♦♦♦♦♦ 2275 MENUE ♦♦♦♦♦

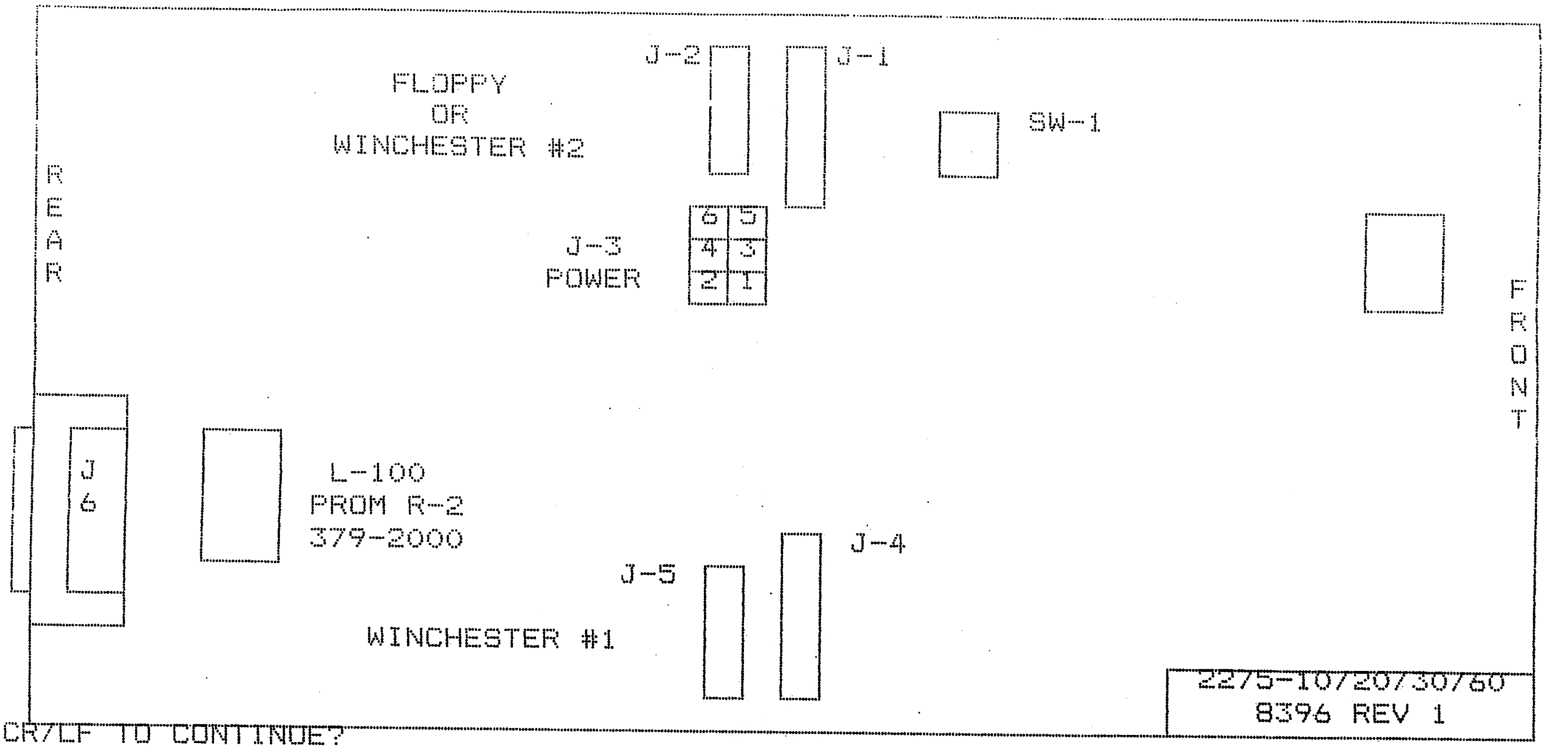
1	NOT USED
2	RETURN TO MENUE #1
3	8396 BOARD
4	8397 BOARD
5	SEAGATE 10 MEG DRIVE
6	IMI 10 MEG DRIVE
7	2275 FUSE LIST
8	2275 SWITCH INFO
9	MPI FLOPPY DRIVE
A	FLOPPY I/O CONNECTOR
B	WINCHESTER CONNECTOR
C	TANDON FLOPPY DRIVE
D	PARTS LISTING
E	QUANTUM 30 MEG DRIVE
F	2275 GENERAL INFO

♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦

KEY IN HEX CHARACTER TO SELECT ITEM ON MENUE

2275 Info

Paul M.

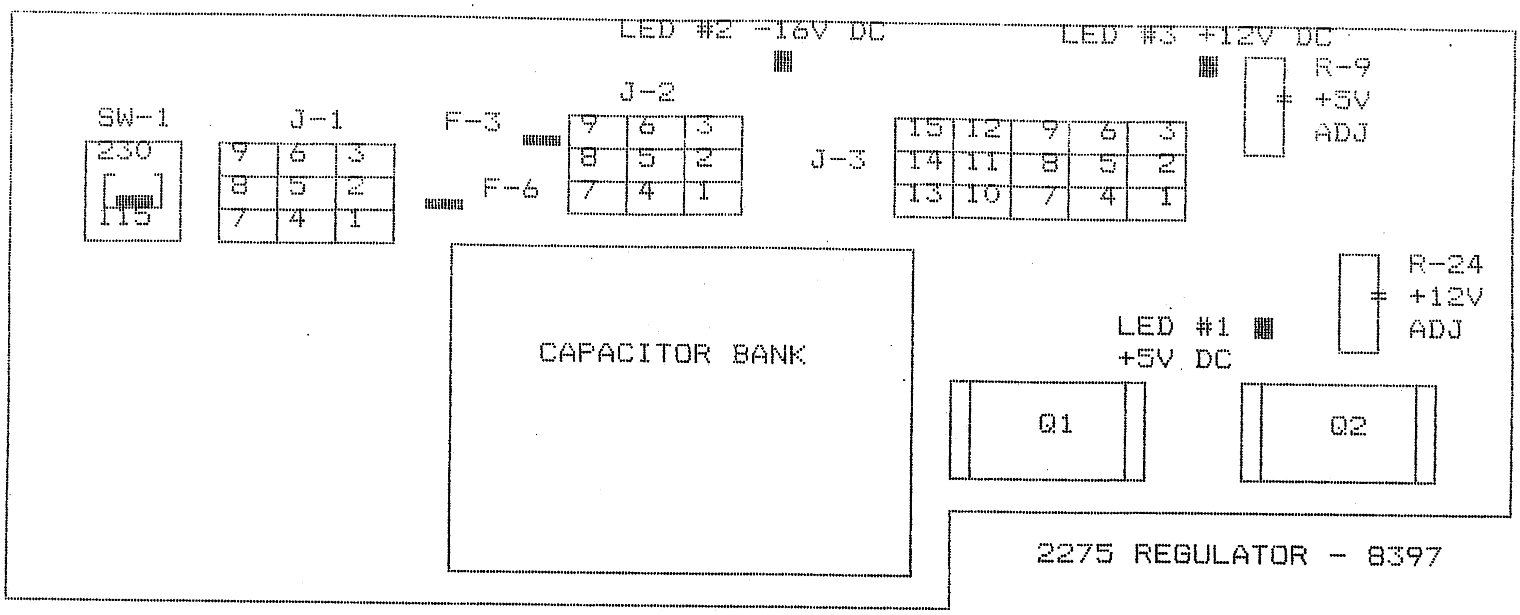


CR/LF TO CONTINUE?

8396 BOARD
CONNECTOR J-3

```
[1]--- +5V REGULATED
[2]--- +16V UNREGULATED - (INPUT TO +12V REGULATOR)
[3]--- +5V REGULATED
[4]--- -16V UNREGULATED - (INPUT TO -12V REGULATOR)
[5]--- 0 VOLTS DC
[6]--- 0 VOLTS DC
```

CR/LF TO RESTART -- FN/TAB TO RETURN TO MENUE?



CR/LF TO CONTINUE?

8397 REGULATOR
CONNECTOR J-1

```
[1]--- AC IN FROM FUSE AND LINE SWITCH  *
[2]--- NOT USED
[3]--- 110/220VAC
[4]--- 110/220VAC
[5]--- AC IN FROM LINE SWITCH
[6]--- 110/220VAC
[7]--- 110/220VAC
[8]--- AC TO FAN
[9]--- AC TO FAN
```

* NOTE:
PIN #1 GOES TO THERMAL CIRCUIT BREAKER ON
HEATSINK OF Q1 THEN RETURNS TO SWITCH S-1

CR/LF TO CONTINUE?

8397 REGULATOR
CONNECTOR J-2

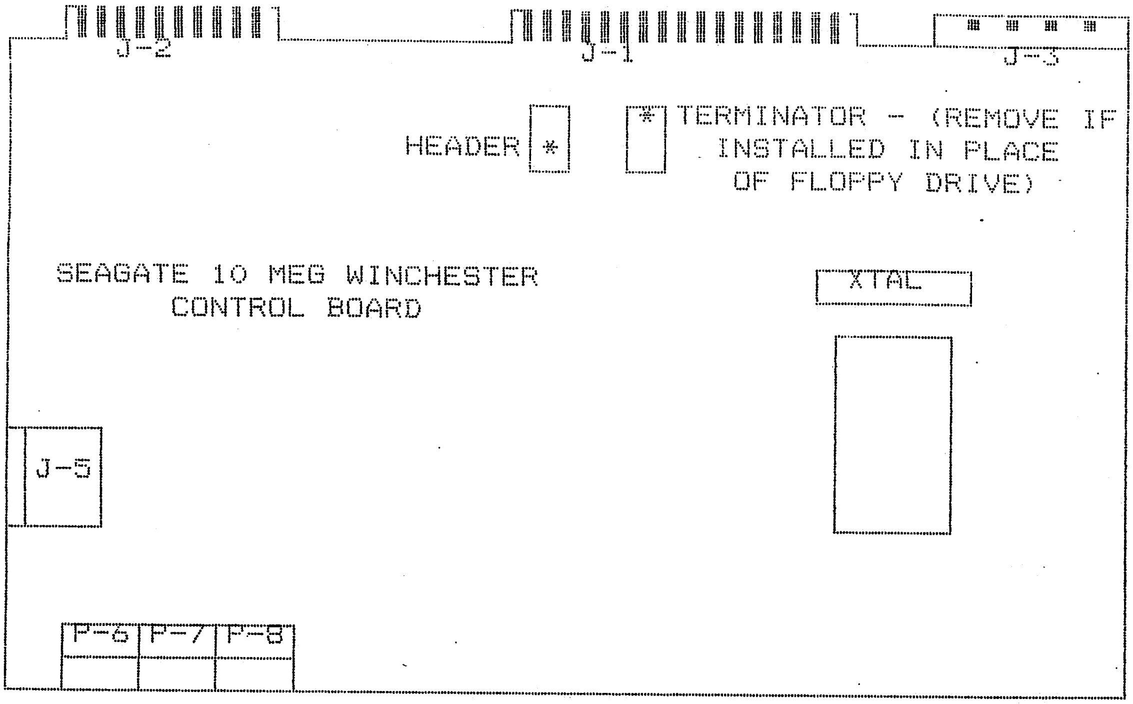
1 --- 0 VOLTS DC
2 --- AC 1
3 --- AC 2
4 --- 0 VOLTS DC
5 --- 0 VOLTS DC
6 --- NOT USED
7 --- AC 4
8 --- AC 5
9 --- AC 3

CR/LF TO CONTINUE?

8397 REGULATOR
CONNECTOR J-3

1	--- +12 VOLTS OUT
2	--- +5 VOLTS OUT
3	--- +5 VOLTS OUT
4	--- +12 VOLTS OUT
5	--- +5 VOLTS OUT
6	--- +5 VOLTS OUT
7	--- 0 VOLTS DC
8	--- 0 VOLTS DC
9	--- 0 VOLTS DC
10	--- 0 VOLTS DC
11	--- 0 VOLTS DC
12	--- 0 VOLTS DC
13	--- +16 VOLTS UNREGULATED
14	--- NOT USED
15	--- -16 VOLTS UNREGULATED

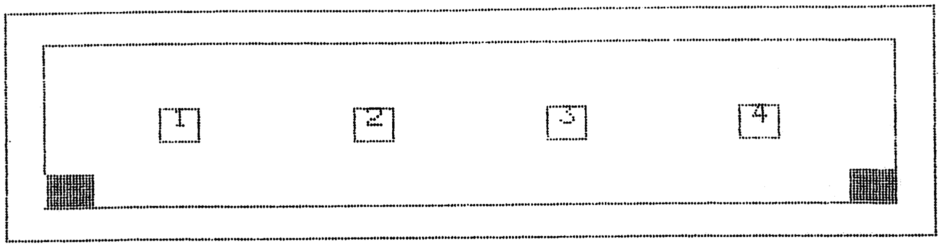
CR/LF TO START PROGRAM AGAIN - FN/TAB TO RETURN TO MENUE?



NOTE:
ALL CONNECTORS
ARE KEYED.

CR/LF TO
CONTINUE
?

