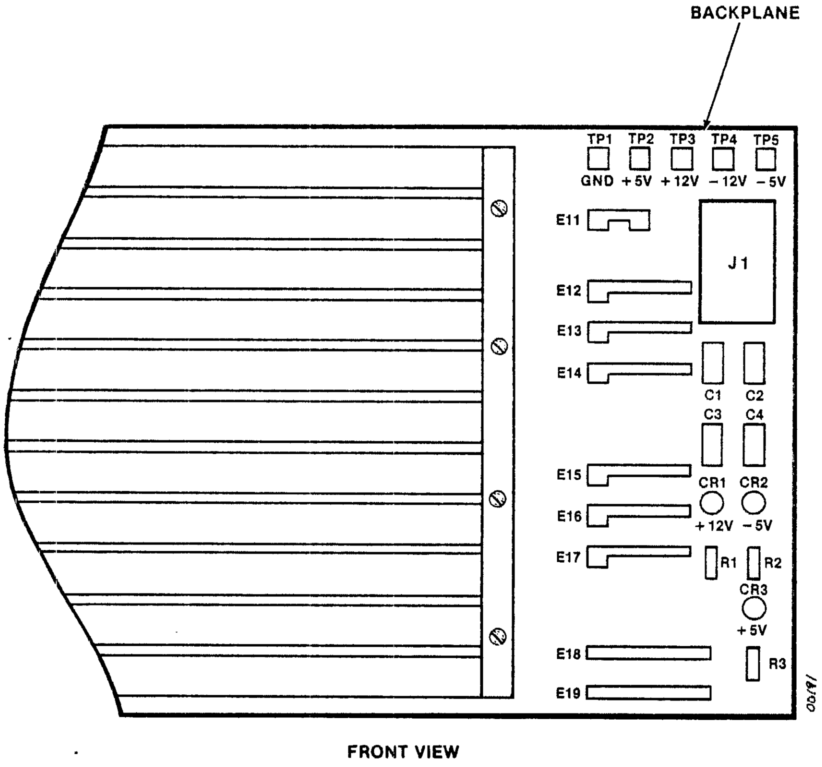


Figure 5-5 Power Supply Voltage Test Points, CPU Module

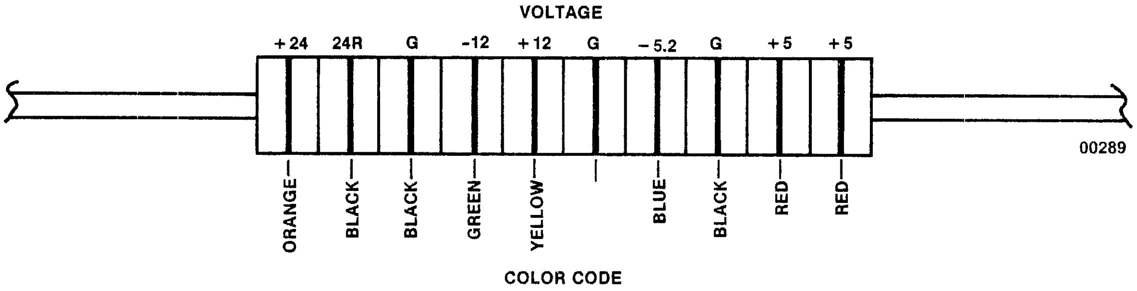


Figure 5-6 Voltage Test Points, Finch DC Power Connector

5.8.4. Disk/Tape Module (Model 31) DC Voltage Test:

1. Remove the CPU Module from the top of the Disk/Tape Module.

2. DC output voltages are checked at the SMD input power cable power connector P2 located on the rear of the SMD (Figure 5-7).

3. SMD Voltage adjustments on the power supply (Figure 5-8) are for +5VDC (+5VDC ADJ), -12VDC (-12VDC ADJ), and +24VDC(+24VDC ADJ). (See Figure 5-8 for locations.)

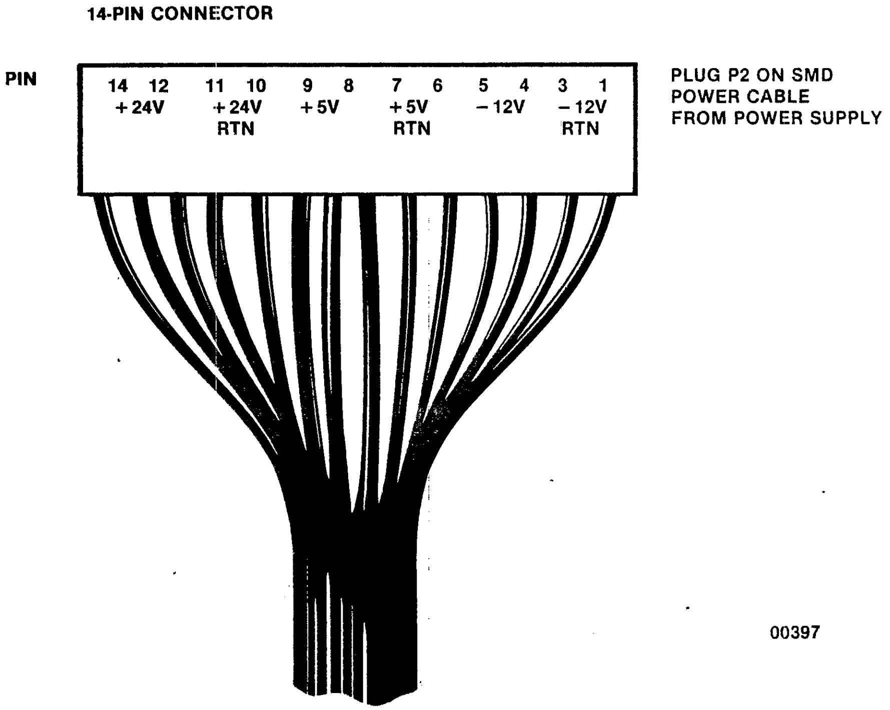


Figure 5-7 Voltage Test Points, SMD Power Connector

NOTE

Ensure that the power connector (P2) is installed with pin 1 oriented as shown in Figure 3-11 to avoid damage to drive.

4. Cartridge Tape Drive voltage adjustments include one additonal power supply output adjustment. Voltage

adjustment for the -24Vdc (-24V ADJ.) can be checked at the power connector labelled 'J2' on the power supply, (see Figure 5-8 for connector location).

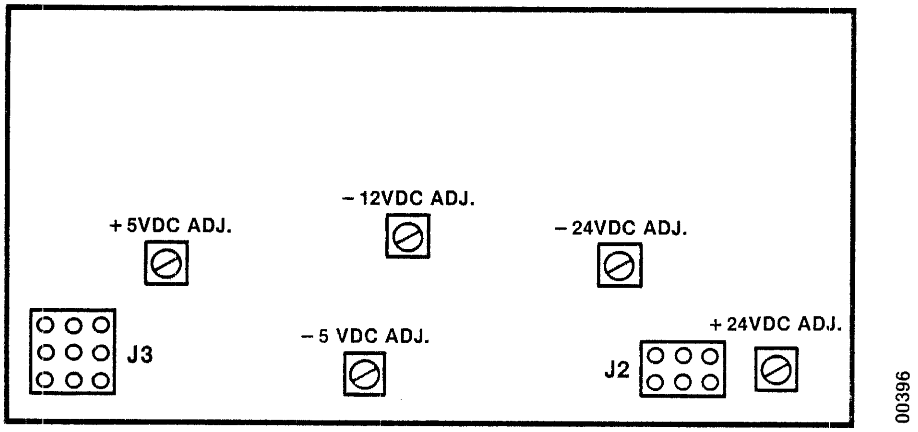


Figure 5-8 Power Supply Voltage Adjustments

5.9. System Monitor

The Monitor Program resides on PROM on the CPU. The monitor includes basic debugging commands, I/O control, and interface software for use with a serial interface to a remote computer system.

5.9.1. Monitor Program Debug Environment: The Monitor Program sets software breakpoints for program debugging. A breakpoint is a command that interrupts or stops program execution at a specified address in the program. The address specified in the breakpoint is the address of the instruction. When encountered during program execution, the breakpoint suspends execution of the user's program and saves all registers, program counters (PC), and the Flag Control Word (FCW) in the memory area provided. It then displays a message reporting the break and the address where it occurred.

Any number of breakpoints can be set manually by setting the instruction at the desired breakpoint address to %7F00 (% indicates the address is in hexadecimal notation). This interrupts the executing program and jumps (traps) to the breakpoint procedure. When the breakpoint is no longer required, the original instruction must be manually restored.

The BREAK command saves the address where the breakpoint is being set and the instruction that it is replacing. When the breakpoint is cleared, the instruction is automatically restored. The BREAK command also stores a repetition counter, n. Execution is not suspended until the nth time this breakpoint is encountered unless another breakpoint is encountered first.

The following restrictions on the user program are necessary to set breakpoints:

1. This program must be able to execute with interrupts enabled after encountering the breakpoint.

2. The program should not be timing-dependent because there will be some timing distortion each time the breakpoint is encountered.

3. The user program must not use Channel 3 of the Z80A Counter Timer Circuit (CTC #0), because it is used to implement the multiple execution feature.

4. The breakpoint cannot be within an interrupt procedure entered by an interrupt from Channels 0 through 2 of the Z80B CTC #0.

The BREAK and the NEXT commands use instruction modification and the interrupt system. Therefore, the program being debugged cannot be in the PROM area and cannot involve modification of the interrupt status.

Any set breakpoints must be cleared by hitting RESET before a new program is loaded from the System 8000; otherwise, previously set breakpoints continue to operate on the new program during debugging.

The user stack is used whenever a JUMP or GO command is executed. The command must be set to some address within writable memory. If the JUMP or GO address has a system breakpoint set, it does not cause suspension of execution.

5.9.2. Monitor Program: The following conventions are used in command descriptions:

< > Angle brackets enclose descriptive names for the quantities to be entered.

[ ] Square brackets denote optional quantities.

¦ A bar denotes an OR condition. For example, W¦B means either W or B can be used.

<CR> Carriage Return and line feed.

[A single square left bracket is the monitor prompt.

% Symbol for hexadecimal value, for example %4F or %4FFF.

Apply the following when entering commands and options:

1. All commands and options must be entered in uppercase.

2. Commands can be abbreviated to the first letter.

3. Numbers are represented in hexadecimal notation.

4. The first character typed on a new line identifies which command is being invoked. If an invalid character is entered, a "?[" is displayed, prompting a new command.

5. Addresses are specified by an optional segment number in angle brackets, followed by a hexadecimal address. If no segment number is specified, segment 0 is assumed. For example, <00>4000 or 4000, <00>0 or 0, <01>F800 or F800 for segment 0.

5.9.3. Monitor Mode Commands: A summary of the commands in the Monitor Mode are as follows:

COMMAND NAME	PARAMETERS
DISPLAY	<address> [<# of long words/words/bytes>] [L¦W¦B] Display and alter memory
REGISTER	[<register name>] Display and alter registers
BREAK	[<address>] [<n>] Set and Clear breakpoint
NEXT	[<n>] Step instruction
GO	Branch to last PC set in user register array
JUMP	<address> Branch to address
FILL	<address1> <address2> <data> Fill memory
IOPORT	<port address> [W¦B] I/O port read/write
MOVE	<address1> <address2> Move memory block
COMPARE	<address1> <address2> Compare memory blocks
QUIT	Enter Transparent Mode
PORT	<port address> [W¦B] Special I/O read/write
TEST	Enter Test Mode
ZBOOT	[D¦S¦T] Read a BOOTSTRAP program(s) from SMD, disk or cartridge tape and execute

NOTE

All outputs in Monitor Mode can be suspended with XOFF (%13 or control-s) and resumed with XON (%11 or control-q).

COMMAND NAME

DISPLAY

SYNTAX

D <address> <# of long words/words/bytes> [L¦W¦B]

DESCRIPTION

This command displays at the terminal the contents of specified memory locations starting at the given address, for the given number of bytes.

If the count parameter is specified, the contents of the memory locations are displayed in hexadecimal notation and as ASCII characters.

If the count parameter is not specified, the memory locations are displayed as words one at a time, with an opportunity to change the contents of each location. For each location, the address is displayed, followed by the contents of L¦W¦B and a space. To change the contents at a given location, enter the new contents in the form long word¦word¦byte. If RETURN is pressed, either alone or after the new contents, the next sequential location is displayed. Entering a "Q" (for QUIT), followed by a RETURN terminates the command.

EXAMPLE

```
    Display memory starting at Segment 0, %5200 for 16 words.
    [D 5200 10 <CR>

    <00> 5200 1808 FE2B 2004 D923 7ED9 CD35 2238 0AED
      *...+..#...5"8..*
    <00> 5210 6F23 ED6F 2B1E 0118 EDD9 2218 14D9 5778
      *o#.o+...H...Wx*
```

EXAMPLE

```
    Display memory starting at Segment 0, %5200 for 16 bytes.
    [D 5200 10 B <CR>

    <00> 5200 18 08 FE 2B 20 04 D9 23 7E D9 CD 35 22
      38 0A ED *...+..#...5"8..*
```

EXAMPLE

Display memory location Segment 0, %5200 and alter its contents.

```
[D 5200 <CR>

<00> 5200 1808 ?1922 <CR>

<00> 5201 FE2B ?<CR>
<00> 5202 2004 ?<CR>
```

COMMAND NAME

REGISTER

SYNTAX

R [<register name>]

DESCRIPTION

The REGISTER command is used to examine or modify a specified register. If no register name is given, all registers R0, R1, R2 ... R15 PC and FCW are displayed. If a register name is given, the specified register name is displayed, followed by a space. To change the contents of that register, enter the new contents followed by <CR>. <CR> alone or after the new contents displays the next register. A "Q" followed by a <CR> terminates the command.

The following register names can be used in the command:

1. Any of the sixteen 16-bit registers named R0, R1, R2 ... R15.

2. Any of the sixteen 8-bit registers named RH0, RL0, RH1, RL1 ... RH7, RL7.

3. Any of the eight 32-bit registers named RR0, RR2, RR4 ... RR14.

4. Program counter register named PC.

NOTE

The new contents of the program counter must be given in even hexadecimal numbers.

5. Flag and control word named FC.

EXAMPLE

```
    Display all registers.
    [R <CR>

    R0   R1   R2   R3   R4   R5   R6   R7   SG   PC   FC   RF
    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

    R8   R9   R10  R11  R12  R13  R14  R15  N4   N5   PS   PO
    0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

EXAMPLE

Display 32-bit word register RR4 and alter its contents.

```
[R RR4 <CR>

RR4 00000000 ?A2557FFFF <CR>

RR6 00000000 ?Q <CR>
```

COMMAND NAME

BREAK

SYNTAX

B [<address>] [<n>]

DESCRIPTION

The BREAK command sets a breakpoint at a given even address after clearing any previously set breakpoint. If <n> is given, program execution is not interrupted until the nth time the breakpoint instruction is encountered (<n> is in the range %1-%FFFF). If <n> is not given, 1 is assumed. If the BREAK command is issued with no parameters, any previously set breakpoint is cleared. When program execution is suspended by the BREAK command, the Monitor Program displays a message reporting the break and the address where it occurred.

EXAMPLE

```
    [B 6A5E <CR>
```

Message: BREAK AT 6A5E

EXAMPLE

```
    [D 8000 <CR>
    <00> 8000 (XXXX)? <8D07> <CR>
    <00> 8002 (XXXX)? <Q> <CR>
    [B 8002 <CR>
    [J 8000 <CR>
    BREAK AT 8002
    [
```

COMMAND NAME

NEXT

SYNTAX

N [<n>]

DESCRIPTION

The NEXT command causes the execution of the next n machine instructions, starting at the current PC, and displays all registers after executing each instruction. (<n> is in the range %1-%FFFF.) If <n> is not given, 1 is assumed.

EXAMPLE

```
[F 8000 9000 8D07 <CR>
[D 9000 <CR>
<00> 9000 8D07 ? <7F00> <CR>
<00> 9002 XXXX ? <Q> <CR>
[R SG <CR>
RSG XXXX ? <0> <CR>
RPC XXXX ? <8000> <CR>
RFC XXXX ? <C000> <CR>
RRF XXXX ? <Q> <CR>
[N <CR>
R0   R1   R2   R3   R4   R5   R6   R7   SG     PC     FC     RF
X    X    X    X    X    X    X    X    0000   8002   C000   X
R8   R9   R10   R11   R12   R13   R14   R15   N4   N5   PS   PO
X    X    X     X     X     X     X     X     X    X    X    X
[N2 <CR>
R0   R1   R2   R3   R4   R5   R6   R7   SG     PC     FC     RF
X    X    X    X    X    X    X    X    0000   8004   C000   X
R8   R9   R10   R11   R12   R13   R14   R15   N4   N5   PS   PO
X    X    X     X     X     X     X     X     X    X    X    X
R0   R1   R2   R3   R4   R5   R6   R7   SG     PC     FC     RF
X    X    X    X    X    X    X    X    0000   8006   C000   X
R8   R9   R10   R11   R12   R13   R14   R15   N4   N5   PS   PO
X    X    X     X     X     X     X     X     X    X    X    X
[
```

COMMAND NAME

GO

SYNTAX

G

DESCRIPTION

This command causes a branch to the current PC, continuing program execution from the location where it was last interrupted. All registers and the FCW are restored before branching.

EXAMPLE

Execute/continue executing program

```
[G <CR>
```

COMMAND NAME

JUMP

SYNTAX

J <address>

DESCRIPTION

The JUMP command branches unconditionally to the given even address. All registers and the FCW are restored before branching.

EXAMPLE

```
    Execute user program starting at %5000.
    [JUMP 5000 <CR>
```

COMMAND NAME

FILL

SYNTAX

F <address1> <address2> <word data>

DESCRIPTION

The FILL command stores the given data word in a memory location, from address1 to address2. The command address must be an even hex number.

EXAMPLE

```
    Store data FFFF in memory from %5400 to %5410.
    [F 5400 5410 FFFF <CR>
    [
```

COMMAND NAME

IOPORT

SYNTAX

```
I <port address> [W¦B]
```

DESCRIPTION

This command reads data in either byte or word form from the given port address and displays the value. Enter a hex value to be output to the specified port or enter only a value to be output to the specified port or enter only a carriage return if no output is to be made. If the W¦B parameter is not given, byte data is read from the I/O port.

EXAMPLE

Output data FF to port address %FF29.

```
[I FF29 <CR>

FF29 00 ? <FF> <CR>
[
```

COMMAND NAME

MOVE

SYNTAX

```
M <address1> <address2> <n>
```

DESCRIPTION

This command moves the contents of a block of memory from the source address specified by <address1> to the destination address specified by <address2>. <n> is the number of bytes to be moved.

EXAMPLE

```
    Move memory from address %5000 to %6000 for 256 bytes.
    [M 5000 6000 100 <CR>
```

COMMAND NAME

COMPARE

SYNTAX

```
C <address1> <address2> <n>
```

DESCRIPTION

This command compares the contents of two blocks of memory. <address1> and <address2> specify the starting addresses of the two blocks, and <n> specifies the number of words to be compared. If any locations of the two blocks differ, the addresses and contents of those locations are displayed.

EXAMPLE

Compare two blocks of memory with starting addresses %4000 and %5000 for 32 words.

```
[C 4000 5000 20 <CR>
[
or on MISCOMPARE:
<00> 5000=XX <00> 4000=YY
[
```

COMMAND NAME

QUIT

SYNTAX

Q

DESCRIPTION

The QUIT command is used to enter Transparent Mode from Monitor Mode. In Transparent Mode, all keyboard inputs and console outputs are passed between the remote computer system and the local system. The console controls the remote computer system operating system. Channels A and B of the SIO2 must be set to the same baud rates when operating in Transparent Mode. (The remote system connects to TTY0 on the rear panel of the local system.)

The START switch on the System 8000 is used to return to Monitor Mode.

COMMAND NAME

PORT

SYNTAX

P <port address> [W¦B]

DESCRIPTION

The PORT command is similar to the IOPORT command; however, it is used to read or write special I/O devices.

EXAMPLE

```
[P FC <CR>
OOFC XX ? <CR>
[
```

COMMAND NAME

TEST

SYNTAX

T

DESCRIPTION

The TEST command executes the SPUD System Power-Up Diagnostic tests. (See section 3.7 for detailed description).

EXAMPLE T <CR>

COMMAND NAME

ZBOOT

SYNTAX

```
Z [D¦S¦T]
```

DESCRIPTION

This command is commonly used to manually bootstrap the ZEUS Operating System. The ZBOOT command reads a 512-byte program from block 0 of the device determined by D/ DISK (Model 21), S/SMD (Model 31), T/ Cartridge Tape. Generally, there is no return to the Monitor.

EXAMPLE

```
Z T <CR>
```

5.9.4. Download Mode Command: Summary of Commands in Download Mode are:

NAME	PARAMETER
LOAD	<filename> Load S/W from System 8000

NOTE

Filenames can be specified in either uppercase or lowercase. Filenames can be full path names. LOAD stores data into segment 0.

Download Mode transfers data between the System 8000 and a remote computer system. Channels A and B of the SIO2 must be set to the same baud rates when operating in Download Mode. (The remote system is connected to TTY0 on the rear panel.) The LOAD program is required on the remote system to perform download functions through console I/O.

The Download Mode uses the Tektronix record format, which uses only ASCII characters. Each record contains two checksum values, a starting address, and a maximum of 30 bytes of data. The format of the record is:

RECORDS 1 to n

```
<address(4)> <count(2)> <checksum1(2)> <data(2)>...
<data(2)> <checksum2(2)> <carriage return>
```

where:

<address(4)>: Is the address of the first byte of data in the record (address is represented in four ASCII characters)

<count(2)>: is the number of <data> in current record (two ASCII characters)

<checksum1(2)>: is the checksum for the address and count field (two ASCII characters)

<data(2)>: is the value of byte data (represented in two ASCII characters)

<checksum2(2)>: is the checksum for the data portion of the record (two ASCII characters)

<carriage return>: indicates the end of the record

No segment information is transferred. All downloaded data is loaded into segment 0 with the LOAD command. Data for segments other than 0 must be transferred to Segment 0 by the MOVE command.

LAST RECORD

<entry address(4)> 00 <checksum(4)> <carriage return>

where:

<entry address>: is the starting execution address for the program

<checksum>: is the checksum for the entry address

NOTE

A record with 00 in the count field indicates the end of load data.

RECORD WITH ERROR MESSAGE

If either the local or remote system has to abort the load process, it sends a record of the form:

/ <error messages in ASCII text> <carriage return>

ACKNOWLEDGE

During the loading process, after each record is received from the remote system, an acknowledge (ASCII 0) is sent when the checksum values are verified. If a nonacknowledge (ASCII 7) is received, the remote system attempts to load the same data record up to ten times. After the tenth try, the Monitor Program returns to Monitor Mode for the next command. An abort-acknowledge (ASCII 9) is sent to the remote system if the escape (ESC) key is pressed, aborting the loading process. The

Monitor Program then returns to Monitor Mode for the next command. The address used in the data record during the loading process is provided by the file description record; it must be greater than %8000.

COMMAND NAME

LOAD

SYNTAX:

L <filename>

DESCRIPTION:

This command downloads a Z8000 program named <filename> that resides in the remote system.

The Monitor Program transmits the exact command line to the remote system. The command causes a remote procedure file (LOAD) to be executed, to open the file specified by <filename>. The binary data in the file is converted to Tektronix record format and transmitted to the local system. The Monitor Program verifies the two checksum values in the receiving record and stores the data in RAM memory as specified by the address indicated in the record. An acknowledgement from the system causes the next record to be downloaded from the remote system. A nonacknowledgement from the remote system causes the current record to be retransmitted up to ten times, after which a record with an error message is sent, and the Monitor Program returns to Monitor Mode. The LOAD program in the remote system is also aborted. When the loading process is completed, the entry point received on the first record is displayed. Pressing ESC aborts the LOAD command. Any breakpoints set from a previous program must be cleared before a new program is loaded from the remote system. (NOTE: Ensure that the remote system is connected to TTY0 on the rear panel).

ERROR MESSAGES:

/ABORT
/UNABLE TO OPEN FILE (XX), where (XX) is the ZEUS error code from the remote system
/FILENAME ERROR
/NOT PROCEDURE FILE
/ERROR IN READING FILE (XX), where (XX) is the ZEUS error code from the remote system
/RECORD CHECKSUM ERROR
/INCORRECT LOAD ADDRESS

EXAMPLE:

Transfer file names MYFILE from the remote system to the local system RAM memory.