J-3 VOLTAGE CONNECTOR



PIN #1 = +12V DC

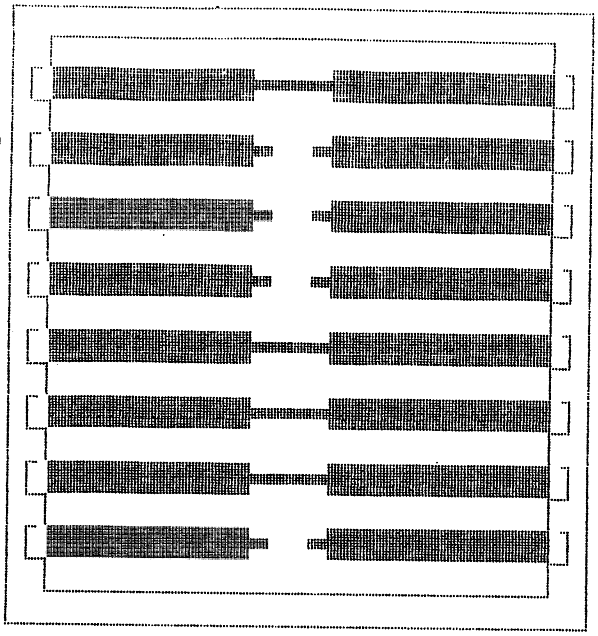
PIN #2 = +12V RETURN

PIN #3 = +5V RETURN

PIN #4 = +5V DC

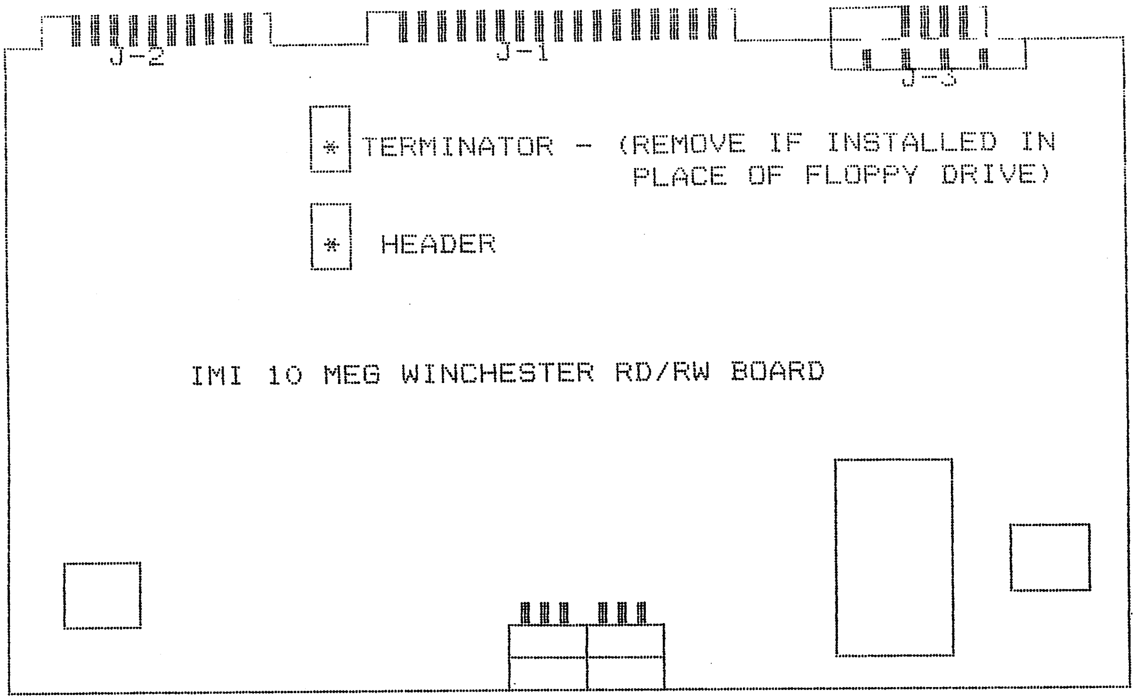
CR/LF TO CONTINUE?

HEADER FOR SEAGATE WINCHESTER



9	8	** DRIVE SELECT #1
10	7	** DRIVE SELECT #2
11	6	** DRIVE SELECT #3
12	5	** DRIVE SELECT #4
13	4	** GROUND - NO CONNECTION
14	3	** GROUND - NO CONNECTION
15	2	** GROUND - NO CONNECTION
16	1	** DRIVE ALWAYS SELECTED

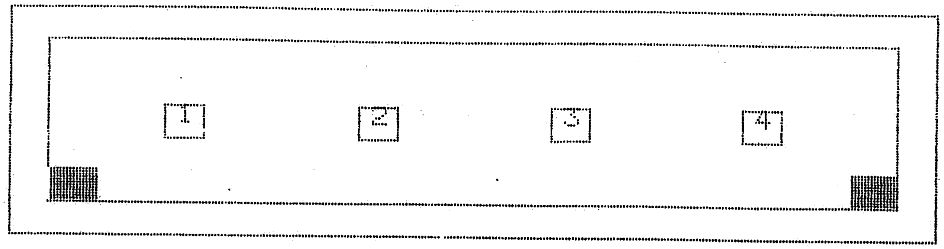
FN/TAB TO RETURN TO MENUE



NOTE:
ALL CONNECTORS ARE KEYED.

CR/LF TO CONTINUE
?

J-3 VOLTAGE CONNECTOR



PIN #1 = +12V DC

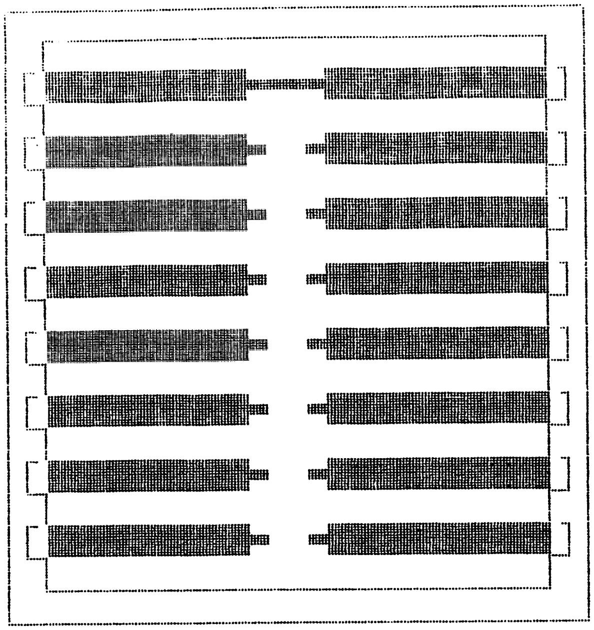
PIN #2 = +12V RETURN

PIN #3 = +5V RETURN

PIN #4 = +5V DC

CR/LF TO CONTINUE?

HEADER FOR IMI WINCHESTER



1	16	**	DRIVE SELECT #1
2	15	**	DRIVE SELECT #2
3	14	**	DRIVE SELECT #3
4	13	**	DRIVE SELECT #4
5	12	**	OPTIONAL RESET
6	11	**	SIZE SELECT 0
7	10	**	SIZE SELECT 1
8	9	**	DRIVE ALWAYS SELECTED

FN/TAB TO RETURN TO MENUE

?

2275 FUSE LIST

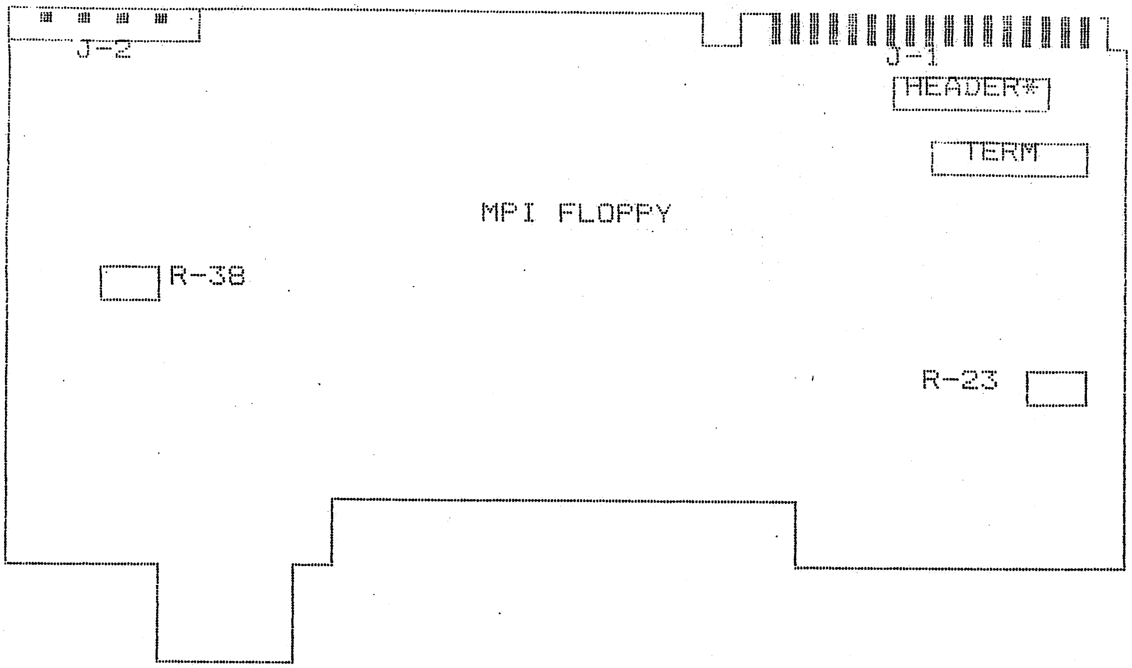
```
MAIN LINE FUSE - 115 VOLTS AC = 2.3 AMPS
 ''   ''   ''    220 VOLTS AC = 1.6 AMPS
FUSE #1 - JUMPER - NEXT TO J-2
FUSE #2 - JUMPER - NEXT TO J-2
FUSE #3 - 24 VOLTS = 4A * NO FUSE ON R-0 BOARDS.
FUSE #3 - 24 VOLTS = 4A * FUSE ON R-2 BOARDS
FUSE #4 - JUMPER - NEXT TO J-2
FUSE #5 - JUMPER - NEXT TO J-2
FUSE #6 - 8.5 VOLTS = 4A * NO FUSE ON R-0 BOARDS
FUSE #6 - 8.5 VOLTS = 4A * FUSE ON R-2 BOARDS
```

FN/TAB TO RETURN TO MENUE?

	2275	SWITCH CONFIGURATION			
Floppy	- XX	8396 BOARD			
	- 10	ON	ON	ON	ON
	- 30	ON	ON	ON	OFF
	- XX	ON	ON	OFF	ON
No Floppy	- 20	OFF	ON	ON	ON
	- XX	OFF	ON	ON	OFF
	- XX	OFF	ON	OFF	ON
	- 60	OFF	ON	OFF	OFF
	SWITCH #	SW-4	SW-3	SW-2	SW-1

SWITCH ON = SWITCH CLOSED

FN/TAB TO RETURN TO MENUE
?

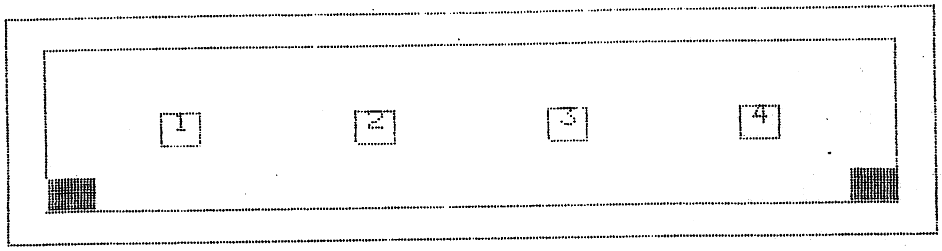


NOTE:
ALL CONNECTORS
ARE KEYED.

* HEADER
JUMPER 2-13
AND 7-8

CR/LF TO
CONTINUE
?

J-2 VOLTAGE CONNECTOR



PIN #1 = +12V DC

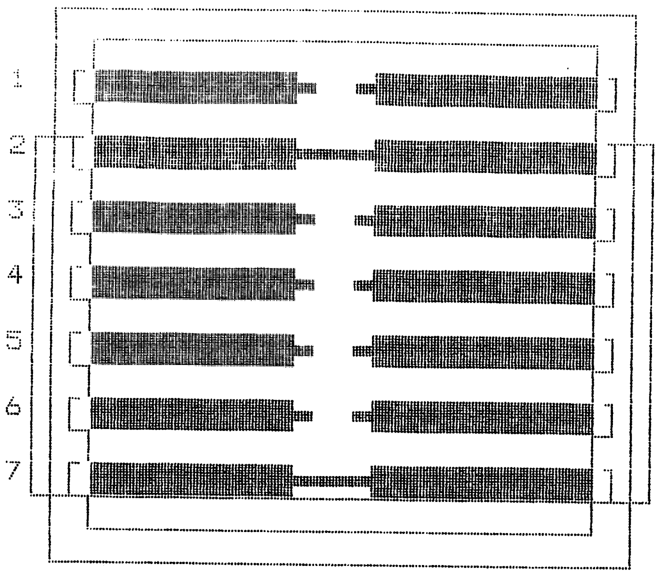
PIN #2 = +12V RETURN

PIN #3 = +5V RETURN

PIN #4 = +5V DC

CR/LF TO CONTINUE?

HEADER FOR MPI 5.25'' FLOPPY



14 ** HEAD SOLONOID WITH DRIVE SELECT

13 ** DRIVE SELECT #1

12 ** DRIVE SELECT #2

11 ** DRIVE SELECT #3

10 ** MUX

9 ** DRIVE SELECT #4

8 ** HEAD SOLONOID WITH MOTOR ON

HEADER IS A 12 PIN BLOCK
IN A 14 PIN SOCKET

FN/TAB TO RETURN TO MENUE?

5.25'' FLOPPY DRIVE P1/J1 I/O SIGNAL CONNECTOR

1	2	---- PIN 2 = NOT USED
3	4	---- PIN 4 = MAY BE USED AS DOOR LOCK OR INDICATOR
5	6	---- PIN 6 = DRIVE SELECT #4
7	8	---- PIN 8 = INDEX
9	10	---- PIN 10 = DRIVE SELECT #1
11	12	---- PIN 12 = DRIVE SELECT #2
13	14	---- PIN 14 = DRIVESELECT #3
15	16	---- PIN 16 = MOTOR ON
17	18	---- PIN 18 = DIRECTION SELECT
19	20	---- PIN 20 = STEP
21	22	---- PIN 22 = WRITE DATA
23	24	---- PIN 24 = WRITE GATE
25	26	---- PIN 26 = TRACK 00
27	28	---- PIN 28 = WRITE PROTECT
29	30	---- PIN 30 = READ DATA
31	32	---- PIN 32 = SIDE SELECT
33	34	---- PIN 34 = NOT USED

FN/TAB TO RETURN
TO MENUE?

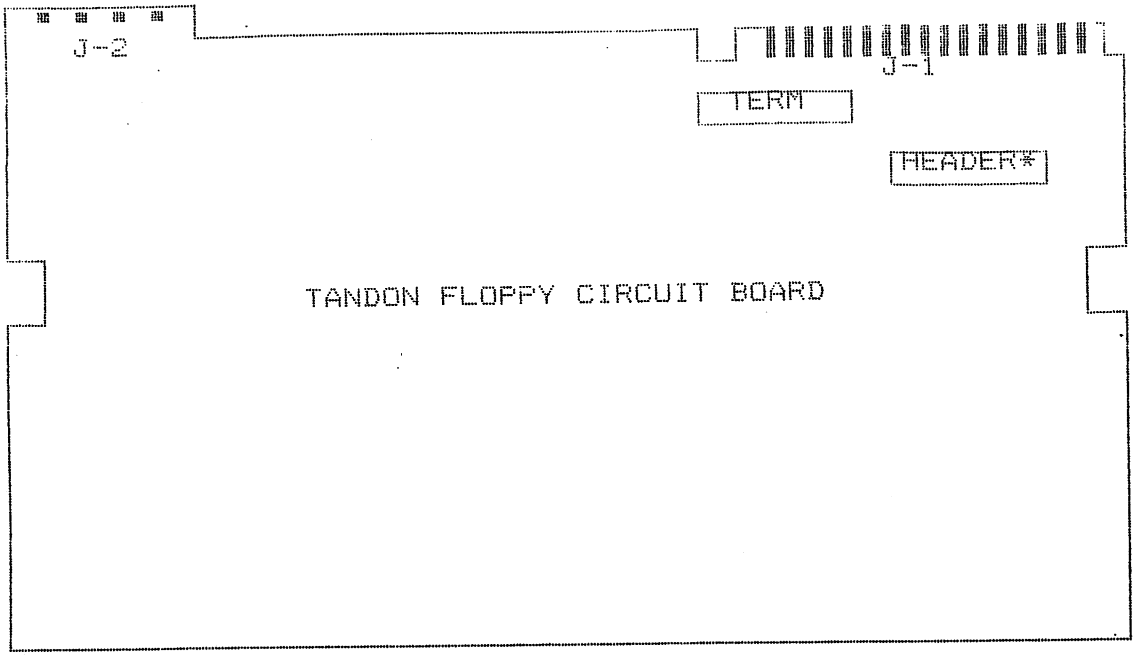
NOTE: ALL ODD NUMBERED PINS ARE 0 VOLT DC RETURN LINES

10 MEG WINCHESTER DRIVE P1/J1 I/O SIGNAL CONNECTOR

1	2	PIN 2 = RESERVED
3	4	PIN 4 = HEAD SELECT 2
5	6	PIN 6 = WRITE GATE
7	8	PIN 8 = SEEK COMPLETE
9	10	PIN 10 = TRACK 00
11	12	PIN 12 = WRITE FAULT
13	14	PIN 14 = HEAD SELECT 0
15	16	PIN 16 = RESERVED
17	18	PIN 18 = HEAD SELECT 1
19	20	PIN 20 = INDEX
21	22	PIN 22 = READY
23	24	PIN 24 = STEP
25	26	PIN 26 = DRIVE SELECT #1
27	28	PIN 28 = DRIVE SELECT #2
29	30	PIN 30 = DRIVE SELECT #3
31	32	PIN 32 = DRIVE SELECT #4
33	34	PIN 34 = DIRECTION SELECT

FN/TAB TO RETURN
TO MENUE?

NOTE: ALL ODD NUMBERED PINS ARE 0 VOLT DC RETURN LINES

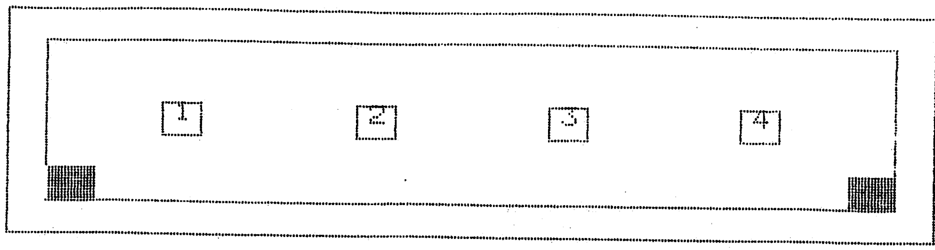


NOTE:
ALL CONNECTORS
ARE KEYED.

* HEADER
JUMPER 2-15
AND 8-9

CR/LF TO
CONTINUE
?

J-2 VOLTAGE CONNECTOR



PIN #1 = +12V DC

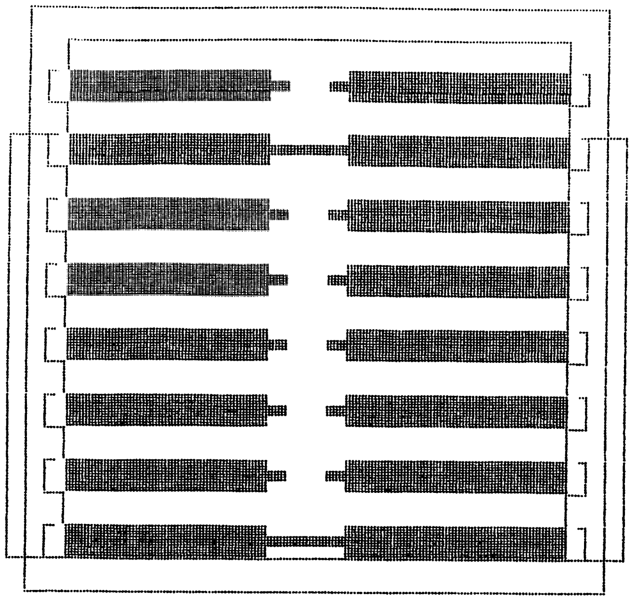
PIN #2 = +12V RETURN

PIN #3 = +5V RETURN

PIN #4 = +5V DC

CR/LF TO CONTINUE?

HEADER FOR TANDON 5.25'' FLOPPY



1 — 16 ** HEAD SOLONOID WITH DRIVE SELECT

2 — 15 ** DRIVE SELECT #1

3 — 14 ** DRIVE SELECT #2

4 — 13 ** DRIVE SELECT #3

5 — 12 ** DRIVE SELECT #4

6 — 11 ** MUX

7 — 10 ** NOT USED

8 — 9 ** HEAD SOLONOID WITH MOTOR ON

HEADER PLUG IS 14 PINS
IN A 16 PIN SOCKET

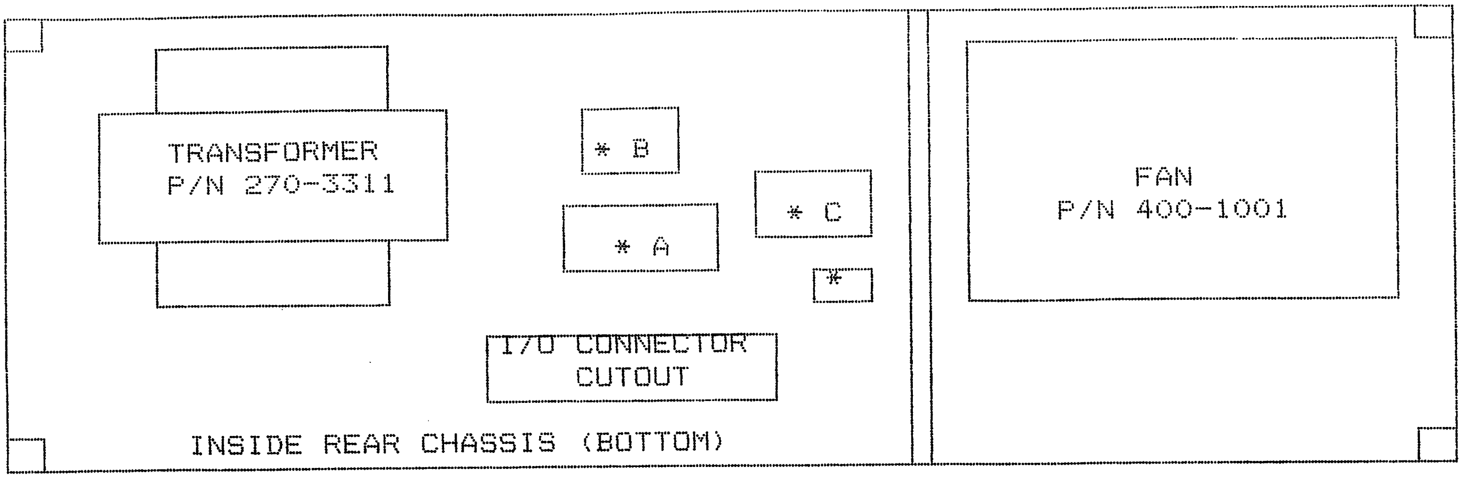
FN/TAB TO RETURN TO MENUE?

2275 PARTS LISTING

P/N 270-0870	CHASSIS AND POWER SUPPLY ASSEMBLY
P/N 278-4030	WINCHESTER 10 MEG DRIVE
P/N 278-4034	30 MEG QUANTUM DRIVE
P/N 278-4026	5 1/4 FLOPPY DRIVE DSDD - (HAS DOOR SWITCH)
P/N 330-0842	TERMINATOR - FLOPPY DRIVE (150 OHM)
P/N 210-8396	DISK CONTROLLER BOARD
P/N 210-8397	REGULATOR BOARD
P/N 220-2013	CABLE - POWER SUPPLY REG TO DRIVES AND 8396 BOARD
P/N 220-3313	RIBBON CABLE - SMALL 8396 BOARD TO WINCHESTER
P/N 220-3239	RIBBON CABLE - LARGE 8396 BOARD TO WINCHESTER OR FLOPPY
P/N 220-3324	RIBBON CABLE - LARGE DAISY CHAIN 2 WINCHESTERS
P/N 379-2000	R-2 PROM ON 8396 BOARD. (FCO KIT # 728-0129)

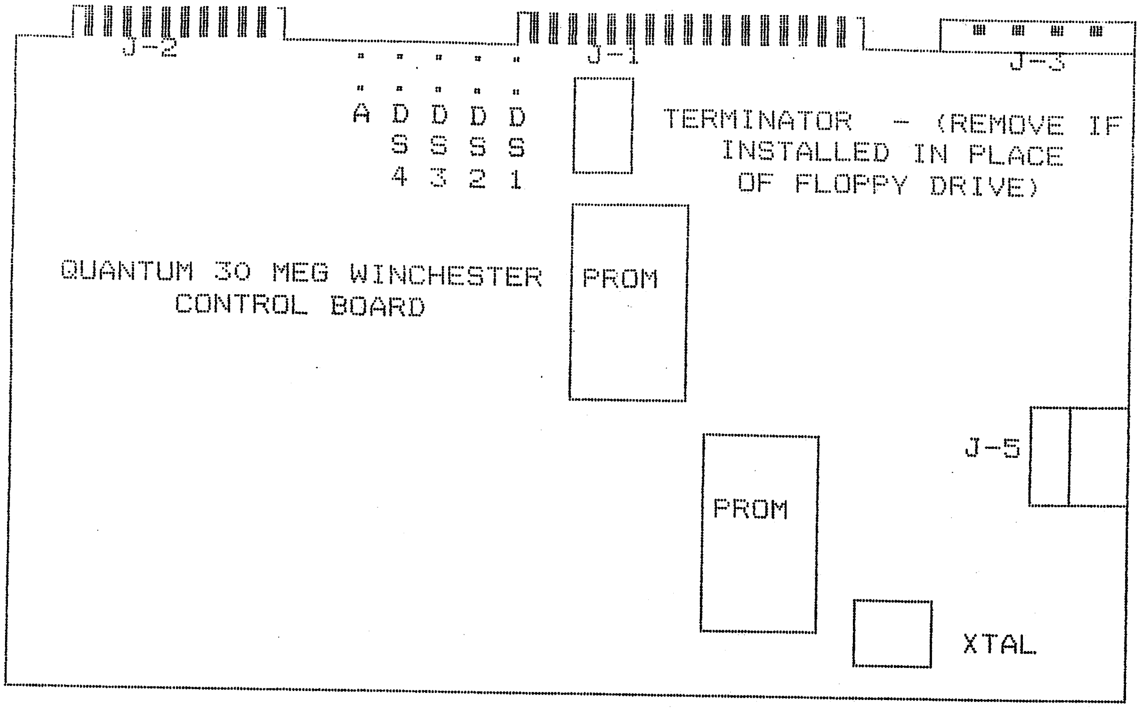
CR/LF TO CONTINUE?