```
[LOAD MYFILE <CR>
```

NOTE

The address of RAM memory and the entry address used in the download process are provided by the information in the descriptor record of the file specified by <filename> in the LOAD command.

5.9.5. System Parameters: The following system parameters are accessible to the user:

NAME **PARAMETER**

NULLCT Null Count %43F6

This address stores the number of null characters that are inserted after a line feed. Modifying the null count adapts the System 8000 to the return delays of various terminals. NULLCT is initialized to 0.

LINDEL Line Delete %43F3

This address stores the character intercepted by the input line procedure as a line delete. When it is read from the terminal, this procedure purges the buffer and continues reading the input stream. LINDEL is initialized to %7F (RUB).

CHRDEL Character Delete %43F2

This address stores the character intercepted by the input line procedure as a character delete. When it is read from the terminal, the last character entered is purged from the input buffer. Multiple character deletes can be used to delete the last n characters entered. CHRDEL is initialized to %08 (control-h).

XOFCHR XOFF Character %43F5

The character stored at this address is interpreted by the input interrupt procedure as a character that stops outputting data to the terminal. When it is read from the terminal, all output is suspended until an XONCHR is received. XOFCHR is initialized to %3 (control-s).

XONCHR XON Character %43F4

The character stored at this address is interpreted by the input interrupt procedure as a character that resumes output after XOFCHR is entered. When it is read from the terminal, all output is resumed. XONCHR is initialized to %11 (control-q).

STACK Stack Pointer %40A0

This address is the base of the user stack set by the Monitor Program at reset. The top of the stack is %4000.

PSAREA Program Status Area %4400

The Program Status Area for entering various interrupts and trap handling procedures starts at this address. This area includes the program status blocks (FCW and PC) for different types of interrupts and traps. The Monitor Program sets up these program status blocks as shown in Table 5-3.

Table 5-3. Program Status Area

WORD	VALUE	COMMENT
0-1	unused	RESERVED
2-3	unused	RESERVED
4-5	unused	Unimplemented instruction
6-7	unused	Unimplemented instruction
8-9	unused	PRIVILEGED INSTRUCTION
A-B	unused	PRIVILEGED INSTRUCTION
C-D	%4000	SYSTEM CALL entered in Segmented Mode
E-F	#BREAK	Address of BREAK interrupt procedure
10-11	unused	SEGMENT TRAP
12-13	unused	SEGMENT TRAP
14-15	%4000	FCW for NONMASKABLE interrupt procedure
16-17	#NMINT	Address of NONMASKABLE interrupt procedure
18-19	unused	NONVECTORED INTERRUPT
1A-1B	unused	NONVECTORED INTERRUPT
1C-1D	%4000	FCW for all VECTORED INTERRUPTS
1E-1F	unused	VECTOR 0
20-21	unused	VECTOR 2
22-23	unused	VECTOR 4 -- CTCO, CH.3
24-25	unused	VECTOR 6 -- BREAK and NEXT
26-27	unused	VECTOR 8
28-29	unused	VECTOR A
2A-2B	unused	VECTOR C
2C-2D	unused	VECTOR E
2E-2F	unused	VECTOR 10
30-31	unused	VECTOR 12
32-33	#PTYINT	VECTOR 14 (SIO Channel B input interrupt procedure address)
34-35	#CHASRC	VECTOR 16 (SIO Channel B special receive condition procedure address)
36-37	unused	VECTOR 18
38-39	unused	VECTOR 1A
3A-3B	#MCZINT	VECTOR 1C (SIO Channel A input interrupt procedure address)
3C-3D	#CHASRC	VECTOR 1E (SIO Channel A special receive condition procedure address)
3E-3F	unused	VECTOR 20
40-41	unused	VECTOR 22
42-43	unused	VECTOR 24
44-45	unused	VECTOR 26
46-47	unused	VECTOR 28
48-49	unused	VECTOR 2A
4A-4B	unused	VECTOR 2C
4C-4D	unused	VECTOR 2E
4E-4F	unused	VECTOR 30

The port addresses shown in Table 5-4 are used in the Monitor Program.

Table 5-4 System Hardware I/O Port Addresses

PORT	ADDRESS
CTC CHANNEL 0	FFA1
CTC CHANNEL 1	FFA3
CTC CHANNEL 2	FFA5
CTC CHANNEL 3	FFA7
SIO DATA CHANNEL A	FF81
SIO DATA CHANNEL B	FF83
SIO CONTROL CHANNEL A	FF85
SIO CONTROL CHANNEL B	FF87
RETI PORT	FFE1
SYSTEM CONFIGURATION PORT (or REGISTER)	FFC1 - 2 bits baud rate, 2 bits boot device, 4 bits other configurations

5.10. Monitor I/O Procedures

The I/O procedures most frequently used in the Monitor Program are given in this section. These procedures are accessed by system calls in user programs to perform console I/O functions.

5.10.1. I/O Procedures:

PROCEDURE NAME

TYIN

DESCRIPTION

Gets a character from the keyboard buffer. If the buffer is empty, this procedure waits for a character to appear. The character is stored in register RL0, and the contents of register RH0 are lost.

EXAMPLE

```
CONSTANT
     TYIN :=%04
          .
          .
          .
     SC    #TYIN
     (character in RL0)
```

PROCEDURE NAME

TYWR

DESCRIPTION

Displays the character in RL0. The character is not displayed if the XOFF character has been received before this procedure is executed. In this case, the procedure waits until an XON character is received from the console before displaying the character in RL0. If the character to be displayed is a carriage return, the zero flag is set, and RH0 is lost.

EXAMPLE

```
CONSTANT
     TYWR :=%06
           .
           .
           .
     SC    #TYWR
     (character in RL0)
```

PROCEDURE NAME

PUTMSG

DESCRIPTION

Sends a character string to the terminal. Register R2 contains the address of the character string buffer, and the first byte in the buffer contains the number of characters to be displayed. If there is no return in the string, the entire specified string is displayed. Otherwise, the string is displayed up to and including the first return. Register contents R0, R1, and R2 are lost.

EXAMPLE

```
CONSTANT
     PUTMSG:=%0C
           .
           .
           .
     (string address in R2)
     SC     #PUTMSG
```

PROCEDURE NAME

TTY

DESCRIPTION

Receives and echoes at the terminal a character string up to the first return. The character string is stored in a buffer pointed to by register R2. Register R1 contains the size of the buffer. If the size of the character string exceeds the size of the buffer, the zero flag is set. All lowercase alpha characters are converted to uppercase characters before they are stored in the buffer. R1 returns the actual number of characters received from the terminal. The contents of registers R0 and R2 are lost.

EXAMPLE

```
CONSTANT
     TTY :%08
          .
          .
          .
     SC   #TTY
     (string address in R2, size in R1)
```

PROCEDURE NAME

CRLF

DESCRIPTION

Outputs a return followed by a line feed to the terminal. The contents of register R0 are lost.

EXAMPLE

```
CONSTANT
     CRLF :=%0A
          .
          .
          .
     SC    #CRLF
```

5.11. Stand-Alone Diagnostic Interactive Executive (SADIE)

NOTE

Refer to Appendix A of this manual, for SADIE tape release 3.2 test descriptions.

SADIE is a stand-alone diagnostic executive and diagnostic library. It provides thorough testing of all mainframe components. SADIE is based on the construction, modification, storage, and execution of test lists. The commands to operate on these test lists are displayed in a menu-oriented format on the CRT to minimize the documentation support required and to allow a first-time user to execute SADIE without constant reference to an instruction manual. The SADIE program code and associated data sets reside on the SADIE diagnostic tape cartridge, which is inserted in the tape slot and run whenever degraded system operational capability is suspected. The object code for SADIE resides in the lower part of a memory segment; the diagnostic test resides in the upper part of the segment.

5.11.1. Purpose of SADIE: The basic purpose of SADIE is to test the system hardware. To do this, SADIE allows construction and execution of test lists. A test list consists of lines executed sequentially. A line contains either a test or a control statement, up to four parameters necessary for execution of the test or control statement, and a repetition count for all noninteractive tests. Normal sequential execution of a test list can be altered by a GOTO control statement or interrupted by a PAUSE control statement in the list. There are control statements to specify line-printer test output as well as the visual display output. Other control statements specify that that the tests that follow should pause on a hard-error condition. Every test list includes a termination line consisting of an EOL (end-of-list) in the test-name field of the display. Single tests can also be executed. SADIE considers a single test to be a two-line test, where the second line is the EOL.

As a test list executes, errors are logged into a cumulative error log and an error summary log for the test. The cumulative log contains the last 20 error messages. The error summary contains the number of laps completed, the number of hard and soft errors, and the completion status of each line in the list. Both logs remain available until the next test list is run or until the end of the SADIE session.

5.11.2. Organization and Principles of Operation: SADIE is organized as a tree of menus to provide its diagnostic functions. Menu items are either submenu names (lower branches of the tree) or functions (the end of the branch) of SADIE. The tree is traversed by entering the choice required to perform a function or to move to another menu on the path to a function. Choices are entered on the console. Choices are interpreted when the line on which they are entered is terminated by a <CR>. In addition, the START and RESET buttons can be used to interact with SADIE.

The characters identifying menu choices are letters, integers, <CR> (carriage return), and ^ (↑ on some terminals). Entry of ^ causes a return to the menu from which the present menu was selected or to the master (COMMAND LEVEL) menu. A <CR> is used when there is only one choice or when there is a default choice, as in verifying that a prior choice is correct. Integers are used to identify tests, test lists, or lines in a test list, or for entry of numerical constants. Numerical constants entered are interpreted in decimal base unless the prompt indicates that the default base is hexadecimal or octal. The default base can be overridden by prepending 0 (zero) and a letter, d for decimal, o for octal, or x for hexadecimal for a specific number base. Letters are used for all other choices. When a letter is used, it is usually the initial letter of a capitalized keyword in the item description. Either upper or lowercase letters can be used.

A sequence of menus can be traversed by entering appropriate choices on a single line. Entries must be separated by one or more spaces. The terminating <CR> is treated as a menu choice if it is preceded by a space.

When a menu is too large to fit onto the CRT screen, entry of + will get the next screen, and - will get the previous screen.

SADIE maintains in memory both the current single test and current test list. Initially the current single test and the current test list only contain an EOL.

5.12. SADIE Tape Organization

The SADIE tape contains the following data sets:

1. The bootstrap loader functions as a tape loading supervisor that loads the SADIE machine code.

2. The SADIE machine code provides I/O support for tests. When invoked, it loads the library and test list cata-

logs, and initializes the CPU. After completing these initial functions, the executive function is available for interactive use.

3. The tape maintenance machine code supports addition and deletion of tests stored on the SADIE tape and updates the test catalog.

4 The diagnostic tape library contains object code files.

   An object code file contains the object code for a stand-alone test. A maximum of 25K bytes of memory is allocated to each object code file. This limit is necessary to allow the SADIE machine code to be coresident and to provide room for the stack that grows downward from the end of the segment.

5. The diagnostic library catalog contains all information necessary to identify the test, prepare for its execution, and load it. Included are the test name (1 to 8 characters), a short description of the test (1 to 40 characters), declarations of base (decimal, hexadecimal, or octal) and default values for up to four integer parameters, a functional classification of the test, and its file and track location on the tape.

6. The test list library contains test lists that have been stored.

7. The test list catalog contains a one- to 50-character description of each test list.

5.13. SADIE Program Initialization

To initialize SADIE:

1. Insert the tape cartridge into cartridge tape drive.

2. Press RESET.

3. Enter T<CR>.

4. Enter Z T<CR>.

This command executes the monitor-resident primary tape bootstrap routine that loads the SADIE bootstrap routine and transfers control to SADIE. After the bootstrap process is complete, a command level menu is displayed on the CRT that includes the version of SADIE that is present.

5.14. SADIE Diagnostic Functions

SADIE diagnostic functions fall into six classes accessible from the COMMAND LEVEL menu. This is the first menu displayed after product invocation. The first class is execution of a test list reached by the LIST command, L, or the CHOOSE command, C. The second class is execution of a single test reached by the TEST command, T, or the REPEAT command, R. The third class is display of SADIE-maintained error logs. The DISPLAY command, D, displays the error summary log for the most recently executed list. The A command displays the cumulative error log for ALL tests in the most recently executed list. The fourth class is the EDIT command, E, which provides editing functions for test lists such as creation, modification, storage, and retrieval. The fifth class is the tape maintenance functions, reached by the M command. These functions allow additions and deletions of tests in the SADIE test library. The sixth class is the QUIT command, Q, which performs an orderly return to the PROM Monitor.

Two functions are controlled by the START and RESET buttons. Pressing RESET disables SADIE and enables the PROM monitor. Presssing START halts the function that is executing by generating a Non-Maskable Interrupt (NMI). In addition to halting the current test, SADIE displays a PAUSE menu, described later.

5.14.1. Console Interactions: RUNNING A TEST LIST

Since single tests are executed as if they were one-line test lists, the running of test lists is treated first. The user can run the test list currently in memory with the LIST command, L, or choose any of the lists stored on tape with the CHOOSE command, C. In the second case, the catalog of test lists is displayed, and the one selected by the user is transferred from tape to memory. In either case, the subsequent actions are identical.

The test list, or the starting portion of a list too long for the screen, is displayed. The user can specify the line where list execution begins; S begins execution at line 1, <CR> begins execution at the current line. The current line is indicated in the display by an asterisk (*) in the left margin. Initially, the current line is line 1; however, by entering another number, the user can change the current line to that number.

After a request for execution of the list, a check of the list is performed by SADIE, and the SADIE error logs are cleared. A line may contain a reference to a test not on the tape due to alteration or corruption of the tape. Any such line will be skipped in execution. The user is notified of any such line or of any line containing a test that would overwrite the disk or tape. If any line overwrites the disk, the user must verify that this is acceptable before execution of the list begins. If any lineprinter control line will cause printed output, and the lineprinter interface has not been identified during the session, SADIE prompts for the interface (Centronics or Data Products). During execution, if any test would overwrite the tape, the test is loaded, but execution halts until the SADIE tape has been replaced with a scratch tape. A request to reload the SADIE tape is not generated until SADIE next attempts to reposition the tape.

NOTE

> **When the SADIE tape is replaced, wait until it is FULLY REWOUND before responding with Y.**

During execution, SADIE updates an error log whenever an error is detected or a test lap is completed. Detailed error messages are logged in the cumulative error log. When execution is complete, enter a <CR> to return to the COMMAND LEVEL.

RUNNING A SINGLE TEST: Any test can be selected and run with the TEST command, T. The catalog of tests and control statements is presented. The selected one is displayed with default parameters set by SADIE. The default repetition count, 1, or thé parameters can be modified. When satisfied, enter a <CR> to change the test into a special one-line test list, and begin its execution. The list check and verification of any attempt to overwrite the disk or tape are then performed just as in running a regular test list. When the test is complete, enter a <CR> to return to COMMAND LEVEL.

To rerun a test previously chosen with the TEST command use the REPEAT command, R. It is now possible to modify the existing repetition count or parameters before executing the test.

DISPLAYING THE TEST LIST ERROR LOGS: The DISPLAY command, D, displays an error summary log generated by execution of a test or test list. This log contains a line for each line of the test list. Each error log line contains the test name, the number of times it was run, the number of hard and soft errors it detected, and an indication of whether or not the line was run to completion, and whether supplied parameters were found inappropriate and replaced.

The error log remains after the list is completed, as the only record of completion states of tests in the list.

The A command displays the cumulative error log for ALL tests in the test list. This log contains the last 20 detailed error messages logged by tests in the list. Each message contains the test name, the lap number, and error number, along with a description of the error. This error log also remains after completion of the test. Refer to Appendix A for the descriptions of the error messages logged by the tests.

EDITING AND EDIT LEVEL: The EDIT level is invoked by entering E, the Edit command. One test list resides in memory and can be altered using the EDIT LEVEL subcommands. This list can be a null list containing only a termination (EOL) line. A window of up to seven lines of the current list is displayed by the editor. One line of the display is the current line, marked by *, upon which line-oriented commands operate. Enter an integer to change the current line of the list.

Previously created and stored test lists can be moved to memory from the tape with the FETCH command, F. The memory resident list can be moved to tape with the STORE command, S. In either case, the test list catalog containing list descriptions is displayed, and the source or destination for the test list can be selected. When storing lists, the list description in the test list catalog can be changed.

The memory-resident list can be altered in five ways. The CLEAR command, C, erases the current list by making it a null list. The DELETE command, D, removes the current line from the list. The MODIFY command, M, allows the user to change the repetition count or parameters of the current line.

The REPLACE command, R, can be used to replace an entire line. The catalog of tests and control statements is displayed and the·user makes a choice. Default parameters and repetition count are supplied and the line is displayed. The repetition count or parameters can now be reset. When the line is correct, enter a <CR> to substitute the new line for the original line of the list and to return to the EDIT LEVEL.

The INSERT command, I, presents the catalog of tests and test lists and then offers an optional reset of default repetition count or parameters just as the R command does. Entering a <CR> when the line is correct, inserts the line in the test list and displays the catalog of tests for another insertion. Entering an E, (for EXIT) when the line is correct, inserts the line in the test list and returns to the EDIT LEVEL menu.

TAPE MAINTENANCE LEVEL: Three choices are available in the maintenance menu

The ADD command, A, queries the user until a catalog entry for a new diagnostic has been formed. The new entry is displayed, and the user may edit it. When the user signals that the entry is correct, the copy of the diagnostic library in memory is updated, and the diagnostic is appended to the diagnostic tape library.

NOTE

> **The new diagnostic must already exist on the SADIE tape at file 0, track d.**

The DELETE command, D, displays the catalog of tests. The selected test is removed from the diagnostic tape library, and the entry for it is removed from the copy of the copy of the diagnostic library catalog in memory.

The third choice returns to the COMMAND LEVEL. If the diagnostic library catalog has changed, the new tape is stored on tape.

QUITTING: The QUIT command, Q, rewinds and unloads the SADIE tape and returns to the PROM Monitor program.

5.14.2. START and RESET Interactions:

RESETTING

Pressing the RESET button forces a hardware reset. It performs the same function as the QUIT command. It is the only sure way to abort a malfunctioning diagnostic.

FORCING A PAUSE WITH THE START BUTTON: When the START button is pressed, a nonmaskable interrupt is generated, and SADIE responds by displaying a PAUSE menu with five choices. This is typically done when running a test or test list. The LIST command, L, displays the current test list. The DISPLAY command, D, displays the error log previously discussed. The ERROR command, E, displays a detailed error log for the current test if it maintains one. The A command displays a similar detailed error log, but this log is for ALL tests in the test list. The MONITOR command, M, brings up a menu of MONITOR commands. MONITOR functions are described in the following paragraph. The SKIP command, S, sets a flag which, when read by the test, causes it to cease execution. SADIE then runs the next line of the test list. Entering a ^ sets the same flag, but SADIE terminates execution of the test list when the test returns control to it. Entering a <CR> resumes the interrupted process.

NOTE

> **SADIE cannot always force an immediate abort of a test in progress.**

MONITOR COMMANDS: The MONITOR provides functions similar to those of the PROM Monitor. Specified locations in memory can be displayed or filled with any single word pattern. Blocks of memory can be moved, and as a special case, a multiword pattern can be replicated through a block of memory. Memory can be stepped through a word at a time, with optional alteration. Any I/O or special I/O port can be read and optionally written. Success of reading and writing depends on the hardware addressed.

5.15. Command Level Test Functions

Stand-alone diagnostics are provided for testing of all mainframe components. These diagnostics are supplied on the SADIE tape. Insert this tape into the tape drive and complete the bootstrap procedure (paragraph 5-11). After an introductory message, the COMMAND LEVEL menu will appear on the display screen as follows:

```
*****  COMMAND LEVEL *****

T       choose and run single TEST
R       REPEAT current single test
L       run current test LIST
C       CHOOSE and run a test list
E       EDIT test lists
```

```
D       DISPLAY error log
A       display cumulative error log of ALL tests in list
M       perform tape MAINTENANCE functions
Q       QUIT

Enter your choice ]=>
```

After this menu appears, the desired function can be selected by keying in its letter code in response to the prompt "Enter your choice]=>" and pressing <CR>.

The following descriptions are presented in the same sequence as they appear in the COMMAND LEVEL menu. If a function has subcommands, they are detailed before proceeding to the next description. This allows for all the information of one function to appear together for convenient reference.

5.15.1. Command Level T: Choose and run a single TEST:

General Description: A test can be chosen from the first CRT display.

A second menu displays the available control lines. in either case, once a test or control line is chosen, a subsequent CRT display allows for parameters and repetition count to be changed.

CRT Display Contents: ** CHOOSE A TEST OR CONTROL LINE ** Followed by a list of test numbers and corresponding titles

Optional Commands: Optional available commands for this display are:

Command	**Description**
+	CRT displays next page of available tests.
-	CRT displays preceding page of available tests.
1	CRT displays the test of control lines.
#	Select the test number of

	your choice.
^	Return to COMMAND LEVEL.

T SUBCOMMAND LEVEL: CHOOSE A CONTROL LINE

General Description: A control line is chosen from the menu. A second CRT display allows for parameters to be changed.

CRT Display Contents: **CHOOSE A CONTROL LINE** followed by a list of control line numbers and corresponding titles.

Optional commands: # selects the control line with that number

^ returns to the CHOOSE A TEST OR CONTROL LINE menu

RESET TEST LINE

General Description: The test or control line is shown with its current values for parameters and repetition count. Parameters and/or repetition count can be changed.

CRT Display Contents: ** RESET TEST LINE (OPTIONAL) **

Optional Commands: Optional available keyword commands for this display are:

Command	**Description**
R	Reset REPETITION count
P	Reset all PARAMETERS
#	Reset parameter #
I	Test name is INCORRECT--choose a different test
<CR>	Test line is correct
^	Return to CHOOSE A TEST

The P and # commands appear only when the test has parameters to be set. E appears only in insert mode. I appears only if this menu was reached immediately after the user chose a test.

T SUBCOMMAND LEVEL: Reset REPETITION Count

General Description: This function allows the user to specify the number of times a particular test function is to be run. The current value of the repetition count is displayed, followed by a prompt for a new value. A new value can be entered. Entering only <CR> leaves the count unchanged.

CRT Display Contents: ***** SET REPETITION COUNT *****

Optional Commands: Optional available commands in this mode are:

Command	**Description**
0	Test runs until START button is pressed
#	A number # of iterations expressed in decimal notation between 1 and 9999
^	Return to CHOOSE A TEST or CONTROL LINE (leave repetition count unchanged)

T SUBCOMMAND LEVEL: PARAMETERS

General Description: Parameters that specify the conditions under which a test is performed can be changed. The current value of a parameter is displayed, and the SADIE prompt for a new value of type hexadecimal or decimal. Entering only <CR> leaves the value unchanged.

CRT Display Contents: ***** SET PARAMETERS *****

```
      .    .
      .    .
```

Optional Commands: Optional available commands in this mode are:

Command	Description
#	A number # representing the value of the parameter.
^	Return to CHOOSE A TEST or CONTROL LINE.

5.15.2. Command Level R: REPEAT previously loaded single test:

General Description: A previously chosen single test is run. Parameters, repetition counts, and options can be reset before the test is re-executed.

CRT Display Contents: ** RESET TEST LINE (OPTIONAL) **

Optional Commands: Optional available keyboard commands in this mode are the same as T (choose and run a single TEST).

5.15.3. Command Level L: Run current test LIST:

General Description: Current test list (previously chosen or selected at edit level) is run.

First, the list is examined and, if it contains tests that overwrite the disk, a warning appears. It must be verified that this overwriting is permissible before execution of the list begins. If a test on the list does not appear in the catalog, this information is displayed and execution continues. If execution of the list will cause lines to be printed, and no prior lists have opened the lineprinter file, a menu of lineprinter interfaces appears. If a test writes to tape, execution of the list pauses after the test has been loaded until the user signals a scratch tape has been installed and the proper command is entered. Upon completion, the scratch tape is rewound and removed, and the SADIE tape should be re-installed.

CRT Display Contents: ***** RUN CURRENT TEST LIST ***** followed by the first seven lines of the test list at the top of the screen and a menu at the bottom.

Optional Commands: Optional available keyboard commands in this mode are:

Command	**Description**
<CR>	Execute list beginning at current line.
S	START execution at line 1
#	Make line # the CURRENT line (>1)
^	Return to COMMAND LEVEL

5.15.4. Command Level C: CHOOSE and run a test list:

General Description: The catalog of test lists is displayed and one is selected and fetched from the SADIE tape. A second menu then appears, which is the same as that described in section 5.15.1, except that ^ returns to the C command level, not the COMMAND LEVEL.

CRT Display Contents: ***** CHOOSE A TEST LIST ***** Followed by a list of test-list numbers, and short descriptions of test lists.

Optional Commands: Optional available keyboard commands in this mode are:

Command	Description
#	Make line # (>1) the current line
^	Return to COMMAND LEVEL

C SUBCOMMAND LEVEL: Run current test list

General Description The test list just chosen at the C command level is run.

CRT Display Contents: ***** RUN CURRENT TEST LIST ***** followed by the first seven lines of the test list at the top of the screen and a menu at the bottom.

Optional Commands Optional available keyboard commands are the same as those described in paragraph 5.15.1, except that ^ returns to the C command level, not the COMMAND LEVEL.

5.15.5. Command Level E: EDIT test list:

General Description: This level performs storage and list modification function options.

CRT Display Contents: ***** EDIT LEVEL *****
followed by one to seven lines of the current test list at the top of the screen and a menu at the bottom. The menu contains a test line number column, a test name column, a repetitions column, and four parameter columns. An asterisk (*) to the left of the test-line number indicates the current line.

Optional Commands: Optional available keyboard commands in this mode are:

Command	**Description**
C	CLEAR current test list
F	FETCH list from tape to current test list
S	STORE current test list
D	DELETE line x
R	REPLACE line x
M	MODIFY repetitions on parameters on line x
I	INSERT line(s) before line x
#	Make line # (<x+1) current line
^	Return to COMMAND LEVEL

E SUBCOMMAND LEVEL: C - CLEAR current test list

General Description: The current test list residing in RAM is deleted before creating a new test list.

CRT Display Contents: The EDIT TEST LIST is displayed, with the test list at the top of the screen showing only an EOL line.

Optional Commands: Not applicable

E SUBCOMMAND LEVEL: F - FETCH list from tape to test list:

General Description: The CRT displays a catalog of test lists stored on tape and a test list is selected and loaded into memory in preparation for desired modification.

CRT Display Contents: ***** FETCH LIST *****
Followed by a menu containing a list of the test lists.

Optional Commands: # the number corresponding to the test list to be fetched

^ to return to the EDIT level

E SUBCOMMAND LEVEL: S - STORE current test list:

General Description: The current catalog of test lists stored on the tape are displayed. The user chooses a list number where the current test list is to be stored. The current test list description is displayed and the user has the opportunity to enter a new description. Then the current test list overwrites the original test list on the tape and the test list catalog is updated.

CRT Display Contents: ***** STORE LIST *****
Followed by a menu containing a list of the 19 test lists.

Optional Commands: # the test list-number where the test should be stored.. SADIE prompts for a new test-list name of 1 to 39 characters.

^ to return to the EDIT level

E SUBCOMMAND LEVEL: D - DELETE line x:

General Description: Line is removed from test list.

CRT Display Contents: The EDIT level menu is displayed with the deleted line removed.

Optional Commands: Not applicable.

E SUBCOMMAND LEVEL: R - REPLACE line x:

General Description: A test or control statement, including parameters and repetition count, are chosen and installed in the test list that replaces the selected line contents.

CRT Display Contents: ** CHOOSE A TEST OR CONTROL LINE **

Optional Commands: The choices available are the same as those described in paragraph 5.15.1 under CHOOSE A TEST OR CONTROL LINE, except that ^ returns to the EDIT level.

on line x

General Description: The number of repetitions or value of any parameter can be changed.

CRT Display: ** RESET TEST LINE (OPTIONAL) **

Optional Commands: The choices available are the same as those described in paragraph 5.15.1 under CHOOSE A TEST OR CONTROL LINE, except that ^ returns to the EDIT level.

E SUBCOMMAND LEVEL: I - INSERT line(s) before line x:

General Description: A line is inserted before the selected line. The list is displayed again, centered above the originally selected line. This procedure continues as long as new lines are entered.

CRT Display: ** CHOOSE A TEST OR CONTROL LINE **

Optional Commands: The choices available are the same as those described in paragraph 5.15.1 under CHOOSE A TEST OR CONTROL LINE, except that ^ returns to the EDIT level, and there is an additional command, E, that Exits insert mode.

E SUBCOMMAND LEVEL: # - Make line # the current line:

General Description: Pointer is moved to line number #.

CRT Display: The EDIT level menu is displayed with an asterisk (*) next to the newly chosen current line. Testlist lines are centered around the new current line.

Optional Commands: Not applicable

5.15.6. Command Level D: DISPLAY error log:

General Description: The error log maintained in SADIE is displayed, including the test name, the number of times a test was executed, the number of errors reported by the test (organized by test list line), and the completion status of each test line. The error log is cleared prior to the start of execution of a test list or single test.

CRT Display Contents:

```
       #     HARD    SOFT
TEST  REPS  ERRORS  ERRORS  STATUS
```

Optional Commands: Not applicable

5.15.7. Command Level A: Cumulative error log - ALL tests in list:

General Description: 0 to 20 error messages are displayed.

CRT Contents:

```
TESTNAME LAP ERROR ERROR MESSAGE
followed by the error messages
```

Optional Commands: Not Applicable

5.15.8. Command Level M: do tape MAINTENANCE:

General Description — The user can modify the SADIE tape, adding or deleting tests.

CRT Contents: — *****PERFORM TAPE MAINTENANCE FUNCTIONS***** followed by a menu of commands

Optional Commands: — Optionally available commands for this display are:

Command	**Description**
D	DELETE a test from tape
A	ADD a test to tape
^	return to COMMAND LEVEL

5.15.9. Command Level Q: QUIT:

General Description: — The QUIT command terminates the SADIE program functions. The SADIE tape is rewound to the physical load point and system control is returned to the PROM Monitor.

CRT Display Contents: — When this command is complete, the Monitor message appears.

Optional Commands: — Not applicable

5.16. SADIE Test Line and Control Statements

5.16.1. SADIE Test List: Any test can be selected and run with the TEST command, T. When selected on the COMMAND LEVEL menu, a catalog of tests and control statements is displayed:

Appendix A contains detailed information of each test on the SADIE diagnostic tape. Control Statements are defined in the following paragraph.

5.16.2. Control Statements: Control statements can be inserted in test lists to modify list execution. Control statements are: GOTO, PAUSE, PRINT OPTIONS, and PAUSE-ON-ERROR OPTIONS.

GOTO

The GOTO statement is useful for setting up test loops. It causes the execution of the test list to jump to a specified line. A loop count of 0 to 9999 is specified. When the specified loop count is reached, control will fall to the next statement in the test list. When special value 0 is specified, GOTO will always be executed.

PAUSE

This statement causes the PAUSE menu to be displayed. SADIE programs are not disturbed; they are only temporarily halted to allow for optional action. The optional available commands are listed in the following PAUSE menu:

```
***** PAUSE *****

L       Display current test LIST

D       DISPLAY error log

E       Display detailed ERROR log

A       Display cumumative log of ALL tests in list

M       MONITOR function

S       SKIP to next line in test list

<CR>    RESUME current test

^       Return to COMMAND LEVEL
```

All the commands operate the same way, except the SKIP command, S, which causes the line of the test list following the PAUSE line to be skipped, rather than the PAUSE line itself.

PRINT OPTIONS

There are five print option statements that determine whether test output will be printed as well as being displayed. Normally, test error messages are displayed but not printed.

The print options are:

NOPRINT	(default)
PRINTMSG	only error and informative messages are printed
PRINTSUM	only lap summaries are printed
PRINTTOT	only the last lap summary is printed
PRINTALL	all test output is printed

PAUSE-ON-ERROR OPTIONS

All tests following a PAUSE-ON-ERROR option statement cause the test to halt when an error occurs, if the PAUSETST option is selected. The PAUSE-ON ERROR OPTIONS are:

NOPAUSE	(default)
PAUSETST	Display PAUSE menu on hard-error.

5.17. Using SADIE

The purpose of this example is to familiarize the user with the CRT displays and interaction with the console during SADIE execution.

Insert the SADIE tape into the tape drive and press the RESET button. The following CRT display appears:

```
    S8000 Monitor 1.2 - Press START to Load System
```

To initialize memory and execute power-up diagnostics, enter T<CR>.

To load the primary bootstrapper, enter the command:

```
    Z T <CR>
```

The primary bootstrapper displays:

```
    BOOTING FROM TAPE
```

When booting is complete, information pertaining to the current version of the diagnostic tape will appear on the CRT display. For example:

```
    SADIE        (Stand Alone Diagnostic Interactive Executive)
    Customer Release:    3.1        Released:  February 19, 1982
```

This display appears momentarily, followed by the COMMAND LEVEL menu:

```
    *****  COMMAND LEVEL  *****

    T    choose and run single TEST
    R    REPEAT current single test
    L    run current test LIST
    C    CHOOSE and run a test list
    E    EDIT test lists
    D    DISPLAY error log
    A    display cumulative log of ALL tests in list
    M    perform tape MAINTENANCE functions
    Q    QUIT

    Enter your choice ]=>
```

After this menu appears, the desired diagnostic function may be selected by simply entering its letter code in response to the prompt "Enter your choice]=>". For example, selecting T, followed by <CR>, produces:

```
*****  CHOOSE A TEST OR CONTROL LINE  *****
```

followed by a submenu of SADIE tests and the prompt:

```
Enter your choice ]=>
```

The CHOOSE A TEST OR CONTROL LINE submenu presents the catalog of tests and control statements available to the user.

In response to the prompt "Enter your choice]=>", enter the 1- or 2-digit test number in the first column, followed by <CR>. In the following example, the MMUTST5 is chosen. The following display appears:

```
***** RESET TEST LINE (OPTIONAL) *****

the test line is currently set as follows:

NAME       #REPS     PARAMETERS 1-4

MMUTST5      1       No parameters to set

R     to reset REPETITION count
I     test name is INCORRECT - choose different test
<CR> if test line is correct
^     to return to COMMAND LEVEL

Enter your choice ]=>
```

The MMUTST5 test line is displayed with default parameters set by SADIE. The default repetition count, 1, or the parameters can be modified. To reset the repetition count, enter R in response to the prompt "Enter your choice]=>", followed by <CR>. The following display appears:

```
***** SET REPETITION COUNT *****

Current repetition count = 1

0     to run test until NMI pressed
#     (positive # <=9999) to run test this many times
^     to return to RESET TEST LINE
<CR> to leave repetition count the same

Enter your choice ]=>
```

To modify the default or existing repetition count, following the prompt "Enter your choice]=>", enter the number of times the test is to run. In this case, the default repetition count, 1, is changed to 5. When <CR> is entered, the following display appears:

```
***** RESET TEST LINE (OPTIONAL) *****

The test line is currently set as follows:

NAME          #REPS     PARAMETERS 1-4
MMUTST5         5       No parameters to set

R     to reset REPETITION count
I     test name is INCORRECT - choose different test
<CR>  if test line is correct
^     to return to COMMAND LEVEL

Enter your choice ]=>
```

When satisfied with the repetition count, enter <CR> in response to the prompt "Enter your choice]=>". This changes the test into a special one-line test list. The list check and verification of any attempt to overwrite the tape is then performed just as in running a regular test list. During the list check and verification process, the following message is displayed:

```
*****  CHECKING TEST LIST...  *****

LINE 1: OK

*****  CHECK COMPLETE  *****
```

After the CHECK COMPLETE message, the following display appears:

```
*******************************************************
  The following test is now running
NAME      #REPS     PARAMETERS 1-4
MMUTST5     5       No parameters to set
*******************************************************

This is MMUTST5 - version 1.01

Now doing a block random data test on all MMUs.
Now testing SDRs with random data and random accesses.
Now testing control registers with random data.
Now testing READ ONLY flags in the data and stack MMUs.
Now testing the LIMIT flags of the stack and data MMUs.
Now testing the DIRW (direction) flag of the stack and
data MMUs.
Now testing TRANSLATION on the DATA MMU seg(1-63).
Now testing TRANSLATION on the STACK MMU seg(1-63).
Now testing TRANSLATION on the CODE MMU seg(1-63).
```

As the first repetition of MMUTST5 progresses, the test line for each MMU register or function is displayed, in sequence, until the test is complete. If an error occurred during the test, error messages will be displayed immediately following the last test line.

When the test repetition is complete, the following lap count summary appears:

```
This is MMUTST5 - version 1.01
   LAPCNT=1 ERROR COUNT=0

CODE MMU ERRORS
SDR: BLOCK 0     CONTROL: SAR  0   FLAGS: RD  0   TRANS: 0
     RNDM  0              DSC  0          LIM 0
                          MODE 0          DIR 0

DATA MMU ERRORS
SDR: BLOCK 0     CONTROL: SAR  0   FLAGS: RD  0   TRANS: 0
     RNDM  0              DSC  0          LIM 0
                          MODE 0          DIR 0

STACK MMU ERRORS
SDR: BLOCK 0     CONTROL: SAR  0   FLAGS: RD  0   TRANS  0
     RNDM  0              DSC  0          LIM 0
                          MODE 0          DIR 0
```

The MMUTST5 test messages and lap count summary screens appear the number of times set in the SET REPETITION COUNT display. When the last repetition is complete, the message

```
Hit <CR> to return to COMMAND LEVEL ]=>
```

appears immediately following the last lap count summary.

When the COMMAND LEVEL menu appears, any of the diagnostic functions may be selected. To facilitate this example, entering the QUIT command, Q, in response to the prompt "Enter your choice]=>" terminates the SADIE diagnostic functions. The tape is rewound to the physical load point and system control is returned to the PROM Monitor.

APPENDIX A
SADIE TEST DESCRIPTIONS

This appendix gives detailed information for the diagnostic tests contained on the System 8000 Diagnostic Tape, SADIE (Part Number 14-0009-05). The following menu is displayed when SADIE is loaded. Some tests do not apply to Models 21 and 31 and do not appear in this appendix.

```
WDCCRC      SMDFMT      NEWMEM3
WDCFMT      SMDMEDIA    MMUTST5
WDCMEDIA    SMDCRC      CENT.PRT
WDCTST3     SMDTEST     DP.PRT
WDCTST7     SMDMON      S16SIO
WDCMON      TCUMON      SIOMODEM
MDCCRC      TCOM        ECCTEST
MDCFMT      TEX         MTCMON
MDCMEDIA    NEWMEM1     MTCON
MDCTEST     NEWMEM2
```

TEST NAME

WDCCRC - a non-destructive verification of the Winchester Disk Data, Cyclic Redundancy Checks (CRCs)

PARAMETERS

Disk drive to be tested (default=0)

DESCRIPTION

WDCCRC reads all tracks of the selected drive. If an error is detected, a message is displayed and logged. WDCCRC repeats n times, where n is the #REPS in the test line.

ERROR MESSAGES

WDCCRC can issue the following error messages

where:

```
      u = unit number (decimal)
    ccc = cylinder number (decimal)
      h = head number (decimal)
     ss = sector number (decimal)
     dr = contents (hex) of drive ready register
     ds = contents (hex) of disk status register
     oe = contents (hex) of operation error status register
     os = contents (hex) of other status register
     xx = value (hex) of command issued to the WDC
```

DISK NOT RESPONDING DURING INIT

WDC failed to respond within a reasonable time after the test began.

INVALID COMMAND DISK=(u,ccc,h,ss) COMMAND ISSUED=xx

Invalid command opcode or out-of-range command parameters.

HARD TRK READ DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

An uncorrectable error occurred during track read operation.

```
SOFT TRK READ  DISK=(u,ccc,h,ss)  DR=dr DS=ds OE=oe OS=os
```

A correctable error occurred during track read operation.

LAP SUMMARY

A lap summary is displayed when each repetition of WDCCRC is completed. The lap summary includes:

- test name
- lap number
- total number of errors
- number of times each WDC command was issued during the test
- tally of the number of times each of the following status bits was returned by the disk controller when a soft error occurred:

TOT	-- total number of errors
DAT	-- CRC errors
POS	-- sector not found
FOR	-- format error (sector header field error)
NR	-- unit not ready
SVE	-- servo error
RWF	-- read or write fault
SPE	-- speed error
PL	-- power loss
WPT	-- write protected
DSE	-- seek error
NCL	-- not on cylinder
GB	-- guard band error
PLE	-- PLO error
UNS	-- unsafe
DCE	-- invalid command
DTO	-- timeout error
P/M	-- POR/MR
ADE	-- address error
DF	-- drive fault
NOL	-- drive not on-line
CTO	-- controller operation timed out
CWP	-- write protect error

VF -- verify failure
BD -- bad disk: excessive media defects
CRDT -- can't read defect table
ME -- map error

⊕ -- a tally of hard errors similar to the soft error tally described previously, but with these additions:

INVC -- invalid command
CMP -- compare error
INIT -- WDC not responding during initialization

For detailed information on the disk controller commands and status registers, refer to the Winchester Disk Controller Hardware Reference Manual (03-3203).

NOTES

WDCCRC does not recognize when the drive selected for testing is off-line, or when the disk medium is not present in the drive. WDCCRC will display an error message for each track it attempts to read on the nonexistent drive. The same results occur when the drive and disk medium are present but the disk is locked. To recover, press the system START button, then respond with "^" to the PAUSE menu.

TEST NAME

WDCFMT - data-destructive formatting of the entire disk

PARAMETERS

Disk drive to be tested (default=0)

DESCRIPTION

WDCFMT formats every sector of the disk. This is accomplished by issuing the format command, FMT, to the Winchester Disk Controller (WDC). WDCFMT reformats the disk n times, where n = #REPS in the test line. After each repetition, WDCFMT displays the defect map. If cylinder-sparing WDC firmware is installed, a list of defective cylinders is displayed. Otherwise, a list of defective sectors is displayed.

ERROR MESSAGES

WDCFMT can issue the following error messages

where:

u = unit number (decimal)
ccc = cylinder number (decimal)
h = head number (decimal)
ss = sector number (decimal)
dr = contents (hex) of drive ready register
ds = contents (hex) of disk status register
oe = contents (hex) of operation error status register
os = contents (hex) of other status register
xx = value (hex) of command issued to the WDC

DISK NOT RESPONDING DURING INIT

WDC failed to respond within a reasonable time after the test began.

INVALID COMMAND DISK=(u,ccc,h,ss) COMMAND ISSUED=xx

Invalid command opcode or out-of-range command parameters.

HARD BAD DISK DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

Number of bad sectors/cylinders exceeds the number of spare sectors/cylinders. The disk is not useable.

HARD FORMAT ERR DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

WDC returned an uncorrectable error status other than HARD BAD DISK.

SOFT FORMAT ERR DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

Error during format was corrected on retry.

HARD ERROR: ENTERING CEMODE DR=dr DS=ds OE=oe OS=os

WDC returned an uncorrectable error status after a CEMODE command was issued.

SOFT ERROR: ENTERING CEMODE DR=dr DS=ds OE=oe OS=os

WDC returned a correctable error status after a CEMODE command was issued.

HARD ERROR: READING DEFECT MAP DR=dr DS=ds OE=oe OS=os

WDC returned an uncorrectable error status after a READ command was issued to cylinder 0. WDCFMT was trying to read a copy of the defect map.

SOFT ERROR: READING DEFECT MAP DR=dr DS=ds OE=oe OS=os

WDC returned a correctable error status after a READ command was issued to cylinder 0. WDCFMT was trying to read a copy of the defect map.

HARD ERROR: EXITING CEMODE DR=dr DS=ds OE=oe OS=os

WDC returned as uncorrectable error status after an RDDT (read-defect-table) command was issued to force the WDC out of CEMODE.

SOFT ERROR: EXITING CEMODE DR=dr DS=ds OE=oe OS=os

WDC returned a correctable error status after an RDDT (read-defect-table) command was issued to force the WDC out of CEMODE.

LAP SUMMARY

At the end of each test repetition, a lap summary as described under the WDCCRC LAP SUMMARY is displayed. Following the lap summary is the defect table. If cylinder sparing firmware is installed on the controller board, a list of bad physical cylinders is shown. Otherwise, a list of bad physical sectors is displayed in the following format: (cc,h,ss), where cc = cylinder number (decimal); h = head number (decimal); ss = sector number (decimal).

NOTES

If the START button is pressed while the FMT command is being executed, requests by the user to skip to the next test line, or return to the COMMAND LEVEL, will not be honored until the FMT command completes execution. The FMT command runs for approximately 90 minutes. To stop this test, press the RESET button.

If the drive selected for testing does not exist, the FMT command aborts promptly with a hard error.

TEST NAME

WDCMEDIA - data-destructive, write-read-compare test of the Winchester disk, using several data patterns

PARAMETERS

Parameter 1 - disk drive to be tested (default=0)
Parameter 2 - compare pattern read with pattern written.
If compare = 1 (default), do compare step.
Otherwise, skip compare step.
Parameter 3 - pattern use. If pattern =1 (default), use all
patterns. Otherwise, use only "worst case" pattern

DESCRIPTION

WDCMEDIA exercises the disk medium by writing three separate data patterns to each sector of the medium. After each data pattern is written, the disk is read for verification. All errors incurred while writing or reading a data pattern are displayed. The three data patterns are:

1. AAAA
2. FFFF
3. The "worst case" pattern B6DB6DB6DB6D.

Each pattern is written to every track of the medium. All errors writing or reading a pattern are displayed.

ERROR MESSAGES

WDCMEDIA can issue the following error messages

where:

pppp= data pattern (hex) read or written to the track
u = unit number (decimal)
ccc = cylinder number (decimal)
h = head number (decimal)
ss = sector number (decimal)
dr = contents (hex) of drive ready register
ds = contents (hex) of disk status register
oe = contents (hex) of operation error status register
os = contents (hex) of other status register
xx = value (hex) of command issued to the WDC
gggg= "good" data (hex) written to the track
bbbb= "bad" data (hex) read from the track

DISK NOT RESPONDING DURING INIT

WDC failed to respond within a reasonable time after the test began.

INVALID COMMAND DISK=(u,ccc,h,ss) COMMAND ISSUED=xx

Invalid command opcode or out-of-range command parameters.

HARD TRK WRITE pppp DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

Invalid command opcode or out-of-range command parameters.

SOFT TRK WRITE pppp DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

A correctable error occurred during track write.

HARD TRK READ pppp DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

An uncorrectable error occurred during track read.

SOFT TRK READ pppp DISK=(u,ccc,h,ss) DR=de DS=ds OE=oe OS=os

A correctable error occurred during track read.

COMPARE ERROR DISK=(u,ccc,h,ss) GOOD DATA=gggg BAD=bbbb

A mismatch was found in comparing the "good" buffer written to the disk with the "bad" buffer read from the disk.

LAP SUMMARY

At the end of each test repetition, a lap summary is displayed showing cumulative statistics for all repetitions. Refer to the description of LAP SUMMARY of the WDCCRC test.

TEST NAME

WDCTST3 - a random queue test of the Winchester disk

PARAMETERS

disk drive to be tested (default=0)

DESCRIPTION

WDCTST3 creates a 128 element queue. Each queue member has the following three addresses:

1) a source buffer address
2) a destination buffer address
3) a disk address

Each source and destination buffer is 512 bytes long. The source buffers are in memory segment 1, and the destination buffers are in memory segment 2.

Initially, the test randomizes the source buffer and disk addresses of the 128 queue elements. For each element, the source buffer contents are written to the disk sector.

Next the test reads the sector at the disk address of each element into the destination buffer. The source and destination buffers are then compared.

Finally, the disk addresses are again randomized to prevent subsequent test repetitions from testing the same disk sectors.

WDCTST3 repeats n times, where n=#REPS in the test line.

ERROR MESSAGES

WDCTST3 can issue the following error messages

where:

```
   u = unit number (decimal)
 ccc = cylinder number (decimal)
   h = head number (decimal)
  ss = sector number (decimal)
  dr = contents (hex) of drive ready register
  ds = contents (hex) of disk status register
  oe = contents (hex) of operation error status register
  os = contents (hex) of other status register
  xx = value (hex) of command issued to the WDC
gggg= "good" data (hex) written to the sector
bbbb= "bad" data (hex) read from the sector
```

DISK NOT RESPONDING DURING INIT

WDC failed to respond within a reasonable time after the test began.

INVALID COMMAND DISK=(u,ccc,h,ss) COMMAND ISSUED=xx

Invalid command opcode or out-of-range command parameters.

HARD SEC WRITE DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

An uncorrectable error occurred during sector write.

SOFT SEC WRITE DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

A correctable error occurred during sector write.

HARD SEC READ DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

An uncorrectable error occurred during sector read.

SOFT SEC READ DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

A correctable error occurred during sector read.

COMPARE ERROR DISK=(u,ccc,h,ss) GOOD DATA=gggg BAD=bbbb

A mismatch was found in comparing the "good" buffer written to the disk with the "bad" buffer read from the disk.

LAP SUMMARY

WDCTST3 tallies a number of statistics whenever a hard or soft error occurs. At the end of each test repetition, these statistics are displayed in tabular form. The lap summary table is described under LAP SUMMARY of the WDCCRC test description.

NOTES

None

TEST NAME

WDCTST7 - a comprehensive, multisector, write-read-compare test of the Winchester disk

PARAMETERS

disk drive to be tested (default=0)

number of test loops equals the number of iterations per lap (default= 10)

DESCRIPTION

WDCTST7 exercises the Winchester Disk Controller by doing large, variable-sized writes and reads to random disk addresses. The test repeats n times, where n=#REPS in the test line. Each test contains the following "loop" iterations:

(1) Fill segment 1-3 with "AAAA"s.

(2) Fill a randomly chosen source buffer of random length with random data.

(3) Write the source buffer into a random disk address.

(4) Read from the disk into a destination file.

(5) Compare the source and destination buffers.

(6) Fill the buffers with "AAAA"s.

(7) Check segments 1-3 for all "AAAA"s to verify that the disk transfers did not corrupt locations outside the buffers.

ERROR MESSAGES

WDCTST7 can issue the following error messages

where:

```
      u = unit number (decimal)
    ccc = cylinder number (decimal)
      h = head number (decimal)
     ss = sector number (decimal)
```

```
  dr = contents (hex) of drive ready register
  ds = contents (hex) of disk status register
  oe = contents (hex) of operation error status register
  os = contents (hex) of other status register
  xx = value (hex) of command issued to the WDC
 llll= number (hex) of words in the random-length buffer written
       or read
 gggg= "good" data (hex) written to the track
 bbbb= "bad" data (hex) read from the track
```

DISK NOT RESPONDING DURING INIT

WDC failed to respond within a reasonable time after the test began.

INVALID COMMAND DISK=(u,ccc,h,ss) COMMAND ISSUED=xx

Invalid command opcode or out-of-range command parameters.

HARD WRITE-llll DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

An uncorrectable error occurred during a random-length write operation. llll is replaced by the number of words (hex) to be written.

SOFT WRITE-llll DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

A correctable error occurred during a random-length write operation. llll is replaced by the number of words (hex) to be written.

SOFT READ-llll DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

A correctable error occurred during a random-length read operation. llll is replaced with the length in words (hex) of the buffer.

HARD READ-llll DISK=(u,ccc,h,ss) DR=dr DS=ds OE=oe OS=os

An uncorrectable error occurred during a random-length operation. llll is replaced with the length in words (hex) of the buffer.

COMPARE ERROR DISK=(u,ccc,h,ss) GOOD DATA=gggg BAD=bbbb

> A mismatch was found in comparing the "good" buffer written to the disk with the "bad" buffer read from the disk.

LAP SUMMARY

WDCTST7 tallies statistics whenever a hard or soft error occurs. At the end of each test repetition, the statistics are displayed in 'tab' form. Refer to diagnostic test WDCCRC for a description of the statistical table.

NOTES

WDCTST7 requires four segments of memory.

TEST NAME

WDCMON

PARAMETERS

None

DESCRIPTION

WDCMON is an interactive monitor for the Winchester Disk Controller. The user interacts with the monitor by issuing commands and parameters in response to the prompt: Command? In order to run this test, the user must know the following:

(1) Commands are entered in upper case.

(2) Numeric parameters are interpreted by the monitor as decimal, unless the digits are followed by an H (which indicates a hexadecimal number).

Example: 100H = 256.

(3) Omitted parameters take on their previous values.