2275 POWER SUPPLY AND CHASSIS ASSEMBLY P/N 270-0870



* = FUSE HOLDER P/N 360-0018   *A = LINE FILTER P/N 410-2024
*B = LINE SWITCH P/N 325-0059   *C = RECTIFIER P/N 380-4005
NOTE: - IF ANY COMPONENT IS DEFECTIVE ORDER COMPLETE CHASSIS ASSEMBLY

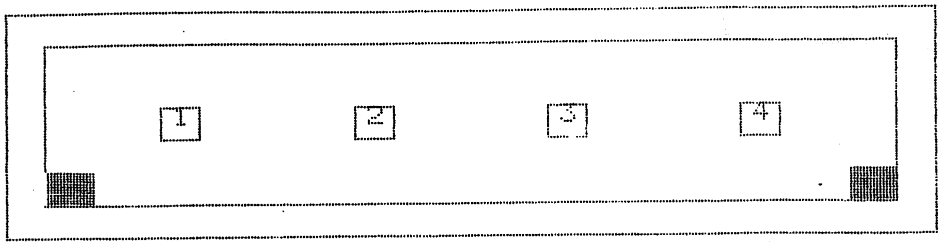
FN/TAB TO RETURN TO MENUE - CR/LF TO RESTART?



NOTE:
ALL CONNECTORS
ARE KEYED.

CR/LF TO
CONTINUE
?

J-3 VOLTAGE CONNECTOR



PIN #1 = +12V DC

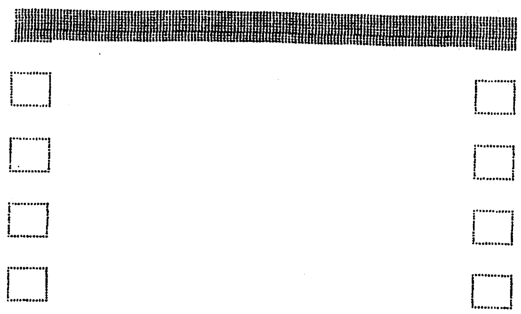
PIN #2 = +12V RETURN

PIN #3 = +5V RETURN

PIN #4 = +5V DC

CR/LF TO CONTINUE?

DRIVE SELECT JUMPERS



** DRIVE SELECT #1

** DRIVE SELECT #2

** DRIVE SELECT #3

** DRIVE SELECT #4

** ALWAYS SELECTED

FN/TAB TO RETURN TO MENUE

GENERAL 2275 INFORMATION

ADDRESS AND SECTOR INFORMATION

1) 5.25 INCH FLOPPY HAS 1279 SECTORS

2) 10 MEG WINCHESTER HAS 38911 SECTORS

3) 30 MEG QUANTUMS HAVE 64023 SECTORS PER PLATTER (2 PLATTERS)

4) IF 2275 ADDRESS IS 310

   A) 2275-10 - FLOPPY IS D10 AND 10 MEG IS D11

   B) 2275-20 - #1 10 MEG IS D10 AND #2 10 MEG IS D11

   C) 2275-30 - FLOPPY IS D10 AND 30 MEG IS D11 AND D12

   D) 2275-60 - 2 30 MEG DRIVES. ADDRESSES D10,D11,D12,D13.

CR/LF TO CONTINUE?

GENERAL 2275 INFORMATION

ADDRESS AND SECTOR INFORMATION

5) IF 2275 ADDRESS IS 320

   A) 2275-10 - FLOPPY IS D20 AND 10 MEG IS D21

   B) 2275-20 - #1 10 MEG IS D20 AND #2 10 MEG IS D21

   C) 2275-30 - FLOPPY IS D20 AND 30 MEG IS D21 AND D22

   D) 2275-60 - 2 30 MEG DRIVES. ADDRESSES D20,D21,D22,D23.

CR/LF TO CONTINUE?

GENERAL 2275 INFORMATION

ADDRESS AND SECTOR INFORMATION

6) IF 2275 ADDRESS IS 330

   A) 2275-10 - FLOPPY IS D30 AND 10 MEG IS D31

   B) 2275-20 - #1 10 MEG IS D30 AND #2 10 MEG IS D31

   C) 2275-30 - FLOPPY IS D30 AND 30 MEG IS D31 AND D32

   D) 2275-60 - 2 30 MEG DRIVES. ADDRESSES D30,D31,D32,D33.

NOTE: 2275-60 ADDRESSES. THE FIRST DRIVE IS DX0 AND DX3.
THE SECOND DRIVE IS ADDRESSED DX1 AND DX2

CR/LF TO CONTINUE?

WANG

3110

Don Pauling
FYI
Wayne J
9/14/83

5-1/4 INCH DISK PERIPHERAL

Model: 2275

ADVANCE COPY

Customer Engineering
Product Maintenance Manual

CHAPTER 3
OPERATION

3.0 INTRODUCTION

The 2275 5-1/4" Disk Peripheral is a small, stand-alone, add-on storage device for the 2200 MVP System, and is intended to substantially reduce the cost of small 2200 systems. This unit is available in two versions, 2275-10 and 2275-20. The 2275-10 is equipped with a 5-1/4" Winchester Disk Drive and a 5-1/4" Diskette Drive, while the 2275-20 is equipped with two 5-1/4" Winchesters.

The unit is housed in a 14.8" x 6.5" x 16" enclosure, and consists of a chassis frame, the Disk Drives, a Disk Controller Board, a Linear Power Supply Regulator Board, power supply components and a cooling fan. The power supply components and cooling fan are mounted on the rear panel of the chassis frame. The frame, fan, and power supply components are considered to be a single assembly for replacement purposes.

3.1 CONTROLS AND INDICATORS

Only one control and two indicators are used on the Disk Peripheral. The single control is a Power ON/OFF switch located on the rear panel. The indicators are LED lamps located on the front panels of the Disk Drives. When either of these lamps is lit, it indicates that the corresponding drive is in operation. That is, data is being written to, or read from the disk.

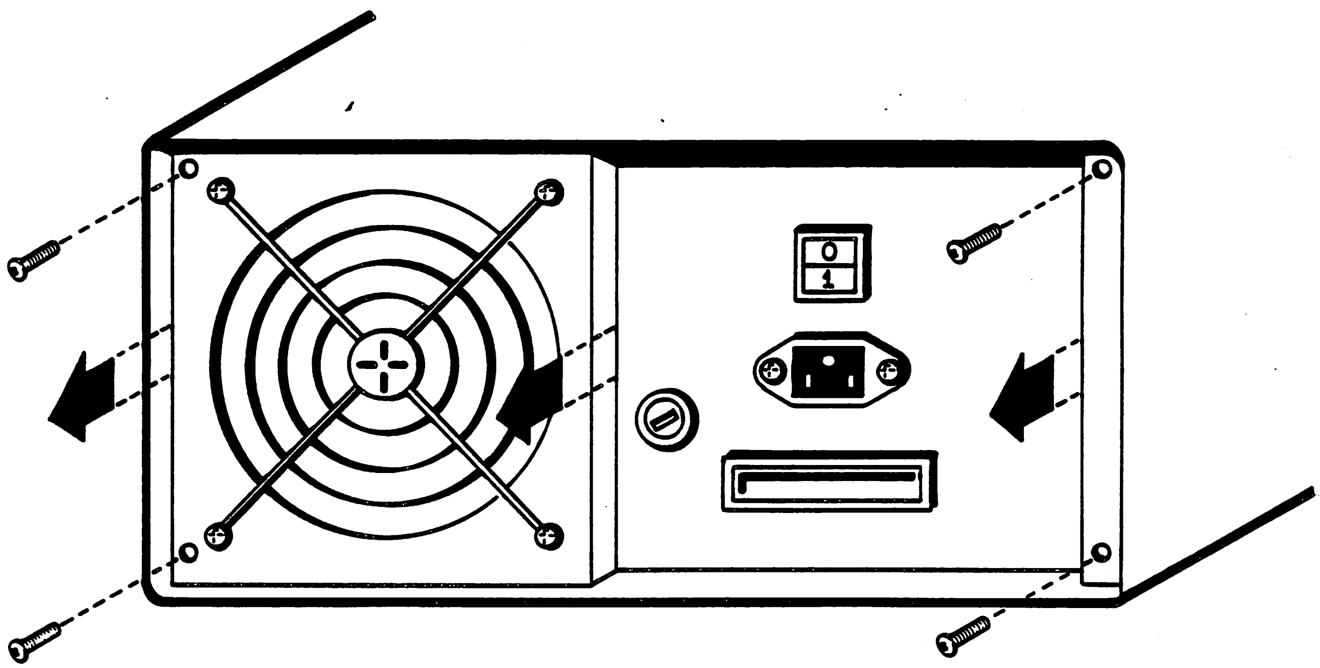


Figure 3-1 Rear Panel

3.2 REQUIRED ACCESSORIES

The following cable is required to connect the 2275 Disk Peripheral to a 2200 MVP system:

PN 220-0105-4

3.3 INITIAL POWER-UP PROCEDURE

The initial power-up procedure for the 2275 Disk Peripheral consists of putting the Power Switch in the ON position. The switch location is shown in Figure 3-1. The unit will be ready for use when the 2200 MVP is ready.

3.4 VERIFY NORMAL OPERATION

To verify normal operation, Format the Winchester Disk according to the procedures outlined in the 2200 BASIC 2 Language Reference Manual (700-40480D); and verify that the disk has, in fact, been formatted. Similarly, write data to the Diskette; and recover the data from the diskette, all according to the procedures outlined in the 2200 BASIC 2 Language Reference Manual.

3.5 NORMAL SHUT-DOWN PROCEDURE

The normal shut-down procedure for the 2275 is to place the Power Switch in the OFF position. Do not shut-down the 2275 while either drive is operating (front panel LED indicator lamp "ON")!

3.6 OPERATOR PREVENTIVE MAINTENANCE

No Operator Preventive Maintenance is required for the 2275 Disk Peripheral.

3.7 EMERGENCY SHUT-DOWN PROCEDURE

The emergency shut-down procedure for the 2275 Disk Peripheral is the same as the normal shut-down procedure.

CAUTION!!

The Winchester Disk, the floppy Diskette, and the recording heads may be severly damaged if the power is removed while a Read or Write operation is in progress.

CHAPTER 4
INSTALLATION

4.0 INSTALLATION SITE CHECK

Determine the voltage level of the available AC Power, and verify that the 2275 Disk Peripheral is set-up to operate at that voltage level.

The 2275 Disk Peripheral will operate from a nominal 115 Vac or 230 Vac power line, as selected by a switch on the 8397 Power Supply Regulator Board. The switch is located on the component side of the Regulator Board, and is accessible only when the unit is out of its enclosure. See Figure 4-1. The actual line voltage level must be within the range of 98 Vac to 128 Vac, or 196 Vac to 256 Vac. The required current capacity for the 115 Vac line is 2 amperes.

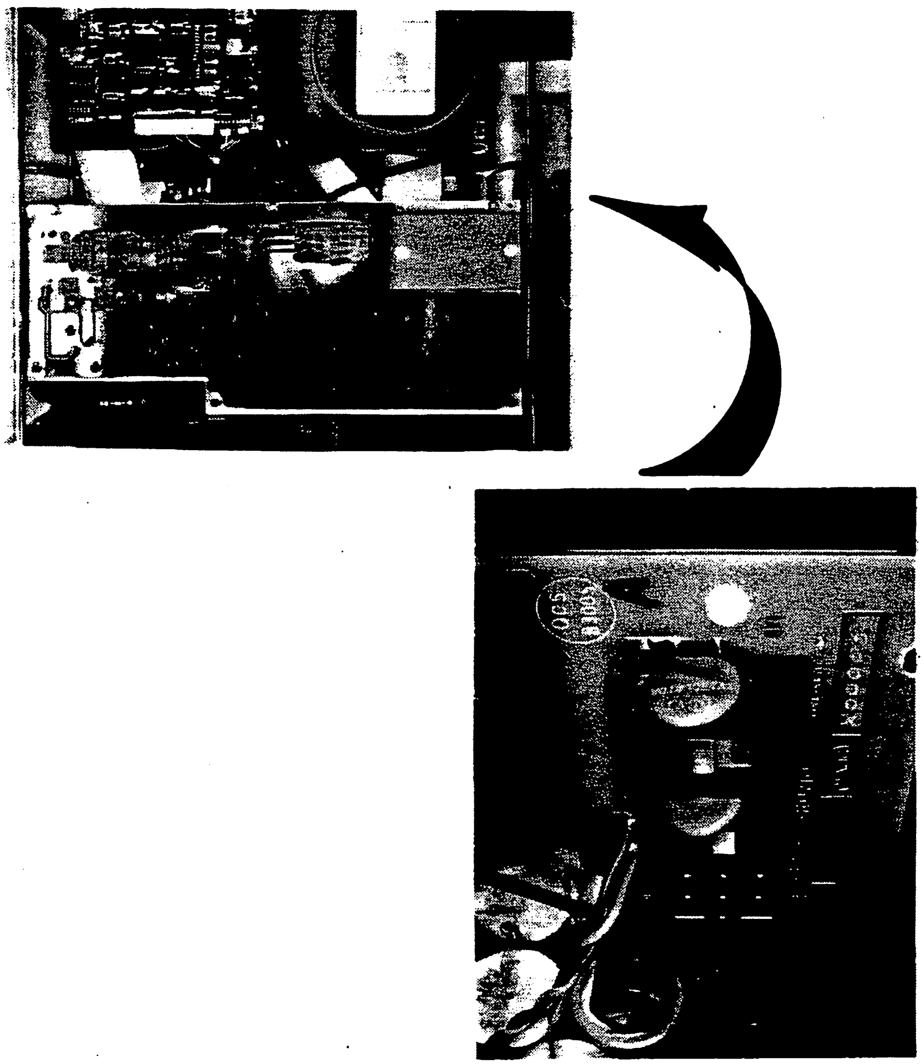


Figure 4-1 Power Select Switch

4.1 SPECIAL TOOLS AND TEST EQUIPMENT

No special tools or test equipment are required for the installation of a 2275 Disk Peripheral.

4.2 UNPACKING

Before opening the shipping container, inspect the package for external indications of damage. Carefully unpack the Disk Peripheral and again check for signs of physical damage. Remove the chassis assembly from the external case (see Figure 4-2), remove any internal packing material, and check the interior for obvious damage.

If any damage is found, file a claim promptly with the delivery carrier. Notify, as well, the WLI Distribution Center (Dept. 90), Quality Assurance Department, Tewksbury, MA 01876. Detail the extent of the damage and arrange for equipment replacement, if necessary.

NOTE

Do not destroy the packing materials until it has been determined that the equipment is in acceptable condition.

Perform the following steps to remove the Disk Peripheral from its enclosure:

1. Remove four screws.
2. Slide rear panel and chassis to the rear, and lift clear.

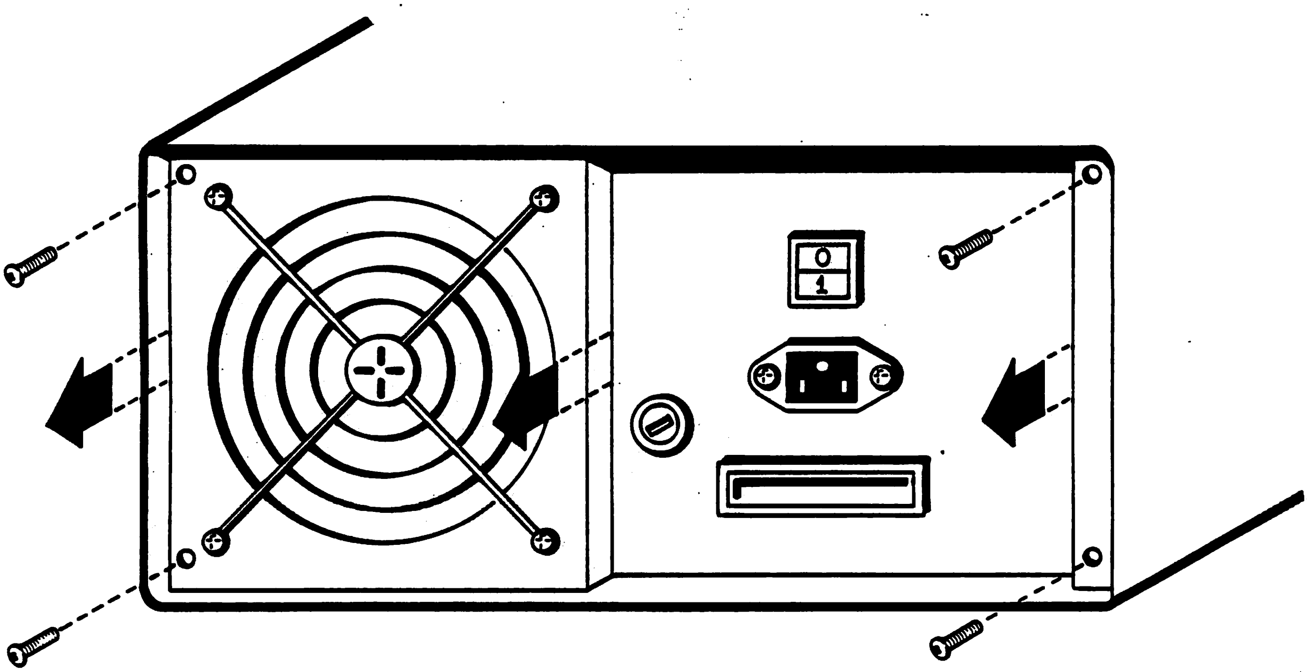


Figure 4-2 Enclosure Removal

Inspect and prepare the Disk Peripheral for operation as follows:

1. Remove any internal packing material.

2. Unfasten the Disk Controller Board, which is held in place by four machine screws and two guide posts that mate with chassis-mounted slotted support brackets (see Figure 4-3).

3. Remove all cable plugs from the Controller Board receptacles, and lift the board clear of the chassis.

4. Locate SW-1 on the 8396 Controller Board, and verify that the switch is set as in Figure 4-4.

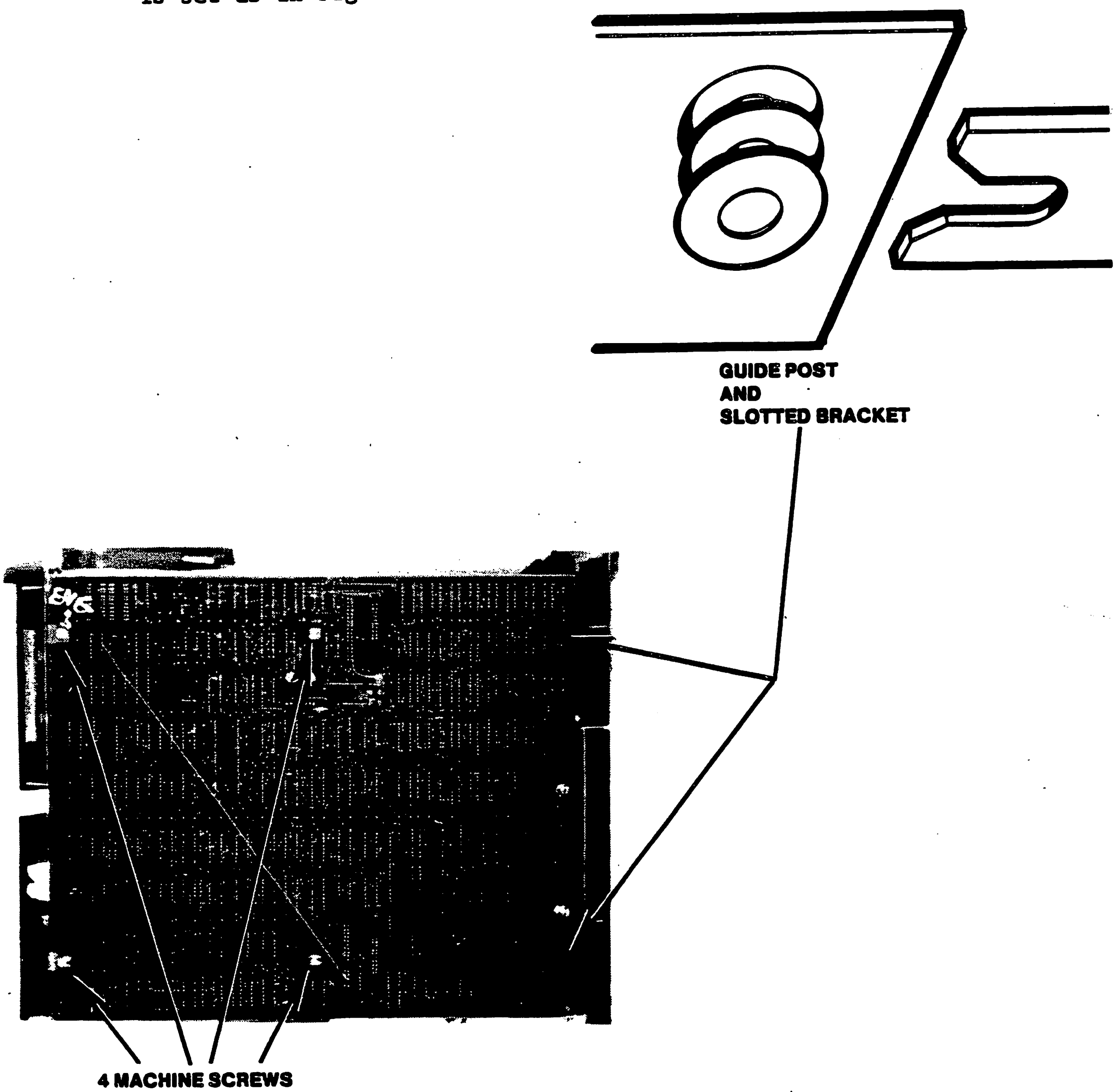


Figure 4-3 Controller Board Removal

4.2 UNPACKING (cont.)

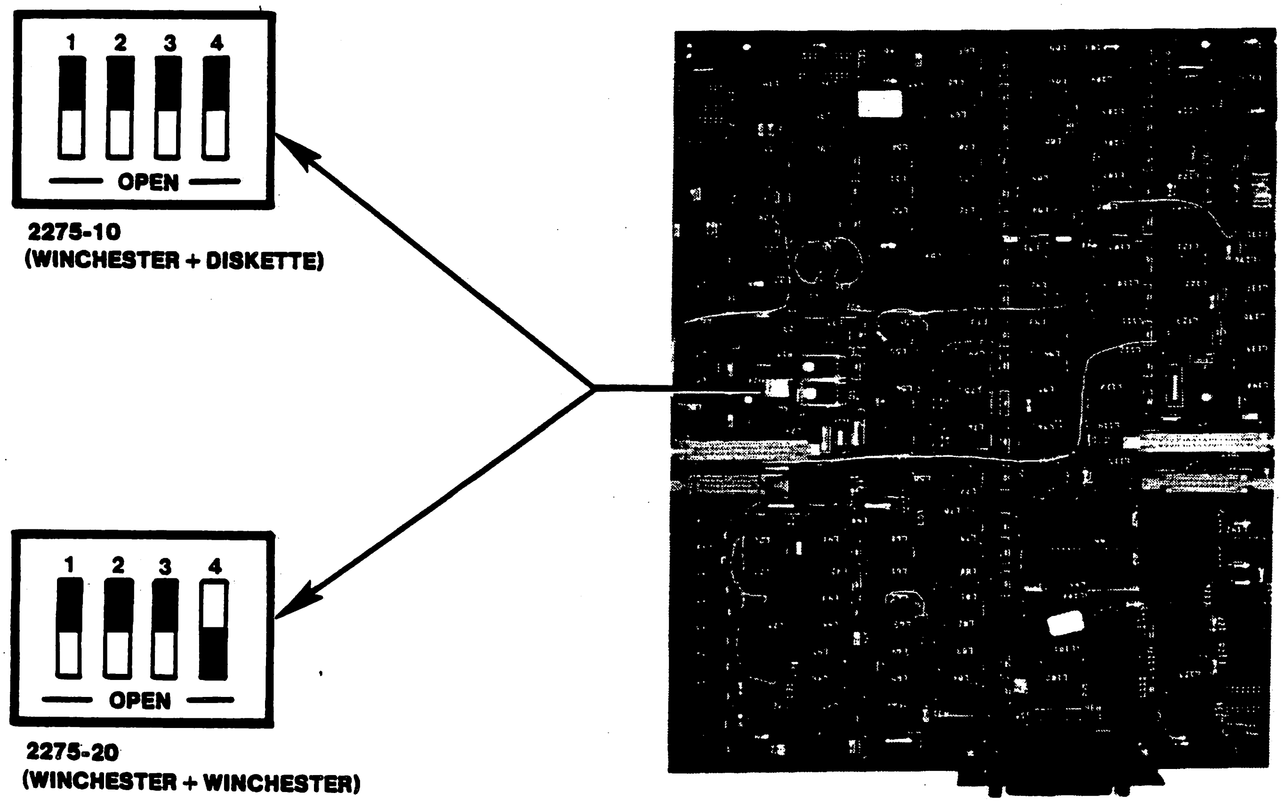


Figure 4-4 Switch Settings

4.2 UNPACKING (cont.)

5. Replace the Controller Board in the chassis. Re-insert and lock all cable plugs into their respective receptacles, and fasten the Controller Board in place with the four machine screws removed in step 2.