```
Example:  Command?
          READ 1 1000H 100H 0 1 0
          Command?
          READ
```

The second READ and the first READ are identical commands.

(4) A number preceding a command is a repeat factor.

(5) Commands can be nested within command lines by using parenthesis. Each command on a command line must be parenthesized (unless there is only one command).

```
Example:  Command?
          READ 1 1000H 100H 0 1 0
          Command?
          10 ((READ) (ISEC 1))
```

The first command reads from cyl:0, head:1, sector:0, 100H words into segment 1, offset 1000H.

The second command does the following:

a. Reads 100H words from the disk address into the memory buffer.

b. Increments the sector number by 1 (carrying into head and cylinder numbers).

c. Decrements the loop counter (initially 10), and returns to step "a" if it is still positive.

(6) If the user forgets the available commands, the command HELP will display the entire list of commands and parameters.

ERROR MESSAGES

WDCMON can issue the following error messages

where:

```
  u = unit number (decimal)
ccc = cylinder number (decimal)
  h = head number (decimal)
 ss = sector number (decimal)
 dr = contents (hex) of drive ready register
 ds = contents (hex) of disk status register
 oe = contents (hex) of operation error status register
 os = contents (hex) of other status register
 xx = value (hex) of command issued to the WDC
```

```
HARD ERR CMD=xx DISK=(u,ccc,h,ss) DS=ds DR=dr OE=oe OS=os
```

An uncorrectable error occurred when command xx was issued.

```
SOFT ERR CMD=xx  DISK=(u,ccc,h,ss) DS=ds DR=dr OE=oe OS=os
```

A correctable error occurred when command xx was issued.

LAP SUMMARY

WDCMON is an interactive monitor; it does not count laps. It does, however, maintain a table of statistics, as described under LAP SUMMARY of the WDCCRC test description. This table is displayed by entering the REC command (see below). It is also displayed by entering E in the SADIE PAUSE menu.

NOTES

WDCMON does not protect the user against commands that will destroy the memory resident SADIE code, the WDCMON code, the SADIE test catalog, and the test list catalog. These reside in segment 0.

COMMAND DESCRIPTIONS

The following list of commands are accepted by WDCMON. Command names are indicated in uppercase letters; parameters are indicated in lowercase letters.

COMMAND: BADCYL/BADSEC

Description

BADCYL reads the defect table residing at physical cylinder 0, and displays a list of bad physical cylinders. BADCYL should only be used with cylinder-sparing firmware installed on the WDC board.

BADSEC reads the defect table residing at physical cylinder 0, and displays a list of bad physical sectors. BADSEC should only be used with sector-sparing firmware installed on the WDC board. The Model 21 WDC board has sector-sparing firmware.

Example

```
BADCYL
```

COMMAND: CEC/REC

Description

CEC clears all error counters; REC displays all error and instruction counters. The counters are displayed in the format shown under LAP SUMMARY of the WDCCRC Test Description.

Example

```
REC
CEC
```

COMMAND: CEMODE, EXITCE

Description

CEMODE enters Customer Engineer mode. This disables defect mapping, and subsequent disk accesses will be to physical, not logical, cylinders. EXITCE exits Customer Engineer mode, enabling defect mapping.

Example

```
CEMODE
EXITCE
```

COMMAND: CLC/ILC/RLC

Description

CLC clears the lap counter; ILC increments the lap counter by 1; and RLC displays the lap counter.

Example

```
CLC
ILC
RLC
```

COMMAND: CMP srcseg srcoff desseg desoff count

Description

This command compares two buffers. Count is a word count.

Example

```
CMP 1 0 2 0 8000H
```

COMMAND: DISP dseg doff dlnth

Description

DISP displays dlnth words beginning at segment dseg and offset doff. Addresses appear at the left margin of the display. Words are displayed in hexadecimal.

Example

```
DISP 1 0 100H
```

COMMAND: DISPRT/ENPRT

Description

DISPRT disables the printing of operational messages from the monitor. ENPRT enables the printing of operational messages from the monitor.

Example

```
DISPRT
```

COMMAND: FMT

Description

This command will format the selected drive. This will destroy any data on the medium.

Example

```
FMT
```

COMMAND: FRD rseg roff cyl head sec

Description

This command is the same as a read, except all disk format header and crc information is also transferred. This reads the exact image of a disk sector into the host memory. This command transfers only one sector.

Example

```
FRD 2 0 1 2 14H
```

COMMAND: HALT/NOHALT

Description

HALT enables all subsequent CMP errors to cause a halt until a <cr> is entered. NOHALT inhibits CMP to halt on an error.

Example

```
HALT
NOHALT
```

COMMAND: HELP

Description

HELP displays a list of the WDCMON commands and parameters.

Example

```
HELP
```

COMMAND: HOME unit

Description

This command will home the selected drive to cylinder 0, and clear any drive fault.

Example

```
HOME 0
```

COMMAND: INHIS/ENAIS

Description

INHIS sets the "inhibit implicit seek" bit in each command to the WDC. ENAIS clears the "inhibit implicit seek" bit.

Example

```
INHIS
ENAIS
```

COMMAND: INHRTY/ENTRY

Description

INHRTY sets the "inhibit retry flag" in each command to the WDC. ENTRY clears the "inhibit retry flag". WDC.

Example

```
INHRTY
ENTRY
```

COMMAND: ININT/ENINT

Description

ENINT sets the "interrupt enable flag" in each command to the WDC. ININT clears the interrupt enable flag.

Example

```
ENINT
ININT
```

COMMAND: INV N

Description

This command forces any value of N as a command to the WDC. N must be less than 100H.

Example

```
INV 23H
```

COMMAND: IROFF, IWOFF, ILNTH, ICYL, IHEAD, ISEC, IDOFF value

Description

These commands are issued in the following format:

```
Ixxxx value
```

The selected variable is incremented by "value". Incrementing SEC beyond the maximum number of sectors for the particular disk model causes a carry over to the head number. Incrementing HEAD beyond the maximum number of heads for the particular disk model causes a carry over into the CYLinder number. Incrementing ROFF or WOFF beyond FFFH causes a carry over into RSEG or WSEG.

Example

If the disk model under test is a BASF Drive and CYL=0 HEAD=2 SEC=3, after entering ISEC 1, CYL=0 HEAD=2 SEC=4. If CYL=0 HEAD=2 SEC=23, after entering ISEC1, CYL=1 HEAD=0 SEC=0.

COMMAND: NULL

Description

NULL sends a NOP command to the WDC.

Example

```
NULL
```

COMMAND: Q, QUIT

Description

QUIT exits WDCMON

Example

```
QUIT
Q
```

COMMAND: RAND rndseg rndoff rlnth

Description

This fills rlnth WORDs of memory, starting at <rndseg>rndoff, with random data.

Example

```
RAND 1 0 8000H
```

COMMAND: RCYL, RHEAD, RSEC, RALL

Description

RCYL sets CYL=random value from 0 to the maximum number of cylinders for the disk model - 1 , inclusive. RHEAD sets HEAD=random value from 0 to the maximum number of heads for the disk model - 1, inclusive. RSEC sets SEC=random value from 0 to the maximum number of sectors for the disk model - 1, inclusive. RALL does RCYL, RHEAD and RSEC all in one command.

Example

```
RCYL
RHEAD
RSEC
RALL
```

COMMAND: RDDT, RBDT

Description

RDDT issues a "read defect table" command to the WDC. RBDT issues a "rebuild defect table" command to the WDC.

Example

```
RDDT
RBDT
```

COMMAND: READ rseg roff lnth cyl head sec

Description

This command reads (lnth) words of data from unit#(unit) into memory <rseg>roff. The first disk address accessed is cyl-head-sec.

Example

```
READ 1 1000H 100H 23 2 20
```

COMMAND: SEEK cyl

Description

This command does an explicit seek, for the cylinder specified, on the selected unit.

Example

```
SEEK 240
```

COMMAND: SETOFF offset

Description

This command sets the strobe offset to the given value.

Example

```
SETOFF 1
```

COMMAND: SROFF, SWOFF, SLNTH, SCYL, SHEAD, SSEC, SUNIT value

Description

These commands are entered in the following format: Sxxxx value

The selected variable is initialized to the given "value".

Example

```
SROFF 1000H   sets the read offset (roff) to 1000 H.
SCYL 500    sets the cylinder to 500.
```

COMMAND: SRSEG, SWSEG segnum

Description

SRSEG sets the read segment number to the given segment number (segnum). SWSEG, sets the write segment number to the given "segnum".

Example

```
SRSEG 1
SWSEG 3
```

COMMAND: STAT

Description

STAT returns the detailed disk controller status registers.

Example

```
STAT
```

COMMAND: WP, UNPROT unit

Description

WP does a software write protect on the selected unit. UNPROT disables the WP command.

Example

```
WP 0
UNPROT 1
```

COMMAND: WRITE wseg woff lnth cyl head sec

Description

This command writes (lnth) data words to the selected unit from memory <rseg>roff. The disk address is cyl-head-sec.

Example

```
WRITE 2 1024 100H 2 0 20
```

INTRODUCTION TO SMD DIAGNOSTICS

There are five diagnostic programs on the SADIE tape that address the Zilog Storage Module Drive (SMD) controller and disk:

SMDCRC - a non-destructive, read-only disk test (sometimes referred to as a format-verify test)

SMDFMT - a data-destructive disk formatting program

SMDMEDIA - a data-destructive disk media test

SMDTEST - two data-destructive random SMD controller tests

SMDMON - a monitor that allows an informed operator to issue single commands and sequences of commands to the SMD controller

These five diagnostic programs are explained separately in the pages that follow this introduction. Since the diagnostics share a common start-up sequence and also share many error messages, these common items are explained in the introduction. Also included in the introduction is a brief outline of the SMD controller-to-host interface.

SMD CONTROLLER-HOST INTERFACE DATA STRUCTURES

The SMD controller has a 16-bit write-only Command Register and a 16-bit read-only Status Register. They share a common I/O address. Ending status for typical disk commands issued in interrupt mode is also reported to the host in the Interrupt Acknowledge Vector. These registers and the Interrupt Acknowledge Vector have the following fields:

Command Register:

```
         15                      8  7  6  5  4  3  2     0
        +-----------------------+--+--+--+--+--+---------+
  CR:   | DATA (depends on CMD) |IN|RI|DI|EI|WK|  CMD    |
        +-----------------------+--+--+--+--+--+---------+

  CR:CMD  --   0 = No operation
               1 = Read Packet Addresses from Dispatch Table
               2 = Not defined
               3 = Not defined
               4 = DATA is Dispatch Table Address l.s. byte
               5 = DATA "       "        "       "    middle byte
               6 = DATA is Dispatch Table Address m.s. byte
```

7 = DATA is Interrupt Vector

CR:WK -- WAKEUP controller; service all packets with Packet Status set to GO

CR:EI -- Enable Interrupts (reset by IU of Status Register)

CR:DI -- Disable Interrupts

CR:RI -- Reset IP and IU of Status Register

CR:IN -- Begin SMD controller initialization and self-test

CR:DATA -- 1 byte of data, whose value is dependent upon the contents of CR:CMD

Status Register:

```
         15 14  13                  8  7  6  5  4  3  2  1  0
        +-----+------------------+--+--+--+--+--+--+--+--+
    SR: | DRV |ES (ending status)|ND| 0| 0| 0| 0|IP|IU|BZ|
        +-----+------------------+--+--+--+--+--+--+--+--+
```

SR:BZ -- Controller busy servicing CR:CMD

SR:IU -- Interrupt Under Service

SR:IP -- Interrupt Pending

SR:ND -- No Dispatch Table Address or Interrupt Vector sent to controller after CR:IN

SR:ES -- Packet Command ending status after CR:WK, or controller self-test ending status after CR:IN For packet command ending status codes, see description of Packet below.

Self-test ending codes:
8 = 2910 sequencer error
9 = 2901 ALU error
A = controller internal memory error

SR:DRV -- Disk Drive number (0 - 3)

Interrupt Acknowledge Vector:

```
         15 14  13                  8  7                      0
        +-----+------------------+-----------------------+
    IV: | DRV |ES (ending status)|   Interrupt Vector     |
        +-----+------------------+-----------------------+
```

IV:ES -- Packet Command ending status after CR:WK See definition of Packet for description of packet ending-status codes.

IV:DRV -- Drive Number

In addition to these registers, the host communicates with the controller via a Dispatch Table and Packets (one per disk unit) in host memory. All disk-control commands and command-completion reports go through this channel. The organization of these structures is:

Dispatch Table:

```
          15                                               0
          +------------------------------------------------+
  DT: + 0¦          PS0 (Drive 0 Packet Status)            ¦
          +------------------------------------------------+
          /          .                         .           /
          /          .                         .           /
          +------------------------------------------------+
         6¦          PS3 (Drive 3 Packet Status)           ¦
          +------------------------------------------------+
         8¦  PACKET 0 ADDRESS M.S.WORD                     ¦
          +------------------------------------------------+
         A¦  PACKET 0 ADDRESS L.S.WORD                     ¦
          +------------------------------------------------+
          /          .                         .           /
          /          .                         .           /
          +------------------------------------------------+
        14¦          PACKET 3 ADDRESS M.S.WORD             ¦
          +------------------------------------------------+
        16¦          PACKET 3 ADDRESS L.S.WORD             ¦
          +------------------------------------------------+
```

```
  DT:PSn  --  Packet Status for packet #n:
              0 = IDLE (set by host)
              1 = GO   (set by host when packet ready to go)
              2 = BUSY (set by controller after packet is read)
              3 = DONE (set by controller after packet command
                  is completed)
```

Packet:

```
            15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
           +--------------+--+--+--+-----------------------+
  + 0  CM: ¦ 0 ---------0 ¦NR¦NE¦NO¦ PCMD (packet command) ¦
           +--------------+--+--+--+-----------------------+
    2  ST: ¦ 0 -----------------------0 ¦ES(ending status)¦
           +----------------------------+--------+--+--+--+
    4  SB: ¦ 0 ------------------------------ 0 ¦RZ¦RT¦EC¦
           +-----------+-------------+--+--+--+--+--+--+--+--+
    6  DS: ¦    SKE    ¦     SEL     ¦ x¦ x¦BZ¦RO¦FT¦SE¦OC¦RY¦
           +-----------+-------------+--+--+--+--+--+--+--+--+
     8    CT: ¦              Byte or Sector Count                ¦
              +--------------------------------------------------+
     A    AH: ¦              DMA Address Bits 23-16              ¦
              +--------------------------------------------------+
     C    AL: ¦              DMA Address Bits 15-0               ¦
              +--------------------------------------------------+
     E    UN: ¦              Unit Number                         ¦
              +--------------------------------------------------+
    10    CY: ¦              Cylinder Number                     ¦
              +--------------------------------------------------+
    12    HD: ¦              Head Number                         ¦
              +--+--+-------+------------------------------------+
    14    VS: ¦FS¦NW¦       ¦ Head Bias and Volume Select        ¦
              +--+--+-------+------------------------------------+
    16    SC: ¦              Sector Number                       ¦
              +--+--+-------+------+--+--+--------+--+--+--+--+--+
    18    OF: ¦ 0 -------------- 0 ¦SL¦SE¦ 0 -- 0 ¦O-¦O+¦ 0¦ 0¦
              +--------------------+--+--+--------+--+--+--+--+--+
    1A - 1F   ¦              reserved                            ¦
              +--------------------------------------------------+
               15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
```

CM:PCMD -- Packet Command:

Opcode	Name	Parameters and Operation
0	NOP	Firmware rev. returned in CT field
1	WRAM	Parameters: CT, AH, AL Write to controller board RAM
2	RRAM	Parameters: CT, AH, AL Read controller board RAM
3	SELECT	Parameters: UN Select drive
4	PRISEL	Parameters: UN Priority Select (dual-access only)
5	PRIREL	Parameters: UN Release Priority (dual-access only)
6	RESET	Parameters: UN Reset Fault on drive
7	SEEK	Parameters: UN, CY Seek to specified cylinder (if CY field = -1, rezeroes drive AND resets fault)
8	FMT	Parameters: UN,CT,AH,AL,CY,HD,SC Format track (host supplies sector header data in buffer defined by AH and AL; CT is # sectors))
9	WLONG	Parameters: UN,CT,AH,AL,CY,HD,SC Write Long (data plus 4 ECC bytes; controller does not compute ECC).
A	WRITE	Parameters: UN,CT,AH,AL,CY,HD,SC

		Write data
C	FRD	Parameters: UN,CT,AH,AL,CY,HD,SC Format Read track (controller reads and transmits sector IDs; CT is # sector IDs to read).
D	RLONG	Parameters: UN,CT,AH,AL,CY,HD,SC Read Long (controller reads data and 4 bytes of ECC)
E	READ	Parameters: UN,CT,AH,AL,CY,HD,SC Read data
F	SIZE	Parameters: UN Size Disk if formatted (controller returns size in CY, HD, SC fields; returns bad ending status if it cannot size disk.)

CM:NO -- No offsets during retries
CM:NE -- No Error Correction
CM:NR -- No Retries; Use data in CM:OF for strobe timing

ST:ES -- Packet Command Ending Status:

Code	Name and optional description
0	No Error OR Soft Error (NOTE: if ES = 0, but any of SB bits are set, there was a SOFT error; if ES = 0 and all SB bits are clear, NO error occurred.)
1	Initialization error Initialization sequence incomplete
2	Sector Overrun
3	DMA Error - Memory Parity Error (NOTE: in case of a DMA Error, controller does NOT update packet; only Status Register will have this code in ES field)
4	Select Error - 0 drives or multiple drives selected
5	Byte/Sec Count Error - Odd byte count or sector count = 0
6	Dual-Access Busy Err
7	Rezero error - A rezero did not clear fault
8	Drive Status - One or more bad SMD status bits in DS
9	Odd DMA Address
10	Pack Overflow
11	Power Fail Detected
12	Invalid Packet Command - CM:PCMD invalid
13	Hard Data Error - Data error was unrecovered

(NOTE: error is not necessarily unrecoverable. If retries and error correction is disabled, the error may be recoverable.)

14 Sector Not Found
15 Write Protect Violation

SMD Timeout errors:

32 Timeout - Controller timeout in idle loop
33 Timeout - waiting for IP & IUS to clear
34 Timeout - waiting for DMA complete
35 Timeout - waiting for On-cylinder
36 Timeout - waiting for Servo clock
37 Timeout - waiting for Data clock
38 Timeout - waiting for Sector/index mark
39 Timeout - waiting for ID sync
40 Timeout - waiting for Data sync

SB:EC -- Error Correction attempted
SB:RT -- Retry attempted
SB:RZ -- Rezero was required to clear a fault

DS:RY -- Selected drive ready
DS:OC -- Selected drive on cylinder
DS:SE -- Selected drive seek error
DS:FT -- Selected drive fault
DS:RO -- Selected drive read only
DS:BZ -- Selected drive busy (dual-access only)
DS:SEL -- Selected ports (port 0 = l.s.bit)
DS:SKE -- Ports with seek complete (port 0 = l.s.bit)

VS: -- Volume Select for multi-volume drives (NOT supported by SADIE diagnostics; this field should always be 0)

OF:O+ -- Servo Offset Plus (only used if CM:NR set)
OF:O- -- Servo Offset Minus (" ")
OF:SE -- Data Strobe Early (" ")
OF:SL -- Data Strobe Late (" ")

INITIALIZING THE CONTROLLER

At power-up, the SMD controller sets SR:NDT. This indicates the controller has not been initialized. SR:NDT will remain on until the following initialization sequence is completed:

1) Host sends CR:IN, then polls SR until SR:BZ is cleared. When SR:BZ is clear, Host checks SR:ES for self-test errors. If SR:BZ never clears, the Controller failed the self test.

2) Host initializes the Dispatch Table with the addresses of the packets for each drive. It also sets all packet status (PS0 to PS3) fields to IDLE. It is recommended that the packets are also initialized to zeroes.

3) Host sends each of the following commands, polling until SR:BZ clears after each command:

   a) CR:CMD = 4 and CR:DATA = l.s.byte of Dispatch Table (DT) address

   b) CR:CMD = 5 and CR:DATA = middle byte of DT address

   c) CR:CMD = 6 and CR:DATA = m.s.byte of DT address

   d) CR:CMD = 7 and CR:DATA = interrupt vector (NOTE: this must be done even if interrupt mode is not used!)

4) Host sends CR:CMD = 1, then polls SR until SR:NDT clears.

ISSUING PACKET COMMANDS

After the controller is initialized, it is ready to receive packet commands. All disk-control operations are packet commands. Packet commands may be issued in either polled mode or interrupt mode. The programming sequence for packet commands is:

1) Host sets up one or more packets with the appropriate packet command opcode in CM:PCMD and sets all applicable parameter fields. Bits which modify the behavior of the controller on error conditions (such as CM:NR, CM:NO, CM:NE and OF: bits) may also be set or cleared.

2) Host sets the PS field of each packet set up in step 1 to GO.

3) Host sends CR:WK with CR:EI optionally set. If CR:EI is set, the controller will interrupt at completion of each packet command.

4) The Controller reads all packets with DT:PS set to GO, and sets the DT:PS field for each such packet to BUSY.

5) The Controller initiates seeks on drives with packet commands requiring a seek. When seek is complete, or if no seek was required, the command is performed and

the Controller updates the packet with completion status. ST, SB and DS fields are always updated unless there is a DMA Error. Other packet fields are also updated by the Controller for some packet commands. The DT:PS field for the completed packet is set to DONE.

NOTE

If a DMA error occurs, the packet and packet status fields in the dispatch table are not updated. The Status Register ES field must be tested by the Host after each command.

6) SR:IP is posted by the Controller. Host may be waiting for SR:IP (polled mode) or waiting for an interrupt if CR:EI was set in step 3.

7) Host reads SR, then issues CR:RI to reset SR:IP and SR:IU. This must be done in polled mode as well as interrupt mode. The Controller may post IP again as soon as CR:RI is sent. If more than one packet was set up before CR:WK was sent, the completed packet is determined by looking for a DT:PS field set to DONE. The host sets this field to IDLE, and the packet command cycle is complete.

SMD DIAGNOSTIC START-UP SEQUENCE

All SMD diagnostics have a similar start-up procedure. Errors encountered during this sequence will cause the diagnostic to display an error message and abort. The sequence is:

1) Initialize the SMD controller. See the earlier section that describes this process. For any errors during this process, the diagnostic displays a message and aborts with a "missing device" status.

2) The SELECT packet command is issued to the disk drive to be tested. If no drive is selected or multiple drives are selected, a message is displayed an the diagnostic aborts with a "missing device" status;

3) The SIZE packet command is issued to the disk drive to be tested. If this command fails, the diagnostic displays a message and aborts with a "test aborted" status.

4) The SEEK packet command is issued with CY = -1. This forces the disk drive to rezero and clears any drive fault. This step prevents spurious soft rezero errors during the first packet command issued by diagnostics.

NOTE

SMDMON is a special trouble-shooting diagnostic. It does not abort if an error occurs during the start-up procedure. It assumes that all disk drives will be tested, and it attempts to SELECT and SIZE all four possible SMD drives. It does not issue the SEEK command.

SMD DIAGNOSTIC ERROR MESSAGES

During the start-up procedure SMD diagnostics may display and log the following error messages:

NO RESPONSE to controller command, SMD CMD = xxxx

SMD controller did not respond to a command in the initialization sequence. The command written to the SMD controller Command Register replaces xxxx (hexadecimal).

During controller initialization, commands are written to the controller command register, and the status register is polled until the command is complete. If the status register contains "FFFF", the controller is not responding to the command. This usually indicates that there is no controller board or the board is not seated properly.

TIMEOUT waiting for Controller Not Busy, SMD CMD = xxxx

SMD controller did not complete a command in a reasonable time. The command was issued in polled mode. The command written to the SMD controller command register replaces xxxx (hexadecimal).

TIMEOUT waiting for SMD interrupt, Packet cmd = cc

A command was issued in interrupt mode, but no interrupt occurred within a reasonable time. During the initialization sequence, the SELECT, SIZE and SEEK commands are issued in interrupt mode. The opcode of the packet command issued replaces cc (hexadecimal). The opcodes of these packet commands are:

SELECT	-- 03
SIZE	-- 0F
SEEK	-- 07

SELECT ERROR -- assume no drive# uu

An error occurred when the SELECT command was issued to the disk unit to be tested. uu is replaced by the disk unit number (decimal). This error may occur if no disk drives are selected or if multiple drives are selected. This message is also displayed if the SMD controller times out after the SELECT packet command is issued.

DISK SIZING ERROR -- assume drive# uu not formatted

The SMD controller could not size the drive, indicating that the disk was probably not formatted. The disk unit to be tested replaces uu (decimal). This error may occur if the disk was never formatted or if one or more sector headers have been destroyed after the disk was formatted. This message is also displayed if the SMD controller times out after the SIZE packet command is issued.

After the start-up procedure, the following messages may appear if the SMD Controller does not respond within a reasonable time after the diagnostic issues a packet command:

NO RESPONSE to controller command, SMD CMD = xxxx

SMD Controller did not respond to a command. The command written to the Command Register replaces "xxxx" (hexadecimal).

NOTE

This message appears only if a command is sent with interrupts disabled. Only SMDMON can send commands in polled mode.

TIMEOUT waiting for Controller Not Busy, SMD CMD = xxxx

SMD controller did not complete a command in a reasonable time. The command was issued in polled mode. The command written to the SMD controller Command Register replaces "xxxx" (hexadecimal).

NOTE

This message appears only if a command is sent with interrupts disabled. Only SMDMON can send commands in polled mode.

TIMEOUT waiting for SMD interrupt, Packet cmd = cc

> A WAKEUP command was issued with interrupts enabled, but no interrupt occurred within a reasonable time. The opcode of the packet command issued replaces "cc" (hexadecimal).

After the start-up procedure, diagnostics will display error messages if the SMD Controller returns ending-status (ES), status-bits (SB) or Drive-Status (DS) fields indicating an error occurred during command execution. These messages have the following three fields:

1) Error Description Field - this field explains the ending status (ES) code returned as well as any other applicable error bits in DS and SB.

2) Operation Field - this field explains what type of operation was being done when the error occurred.

3) Disk Address Field - this field shows what disk address was being accessed when the error occurred. When a multi-sector transfer command is given to the controller, the Disk Address Field shows the START sector address.

NOTE

The error may have occurred in a sector other than the start sector.

The following tables show what values may appear in the Error Description Field and the Disk Address Field. The contents of the Operation Field vary from diagnostic to diagnostic and are shown in the documentation for each diagnostic.

ERROR DESCRIPTION FIELD

Field Contents	Explanation
Soft Error, SB=x	ES = 0 and one or more bits in SB is set; the low-order nibble of SB replaces "x" (hexadecimal). bit 0 = error correct. bit 1 = retries att. bit 2 = rezero req'd.
Initialization Error	ES = 1
Sector Overrun	ES = 2
DMA Error @ <sss>oooo	ES = 3; The segment and offset where the memory buffer began replace "sss" (decimal) and "oooo" (hexadecimal), respectively.
Select Error	ES = 4
Byte/Sec Count Error	ES = 5
Rezero Error	ES = 6
Odd DMA Address	ES = 7
Drive Status, DS=xxxx	ES = 8; DS replaces "xxxx" (hexadecimal). bit 0 = drive ready bit 1 = on cylinder bit 2 = seek error bit 3 = drive fault bit 4 = read only bit 5 = dual-acc. drive busy bits 8-11 = ports selected bits 12-15 = ports w/ seek end
Pack Overflow	ES = 10
Power Fail Detected	ES = 11
Inv Pkt Cmd, CMD=xx	ES = 12; the packet command opcode replaces "xx" (hexadecimal).
Hard Data Error, SB=x	ES = 13; the low nibble of SB replaces "x" (hexadecimal). See Soft Error above for bit definitions.
Sector Not Found	ES = 14
Write Protect Viol.	ES = 15
Timeout, ES=dd	ES is 32-40; The actual ending status code replaces "dd" (decimal).
Unknown, ES=xx	ES is non-zero, but not one of the defined ending-status codes. The

ending status code replaces "xx" (hexadecimal).

DISK ADDRESS FIELD

Message	Explanation	
DISK=(u,cccc,hh,ss)	u = disk unit number	(decimal)
	cccc = cylinder number	(")
	hh = head number	(")
	ss = sector number	(")

SMD DIAGNOSTIC LAP SUMMARY

At the conclusion of each repetition (lap) of a diagnostic, a table is displayed summarizing the numbers of packet commands and command errors. The lap summary is cumulative for all laps. It shows:

- The test name
- The disk unit# under test
- The lap count
- The total number of errors of all kinds
- The number of times each packet command was issued, including:
    - TOTAL = total packet commands issued
    - nop = NOP commands
    - wram = WRAM commands
    - rram = RRAM commands
    - sel = SELECT commands
    - pri = PRISEL commands
    - rel = PRIREL commands
    - rst = RESET commands
    - seek = SEEK commands
    - fmt = FMT commands
    - wlng = WLONG commands
    - wri = WRITE commands
    - frd = FRD commands
    - rlng = RLONG commands

- o read = READ commands
- o size = SIZE commands
- o inv = "invalid" packet commands

NOTE

Only SMDMON can issue invalid commands through the DOPKT command. These commands are tallied as invalid, even though they may be valid commands.

- Tallies of all soft errors reported by the SMD controller after execution of a packet command, including:
  - o TOTAL = the number of times one or more error status bits (SB) were returned, but the ending status (ES) indicated no uncorrectable errors occurred.
  - o rty = soft errors with retries attempted
  - o corr = soft errors with correction attempted
  - o rz = soft errors with a disk rezero attempted

- Tallies of all uncorrected (hard) data errors reported by the SMD controller after execution of a packet command, including:
  - o TOTAL = the number of times a hard data error was returned in the ending status
  - o rty = hard data errors with retries attempted
  - o corr = hard data errors with correction attempted
  - o rz = hard data errors with a disk rezero attempted

- Tallies of drive errors encountered during execution of packet commands, including:
  - o TOTAL = the number of times one or more drive errors occurred during execution of a packet command
  - o df = drive faults
  - o se = drive seek errors
  - o noc = drive not-on-cylinder errors
  - o nrdy = drive not ready errors
  - o busy = dual-access busy errors

- Tallies of all SMD controller timeout errors during packet command execution, including:
  - o TOTAL = the total number of timeout errors
  - o idle = timeouts in the controller's idle loop
  - o ius = timeouts waiting for IP/IUS to clear

- dma = timeouts waiting for DMA complete
- oc = timeouts waiting for drive on cylinder
- sclk = timeouts waiting for Servo clock
- dclk = timeouts waiting for Data clock
- mark = timeouts waiting for Sector/index mark
- isnc = timeouts waiting for ID sync
- dsnc = timeouts waiting for Data sync

- Tallies of other packet command errors, including:

  - TOTAL = number of other packet command errors
  - init = initialization errors (SMD controller reports it does not have complete Dispatch Table address and/or Interrupt vector.)
  - so = sector overruns
  - me = memory parity errors
  - sel = select errors (0 or multiple drives)
  - ct = byte/sector Count errors
  - rze = rezero errors (rezero did not clear a fault)
  - ae = address errors (odd address parameter)
  - ofl = pack overflows
  - pf = power-fail detected
  - snf = sector not found errors
  - wpv = write Protect violations
  - inv = invalid Packet commands reported by SMD controller
  - unk = unknown ending status code returned by SMD controller

- Tallies of errors detected by the diagnostic, including:

  - TOTAL = number of diagnostic detected errors
  - cmp = compare errors encountered by the diagnostic when comparing two buffers
  - htrk = the number of tracks with more than one hard defect (only SMDFMT will report this)
  - strk = the number of tracks with soft defects that cannot be flagged as bad, because there are multiple soft defects or there is a hard defect that must be flagged as bad (only SMDFMT will report this)
  - resp = SMD controller not responding errors (polled mode commands only)
  - sto = SMD controller timeout -- controller did not interrupt within expected time (polled or interrupt mode)

TEST NAME:

SMDCRC - a non-destructive, read-only test of all tracks on a SMD disk.

PARAMETERS:

Parameter 1 = disk unit to be tested (Default: unit = 0). The range of valid unit numbers is 0 - 3.

DESCRIPTION:

Before the read test begins, SMDCRC goes through the start-up sequence described in the INTRODUCTION TO SMD DIAGNOSTICS. Any errors during that procedure cause SMDCRC to abort the test.

For each repetition, SMDCRC reads the entire disk track by track. All READ packet commands are issued in interrupt mode. For each error encountered, an error message is displayed and logged to SADIE, and error status bits are tallied. At the end of each repetition of the test, SMDCRC displays a lap summary table, which shows a cumulative tally of error status bits and packet commands issued to the SMD controller.

SMDCRC repeats the read test n times, where n is the value in the #REPS field of the test line. If REPS=0, SMDCRC repeats until the user aborts the test.

ERROR MESSAGES:

SMDCRC displays error messages with the format described in the INTRODUCTION TO SMD DIAGNOSTICS. During the read test, the possible contents of the Operation Field of error messages is: READING TRACK.

NOTE

SMDCRC reads entire tracks. The disk address in an error message is the address of the first sector on the track where the error occurred. The error may have occurred on any sector on that track.

LAP SUMMARY:

SMDCRC displays a cumulative tally of packet commands and errors at the completion of each repetition of the read test. See the INTRODUCTION TO SMD DIAGNOSTICS for a complete description of the lap summary table.

TEST NAME:

SMDFMT - a utility that formats the SMD disk drive after extensive surface analysis. Bad sectors are flagged and will be ignored during subsequent writes and reads.

PARAMETERS:

Parameter 1 = disk unit to be analyzed and formatted (Default: unit = 0). The range of units is 0 - 3.

Parameter 2 = number of scans of the disk during surface analysis (Default: scans = 1). The range of scans is the positive integers.

Parameter 3 = interactive mode. If interact = 0, SMDFMT attempts to find the SMD disk size using an algorithm similar to the one the SMD controller uses in the SIZE command. If interact ≠ 0, SMDFMT polls the user for the disk size.

DESCRIPTION:

SMDFMT goes through most of the standard start-up procedure described in the INTRODUCTION TO SMD DIAGNOSTICS. However, since its purpose is to format a drive, it does not issue the SIZE packet command. It also omits the SEEK to CY = -1 step of the start-up procedure.

If interact = 0, SMDFMT will then do some investigation to find the numbers of cylinders, heads and sectors-per-track present on the drive. The results of its investigation are displayed before formatting commences. For the Model 31 80 Megabyte drives, the correct numbers are:

```
# cylinders = 589
# heads = 7
# total sectors/track = 33
# active sectors/track = 32
```

If interact ≠ 0, SMDFMT prompts the user for the disk size parameters. Once the disk size is known, SMDFMT will format the disk. The disk will initially be formatted with no sectors flagged as bad or spare (i.e. all sectors will be formatted as active sectors).

If the initial disk format succeeds, SMDFMT will do track-by-track surface analysis on the disk to discover those sectors with media errors. The first step in the analysis makes certain that the sector headers are readable. The sector IDs of all sectors on the track are read and compared with the data that should be in each field. This read-and-compare test repeats the number of times given by Parameter 2. The second step makes certain that the data fields of the sectors are good. For this test, six patterns are used: 'AAAA', '5555', 'FFFF' '6DB', 'DB6', and 'B6D'. Each pattern is written to the track, then read a number of times. Parameter 2 determines the number of reads per pattern. Any error during track writes and reads cause SMDFMT to do writes and reads to each individual sector on the track.

During surface analysis, SMDFMT keeps track of all sectors with hard and soft errors. After surface analysis, SMDFMT reformats the disk, flagging the bad sectors. During the reformat step, each track has one sector flagged as either bad or spare, with all other sectors being active. If there is an uncorrectable (hard) error in the header or data portion of a single sector on the track, that sector is flagged as bad and all others are active. If there are correctable (soft) errors on one or more sectors on the track, and no sectors with hard errors, one of the sectors with a soft defect is flagged as bad and all others are active. If there are no defects on any sectors on a track, all sectors are active except the last, which is flagged as a spare.

If a track has more than one sector with hard error(s), the track, and in fact the entire disk, is unusable. A message will appear at the conclusion of SMDFMT if this happens.

After the disk is reformatted, SMDFMT will display a lap summary table. The entire format sequence will repeat n times, where n is the value in the #REPS field on the SADIE test line. If #REPS = 0, SMDFMT will repeat the format sequence until the user aborts the diagnostic.

ERROR MESSAGES:

SMDFMT displays and logs error messages with the format described in the INTRODUCTION TO SMD DIAGNOSTICS. The possible contents of the Operation Field of error messages are described in the following table:

OPERATION FIELD

Field Contents	Explanation
WRITE FORMAT REWRITE FORMAT	The error occurred writing the sector headers for a full track with the FMT packet command. SMDFMT rezeroes the drive and retries the format if it gets an error status the first time.
READ TRACK IDS READ SECTOR ID	The diagnostic was reading the sector headers for a full track or a single sector with the FRD packet command. SMDFMT first tries to FRD an entire track. If it gets a bad ending status, it tries FRD sector by sector.
WRITE TRACK-xxxx WRITE SEC-xxxx READ TRACK-xxxx READ SEC-xxxx	The diagnostic was writing or reading a full track or a single sector. The pattern being written replaces xxxx (hexadecimal). The diagnostic will try to write and read tracks unless an error occurs. When an error occurs, it will try to pinpoint bad sector(s) by writing/ reading single sectors.

SMDFMT also displays and logs the following error messages:

```
COMPARE ERROR IN SECTOR ID     DISK=u,cccc,hh,ss
```

The sector header for the disk address given did not contain the correct data in the cylinder, head and sector fields. This is considered an uncorrectable error in the sector. "u", "cccc", "hh" and "ss" replace the disk unit, cylinder, head and sector numbers where the compare error occurred.

UNMAPPED SOFT ERROR(S) ON CYL# cccc, HEAD# hh

The diagnostic can flag only one sector on a track as bad. The presence of more than one soft error or one hard error and one or more soft errors means that one or more of the soft defects can not be flagged. This is not a fatal error, but unmapped soft defects may degrade system performance.

BAD TRACK--MORE THAN 1 HARD ERROR ON CYL# cccc, HEAD# hh

The diagnostic can only flag one sector on a track as bad. The presence of more than one hard error means that the track is bad, and hence the disk is bad. The cylinder and head numbers of the bad track replace "cccc" and "hh" (decimal).

LAP SUMMARY:

At the conclusion of the re-format step, SMDFMT is finished formatting the disk. It displays the lap summary table of commands and errors described in the INTRODUCTION TO SMD DIAGNOSTICS before exiting or repeating the format sequence.

SMDFMT follows the lap summary table with a list of all defects. For each defect the cylinder, head and sector numbers are given as well as the severity (SOFT or HARD) and whether the defect was mapped or unmapped.

NOTES:

SMDFMT does extensive surface analysis in an attempt to find all defects on the disk. This analysis takes time, and the time depends upon the number of scans indicated by Parameter 2. It is recommended that the number of scans be at least two. SMDFMT will take approximately 2 hours to format an 80 Megabyte drive with 2 scans.

TEST NAME:

SMDMEDIA - a thorough write-read-compare test of a drive, using various data patterns.

PARAMETERS:

Parameter 1 = the disk unit to be tested (Default: unit = 0). The range of unit numbers are 0 - 3.

Parameter 2 indicates whether the test should compare source and destination buffers after write and read. If compare = 0, the compare step is skipped. Otherwise, the compare is done. With the compare step, the test is more thorough, but it takes considerably more time to execute. (Default: compare = 1)

Parameter 3 indicates whether the test should use all data patterns or only the worst-case data pattern. If allpat = 0, only the worst-case pattern ('B6D') is used. Otherwise, all patterns ('AAAA', 'FFFF' and 'B6D') are used. (Default: allpat = 1)

DESCRIPTION:

SMDMEDIA begins with the start-up procedure described in the INTRODUCTION TO SMD DIAGNOSTICS. Any errors during the start-up procedure cause SMDMEDIA to abort the test.

During each repetition of the media test, the following steps are done for each data pattern to be used:

1) A source buffer the size of a full track is filled with the data pattern.

2) For every track on the medium:

   a) The source buffer is written to the track.

   b) The track is read into a destination buffer.

   c) The source and destination buffers are compared.

SMDMEDIA repeats the media test n times, where n is the value in the #REPS field in the SADIE test line. If #REPS=0, SMDMEDIA repeats until the user aborts the test.

ERROR MESSAGES:

SMDMEDIA displays and logs error messages with the format described in the INTRODUCTION TO SMD DIAGNOSTICS. The possible contents of the Operation Field are:

OPERATION FIELD

Field Contents	Explanation
WRITE TRK,xxxx	Error occurred while writing a track. The data pattern written replaces "xxxx" (hexadecimal).
READ TRK, xxxx	Error occurred while reading a track. The data pattern read replaces "xxxx" (hexadecimal).

SMDMEDIA will also display the following message:

```
COMPARE WR:<ws>woff=wd RD:<rs>roff=rd DISK=u,cccc,hh,ss
```

A mismatch was found between a buffer written and a buffer read from the same disk address. The mismatch was found at the write and read buffer addresses given by "ws", "woff" (write segment (decimal) and offset (hex)) and "rs", "roff" (read segment and offset). The data written replaces "wd" (hexadecimal). The mismatching data read replaces "rd" (also hexadecimal).

LAP SUMMARY:

At the conclusion of each complete repetition of SMDMEDIA, a lap summary table is displayed. This table is described in the INTRODUCTION TO SMD DIAGNOSTICS.

TEST NAME:

SMDTEST - two write-read-compare tests of random single-sector disk accesses and random-length disk transfers.

PARAMETERS:

Parameter 1 = disk unit to be tested (Default: unit = 0). The range of valid unit numbers is 0 - 3.

Parameter 2 = test to be performed. (Default: testnum = 1). If testnum = 1, the random single-sector seek test is done. If testnum = 2, the random-length disk transfer test is done.

DESCRIPTION:

Before the test begins, SMDTEST goes through the start-up sequence described in the INTRODUCTION TO SMD DIAGNOSTICS. Any errors during that procedure cause SMDTEST to abort.

SMDTEST is actually a pair of random tests. Which test is performed during any invocation of SMDTEST depends upon the value of Parameter 2.

TEST 1:

The single-sector test is a good test of random seeks. Initially, a queue is formed of 128 elements. Each element is initialized to a random, unique disk address. Associated with each queue element is a unique source buffer and a unique destination buffer. The source buffers are initialized with random data.

Each lap proceeds as follows:

1) For each of the 128 queue elements, the source buffer for that element is written to the disk address. This is a single-sector write.

2) For each of the 128 queue elements:

    a) the sector address is read into the destination buffer associated with that queue element.

b) the source and destination buffers are compared.

c) the queue element is re-randomized, and the new disk address is compared with all other addresses in the queue for uniqueness.

d) the source buffer is filled with new random data.

TEST 2:

The random-length transfer test tests random-length writes and reads. In this test, the source and destination buffer addresses are random, and the buffers may cross segment boundaries (not allowed in the single-sector test). The buffer sizes are random.

During initialization of the random-length transfer test, segments 1 - 3 are filled with a background pattern ('AAAA'). For each lap, there are 64 repetitions of the following:

1) A randomly chosen source buffer is filled with random data. The buffer must be in segments 1 - 3.

2) The source buffer is written to a random disk address.

3) The disk address is read into a random destination buffer. The destination buffer must be in segments 1 - 3.

4) The source and destination buffers are compared.

5) The source and destination buffers are filled with background data. Then the entire background is checked to be sure there has been no corruption of memory outside of the source and destination buffers.

SMDTEST repeats test 1 or test 2 n times, where n is the value in the #REPS field of the SADIE test line. If #REPS = 0, SMDTEST repeats the test until the user aborts it.

SMDTEST always begins with the same random-number seed at each invocation, but it does not reset the seed between laps. Therefore, running SMDTEST 100 times is not the same as running SMDTEST with #REPS = 100.

ERROR MESSAGES:

SMDTEST displays and logs error messages with the format described in the INTRODUCTION TO SMD DIAGNOSTICS. The possible contents of the Operation Field are:

OPERATION FIELD

Field Contents	Explanation
WRITE SECTOR	The diagnostic was writing a single sector in test 1.
READ SECTOR	The diagnostic was reading a single sector in test 1.
WRITE xxxx BYTES	The diagnostic was writing a random-length buffer in test 2. The length in bytes replaces "xxxx" (hexadecimal).
READ xxxx BYTES	The diagnostic was reading a random-length buffer in test 2. The length in bytes replaces "xxxx" (hexadecimal).

SMDTEST will also display the following messages:

```
CMP ERR IN BUFFERS: SRC<ss>soff=sd DST<ds>doff=dd
ORIG BUFS: WR<ws>woff RD<rd>roff L=1111 DISK=u,cccc,hh,ss
```

A mismatch was found comparing the buffer written to the buffer read. The addresses and data where the data mismatch occurred are given by SRC and DST. The address and data in the write buffer where the mismatch occurred replace "ss" (segment, decimal), "soff" (offset, hex) and "sd" (data, hex). The address and data in the read buffer where the mismatch was found replace "ds" (segment, decimal), "doff" (offset, hex) and "dd" (data, hex). Similarly, the beginning addresses of the write and read buffers are given by the values following WR and RD. The buffer length in bytes replaces "1111" (hexadecimal).

```
CMP ERR IN BACKGR: SRC<ss>soff=sd DST<ds>doff=dd
ORIG BUFS: WR<ws>woff RD<rd>roff L=1111 DISK=u,cccc,hh,ss
```

A mismatch was found in the background (all memory locations outside the write and read buffers are the background). The data given in this message is the same as that described for the compare errors within the read and write buffers (see above). Only test 2 does a background check after a write and read.

LAP SUMMARY::

At the end of each complete lap of SMDTEST, a table of command and error statistics is displayed. This lap summary table is described in the INTRODUCTION TO SMD DIAGNOSTICS.

TEST NAME:

SMDMON - an interactive monitor for the Storage Module Drive (SMD) Controller.

PARAMETERS:

SMDMON has no parameters.

DESCRIPTION:

SMDMON is a diagnostic tool useful for troubleshooting and executing custom tests. For maximum utility, it requires knowledge of the host interface. Refer to the INTRODUCTION TO SMD DIAGNOSTICS for a brief description of the SMD Controller interface. More detailed information is included in Section 5.

The user interacts with SMDMON by entering command lines into the console keyboard. SMDMON interprets and executes commands. SMDMON takes care of setting up packets, issuing WAKEUP, waiting for command completion and error handling.

SMDMON begins with a start-up sequence similar but not identical to the other diagnostics. The differences are:

1) SMDMON issues a warning, but does not abort if an error occurs during the start-up seqeunce.

2) SMDMON is based on the use of four disk units. Therefore, it issues the SELECT and SIZE commands to all four units and keeps a table of the drive size parameters for all four units.

3) SMDMON does not issue the SEEK to CY = -1 to any drives during start-up.

After SMDMON has gone through its start-up procedure, it prompts with: Command?. SMDMON accepts and executes command lines until the user enters the QUIT command. A HELP command is included, which displays the command names and parameters.

To enter command lines to SMDMON, the user must know the following:

1) Command names are entered in upper case.

2) Repeat factors and numeric parameters are interpreted by SMDMON as decimal unless the digits are followed by an H, which indicates the number should be interpreted as a hexadecimal number.

   Example:
   a) 256 is interpreted as 256 decimal.
   b) 100H is interpreted as 100 hexadecimal, which i equivalent to 256 decimal.

3) A simple command is defined as a command name followed by zero or more numeric parameters, separated by blanks. (Command names and parameters are defined under COMMAND DESCRIPTIONS below).

```
    Example:  The following are simple commands:
              a)  EINT
              b)  SUNIT  1
              c)  RD  1  0  512  400  3  27
              d)  WR  2  0  512  400  3  27
```

4) A complex command is one of the following:

   A) An optional repeat factor followed by a simple command

   B) An optional repeat factor followed by a parenthesized complex command

   C) A parenthesized complex command followed by any number of parenthesized complex commands

   Example:

   All of the following are complex commands:
   (Note: (a) and (b) are also simple commands!)

```
   a)  DINT
   b)  RD  1  0  1024  400  3  27
   c)  10  RD  1  0  1024  400  3  27
   d)  (WR 2 0 512 400 3 27)(RD 1 0 512 400 3 27)
   e)  10((WR 2 0 512)(RD 1 0 512)(CMP 2 0 1 0 512)
       (ISEC 1))
```

5) A command line is a simple or complex command followed by a carriage return. It can include up to 127 characters.

6) Unless other-wise specified in the COMMAND DESCRIPTION section below, all parameters are optional.

7) Omitted parameters take on their previous values. Most parameters have default values, shown in the COMMAND DESCRIPTION section.