4.3 INSTALL THE EQUIPMENT

The 2275 Disk Peripheral must be connected to the 2200 MVP system with the special cable supplied for that purpose (PN 220-0105-4). The cable connects between the 2200 MVP disk controller (Triple Controller Board or single connector Disk Interface Board) and a 36 pin "D" receptacle accessible at the rear panel of the Disc Peripheral. The "D" receptacle is actually attached to the edge of the 8396 Controller Board, and protrudes through a slot cut in the rear panel of the unit.

The 2275 Disk Peripheral must also be connected to the selected source of AC Power by means of the power cable supplied with the unit.

4.4 HARDWARE COMPATIBILITY

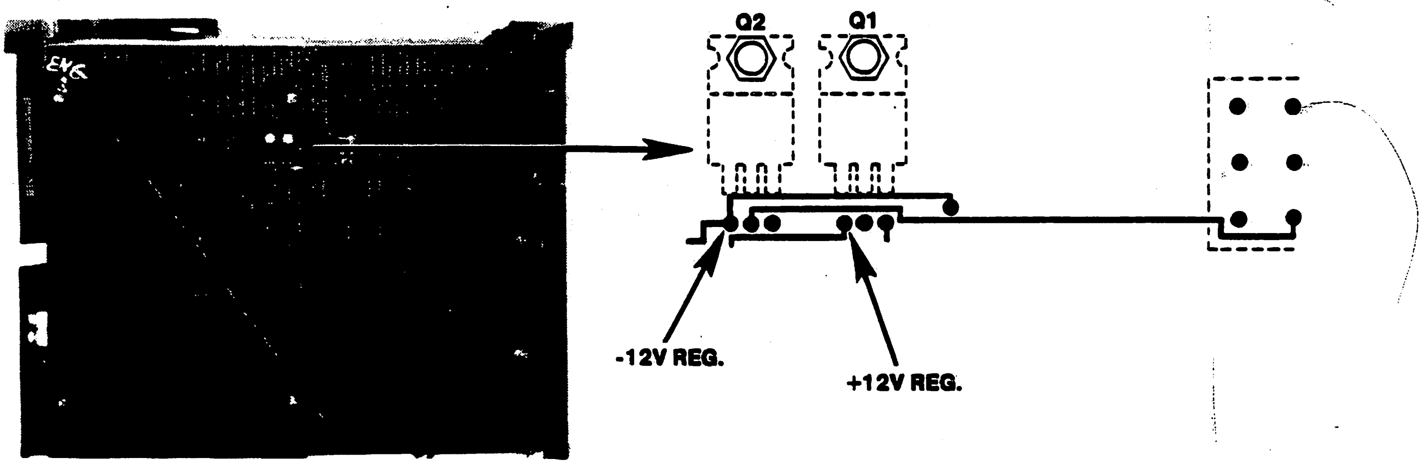
The following describes the minimum hardware and software compatibility requirements for the use of the 2275 Disk Peripheral with a 2200 MVP System.

1. 2200 MVP Operating System Release 2.4, or later.
2. 2200 MVP Triple Controller Board (3012);
   <u>or</u>
   Single Connector Disk Interface Board (6541), minimum Rev R3.

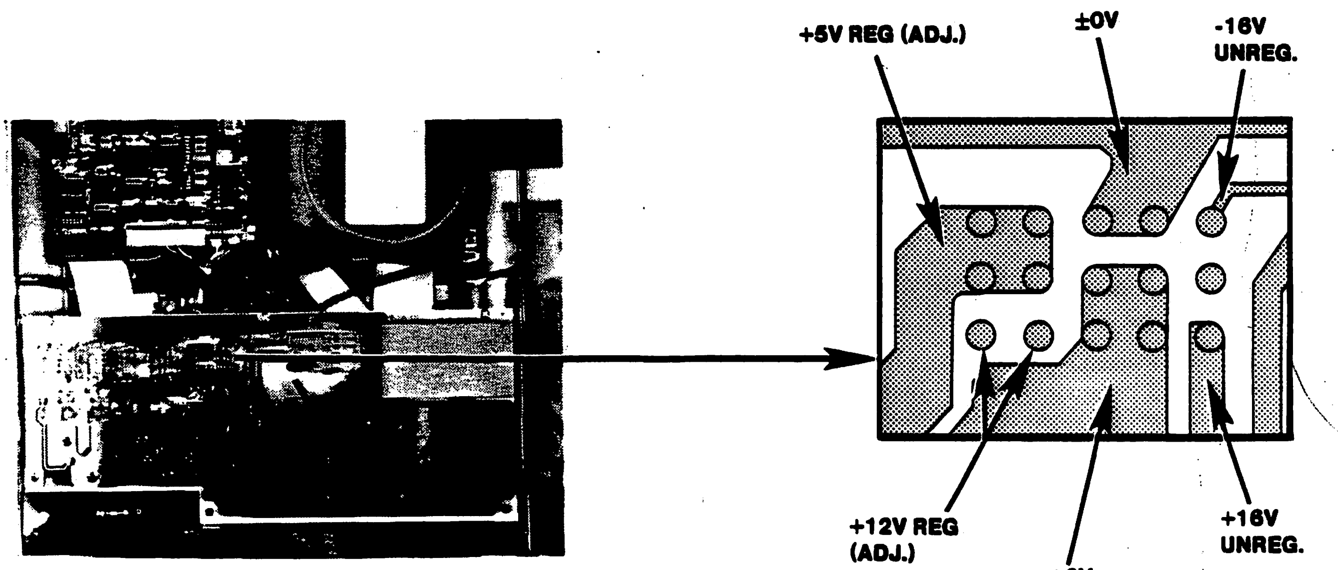
4.5 INITIAL POWER-UP PROCEDURE

Before connecting the power cord, verify that the power line level-select switch is in the correct position. See Figure 4-1.

Place the power switch in the ON position, and proceed to the Voltage Checks section.



8396 CONTROLLER BOARD



8397 REGULATOR BOARD

Figure 4-5 Voltage Test Points

4.6 VOLTAGE CHECKS

There are four regulated, and two unregulated DC voltage sources to be checked in this unit. Four of the test points are located on the 8397 Regulator Board, and two test points are on the 8396 Controller Board. The regulated voltages that appear on the Regulator Board are adjustable, but those on the Controller Board are fixed. The test points are shown in Figure 4-5.

With the AC Power switch ON, measure the DC voltage levels at the test points shown in Figure 4-5. Adjust the +5 VDC and +12 VDC levels, if required. The +5 VDC and +12 VDC adjusting trimpots are shown in Figure 4-6.

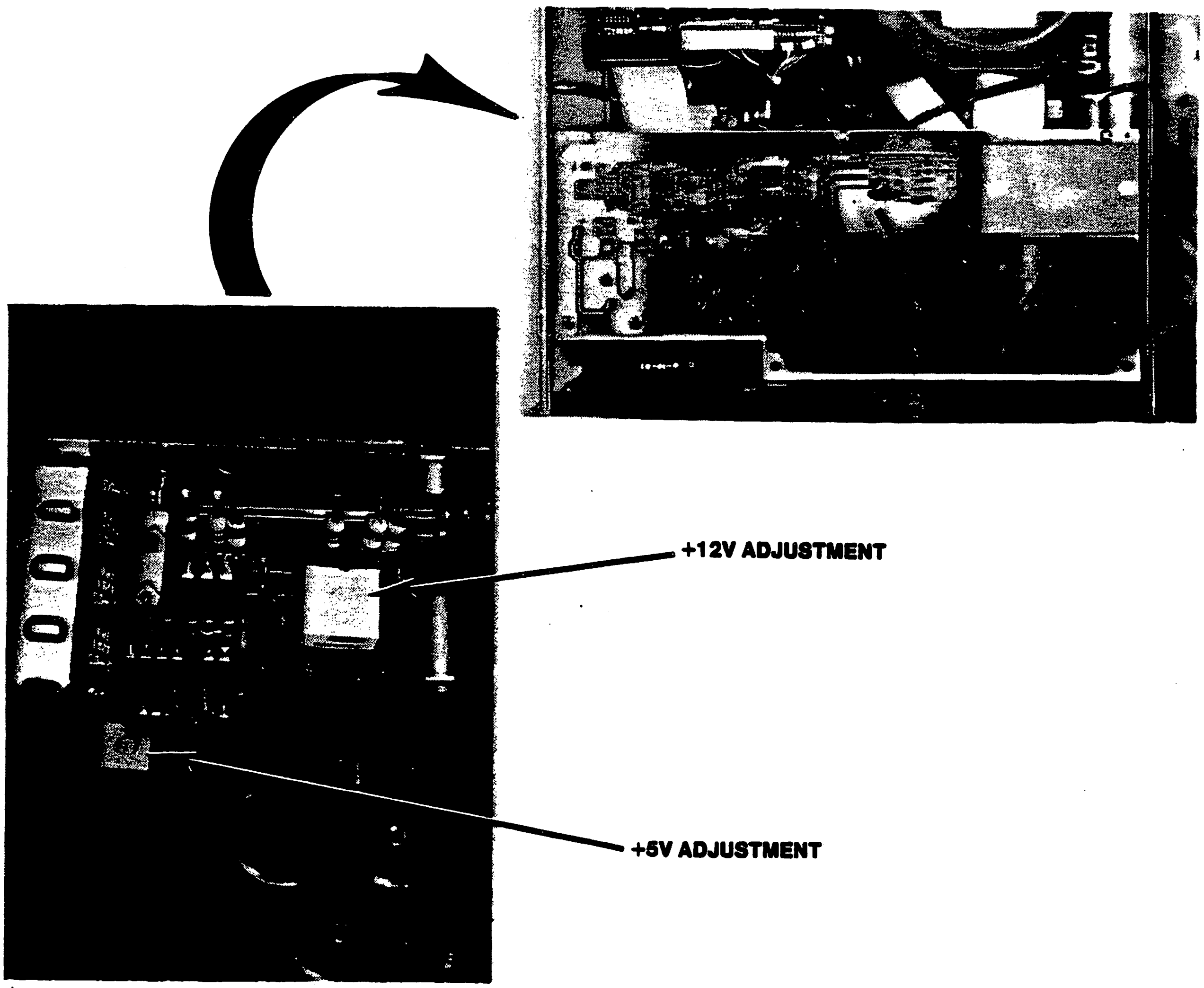


Figure 4-6 Regulated Voltage Adjustment Controls

CHAPTER 5
PREVENTIVE AND CORRECTIVE MAINTENENCE

5.0 PREVENTIVE MAINTENANCE

There are no preventive maintenance procedures for the 2275 Disk Peripheral.

5.1 CORRECTIVE MAINTENANCE

5.1.1 Special Tools and Test Equipment

There are no special tools or test equipmemt required to perform corrective maintenance on the 2275 Disk Peripheral.

5.1.2 Removal and Replacement Procedures

Power Supply Components
All of the power supply components for the 2275 Disk Peripheral are mounted on the rear panel of the chassis, along with the cooling fan. If any of these chassis-mounted components fail in service, the entire Chassis Assembly must be replaced. There are no individually replaceable components within the Chassis Assembly.

Disk Drives
The Disk Drives are mounted in slots built into the chassis. Each drive held in place by a single machine screw located below its front panel. The screws are accessible only when the unit is removed from its housing. Movement of the drives is further constrained by ribbon cables that connect to the Controller Board, and by stranded power cables that connect to the Power Supply Regulator Board. Remove and replace the Disk Drives by performing the following steps:

1. Remove the Disk Peripheral from its enclosure.

2. Remove the ribbon cable(s) from the rear of the Disk Drive to be replaced.

3. Remove the power cable from the rear of the Disk Drive to be removed.

4. Remove the single machine screw centered below the front panel of the Disk Drive to be removed.

5. Slide the Disk Drive out of the chassis.

6. Reverse steps 2 through 5 to replace the Disk Drive.

5.1.2 Removal and Replacement Procedures (Continued)

8397 Power Supply Regulator Board

The Power Supply Regulator Board is held in place by three machine screws with kep nuts located along the front edge, a single machine screw located at the right-rear corner, and two guide posts located along the rear edge. The guide posts mate with slotted brackets fastened to the chassis rear panel. Refer to Figure 5-1.

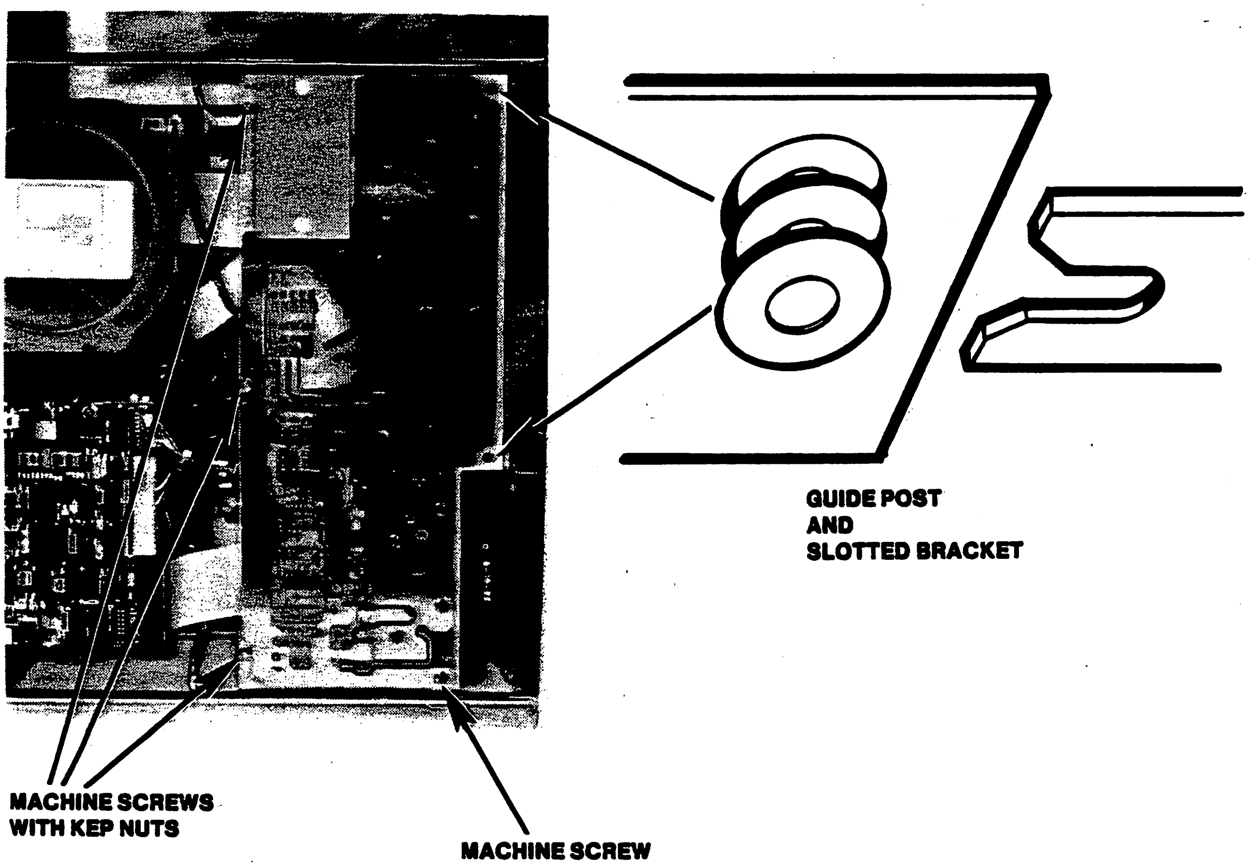


Figure 5-1 Power Supply Regulator Board Fasteners

5.1.2 Removal and Replacement Procedures

8397 Power Supply Regulator Board (Continued)

Movement of the Power Supply Regulator Board is further constrained by power distribution cabling connected to the component side of the board. Remove and replace the Power Supply Regulator Board by performing the following steps:

1. Remove the Disk Peripheral from its enclosure.
2. Remove three machine screws and kep nuts from the front edge of the board.
3. Remove one machine screw from the right rear corner of the board.
4. Slide the board towards the front of the unit, disengaging the guide posts from the slotted brackets at the rear of the unit.
5. Carefully swing the board away from the chassis, to the limit allowed by the internal power distribution cables.
6. Unplug three power cable connectors (J1, J2, and J3) from the board. Refer to Figure 5-2.

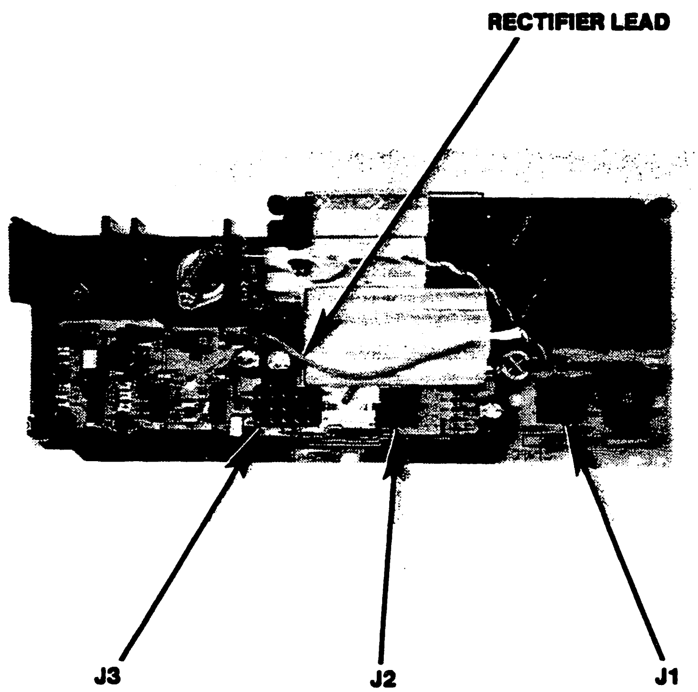


Figure 5-2 Power Supply Regulator Board Connection Points

5.1.2 Removal and Replacement Procedures

8397 Power Supply Regulator Board (Continued)

7. Remove the single wire extending from the board to a quick-connect terminal on the rectifier (The rectifier is mounted on the rear panel of the unit).

8. Lift the board clear of the chassis.

9. Reverse steps 2 through 8 to replace the Power Supply Regulator Board.

10. Check the voltage levels, and adjust if necessary. Refer to Figure 5-3.

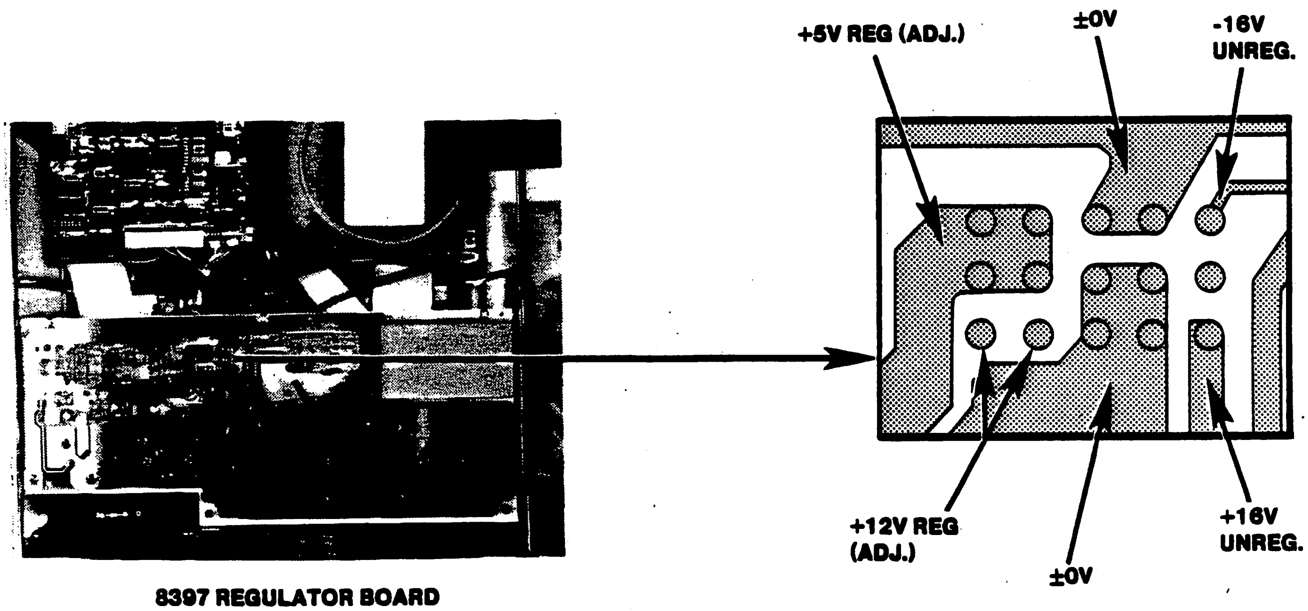


Figure 5-3 Power Supply Regulator Board Voltage Test Points

5.1.2 Removal and Replacement Procedures (Continued)

8396 Disk Controller Board
The Disk Controller Board is held in place by four machine screws, and two guide posts. The guide posts mate with slotted brackets attached to the chassis rear panel. Movement of the Disk Controller Board is further constrained by plug-in ribbon cables connected to the Disk Drives, and a stranded power cable connected to the Power Supply Regulator Board. Remove and replace the Disk Controller Board by performing the following steps:

1. Remove the Disk Peripheral from its enclosure.

2. Remove four machine screws. Refer to Figure 5-4.

3. Slide the board toward the front of the unit, disengaging the guide posts from the slotted brackets at the rear of the unit.

4. Swing the board away from the chassis to the limit allowed by the attached cables.

5. Remove the cable plugs from the component side of the board, and lift the board clear of the chassis

6. Reverse steps 2 through 5 to replace the Controller Board.

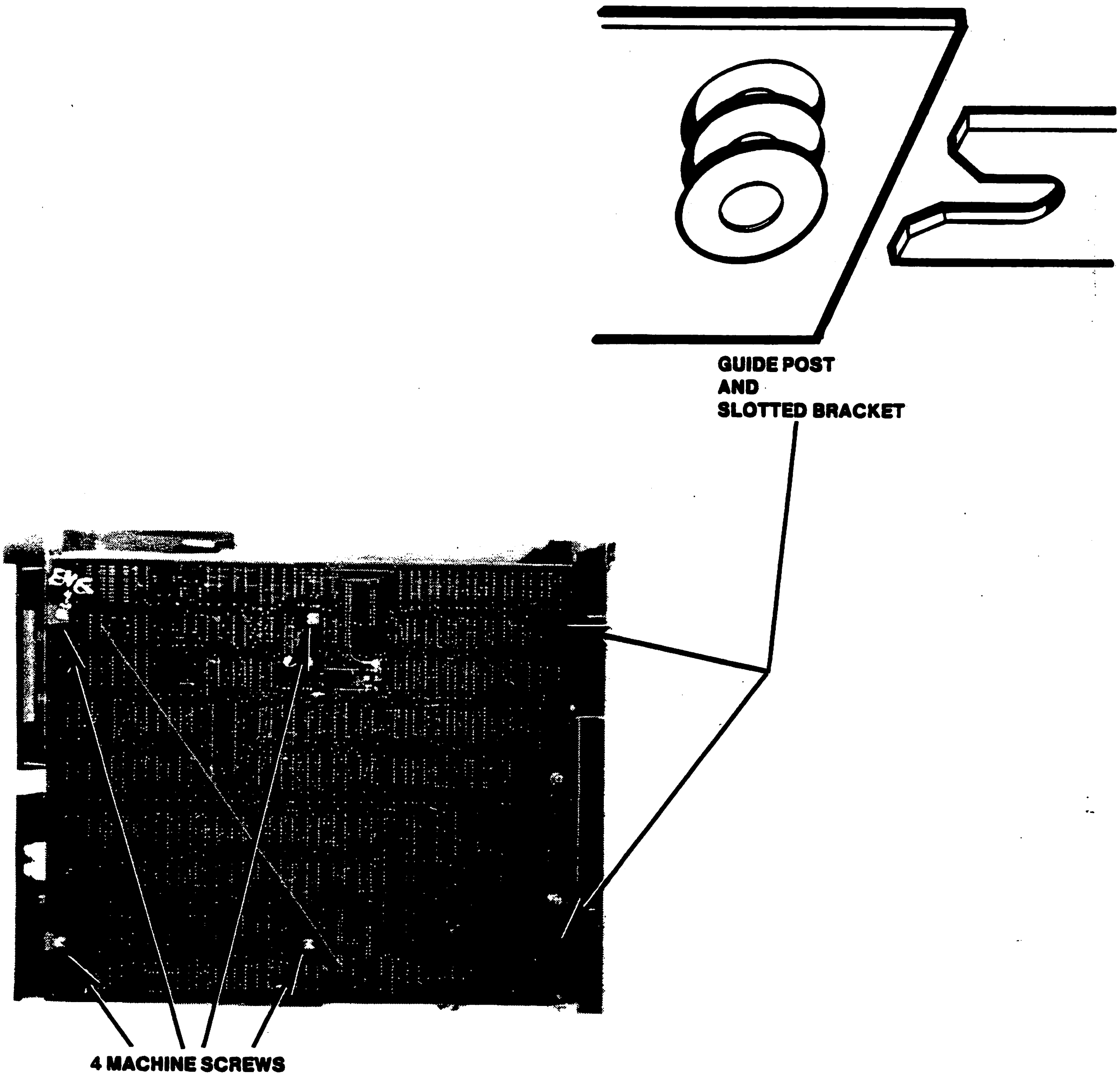


Figure 5-4 Disk Controller Board Fasteners

CHAPTER 7
ILLUSTRATED PARTS BREAKDOWN

The following items are the replaceable parts for the 2275 Disk Peripheral. Note that the 2275 Chassis Assembly includes the cooling fan and all power supply components mounted on the rear panel.