```
Example:
   Command?
   RD  1  0  512  400  3  27
   Command?
   RD
```

The first RD (read) command reads 512 bytes, beginning at disk cylinder 400, head 3, sector 27 into memory segment 1, offset 0. The second RD command does the same thing, because the omitted parameters have the previous values.

8) In some commands SMDMON checks for boundary conditions. Commands that increment the disk address (ICYL, IHEAD, ISEC) or the memory address (IROFF, IWOFF) check for overflows of the disk pack and available memory. If the increment would cause overflow, a warning message appears and the command is ignored.

9) SMDMON does not, in general, protect the user from commands that could be destructive. For example, SMDMON allows the SMD controller to do a DMA transfer into segment 0. This can to cause SMDMON or SADIE to be overwritten.

ERROR MESSAGES

SMDMON displays and logs error messages with the format described in the INTRODUCTION TO SMD DIAGNOSTICS. The Operation Field of packet command error messages shows the packet command opcode (CM:CMD) in hexadecimal.

SMDMON can also display and log the following message:

```
COMPARE WR:<ws>woff=wd RD:<rs>roff=rd DISK=u,cccc,hh,ss
```

A mismatch occured comparing two buffers. The buffers are called the write and read buffers, because that is what is typically being compared. The addresses and data where the mismatch occurred replace ws, woff, wd (write segment, offset and data) and rs, roff, rd (read segment, offset and data). The current disk unit, cylinder, head and sector address replace u, cccc, hh and ss.

LAP SUMMARY

SMDMON does not execute in laps in the same sense as the other SMD diagnostics. However, it maintains the same packet command and error statistics as the other diagnostics. This lap summary table is displayed whenever the REC command is entered. Refer to the INTRODUCTION TO SMD DIAGNOSTICS for a description of the lap summary table contents.

A separate lap summary table is maintained for each disk unit. REC displays the summary for the currently selected unit.

COMMAND DESCRIPTIONS:

The following notes apply to the descriptions of SMDMON commands:

1) Command names are shown in upper-case letters, exactly as they must be entered. Parameter names are shown in lower-case letters, and represent numbers. Except as noted, parameters are optional. All parameters are positional.

Example:

```
          a)  RCYL
          b)  SUNIT  unit
          c)  RAND rndseg rndoff rlnth
```

```
Example (a) is a command with no parameters.
Example (b) has command name SUNIT and a
            single numeric parameter, a unit number.
Example (c) has command name RAND and
            three positional numeric parameters.
```

2) Command names may be abbreviated to the shortest character string that uniquely identifies the command.

Example: The following names are all equivalent:

```
WR, WRI, WRIT, WRITE
```

3) Some commands are discussed together for convenience. In these cases, a slash (/) separates the commands.

Example:

a) DINT / EINT
b) SWSEG wseg / SWOFF woff

In each example, there are two distinct commands. They are discussed together because they are related in function.

4) Memory addresses may be shown with the segment enclosed in < and >, with the offset immediatly following. <1>0 refers to segment 1, offset 0.

5) Command parameter values retain their previous values until changed. Most parameters have default values, described in the command descriptions below. Some parameters are shared by more than one command.

Example: The following commands all share the write buffer start address parameters, wseg and woff:

a) WR wseg woff lnth cyl head sec
b) SWSEG wseg
c) SWOFF woff

6) Parameter defaults are:

```
rseg = 1         wseg = 2        dseg = 1
desseg = 1       srcseg = 2      rndseg = 2
rramseg = 1      wramseg = 2     putseg = 2
fseg = 2
roff = 0         woff = 0        doff = 0
desoff = 0       srcoff = 0      rndoff = 0
rramoff = 0      wramoff = 0     putoff = 0
foff = 0         fpat = ffff
lnth = 512       rlnth = 512     cmplnth = 512
unit = 0         cyl = 0         head = 0
sec = 0          maxseg = 3      offset_val = 0
intvec = 0       fstart = 0      fintvl = 1
rramlnth = 2048
wramlnth = 1280
```

7) Some parameters are dependent upon the current disk unit. SMD disks have different numbers of

cylinders, heads and sectors/track. During the start-up procedure SMDMON attempts to select and size each of the four possible SMD units. The size parameters are stored in a table, and restored whenever a SUNIT command changes the current unit number. To override the drive size parameters use the SDRIVE command. New parameters entered with the SDRIVE command are stored in the table.

Example: After SMDMON start-up, the drive size parameters for unit# 0 are:

#cyls = 589, #heads=7,
#Active sectors/track = 32

To override these values, enter:

```
SDRIVE 600 8 34
```

The values 600, 8 and 34 are stored in the drive-size table.

8) Format parameters are closely tied to drive-size parameters. Whenever the drive-size changes, the format parameters are changed to make sense for the new drive parameters. The format parameter values are:

fmtsec = maxsec + 1
(total number of sectors/track, including spare and bad sectors)

nsec = maxsec + 1
(number sectors/track to be formatted with FMT, FVOL or read with FRD command)

fspare = maxsec
(spare sector number)

fbad = ffffH
(bad sector number; defaults to an invalid number, so there will not be a sector flagged as bad during FMT or FVOL commands)

Drive size parameters may change if:

1) User enters a unit number (SUNIT).
2) User enters new size parameters (SDRIVE).
3) User enters the SIZE command.

The following commands are recognized by SMDMON:

COMMAND: CEC / REC / ILC
DESCRIPTION:

CEC clears the lap summary command, error and lap counters.

REC displays the lap summary counters in tabular form. Refer to the description of the lap summary in the INTRODUCTION TO SMD DIAGNOSTICS.

ILC increments the current lap counter by 1. This is useful in command lines with large repetition counts.

EXAMPLE:

```
CEC 9999((RAND)(RALL)(WR)(RD)(CMP)(ILC))
REC
```

The first command clears all fields in the lap summary. The second line repeats a random write-read-compare test 9999 times, each time incrementing the lap counter by 1. The REC command displays the lap summary.

```
COMMAND:  COMPARE  srcseg srcoff desseg desoff cmplnth
          CMP      srcseg srcoff desseg desoff cmplnth
```

DESCRIPTION:

If NO parameters are entered:

Compares the current write buffer and read buffer beginning at <wseg>woff and <rseg>roff. lnth bytes are compared.

If ANY parameters are in the command string:

Compares the buffer at <srcseg>srcoff with the buffer at <desseg>desoff for cmplnth bytes.

A message tells if and where the first mismatch is found.

CMP is a special abbreviation of COMPARE.

EXAMPLE:

```
WR 3 0 1024
RD 2 0
CMP 2 0 1 0 512
CMP
```

The first CMP command compares the buffer at <2>0 with the buffer at <1>0 for 512 bytes. The second CMP command compares the current write buffer with the current read buffer. In this case, the write buffer is at <3>0 and the read buffer is at <2>0. 1024 bytes are compared.

COMMAND: DISPLAY dseg doff dlnth
DESCRIPTION:

If NO parameters are entered:

Displays the current read buffer at <rseg>roff for lnth bytes.

If ANY parameters are in the command string:

Displays the contents of memory at <dseg>doff for dlnth bytes.

EXAMPLE:

```
RD 2 0 512
DISP 1 0 20H
DISP
```

The first DISP command displays 32 (20 hex) bytes beginning at <1>0. The second DISP displays the read buffer at <2>0 for 512 bytes.

COMMAND: DOPKT
DESCRIPTION:

The current unit packet-status in the dispatch table is set to GO, and WAKEUP is sent to the SMD controller. SMDMON will not alter the packet in any way.

A packet can be set up manually with the PUTMEM command. First, the packet address is found by entering the STATUS command, then PUTMEM is used repeatedly to set each byte in the packet.

EXAMPLE:

```
DOPKT
```

The current unit packet is issued to the SMD controller as is.

COMMAND: EABORT / DABORT
DESCRIPTION:

EABORT enables an abort-on-<CR> feature. If the user enters a carriage-return on the console, the current command line s aborted. This the default for SMDMON.

DABORT disables the abort-on-<CR> feature.

EXAMPLE:

```
EABORT
1000((READ 1 0 512)(IBLK 1))
```

During the second command line, entering a <CR> on the console aborts the line.

COMMAND: ECORR / DCORR
DESCRIPTION:

ECORR enables error correction on subsequent packet commands. The CM:NC bit is 0. This is the default mode for SMDMON.

DCORR disables error correction on subsequent packet commands. The CM:NC bit is set to 1.

EXAMPLE:

```
ECORR
READ
DCORR
READ
```

During the first READ error correction is enabled. During the second READ, no error correction takes place.

COMMAND: EINT / DINT
DESCRIPTION:

EINT enables interrupts at the completion of packet commands. CR:EI is set whenever WAKEUP is sent to the controller. This is the default in SMDMON.

DINT disables interrupts at the completion of packet commands. CR:DI is set whenever WAKEUP is sent.

EXAMPLE:

```
DINT
READ
```

The READ command is issued in polled mode.

COMMAND: EOFFSET / DOFFSET
DESCRIPTION:

EOFFSET enables use of head offsets during retries. CM:NO is set to 0 on all subseqeunt packet commands. This is the default mode for SMDMON.

DOFFSET disables head offsets during retries. CM:NO is set to 1 on all subsequent packet commands.

EXAMPLE:

```
DOFFSET
READ
```

During the READ and all subsequent packet commands, no offsets are used during retries.

COMMAND: EPAUSE / DPAUSE
DESCRIPTION:

EPAUSE enables a pause-on-error mode in SMDFMT. If SMDFMT detects an error upon command completion or a compare error, it will pause and request entry of a <CR> to continue. This is useful when a command line with a large repetition count is entered, and the test is to halt temporarily when an error is detected.

DPAUSE turns off the PAUSE feature. It is the default condition in SMDMON.

EXAMPLE:

```
EPAUSE
1000 ((SALL 0 0 0)(589(7(32((READ 1 0 512)(ISEC 1))))))
```

The second command line reads the entire disk sector-by-sector 1000 times. The user wants the test to halt if an error occurs during the READ, and so EPAUSE is entered before the command line.

COMMAND: EPRINT / DPRINT
DESCRIPTION:

EPRINT causes SMDMON to be verbose. This is the default condition in SMDMON.

DPRINT causes SMDMON to be quiet except when errors occur. Error and warning messages are not disabled by entering DPRINT.

DPRINT is useful when subsequent commands have repetition counts.

EXAMPLE:

```
DP
1000((WR)(RD)(CMP))
```

The second command line does 1000 writes, reads and compares. Since the verbose option was disabled by the DP command, only errors are displayed.

COMMAND: ERETRY / DRETRY
DESCRIPTION:

ERETRY enables retries on all subsequent packet commands. CM:NR is set to 0. This is the default mode in SMDMON.

DRETRY inhibits retries on all subsequent packet commands. CM:NR is 1.

EXAMPLE:

```
DRETRY
READ
```

The READ is done without retries. If SOFFSET was issued earlier to set the OF field in the packet, the user strobe and offset values are used by the controller when reading.

COMMAND: FBLK
DESCRIPTION:

The write buffer beginning at <wseg>woff is filled with the current disk address logical block number. The logical block number is unique for each sector on the disk, and has this formula:

```
log. block = cyl * (maxhead * maxsec)
+ (head * maxsec)
+ sec
```

Logical block numbers may exceed 64K; they are double words.

This command is useful for filling each sector on the disk with data that is both unique and identifies the sector.

EXAMPLE:

```
((SALL 0 0 0)(SWS 3)(SWO 0)(SLNTH 512))
589(7(32((FBLK)(WRI)(IBLK 1))))
```

The second command line writes the unique logical block number to each sector on an 80 megabyte disk with 589 cylinders, 7 heads and 32 active sectors/track.

COMMAND: FILL fseg foff flnth fpat
DESCRIPTION:

If NO parameters are entered, this command fills the current write buffer beginning at <wseg>woff with lnth bytes of the word pattern: fpat.

If ANY parameters are entered, this command fills the buffer at <fseg>foff with flnth bytes of the word pattern: fpat.

EXAMPLE:

```
WR 3 0 3000
FILL 1 0 512 AAAAH
FILL
```

The first FILL fills <1>0 with 512 bytes of 'AAAA's. The second FILL fills <3>0 with 3000 bytes of 'AAAA's.

COMMAND: FMT nsec cyl head sec
DESCRIPTION:

Formats nsec sectors beginning at the disk address: cyl, head, sec. FMT uses the format parameters set in the SFMT command: fmtsec, fspare, fbad, fstart and fintvl.

A sector can be flagged as spare or bad if its logical sector number equals fspare or fbad.

Sector numbers can be interleaved. The default format does not interleave sectors (physical sector# N on a track is also logical sector# N). By setting fstart (starting logical sector number) and fintvl (logical sector number difference between physically contiguous sectors) appropriately, sectors can be interleaved.

EXAMPLE:

```
SFMT 33 32 ffffH 0 3
FMT 33 2 3 0
```

The first line sets up the format parameters: fmtsec = 33 sectors/track, fspare = 32, fbad = ffffH, fstart = 0, fintvl = 3. The second line formats 33 sectors on the track at cylinder 2, head 3. The sectors are interleaved (0, 3, 6, 9 and so on) and the last logical sector (sector# 32) is flagged as a spare sector. No sectors are flagged as bad.

COMMAND: FVOL
DESCRIPTION:

Formats the entire volume, track by track. The format parameters of the SFMT command determine how the disk is formatted.

EXAMPLE:

```
SFMT 33 32 FFFFH 0 1
FVOL
```

The first line sets up the format specifications for an 80 Megabyte drive with 33 total sectors per track. The last sector (sector 32) is designated as a spare. No sectors are designated as bad. Logical sector numbers start with 0, and the interleave factor is 1 (there is no interleaving of sectors).

The second line formats the current disk unit track by track using the format parameters defined in the SFMT command.

COMMAND: FRD rseg roff nsec cyl head sec
DESCRIPTION:

Issues the FRD (format read) packet command.

This reads only the header portions of sectors beginning at: cyl,head,sec. nsec sector headers are read. Six words of data are returned for each sector read, and that data begins at <rseg>roff.

The six words returned for each sector are:

```
     15 14 13                        0
    +--+--+--+-------------------------------+
  0 |ET|EC|EP|                               |
    +--+--+--+-------------------------------+
  2 |              CYLINDER                  |
    +----------------------------------------+
  4 |                HEAD                    |
    +--+--+----------------------------------+
  6 |FL|SP|         SECTOR                   |
    +--+--+----------------------------------+
  8 |             ECC M.S.H.                 |
    +----------------------------------------+
 10 |             ECC L.S.H.                 |
    +----------------------------------------+
```

where,

```
          ET flags the last sector on a track
          EC flags the last sector on a cylinder
          EP flags the last sector on a pack
          FL flags a bad sector
          SP flags a spare sector
          ECC M.S.H. and ECC L.S.H. are the
                     high and low parts of the
                     ECC (checksum) for the
                     header portion of the sector.
```

EXAMPLE:

FRD 1 0 33 400 3 0

The headers of 33 sectors beginning with cyl# 400, head#3, sector# 0 are read into <1>0. Each header is 6 words long.

COMMAND: HELP
DESCRIPTION:

Displays a list of command names and their parameters.

EXAMPLE:

```
HELP
```

COMMAND: ICYL incval / IHEAD incval / ISEC incval
DESCRIPTION:

ICYL, IHEAD and ISEC increment the current values of cyl, head, and sec by incval, respectively.

An increment of sec up to or greater than maxsec causes SMDMON to increment head and cyl by the correct amounts. An increment of head up to or greater than maxhead causes SMDMON to increment cyl the correct amount. If cyl will be equal or greater than maxcyl, however, SMDMON ignores the command.

EXAMPLE:

```
SALL 588 0 0
ISEC 45
IHEAD 8
```

Assuming the drive is an 80 Megabyte drive with maxcyl = 589, maxhead = 7, maxsec = 32, the ISEC command causes SMDMON to set cyl=588, head=1, sec=12. The IHEAD command is ignored, because it would set cyl to 589, which is equal to maxcyl.

COMMAND: ILNTH offinc / IROFF offinc / IWOFF offinc
DESCRIPTION:

ILNTH increments the current read/write buffer length by offinc.

IROFF increments the current read buffer offset, roff, by offinc.

IWOFF increments the current write buffer offset, woff, by offinc.

offinc is the parameter value supplied, or takes on its previous or default value, if absent. It must always be an even number.

EXAMPLE:

```
IROFF 512
IWOFF
```

The first line increments the read buffer offset by 512 bytes. The second line increments the write buffer offset, also by 512, because the last value assigned to offinc was 512.

COMMAND: INIT
DESCRIPTION:

Reinitializes the controller. The sequence of events is:
1) Write CR:INIT to the command port. Wait for SR:BZ to clear.
2) Reinitialize all packets to zeroes.
3) Reinitialize the dispatch table
4) Send the segment number and offset bytes for the dispatch table to the controller.
5) Send the interrupt vector in intvec to the controller.
6) Send the read-packet-addresses command to the controller.

EXAMPLE:

```
INIT
```

COMMAND: NOP / PRISEL / PRIREL / RESET / SELECT
DESCRIPTION:

Each of these commands issues the packet command of the same name in the current unit packet. SMDFMT waits for command completion and reports any error status returned by the SMD controller.

EXAMPLE:

```
NOP
RESET
SELECT
```

COMMAND: PUTMEM putval putseg putoff
         P   putval putseg putoff
DESCRIPTION:

PUTMEM sets the 8-bit value in putval into the byte addressed by <putseg>putoff. It also sets putoff up by 1. Using PUTMEM sets putseg and putoff the first time the command is used, then repeats it with only the putval for subsequent memory bytes.

P is a special abbreviation for PUTMEM.

EXAMPLE:

```
P 7fH 2 10a0H
P 00
P 0cH
```

Sets 7f into <2>10a0, 00 into <2>10a1 and 0c into <2>10a2.

COMMAND: QUIT
DESCRIPTION:

Exits SMDMON. Returns to the program that invoked SMDMON (SADIE or the PROM monitor).

EXAMPLE:

```
QUIT
```

COMMAND: RAND rndseg rndoff rndlnth
DESCRIPTION:

If NO parameters are present, this command fills the current write buffer beginning at <wseg>woff with lnth bytes of random data.

If ANY parameters are present, this command fills the buffer beginning at <rndseg>rndoff with rndlnth bytes of random data.

EXAMPLE:

```
((SWSEG 3) (SWOFF 0) (SLNTH 1024))
RAND 2 0 512
RAND
```

The first RAND command randomizes the buffer beginning at <2>0 for 512 bytes. The second RAND command randomizes the write buffer at <3>0 for 1024 bytes.

COMMAND: RCYL / RHEAD / RSEC / RALL
DESCRIPTION:

RCYL randomizes the current disk cylinder number to a number between 0 and maxcyl-1, inclusive.

RHEAD randomizes the current disk head number to a number between 0 and maxhead-1, inclusive.

RSEC randomizes the current disk sector number to a number between 0 and maxsec-1, inclusive.

RALL randomizes the current disk address. cyl, head and sec are all randomized as if the user had entered RCYL, RHEAD and RSEC commands.

EXAMPLE:

```
RCYL
```

Assuming maxcyl = 589, this command sets cyl to a value in the range (0, 588).

```
COMMAND:  READ  rseg roff lnth cyl head sec
          RD    rseg roff lnth cyl head sec
```
DESCRIPTION:

Issues the READ packet command. The read buffer begins at <rseg>roff. The disk address that will be read is: cyl, head, sec. lnth bytes will be read.

RD is a special abbreviation for READ.

EXAMPLE:

```
READ 1 0 200H 5 2 0
```

reads 512 (200 Hex) bytes into memory beginning at <1>0, from cylinder# 5, head# 2, sector# 0.

COMMAND: RLONG rseg roff lnth cyl head sec
DESCRIPTION:

Issues the RLONG (read long) packet command. The read buffer begins at <rseg>roff. The disk address that will be read is: cyl, head, sec. lnth bytes will be read.

EXAMPLE:

```
RLONG 1 0 516 3 2 20
```

Reads 512 data bytes plus 4 checksum bytes into memory beginning at <1>0. The disk address is cyl# 3, head# 2, sector# 20.

COMMAND: RSTATUS
DESCRIPTION:

The SMD controller status port is read and displayed.

EXAMPLE:

```
RSTAT
```

COMMAND: SEEK cyl
DESCRIPTION:

Issues the SEEK packet command to the SMD controller. If cyl = FFFFH (minus one), the controller rezeroes the drive and resets any drive faults. If cyl is >= 0, the controller seeks to that cylinder.

EXAMPLE:

```
SEEK FFFFH
SEEK 300
```

The first SEEK rezeroes the current disk drive and resets a drive fault.

The second SEEK positions the heads to cylinder 300.

COMMAND: SIZE
DESCRIPTION:

Issues the SIZE packet command to the SMD controller. The drive size values returned by the controller replace maxcyl, maxhead, and maxsec. These values are stored in the table of drive size parameters for the current unit. In addition, the format parameters are changed to correspond to the new size values. See the SUNIT command for a description of how the format parameters change.

EXAMPLE:

```
SIZE
```

COMMAND: SMEM maxseg
DESCRIPTION:

SMEM sets the number of memory segments (in this case, segment means 64K of memory) available to SMDMON. Several commands check against maxseg to see if an address increment would cause a memory address to go beyond the available memory (IROFF and IWOFF commands).

EXAMPLE:

```
SMEM 16
```

maxseg is set to 16. An IROFF or IWOFF command is ignored only if the buffer segment number is >= 16.

COMMAND: SRSEG rseg / SROFF roff
DESCRIPTION:

SRSEG shows the current read buffer address if no parameter is given. If a parameter is given, the read buffer segment is changed to that value.

SROFF shows the current read buffer addres if no parameter is given. Otherwise, the read buffer offset is changed to the parameter value.

EXAMPLE:

```
SRSEG
((SRSEG 2)(SROFF 8000H))
```

The first line does not affect the read-buffer address, but displays its current value.

The second line sets the read-buffer segment to 2 and the offset to 8000 hex.

COMMAND: STATUS
DESCRIPTION:

Complete status is displayed for the current disk unit, including: the Status Register value after the last-completed command, the packet status (0=IDLE, 1=GO, 2=BUSY, 3=DONE), the packet address and the packet contents.

EXAMPLE:

```
READ
STATUS
```

The STATUS shows the current unit status.

COMMAND: SUNIT unit
DESCRIPTION:

If NO parameter is given, this command shows the current unit number. If a parameter is given, SUNIT sets the current disk unit to the parameter value as well as retrieving the drive-size parameters stored in a table. The format parameters are also changed. The drive-size and format parameters change as follows:

```
maxcyl = # of cylinders on drive (per table)
maxhead = # of heads on drive (per table)
maxsec = # Active sectors/track (per table)
fmtsec = new maxsec value + 1
nsec = new maxsec value + 1
fspare = new maxsec value
fbad = ffffH
```

maxcyl, maxhead and maxsec are used to do boundary checking on commands that increment the disk address. fmtsec, nsec, fspare & fbad are parameters used during FMT, FVOL and FRD commands.

EXAMPLE:

```
SUNIT
SUNIT 1
```

The first line does not change the current unit, because there is no parameter given. It simply displays the current disk unit. No drive-size or format parameters are affected.

The second line sets the current unit to 1. The drive-size and format parameters change according to the values stored in a table in SMDMON.

COMMAND: SWSEG wseg / SWOFF woff
DESCRIPTION:

SWSEG shows the current write buffer address if no parameter is given. If a parameter is given, the write buffer segment is changed to that value.

SWOFF shows the current write buffer address if no parameter is given. Otherwise, the write buffer offset is changed to the parameter's value.

EXAMPLE:

```
SWOFF
((SWSEG 3)(SWOFF 0))
```

The first line does not affect the write-buffer address, but displays its current value.

The second line sets the write-buffer segment to 3 and the offset to 0.

COMMAND: SLNTH lnth
DESCRIPTION:

If no parameter is given, the length of the write and read buffers is shown in bytes.

If a parameter is given, the length of the write and read buffers is set to its value.

EXAMPLE:

```
SLNTH 1024
SLNTH
```

The first line sets the read/write buffers to be 1024 bytes long. The second line shows the current read/write buffer length, but does not change its value.

COMMAND: SCYL cyl / SHEAD head / SSEC sec /
SALL cyl head sec
DESCRIPTION:

SCYL sets the current cylinder number to cyl, if the parameter is supplied; otherwise, it shows the current disk address.

SHEAD sets the current disk head number if the parameter is supplied; otherwise, it shows the current disk address.

SSEC sets the current sector number on the current track if the parameter is given; otherwise, it shows the current disk address.

SALL sets the current cylinder, head and sector numbers, if parameters are given. Otherwise, it shows the current disk address.

EXAMPLE:

```
SHEAD 5
SALL 300 4 23
SCYL
```

The first line sets the head number to 5. The second line sets the current disk address to: cyl=300, head=4, sec=23 The third line shows the current disk address. It will be 300,4,23.

```
COMMAND:  SDRIVE  maxcyl maxhead maxsec /
          SDRV    maxcyl maxhead maxsec
```

DESCRIPTION:

If NO parameters appear, the current drive-size parameters are displayed, and no values are affected.

If any parameters are given, SDRIVE replaces the default values for the maximum numbers of cylinders, heads, and active sectors/track to maxcyl, maxhead and maxsec. The format parameters: fmtsec, nsec, fspare and fbad. are also changed.

The commands that increment the disk address (ICYL, IHEAD and ISEC) check against these values to prevent a non-existent disk sector from being accidentally addressed.

SDRV is a special abbreviation of SDRIVE.

EXAMPLE:

```
SDRIVE 1024 8 32
SDRV
```

The first line sets the drive-size parameters for the current unit to 1024 cylinders, 8 heds and 32 Active sectors/track. The format parameters are also changed. The second line does not change any values, but causes SMDMON to display the current drive-size values.

COMMAND: SFMT fmtsec fspare fbad fstart fintvl
DESCRIPTION:

If NO parameters are entered, the current format parameters are displayed.

If ANY parameters are entered, these parameters replace the format parameters. Subsequent FMT, FRD and FVOL commands use the newly entered parameters. See the descriptions of these commands for the functions of the format parameters.

Entering some commands implicitly resets the format parameters to values based upon the drive size. The commands that may change these parameters are: SDRIVE, SIZE, SUNIT.

EXAMPLE:

```
SFMT 33 FFFFH 20
```

This command sets the number of sectors/track to 33, the spare sector to ffffH (out of range) and the bad sector number to 20. A subsequent FMT command flags sector 20 as a bad sector.

COMMAND: SOFFSET offset
DESCRIPTION:

If a parameter is given, its value replaces the current drive offsets in the current unit packet. This value remains there until the packet is cleared (by issuing the INIT command) or until a subsequent SOFFSET command changes it. If no parameter is given, the current offsets are shown.

EXAMPLE:

```
OFFSET CH
```

000C (hex) replaces the current value in the OF field in the current unit packet.

COMMAND: WCMD cmdval
DESCRIPTION:

Writes cmdval to the SMD Controller's command register.

EXAMPLE:

```
WCMD 47H
```

Writes 47 (hexadecimal) to the command port. This corresponds to CR:CMD = 7 (set interrupt vector) and CR:DATA = 4. This instructs the controller to use interrupt vector 4.

```
COMMAND:  WRITE  wseg woff lnth cyl head sec
          WR     wseg woff lnth cyl head sec
```

DESCRIPTION:

Issues the WRITE packet command. The write buffer begins at <wseg>woff. The disk address written to is: cyl, head, sec. lnth bytes are written.

WR is a special abbreviation of WRITE.

EXAMPLE:

```
WRITE 2 0 512 10 0 2
```

Writes 512 bytes from memory beginning at <2>0 into cyl# 10, head# 0, sector# 2.

```
COMMAND:  WLONG  wseg woff lnth cyl head sec
```

DESCRIPTION:

Issues the WLONG (write long) packet command. The write buffer begins at <wseg>woff. The disk address that is written to is: cyl, head, sec. lnth is the number of data bytes to be written.

EXAMPLE:

```
WLONG 2 0 516 550 7 0
```

Writes 512 data bytes plus 4 checksum bytes from memory at <2>0 into cyl# 550, head# 7, sector# 0.

TEST NAME

TCUMON

PARAMETERS

None

DESCRIPTION

This program is an interactive exercise monitor for the Tape Controller Unit.

All commands must be entered in uppercase letters. Some commands require no parameters; but those that do, interpret the values entered as decimal numbers, unless followed by an H (hex number).

Example

200H = 512

For some commands to execute properly, other commands must be issued first. These dependencies are described in the command descriptions.

The command line may contain multiple commands. Repetition counts can be specified for the commands, and parentheses are used to force command groupings. Some parameters are maintained from command to command.

NOTES

TCUMON allows almost complete control of the Tape Controller Unit. It is possible to transfer data into segment 0, and crash the current invocation of SADIE.

EXAMPLES

(1) LOAD

This command loads the tape from the physical beginning-of-tape to the logical beginning-of-tape.

(2) STRK 1

This command performs a rewind and select track 1 (tracks are numbered 0-3).

(3) WRITE 1 A000H 1000H

If this command is executed after examples 1 and 2, it writes 1000 hex bytes of data, starting from location A000 of segment 1, onto track 1 of the tape.

(4) 4((1WOFF 1000H) (WRITE)) (WFM)

If this command is executed after examples 1, 2, and 3, locations B000 through EFFF of segment 1 are written as the second through fifth blocks of track 1. Each block is 1000 hex bytes long, and there is a file mark terminating this file. This illustrates the use of parentheses, implied parameters, and repetition counts.

COMMAND DESCRIPTIONS

The following list of commands are used in TCUMON. All parameters, when applicable, are indicated as lowercase variable names. All variables are assumed to be hexadecimal numbers. If the conditions described under dependencies are not met, the command will be rejected.

COMMAND: CEC

Description

CEC clears all command, lap, and error counters.

Dependencies

None

COMMAND: CCC

Description

CCC clears all command counters.

Dependencies

None

COMMAND: CLC

Description

CLC clears the lap counter, and displays the command and error summary.

Dependencies

None

Example

```
CLC
```

COMMAND: CMP srcseg srcoff desseg desoff count

Description

This command compares two buffers. The memory locations are addressed by segments (srcseg) and (desseg), and offsets within segments (srcoff) and (desoff). The number of words compared is (count). The addresses and contents of the first nonmatching locations, if any, are displayed.

Dependencies

None

Example

```
CMP 1 0 1 100H 100H
```

COMMAND: DIAG1

Description

DIAG1 performs a quick check of the I/O ports that interface with the TCU.

Dependencies

None

Example

```
DIAG1
```

COMMAND: DISPRT

Description

DISPRT disables the echoing of commands, and prevents the displaying of the command and error summaries.

Dependencies

None

Example

```
DISPRT
```

COMMAND: EGP

Description

EGP erases a three inch gap on the tape. The purpose is to get past a defect in the media.

Dependencies

The tape must be at, or beyond, the logical beginning of tape.

Example

```
EGP
```

COMMAND: ENPRT

Description

ENPRT enables the echoing of commands, and prevents the displaying of the command and error summaries.

Dependencies

None

Example

```
ENPRT
```

COMMAND: F fseg foff flen pat

Description

This command fills memory from segment (fseg), address (foff), for a length of (flen) words, with data pattern (pat).

Dependencies

None

Example

```
F 1 8 50 1234H
```

COMMAND: HELP

Description

HELP displays the available commands with their parameters, and gives terse explanations of their use.

Dependencies

None

Example

HELP

COMMAND: ILC

Description

ILC increments the lap counter, and displays the command and error summaries.

Dependencies

None

Example

ILC

COMMAND: ILNTH addval

Description

This command increments the length counter for read or write operations by (addval) words. The length must be less than 0x8000.

Dependencies

None

Example

ILNTH 1000H

COMMAND: INV invcom

Description

This command forces any 2 digit hex number into the command register. This verifies that invalid commands are properly rejected.

Dependencies

None

Example

```
INV 2A
```

COMMAND: IROFF addval

Description

This command increments the destination offset for read operations by (addval) words. It also increments across segment values. The segment value must be 3 or less.

Dependencies

None

Example

```
IROFF 80H
```

COMMAND: ITRK

Description

ITRK increments the track on which succeeding commands will operate, and sets the track. If on track 3, it sets track to 0.

Dependencies

Tape must be at, or beyond, the logical load point.

Example

```
ITRK
```

COMMAND: IVNT

Description

IVNT increments the unit on which succeeding commands will operate. If on unit 3, unit 0 is selected.

Dependencies

None

Example

IVNT

COMMAND: IWOFF addval

Description

This command increments the source offset address for write operations by (addval) words. It also increments across memory segment values. The segment value must be 3 or less.

Dependencies

None

Example

IWOFF 64

COMMAND: LOAD

Description

LOAD moves the tape from the physical beginning-of-tape to the logical beginning-of-tape.

Dependencies

Tape must be at the physical beginning-of-tape.

Example

LOAD

COMMAND: MODE m

Description

This command changes the mode of the tape controller to 0 or 1.

Dependencies

None

Example

```
MODE 1
```

COMMAND: MRTRY rtrycnt

Description

This command sets the maximum number of retries allowed for reads and writes. Default at power-up is 10 (rtrycnt = 0-15).

Dependencies

None

Example

```
MRTRY 8
```

COMMAND: Q

Description

This command quits TCUMON and returns to SADIE.

Dependencies

None

Example

```
Q
```

COMMAND: RAND rndseg rndoff rlnth

Description

This command fills memory with random data from segment (rndseg), address (rndoff), for a length of (rlen) words.

Dependencies

None

Example

```
RAND 3 0 1000H
```

COMMAND: READ rhad rlad rlen

Description

This command reads a block of data from tape and transfers it to segment (rhad), address (rlad). The (rlen) bytes are transferred to memory.

Dependencies

Tape must be at, or beyond, the logical beginning-of-tape.

Example

```
READ 0 B800H 2000H
```

COMMAND: REC

Description

REC displays all error counters, lap counters, and command counters.

Dependencies

None

Example

```
REC
```

COMMAND: REWIND

Description

This command rewinds the tape to the logical beginning of tape.

Dependencies

The tape must be at, or beyond, the logical beginning of tape.

Example

```
REWIND
```

COMMAND: SEL selcnt

Description

Controller selects a new drive (address selcnt, selcnt=0-3).

Dependencies

None

Example

```
SEL 1
```

COMMAND: SKBF skpcnt

Description

This command skips (skpcnt) blocks forward on the tape, or until either a file mark or the end of tape is detected.

Dependencies

Tape must be at, or beyond, the logical beginning-of-tape.

Example

SKBF 12

COMMAND: SKBR skpcnt

Description

This command skips (skpcnt) blocks backward on the tape, or until either a file mark or the logical beginning-of-tape is detected.

Dependencies

Tape must be beyond the logical beginning-of-tape.

Example

SKBR 14

COMMAND: SKFF skpcnt

Description

This command skips (skpcnt) files forward on the tape, or until the end of tape is detected.

Dependencies

Tape must be at, or beyond, the logical beginning-of-tape.

Example

SKFF 4

COMMAND: SKFR skpcnt

Description

Skips (skpcnt) files backward on the tape, or until the end of tape is detected.

Dependencies

Tape must be beyond the logical beginning of tape.

Example

```
SKFR 5
```

COMMAND: STAT

Description

STAT displays all tape controller interface registers.

Dependencies

None

Example

```
STAT
```

COMMAND: STRK trkcnt

Description

Controller rewinds the tape and selects a new track. (trkcnt = 0-3.)

Dependencies

Tape must be at, or beyond, the logical beginning-of-tape.

Example

```
STRK 3
```

COMMAND: UNLOAD

Description

UNLOAD moves the tape to the physical beginning-of-tape.

Dependencies

Tape must be at, or beyond, the logical beginning-of-tape.

Example

```
UNLOAD
```

COMMAND: WFM

Description

WFM writes a file mark on the tape.

Dependencies

The tape must be at, or beyond, the logical beginning-of-tape.

Example

```
WFM
```

COMMAND: WRITE whad wlad wlen

Description

This command writes one block of (wlen) bytes to the tape, from segment (whad), address (wlad).

Dependencies

Tape must be at, or beyond, the logical beginning-of-tape.

Example

```
WRITE 3 4000H 1A00H
```

COMMAND: WUP

Description

WUP moves tape to the end-of-tape, and back to the beginning, to establish tape tension.

Dependencies

None

Example

WUP

TEST NAME

TCOM- Tape Command Exerciser

PARAMETERS

Start= The first module of tape commands to be executed (default=0).
End = The last module of tape commands to be executed (default=0).
Unit = The unit number of the tape to be exercised (default=0)

NOTE

The default values cause all modules between Start and End to be executed.

DESCRIPTION

The following TCOM modules exercise the tape controller call commands:

Module 1: LOAD and UNLOAD commands
Module 2: REWIND and SKBR commands in Mode 0 and Mode 1
Module 3: READ, WRITE, and STRK commands
Module 4: SKBF and SKBR commands
Module 5: SKFF, SKFR, and WFM commands
Module 6: SEL, MRTRY, and EGP commands

The DIAG1 command is executed before entering the Start Module.

ERROR MESSAGES

TCOM issues warnings for correctable errors (e.g., no tape in drive, tape write-protected, etc.), and self-explanatory error messages if a tape operation fails. In addition to error messages, the TCU registers are also displayed.

LAP SUMMARY

Each module includes an introductory statement, a running commentary on the test progress, and a message signalling successful completion. There is no lap summary for the entire TCOM test.

NOTES

TCOM halts execution when a tape operation fails.

TEST NAME

TEX - tests magnetic tape cartridges

PARAMETERS

datpat: Entered in hexadecimal. The data pattern written on the tape (default =0x5555).

pr_wr_rr: Entered in hexadecimal. The first two nibbles, pr, are the number of times the pattern written to tape is read and verified (default=0x10). The second nibble, wr, is the number of retries used in writing the data (default=0). The third nibble, rr, is the number of retries permitted during a read operation (default=0xA).

numblk: Entered in hexadecimal. The number of 0x10 byte blocks which are written, read, and verified as a group.

u_st_et_w: Entered in hexadecimal. The first nibble selects which unit will test the tape. The second nibble specifies the track where testing begins. The third nibble is the track where testing ends. The fourth digit, if non-zero, causes a tape warmup. A tape warmup moves the tape to the logical end-of-tape, and then rewinds. Default values are 0, 0, 3, and 1, respectively.

DESCRIPTION

This test performs the following:

(1) Retries are set to wr.

(2) Write numblk blocks of 1000H bytes (starting at track st) with a data pattern of datpat.

(3) Retries are set to rr. Reads the blocks just written pr times, and compares to ensure valid data.

(4) If EOT is encountered, proceed to step 5; if not, return to step 1.

(5) Rewind and move to next track; if at track et+1, stop test; if not, return to step 1.

Counters and error totals are displayed at the end of each read and write. A detailed log of the last 17 errors recorded, is available by pressing START and entering "E" to the menu prompt.

ERROR MESSAGES

The user is warned if the unit selected does not exist, is not loaded with tape, or if the tape is write protected. If a tape operation fails, the TCU registers are displayed. Normal error messages from the test are displayed if a verification yields a compare error. The track, block, and data are displayed.

LAP SUMMARY

At the end of each read and write of "numblk" blocks; a table is displayed containing lap and parameter information, number of retries attempted, and errors classified as read or write, or hard or soft. This information is cumulative.

NOTES

This test is based on tape distributor's tape screening program.

TEST NAME

NEWMEM1

PARAMETERS

maxseg=maximum segment number to be tested (default=3)
minseg=minimum segment number to be tested (default=0)

DESCRIPTION

NEWMEM1 does a random data test on the memory segments given in the parameters. Each memory segment is 64K bytes. The test gets repeated n times, where n=#REPS in the test line.

NEWMEM1 must relocate the test code if segment 0 is tested. If minseg=maxseg=0, the code is relocated to segment 1 and the test runs from segment 1. If minseg=0 and maxseg>0, the code rotates through the segments to be tested. On the first lap, the code is in segment 0; on the second lap, it is in segment 1, and so on. On each lap all the segments from 0 to maxseg are tested except, of course, the segment where the code resides currently. This implies that at least two repetitions must be done for thorough testing.

The test fills each segment with random data, then reads it back for verification.

ERROR MESSAGES

If the data found in a memory location is not correct, a RANDOM TEST ERROR message appears. It gives the address where the error occurred, what the data should be (DATA=), and what was found in memory (BAD=).

All NEWMEM1 errors are reported as HARD errors.

LAP SUMMARY

After each repetition of NEWMEM1, the last 19 error messages are displayed. The errors are accumulated from one repetition to the next.

NOTES

NEWMEM1 turns the MMUs ON during the test. If NEWMEM1 indicates errors, SADIE diagnostic MMUTST5 or the SYSTEM POWER-UP DIAGNOSTICS (SPUD) should be run to ensure the MMU's integrity.

TEST NAME

NEWMEM2 - a quick memory test of memory segments

PARAMETERS

maxseg=maximum segment to be tested (default=3)
minseg=minimum segment to be tested (default=0)

DESCRIPTION

NEWMEM2 does three (3) write-read-compare tests on each segment tested. The three tests repeat n times, where n=#REPS in the test line. The tests are:

```
Test 1:  A simple data line test.
Test 2:  A simple address line test.
Test 3:  A block test where the segment is filled
           with x5555s; read and verified; filled
           with AAAAs; and read and verified.
```

Like NEWMEM1, NEWMEM2 relocates the code segment.

ERROR MESSAGES

The tests described above, display these messages if an error occurs:

```
DATA ERROR (test 1)
ADDR LINE TEST ERROR (test 2)
BLOCK ERROR (test 3)
```

Each message gives the address and data which is written (GOOD=xxxx), and the incorrect data which is read back (BAD=xxxx).

All NEWMEM2 errors are reported to SADIE as HARD errors.

LAP SUMMARY

After each repetition of NEWMEM2, the last 19 errors are displayed in tabular form. The errors accumulate from one repetition to the next.

NOTES

NEWMEM2 turns the MMUs ON during the test. If NEWMEM2 indicates errors, SADIE diagnostic MMUTST5 or the SYSTEM POWER-UP DIAGNOSTICS (SPUD) should be run to ensure the MMU's integrity.

TEST NAME

NEWMEM3 - a thorough n-cell-coupling test of memory segments

PARAMETERS

maxseg=maximum segment number to be tested (default=3)
minseg=minimum segment number to be tested (default=0)

DESCRIPTION

NEWMEM3 is a slow, but thorough, n-cell-coupling test. It flags problems the other memory tests may not reveal. The test takes approximately 20 minutes per segment tested. NEWMEM3 repeats n times, where n=#REPs in the test line.

NEWMEM3 relocates the code segment the same as NEWMEM1.

ERROR MESSAGES

NEWMEM3 displays a DATA ERROR message for each error. The error message gives the address where the error occurred, the value written (GOOD=xxxx), and the incorrect value read back (BAD=xxxx).

All NEWMEM3 errors are reported to SADIE as HARD errors.

LAP SUMMARY

At the conclusion of each test repetition, the last 19 error messages are displayed. The errors accumulate from one repetition to another.

NOTES

NEWMEM3 turns the MMUs ON during the test. If NEWMEM3 indicates errors, SADIE diagnostic MMUTST5 or the SYSTEM POWER-UP DIAGNOSTICS (SPUD) should be run to ensure the MMUs integrity.

TEST NAME

MMUTST5 - a series of tests of the MMUs

PARAMETERS

None

DESCRIPTION

MMUTST5 performs a series of tests on the MMUs. For each test, MMUTST5 displays a message informing the user that it is about to begin the test. The tests are performed on each of the three MMUs (CODE, DATA and STACK) unless otherwise specified.

(1) A block random data test, where 256 bytes of random data are written to memory, read back, and compared.

(2) A SDR test, where random data is written to each SDR, then read and verified.

(3) A "5555" and "AAAA" data test on all MMUs.

(4) A CONTROL register test, where random data is written to each control register, then read and verified.

(5) A test of the read-only flag, in the DATA and STACK MMU SDR's.

(6) A test of the LIMIT register, in the DATA and STACK MMU SDR's.

(7) A test of the DIRW flag, in the DATA and STACK MMU SDR's.

(8) A test of address translation using each SDR.

ERROR MESSAGES

MMUTST5 displays messages whenever an access violation causes a segment trap. The messages indicate where the MMU error occurred.

All MMUTST5 errors are reported to SADIE as HARD errors.

LAP SUMMARY

On completion of each repetition of the series of tests, MMUTST5 displays the lap number, error count, and for each MMU:

(1) The number of block data errors

(2) The number of random SDR errors

(3) The number of "5555" and "AAAA" data errors.

(4) The number of control register errors (SNR, DSC, and MODE registers)

(5) The number of access violations of the following types:

a. read only

b. limit

c. direction

d. translation

For information about the MMUs, refer to:

Z8010 Z-MMU Memory Management Unit Product
Specification, March, 1981
(Product Number 00-2046-A)

NOTES

None

TEST NAME

CENT.PRT (CENTRONICS printer interface test)
DP.PRT (DATA PRODUCTS printer interface test)

PARAMETERS

None

DESCRIPTION

CENT.PRT and DP.PRT are interactive tests of the Centronics and Data Products printer interface, respectively. The printer port tests, prompt the user to verify that the printer is online. If the printer is online, the tests send the printable character set to the PIO Channel B, n times, where n is the number of repetitions in the test line.

During these tests, PIO interrupts are disabled; the PIO is polled by the test.

ERROR MESSAGES

A message is displayed if the proper connection does not exist between the System 8000 and the printer, or if the printer port is busy too long.

Other error messages are self explanatory.

LAP SUMMARY

None

NOTES

None

TEST NAME

S16SIO

PARAMETERS

None

DESCRIPTION

S16SIO is an interactive, menu-driven test of the SIOs and CTCs not used by the console. (SADIE uses SIO #0, Channel B, to communicate with the console. The test assumes that SIO #0, Channel B, is functioning.)

A choice on the test menu is to exit the test. This is the only way the test should be exited; DO NOT PRESS "START" TO EXIT.

This test requires an auxiliary terminal, referred to by the test as AUX. The user selects the SIO to be tested, and the test prompts the user to plug AUX into a specified port on the system rear panel. Entering <CR> on the console signals the test to proceed.

The AUX terminal must not be connected to TTY0 until SADIE transfers control to S16SIO.

The test proceeds by displaying the entire set of ASCII printable characters continuously, until any key is pressed on the console. Then the test is in "echo mode", and any character pressed on AUX is echoed back to AUX by the test. Pressing any key on the CONSOLE terminates "echo mode", and returns to the S16SIO menu.

ERROR MESSAGES

None

LAP SUMMARY

Not applicable.

NOTES

S16SIO turns OFF interrupts from the console, when the user signals for the test to proceed. Therefore, DO NOT PRESS "START" TO EXIT this test, as interrupts will continue to be disabled when the PAUSE menu is displayed.

If S16SIO does not respond after the user signals for the test to proceed, check that AUX is connected to the correct port. To recover, hit several keys in succession on the console.

The test of SIO #0, Channel A, is slightly different from the other SIO tests. The first key pressed on the console, returns control to the menu. There is no "echo mode" test on SIO #0, Channel A.

TEST NAME

SIOMODEM - a test of the SIOs, including modem signals

PARAMETERS

SIOMODEM is an interactive test; it receives no parameters from SADIE.

DESCRIPTION

SIOMODEM tests the following SIO functions:

- Character transmission, polled mode.
- Character transmission, interrupt mode ("status affects vectors" false).
- Character transmission, interrupt mode ("status affects vectors" true).
- SIO modem signals (RTS,DCD,DTR,CTS), polled mode.
- SIO modem signals, interrupt mode.
- Transmit interrupts.
- External status interrupts.
- Character transmissions, mismatched baud rates.
- Character transmission at all standard baud rates.

The tests are arranged in three different test sequences, each of which is performed for each pair of tty ports. The tests are:

(1) Polled mode character/modem test. Performs a test of character transmission and SIO modem signals.

(2) Interrupt mode character test. Performs a test of character transmission in interrupt mode where "status affects vectors" false; tests character transmission of mismatched baud rates; and tests all standard baud rates.

(3) Modem/character interrupt test. Performs a test of character transmission in interrupt mode where "status affects vectors" true; tests SIO modem signals;

transmit interrupts; and external status interrupts.

Each test continues indefinitely, until the user terminates it by pressing a key on the console. (Do not "type ahead" during the test, or unexpected/unintended choices might be made accidentally.)

All errors result in an audible "beep", and a message on the console screen explaining the error. Beeps also occur when the program expects input from the console.

All SIO ports, including the console port, remote line port, and CPU expansion ports, can be tested. SIOMODEM (also known as SIO Test #3) performs all tests that are in the earlier version (SIO Test #2, or S16SIO).

SIOMODEM requires a special interconnecting cable to carry the modem signals between SIO ports (for example, port TTY2 might be looped back to TTY6). The test is interactive, and gives instructions to the user concerning how the SIOs are to be interconnected. It also explains what input is expected and at what time. The interconnecting cable consists of two standard RS232, 25-pin male connectors, wired as follows:

	A		B	
xmitted data	2	-----	3	received data
rcvd data	3	-----	2	transmitted data
RTS	4	-----	6	DCD
CTS	5	-----	20	DTR
DCD	6	-----	4	RTS
signal ground	7	-----	7	signal ground
DTR	20	-----	5	CTS

The other signals should not be wired. The length of the interconnecting cable should be about 8 feet. If the cable is not available, modem tests cannot be performed, and diagnostic test S16SIO should be run.

SIOMODEM offers the capability of moving the console port, so the standard console channel (TTY1, labelled "CONSOLE") can be tested the same as all other SIO channels. The console may be moved to any SIO channel which has already been tested. The remote line (TTY0) can also be tested when the console is moved. Testing of any pair of SIO ports may be skipped if desired. The test continues in a circular fashion, testing pairs of SIO channels, until the user responds to a prompt with "Q", to end all tests.

Console interrupts are disabled during the test, so NMI is not to be utilized to terminate the test. The NMI function does not normally operate while the test is executing.

ERROR MESSAGES

All error messages are preceded by an audible "beep" on the console.

ERROR: NO CHARACTER RECEIVED, POLLED MODE, TTY# FAILED ON CHARACTER XX

This is the first test performed. No character was transmitted between the ports in a polled-mode transmission. All possible ASCII characters from 00 to FF are transmitted, in both directions, and the error text reveals which character was attempting to be transmitted. If the failed character was 00, check the cable. This message occurs if one of the connectors is loose, a wire is broken, a pin has come loose, or if the cable is connected to the wrong port.

POLLED TTY# ###CHARS, ## ERRS, POLLED TTY# ###CHARS, ## ERRS

This message is displayed on the completion of each cycle of the polled mode test (after 256 characters have been successfully transmitted in each direction). The numbers represent cumulative totals since the test for the current pair of SIO ports was initiated. If the error count is zero in both directions, the test is successful. The error count represents the number of times that the character received differed from the character transmitted. The interrupt mode test continues until a character is input on the console.

TTY# RECEIVE MODEM ERROR STATUS1 = ##, DCD AND/OR CTS SET

An error occurred in polled-mode testing of the transmit modem signals. RTS and DTR were reset in the receive modem, but DCD and/or CTS failed to be reset in the transmit modem. Check the interconnecting cable for broken wires or loose pins.

TTY# TRANSMIT MODEM ERROR STATUS1 = ##, DCD AND/OR CTS SET

An error has occurred in polled-mode testing of the transmit modem signals. RTS and DTR were reset in the receive modem, but DCD and/or CTS failed to be reset in the transmit modem. Check the interconnecting cable.

TTY# RECEIVE MODEM ERROR STATUS2 = ##, DCD NOT TURNED BACK ON

Error in polled-mode testing of medem signals. RTS and DTR were set high in the transmit modem, but the expected status in the receive modem, CTS low and DCD high, did not occur.

TTY# TRANSMIT MODEM ERROR STATUS2 = ##, CTS NOT TURNED BACK ON

Error in polled-mode testing of modem signals. DTR was set in the receive modem, but the expected status in the transmit modem, CTS high and DCD low, did not occur.

SPEED= #### BAUD, TEST CYCLE ##

Displayed at the beginning of each interaction of the interrupt mode test. The interrupt mode test is similar to the polled mode test, except that SIO character received interrupts are used to signal the receipt of each character. All characters from 00 to FF are transmitted in each direction for each of the standard line speeds: 19,200, 9600, 4800, 1200, 300, and 110 baud. The interrupt mode test continues until a character is input on the console. When a character is input, the test terminates on completion of the current cycle. The LAST CYCLE display appears after the test cycle number, indicating that no new cycle will be started.

ERROR: INTERRUPT NOT RECEIVED, TTY#, FAILED ON CHARACTER ##

Character-received interrupt failed to occur in interrupt mode testing ("status affects vectors" =

false, interrupt vector =0x20). The TTY number of the receive port is given.

INTERRUPT TEST ON TTY#, ###CHARS, ##ERRS, ON TTY#, ##CHARS, ##ERRS

This message is displayed at the end of each iteration of each cycle of the interrupt mode test, and is analagous to the similar display of the polled mode test. Character and error counts are cumulative for all test cycles. Zero error counts in both directions indicate a successful test. The error count is the number of times that the transmitted character failed to match the received character.

ERROR: CHARACTERS MATCH WITH MIS-MATCHED BAUD RATES!!

Indicates that 256 characters were successfully transmitted despite differing baud rates in the transmit and receive ports. The baud rate clocks are not correctly set.

ERROR: DCD INTERRUPT NOT RECEIVED, TTY#, STATUS= ##

An error occurred in interrupt-mode testing of modem signals. RTS and DTR were reset in the transmit modem, which should have generated an interrupt in the receive modem when DCD and CTS are reset. If both polled mode and interrupt mode modem tests fail, the interconnecting cable should be carefully checked; however, if only one modem test fails, the SIO is probably at fault.

ERROR: DCD AND/OR CTS NOT CLEARED AFTER RTS/DTR CLEARED, TTY#, STATUS= ##

The interrupt described in the previous error occurred in the receive SIO modem; however, the expected status of zero for both DCD and CTS did not occur.

ERROR: DCD INTERRUPT NOT RECEIVED, TTY#, STATUS= ##

RTS was set on the transmit modem, which should have caused an interrupt on the receive modem when

DCD is set.

ERROR: TTY# DCD NOT SET AND CTS CLEARED AFTER RTS SET, STATUS = ##

The interrupt described in the previous error occurred; however, the expected status, DCD set and CTS reset, did not occur.

ERROR: CTS INTERRUPT NOT RECEIVED, TTY#, STATUS= ##

DTR was set on the receive port, which should have caused an interrupt on the transmit port when CTS was set true.

ERROR: TTY# CTS NOT SET AND DCD RESET AFTER DTR SET, STATUS= XX

The interrupt described in the previous error occurred, but the expected status, CTS true and DCD false, did not occur.

ERROR: NO CHAR RECEIVED, MODEM INTERRUPT TEST ON TTY#, RECEIVE STATUS= ## FAILED ON CHARACTER= ##, TRANSMIT STATUS= ##

A character-received interrupt failed to occur on the receive SIO during interrupt-mode modem testing. The interrupt-mode character test is repeated, except that "status affects vectors" is true, and interrupt vectors 0x24 and 0x2C are used for character interrupts, instead of 0x20. The status of the transmit and receive modems and the character attempting to be transmitted are given.

ERROR: MODEM LINES HAVE DROPPED BETWEEN TTY# AND TTY#, RECEIVE STATUS= ##, TRANSMIT STATUS= ##

An external/status interrupt, indicating a change in one of the states of the modem signals, occurred during the transmission of characters. The status of the receive and transmit modems is given. It may not be possible to recover from this error in which case the test is restarted.

ERROR: NO TRANSMIT INTERRUPTS OCCURRED, MODEM INTERRUPT TEST, TTY#, STATUS= ##

No transmit buffer empty interrupts have occurred on the transmit modem during the transmission of 256 characters.

MODEM INTERRUPTS ON TTY#, ##CHRS, ##ERRS, ON TTY#, ##CHRS, ##ERRS

Displayed on completion of each iteration of the modem interrupt test (after 256 characters are successfully transmitted in each direction). Analagous to the messages during the polled mode and interrupt mode character testing. The character and error counts are cumulative from the beginning of the test. The error count is the number of times the character transmitted did not match the character received. The test is successful when the error count is zero in both directions.

ERROR: SPECIAL RECEIVE CONDITION INTERRUPT

No test currently implemented should cause this interrupt to occur. The receive SIO modem believes it is detecting a serious error condition, such as a parity or framing error.

LAP SUMMARY

This is an interactive test and does not use a lap count or summary. The test continues until the user responds to a prompt with "Q", to quit (or terminate) all tests.

NOTES

A special SIO interconnecting cable, described in the Description of this test, is required.

Console interrupts are disabled during this test. START (NMI) is not to be used to terminate the test, and should not be pressed. The test will terminate upon responding to a prompt with a "Q".

Errors which occur during polled-mode character, or modem signal testing, are usually caused by broken wires or loose pins in the cable, or by a connector which is either loose or connected to the wrong I/O port.

The most all-inclusive test is the Interrupt-Mode Modem Control test. This test is recommended for extended testing, since it tests for the greatest number of possible errors.

TEST NAME

ECCTEST - tests of ECC error correction and detection and ECC memory

PARAMETERS

Parameter 1 = segment -- the segment where the 1-bit and 2-bit error tests will take place. Default = 1. Range is 1 to 127.

Parameter 2 - offset. The offset where the 1-bit and 2-bit error test will take place. Default = 0. Range is 0 to FFFC (hex).

Parameter 3 - maxseg. The maximum segment to be tested in the ECC memory test. Default = 3. Range is 0 to 127. (If maxseg = 0, however, no ECC memory test will occur.)

DESCRIPTION

ECCTEST is a series of three tests. The series repeats n times, where n is the number of repetitions specified in the SADIE test line.

Test 1 tests the ability of the ECC to correct and report 1-bit memory errors. A single-bit error is forced into memory by writing a 32-bit pattern that is corrupted in one bit. The correct check byte for the uncorrupted data is written to the check-byte register. Then error correction is enabled and the memory location is read. The data read should match the original, uncorrupted data. If not, an "uncorrected" error message is displayed. If error correction occurred, the ECC error count register is checked to be sure the ECC has recorded the error. If not, and "unreported" error message appears.

Test 1 repeats 64 K times during each lap. Each repetition is done using a different data pattern and different corrupted bit.

Test 2 tests the ECC's ability to detect and report 2-bit errors. A 2-bit error is forced by corrupting any 2-bits in a 32-bit data pattern and 7-bit check byte. The corrupted data pattern is written to memory and the corrupted check byte is written to the check byte register. Error detection is enabled and the System Configuration Register is set so that ECC errors generate a Non-Maskable Interrupt (NMI). Then the memory location is read. If no NMI occurs, an "undetected" error message is displayed. If the NMI occurs, the ECC error count register is checked to verify that the ECC recorded the 2-bit error. If not, an "unreported" error message is displayed.

Test 2 repeats 64 K times during each lap of ECCTEST. Each repetition is done using a different data pattern and a different pair of corrupted bits.

Test 3 is an ECC memory test. Every double-word location in segments 1 through maxseg is tested using three patterns. The patterns are:

```
        0, 0x01090000 and 0x00000106
```

For each pattern, the correct pattern is written to the location and the correct check byte is written to the check-byte register. Then the location is read with memory correction enabled. If the data read does not match the data written, a data error message is displayed. If the data matches the original, the ECC error count register is checked to see if the ECC recorded error correction occurred. If so, a "corrected" error message is displayed.

ERROR MESSAGES

Test 1:

```
    SB--orig: xxxxxxxx-cc, wrote: xxxxxxxx-cc, read: xxxxxxxx-cc

         where xxxxxxxx = the data pattern (hex)
                     cc = the check-byte (hex)
                     SB = "SINGLE BIT"

    SB--Corrected 1-bit error unreported: count = cccc, report
        = xxxx

         where cccc = the test's count of 1-bit errors (hex)
               xxxx = the error count register content (hex)
```

Test 2:

```
    TB--orig: xxxxxxxx-cc, wrote: xxxxxxxx-cc, read: xxxxxxxx-cc

         where xxxxxxxx = the data pattern (hex)
                     cc = the check-byte (hex)
                     TB = "TWO BIT"

    Detected 2-bit error unreported: count = cccc, report = xxxx
```

```
          where cccc = test's error count (hex)
                xxxx = the error count register (hex)
```

Test 3:

```
    MEM--<sss>ooooH  wrote: xxxxxxxx-cc read: xxxxxxxx-cc

         where sss = segment number (decimal)
              0000 = the offset (hex)
          xxxxxxxx = the data pattern (hex)
                cc = the check-byte (hex)
               MEM = "MEMORY TEST"

    MEM--correction at <sss>ooooH  pattern = xxxxxxxx-cc

         where sss = segment (decimal)
              oooo = offset (hex)
          xxxxxxxx = data pattern (hex)
                cc = check-byte (hex)
```

LAP SUMMARY

At the completion of each lap of ECCTEST, a table of statistics is displayed. It shows:

1) For single-bit errors:

   a) number of 1-bit errors forced

   b) number of 1-bit errors corrected & reported

   c) number of 1-bit errors uncorrected

   d) number of 1-bit errors corrected, but unreported

If the number of forced errors = the number corrected and reported (a=b), then no errors have occurred.

2) For two-bit errors:

   a) number of 2-bit errors forced

   b) number of 2-bit errors detected & reported

   c) number of 2-bit errors undetected

   d) number of 2-bit errors detected but unreported

If the number of errors forced = the number detected and reported, no ECC errors have occurred.

3) Memory test:

   a) number of trials the memory test has completed

   b) numbers of data errors (data read does not equal data written)

   c) number of 1-bit errors reported by the ECC. (data read = data written, but error count register indicates an error was corrected).

NOTES

Test 3 may show spurious errors if memory has not been initialized after power on. To be sure memory is initialized, enter the **T** command to the Prom Monitor before booting SADIE with the **Z T** command. Alternately, memory can be initialized by running NEWMEM2 on segments 0 through maxseg.

APPENDIX B
WINCHESTER DISK CONTROLLER COMMANDS

B.1. General

Appendix C describes all the commands that the host CPU sends to the controller through the command registers. Each description indentifies the registers that the host CPU uses. Bits 0 through 4 of the command byte form the command field. Refer to Section 4 for additional information. The term CP (Command Port) means command register as used in Section 4. For example, CP0 is command register xx00 and CP1 is command register xx01.

B.2. Format / UNIT /

The controller formats an entire disk drive with a single controller command. The CPU sends the FORMAT command to the controller by writing the unit number of the drive to be formatted as CP1 and by issuing the Format command to the command port, CP0. Format writes the data field of each sector with a pattern of alternating ones and zeros (0AA hexadecimal) and rewrites each sector's header. After formatting the target unit, the entire format is read back and the header field of each sector is validated. If an invalid header is found, the Unrecoverable Error bit in the operation ending status byte is set, together with the Verify Failure bit in the Operation Error Status Byte.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

B.3. Read Sector

/ UNIT / HEAD / CYLINDER / SECTOR / WORD COUNT / ADDRESS

The Read Sector command requires both a disk address [unit, head, cylinder, sector] and a host system buffer address [word count, address]. Less than one sector of information is transferred if the word count is less than a sector (512 bytes) in length. The controller command for multiple sector reads and single sector reads is equivalent with the word count determining the number of sectors for the transfer.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

/ HEAD / Head is written to the controller through CP2.

/ CYLINDER / The target cylinder number is written to the controller through command ports CP3 and CP4. The low order eight bits of the ten-bit cylinder address are issued to the controller through CP3 and the high order two bits are issued through CP4 bits zero and one.

/ SECTOR / The sector number of the sector to be read is passed to the controller through the sector register, CP5.

/ WORD COUNT / The number of words to read from the disk is sent to the controller through the word count control register pair, CP8 and CP9. A word count greater than a single sector length is interpreted as a request for a multiple sector read.

/ ADDRESS / The 24-bit address of the host data buffer is passed to the controller through a register triplet; CP10, CP11, and CP12.

B.4. Write Sector

/ UNIT / HEAD / CYLINDER / SECTOR / WORD COUNT / ADDRESS/

The Write Sector command requires both a disk address [unit, head, cylinder, sector] and a host system buffer address [word count, address]. Less than one sector of data is read from host memory if the word count is less than a sector in length. The controller command for multiple sector writes and single sector writes is equivalent to the word count determining the number of sectors for the transfer.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

/ HEAD / Head is written to the controller through CP2.

/ CYLINDER / The target cylinder number is written to the controller through command ports CP3 and CP4. The low order eight bits of the ten-bit cylinder address are issued to the controller through CP3 and the high order two bits are issued through CP4 bits zero and one.

/ SECTOR / The sector number of the sector to be read is sent to the controller through the sector register, CP5.

/ WORD COUNT / The number of words to be written to the disk is sent to the controller through the word count control register pair, CP8 and CP9. A word count greater than a single sector length is interpreted as a request for a multiple sector write.

/ ADDRESS / The 24-bit address of the host data buffer is sent to the controller through a register triplet; CP10, CP11, and CP12.

B.5. Read Detailed Status / UNIT / ADDRESS /

The Read Detailed Status command transfers three words of the detailed disk status for unit number UNIT to the host system memory beginning at location ADDRESS. The act of reading detailed status clears the operation error status byte unless the read status command itself times out. In this case the time-out error will be set. See Table 11 for the error status byte definition.

/ UNIT / Unit is written to the controller through CP1. Any value between 0 and 3 is valid.

/ ADDRESS / The 24-bit address of the host data buffer is sent to the controller through a register triplet; CP10, CP11, and CP12.

B.6. Restore / UNIT /

The Restore command recalibrates the drive seek circuitry by positioning the heads at track zero, and clears the fault status bit in the drive. The restore operation executes at a slower rate than a seek to track zero. Recalibrate should only be used in response to a drive fault. The controller automatically provides a restore and command retry when a drive fault occurs unless that option has been reselected.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

B.7. Null

The null command presents an operation ending status code "operation complete" and performs no disk or controller operation.

B.8. Seek / UNIT / CYLINDER /

The controller generates an automatic seek before disk read and write operations. This feature can be deselected under host software control. The host system can opt to defeat the automatic seek before read/write and issue its own seek commands. Deselection of the automatic seek facility has

the side affect of deselecting overlapped seeks in the controller.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

/ CYLINDER / The target cylinder number is written to the controller through command ports CP3 and CP4. The low order eight bits of the ten-bit cylinder address are issued to the controller through CP3 and the high order two bits are issued through CP4 bits zero and one.

B.9. Set Strobe/Offset / UNIT / SO /

The Set Strobe/Offset command is used to select a value for either or both the data strobe timing or the head positioning during data read operations. Both the data strobe and head offset are reset during disk write operations as required by the disk drive circuitry. This command reduces the sector read time dramatically on media that require a strobe/offset value to recover the data. The set strobe/offset command is useful with removable media since it can correct small differences between head alignment and/or data strobe times between drives. It is of questionable merit for use with fixed media. The Strobe/Offset value is input to the controller through the low byte of CP1, the HEAD select register.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

/ SO / The Strobe/Offset value is issued to the controller by command register CP2.

B.10. Set Write Protect / UNIT / SURFACES /

The Set Write Protect command allows each unit to be selectively write-protected at the controller. Write commands to a protected surface are inhibited at the controller.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

B.11. Format Read / UNIT / HEAD / CYLINDER / / SECTOR / ADDRESS /

When the Format Read command is issued, the host diagnostics can read an entire sector, both sector header information and data.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

/ HEAD / Head is written to the controller through CP2.

/ CYLINDER / The target cylinder number is written to the controller through command ports CP3 and CP4. The low order eight bits of the ten bit cylinder address are issued to the controller through CP3 and the high order two bits are issued through CP4 bits zero and one.

/ SECTOR / The sector number of the sector to be read is sent to the controller through the sector register, CP5.

/ ADDRESS / The 24-bit address of the host data buffer is passed to the controller through a register triplet; CP10, CP11, and CP12.

B.12. Set Interrupt Address. / SECTOR /

The Set Interrupt Address allows the host system to specify the low order eight bits of the address for interrupt transfer. That address and an eight-bit Operation Ending Status byte are sent to the address bus lines when the controller issues an interrupt. Typically, the Set Interrupt Address command is issued once shortly after the controller is powered up.

/ SECTOR / The sector number of the sector to be read is sent to the controller through the sector register, CP5.

B.13. Self Test

This command forces the controller to execute its self diagnostics. Both the command reject and the command accepted bits are set in the command status byte if a self test error is detected. It is possible that the self test error will preclude the setting of the bits in the command status byte.

B.14. Format Verify / UNIT / HEAD / CYLINDER /

This is a diagnostic command that verifies the integrity of the disk format on a track. The controller reads each individual sector of the track and ensures that its header is good. If a format error is detected, the Verification Error bit is set in the Operation Error Status Byte and the Unrecoverable Error on Operation bit in the Operation Ending Status byte is set.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

/ HEAD / Head is written to the controller through CP2.

/ CYLINDER / The target cylinder number is written to the controller through command ports CP3 and CP4. The low order eight bits of the ten-bit cylinder address are issued to the controller through CP3 and the high order two bits are issued through CP4 bits zero and one.

B.15. Unit Format Verify / UNIT /

This is a diagnostic command. It is used to verify the format of an entire unit. If a format error is detected, the Verification Error bit is set in the Operation Error Status Byte, and the Unrecoverable Error on Operation bit in the Operation Ending Status Byte is also set. The cylinder where the failure occurred is read back by the Read Detailed Status command.

/ UNIT / Unit is written to the controller through CP1. Any value between zero and three is valid.

B.16. Command Notes

An explicit format write command is not provided in the controller's command repertoire. All disk write operations write the sector address field as well as the data. Therefore, a write sector command performs a format write operation.

APPENDIX C
CARTRIDGE TAPE ERROR CONDITIONS

C.1. General

Table C-1 lists the error conditions that can result from the commands that the tape controller receives from the host CPU. The table lists the commands, the resulting conditions, and the status bits that the conditions set.

Table C-1 Cartridge Tape Error Conditions

COMMAND RECEIVED	CONDITION	STATUS BITS SET
Initialization	Tape (if present) rewinds for more than 88 sec	INTV and HWERR
DIAG	ROM checksum error	INTV and HWERR
DIAG	Fifo test error	INTV and HWERR
DIAG	Handshake test failed	INTV and HWERR
Any command	Host wrote to ports in byte or double word mode	CMDREJ
Any undefined command		CMDREJ and INVAL
Any command except DIAG	Drive busy, or drive not selected	CMDREJ, INTV, INAP, and HWERR
Any command except DIAG, MRTRY, SEL, MODE	No tape	CMDREJ, INTV, and NOTAP
Any command except DIAG, MRTRY, SEL, LOAD, MODE	Tape not logically loaded	CMDREJ and INAP
WRITE, WFM EGP	Tape write-protected	CMDREJ, INAP, and PROT

LOAD	Tape already logically loaded	CMDREJ and INAP
READ, WRITE, SKBF, SKFF, EGP	Tape at logical end of tape	CMDREJ, INAP, and LEO
SKBR, SKFR, REWIND	Tape at logical beginning of tape	CMDREJ, INAP, and LBO
READ, WRITE	DMA buffer length greater than or equal to 32K bytes (k=1024 bytes)	CMDREJ and BPARM
READ, WRITE	DMA start address and DMA length greater than 24 bits	CMDREJ and BPARM
READ	DMA buffer length not even (bit 0=0)	CMDREJ and BPARM
READ	Blank tape (more than 48 inches) encountered	DATERR and BLKTAP
READ	Attempted buffer overflow during DMA	OVERFL
READ, WRITE, WFM	Bad read (read after write for WRITE and WFM) as indicated by a bad CRCC after retrying the operation the maximum permissible number of times	DATERR
READ	File mark encountered	CMDREJ and FMDET
READ, WRITE	Fifo error (overflow or underrun) after retrying the operation the maximum permissable number of times	FFERR
WRITE	Deck stopped taking data during write	INTV and HWERR
READ, WRITE, WFM	One or more retry attempts made	RTRYAT and number of retries in low byte of status 1

		register
Any command except READ, WRITE, WFM		Number of retries in low byte of status 1 register = 0
WRITE, WFM	Encountered end of tape before WRITE began or retry after error pushes the beginning of the block past the end of tape.	CMDREJ, INAP, and LEO (and RTRYAT and numbe of retries in low byt of status 1 register if retries attempted)
SKBF, SKFF	Encounterd logical end of tape	SKNDE and LEOT and number of blocks/files skipped in high byte of status 1.
SKBR, SKFR	Encountered logical beginning of tape	SKNDNE and LBOT and number of blocks/files skipped in high byte of status 1.
SKBF, SKFF	Blank tape (more than 48 inches) encountered	SKNDNE and BLKTAP an number of blocks/files skipped in high byte of status 1 register.
SKBF, SKBR	File mark encountered SKNDNE	and FMDET and and number of blocks skipped in high byte of status 1 register.
SKBF, SKBR, SKFF, SKFR		Number of blocks/files skipped in high byte of status 1 register.
Any command except SKBF, SKBR, SKFF, SKFR		High byte of status 1 register = 0.
READ, WRITE, SKBF, SKFF, UNLD, REWIND, STRK	Rewind takes more than 88 seconds. Rewind occurs whenever track boundaries are	INTV and HWERR

	crossed.	
SKFR	During forward motion (after FM detected) blank tape or end of tape encountered	INTV and HWERR
SKBR, SKFR, READ WRITE, WFM	During forward motion (after track boundary) tape failed to move off BOT for greater than 166 ms.	INTV and HWERR
Any command	Tape write-protected	PROT
Any command	Tape at logical load point	LLP
Any command	Tape at end of tape	LEOT
Any command		UNITO, 1 and TRKO, 1 set to indicate unit and track selected. TRKO, 1 = 0 if drive not logically loaded
READ, WRITE, WFM, SKBF, SKBR, SKFF, SKFR	Data detected for > 40" (32K bytes)	INTV and HWERR
REWIND, STRK, READ, WRITE, WFM, EGP, SKBF, SKBR, SKFF, SKFR	Tape moves forward > 2.1 sec and fails to move from Bot to LPS	INTV and HWERR
READ, WRITE, WFM, SKBR, SEL, MODE	No block found where one is know to exist	INTV and HWERR
READ, WRITE, WFM	During retry blank tape found while moving in reverse	INTV and HWERR

03-0244-01

NOTE TO USER FOR SADIE VER. 3.2
April 20, 1983

Purpose:

This note describes some major changes between SADIE 3.1 and SADIE 3.2. It also fills in some gaps between the release date of SADIE 3.2 and the release of the latest SADIE documentation in the System 8000 Hardware Reference Manuals for Models 11 and 21/31.

Item 1:

NEWMEM1, NEWMEM2 and NEWMEM3 memory diagnostics have been replaced by a new diagnostic, MEMTEST. MEMTEST supports all memory array boards currently available for the System 8000. The version of MEMTEST in SADIE 3.2 does not, however, test the ICP 8/02 slave memory. The boards supported are:

PARITY	-- 256K and 512K
ECC	-- 256K and 1 Megabyte

MEMTEST is actually a series of tests. Included are a segment uniqueness test, an address bus test, a data bus test, several board trace and pin continuity tests and a ram test. If these tests go smoothly, parity ram or checkbyte ram are also tested. All tests are done with mmu's disabled.

MEMTEST is a substantial improvement over previous memory tests. It is much better at narrowing memory problems to a single chip or a small region of the memory array board. Ram errors are unlikely to escape from MEMTEST as they could in earlier tests. But the primary improvement of MEMTEST is in its error displays, which are visual representations of the memory array board with bad chip positions clearly marked.

MEMTEST does have its limitations. Its primary goal was to be a fast but thorough ram test, with secondary emphasis placed on other memory array board functions. While it is very good at finding bad ram chips, it is not as good at narrowing down other types of errors. This is particularly true when it is used with ECC memory array boards.

Item 2:

ECCTEST test 3, the checkbyte ram test, has been removed. The checkbyte ram test is now included in MEMTEST.

Item 3:

Two new diagnostics test the new 9-track magnetic tape controller board (MTC). They are MTCOM and MTCMON. MTCOM runs through the controller's repertoire of commands. It is divided into 7 modules, of which modules 1 through 4 are run by default. Parameters control which modules are run as well as the tape unit number tested.

MTCOM modules are:

Module 1: WRITE and WFM (write file mark only)

Module 2: READ, SFF (skip file forward), READR (read reverse) and SFR (skip file reverse)

Module 3: SBF (skip block forward), and SBR (skip block reverse)

Module 4: SFFR (skip file forward, reading) and SFRR (skip file reverse, reading)

Module 5: EFL (erase fixed length gap) and EVL (erase variable length gap)

Module 6: SE (security erase) and RUO (rewind, unload and offline)

Module 7: An interactive test of the drive's handling of the write enable ring

MTCMON is an interactive monitor that enables the user to issue single commands or strings of commands to the MTC. A HELP command allows the user to see a menu of the available commands. MTCMON commands have the same basic syntax rules as the other interactive monitors in SADIE: TCUMON, WDCMON, MDCMON and SMDMON.

MTCMON and MTCOM documentation is included in the NINE-TRACK TAPE SUBSYSTEM O&M MANUAL. The "WUP" (tape warmup) command described in the documentation of MTCMON is no longer a valid command.

Item 4:

S16SIO test has been renamed to SIOTEST. It now supports a 24-user system. SIOMODEM also supports 24 users.

Systems Publications

Reader's Comments

Your feedback about this document helps us ascertain your needs and fulfill them in the future. Please take the time to fill out this questionaire and return it to us. this information will be helpful to us and, in time, to future users of Zilog products.

Your Name: ______________________________

Company Name: ______________________________

Address: ______________________________

Title of this document: ______________________________

Briefly describe application: ______________________________

____

____

Does this publication meet your needs? □ Yes □ No If not, why not? ______________________________

____

____

How are you using this publication?

□ As an introduction to the subject?

□ As a reference manual?

□ As an instructor or student?

How do you find the material?

	Excellent	Good	Poor
Technicality	□	□	□
Organization	□	□	□
Completeness	□	□	□

What would have improved the material? ______________________________

____

____

Other comments and suggestions: ______________________________

____

____

If you found any mistakes in this document, please let us know what and where they are: ______________________________

____

____