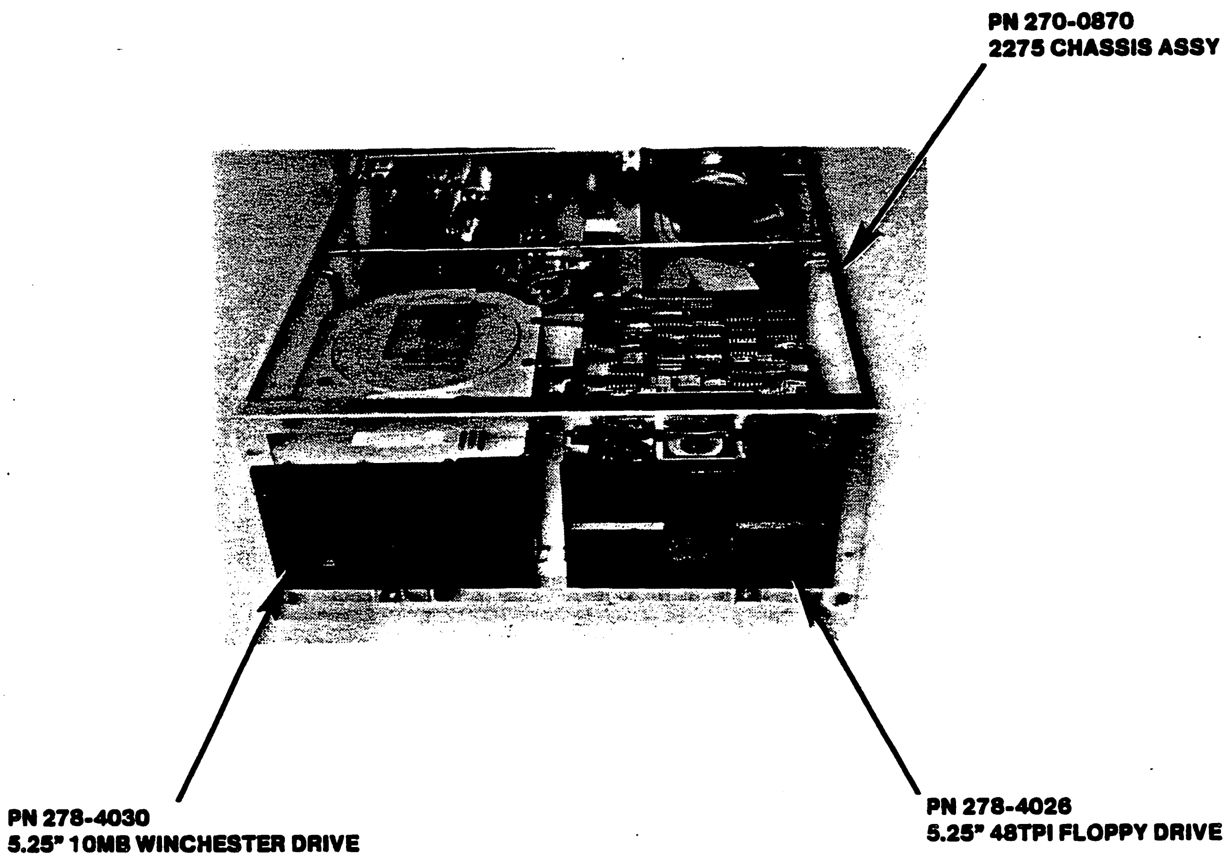
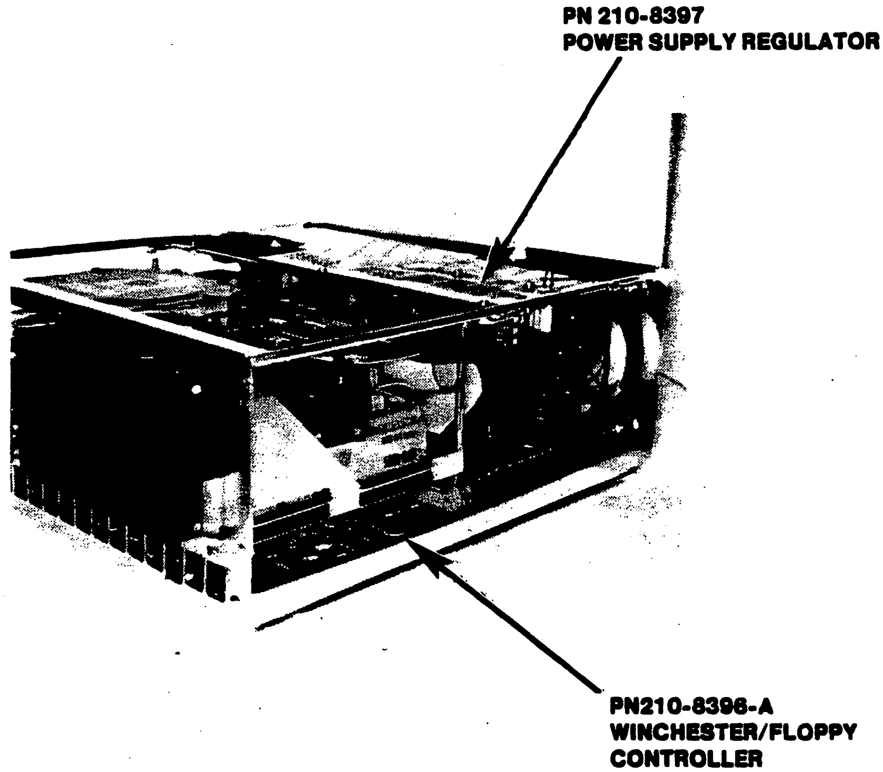


Figure 7-1 Chassis Assembly and Disk Drives

Replaceable Parts (Continued)



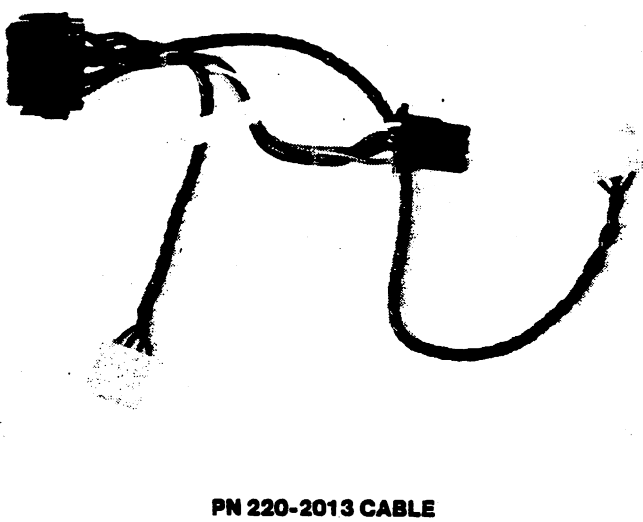
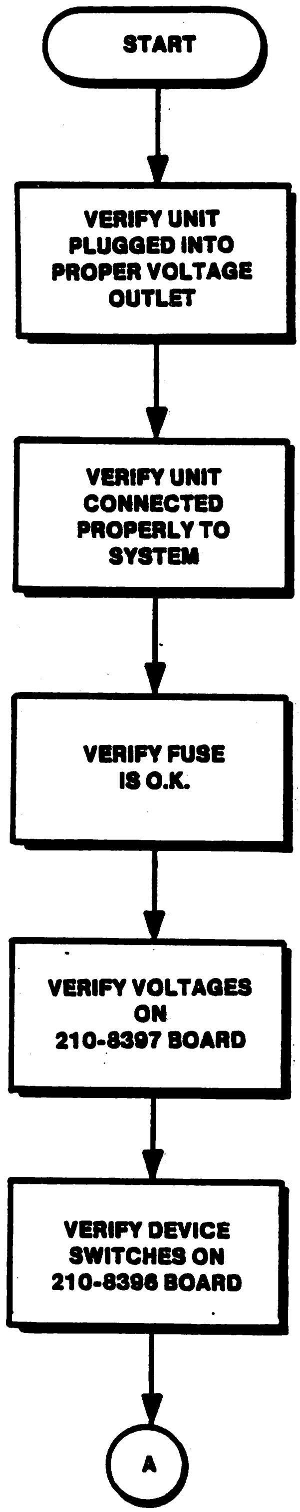
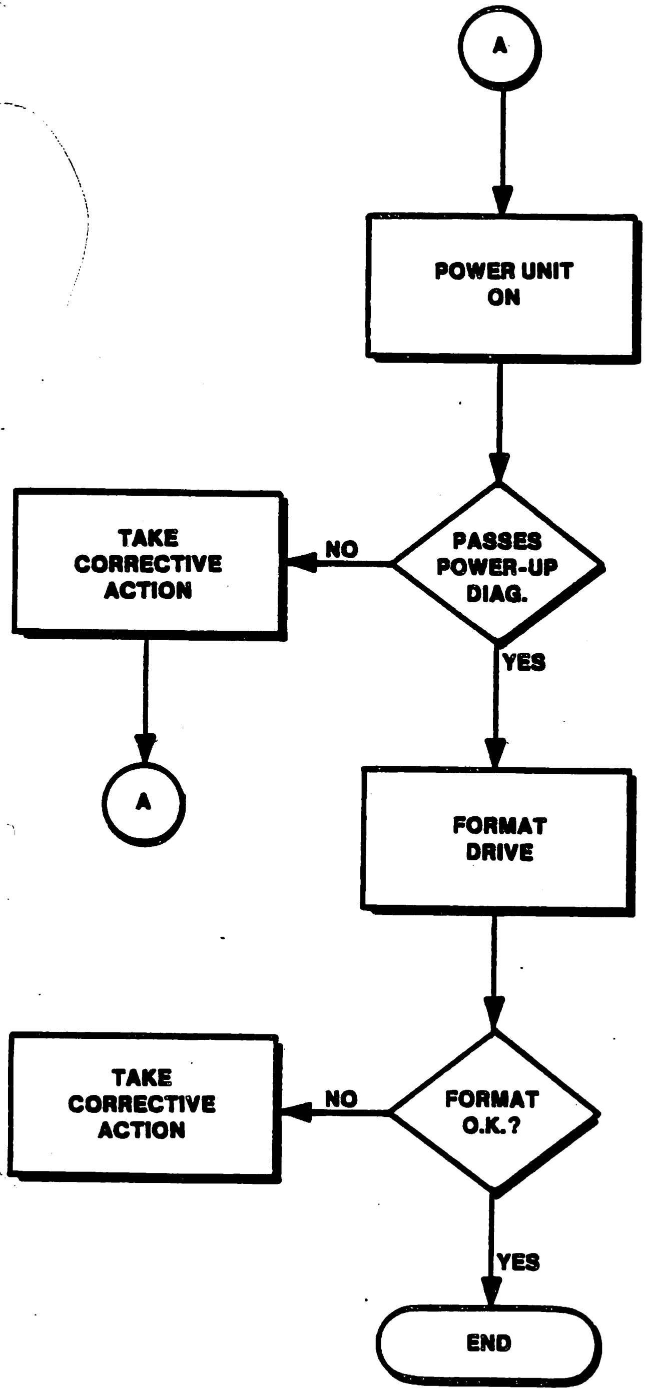


Figure 7-2 Circuit Boards and Cable

CHAPTER 8
TROUBLESHOOTING

The following pages contain a troubleshooting Flow Chart that will assist the Customer Engineer in making a systematic analysis of 2275 Disk Peripheral malfunctions.


START
VERIFY UNIT PLUGGED INTO PROPER VOLTAGE OUTLET
VERIFY UNIT CONNECTED PROPERLY TO SYSTEM
VERIFY FUSE IS O.K.
VERIFY VOLTAGES ON 210-8397 BOARD
VERIFY DEVICE SWITCHES ON 210-8396 BOARD
A


A
POWER UNIT ON
PASSES POWER-UP DIAG.
NO
TAKE CORRECTIVE ACTION
A
YES
FORMAT DRIVE
FORMAT O.K.?
NO
TAKE CORRECTIVE ACTION
YES
END

To: Distribution

From: Jerry Sevigny

Date: 31 March 1984

Subject: 2275 Meeting Minutes

This memo documents the minutes of the BETA release meeting held Friday the 30th of March at 3PM in Conference room II 8th floor tower II.

Status:

Don Logan. PC diskette support completed. Only 8 bytes of memory left. Read alternate sector map not implemented.

Need latest powerup diagnostic to merge with code as well as a number of new 2764-2 proms.

Customer Engineering.

There are currently 2 domestic and 2 international critical accounts as well as 12 other sites currently being escalated.

BETA Release:

The BETA shall include the 16 sites (4 critical and 12 being escalated) being monitored by customer engineering as well as Harvey Worthington, Mike Riley and Jerry Sevigny. This requires a total of 20 sets of proms.

Internal testing of the proms shall be conducted until the 15th of April. After that time, sets will be sent out by CE to the identified 16 sites.

Action Items:

1. Jerry Sevigny and/or Mike Riley to get Dave Netzel to provide Don Logan with the latest diagnostics. This must happen Monday the 2nd of April. NO EXCUSES!!!

2. Harvey Worthington and/or Mike Riley are to provide Don Logan with at least 20 sets of new PROMs for the new code.

3. Don Logan will make the 20 or so sets of proms and provide these to Harvey Worthington as soon as possible after receiving the proms and the latest powerup diagnostic.

4. Harvey Worthington to distribute the new code to the designated BETA sites -

   2 domestic & 2 international critical accounts
   12 accounts currently being escalated
   Jerry Sevigny

30 Meg Support.

The microcode needs to be tested with 30 meg drives in the 2275-30 and 2275-60 configurations. Mike Riley and Harvey Worthington will get their own units. Gene Mantoni will get Don Logan and myself 30 meg units.

PC FORMAT UTILITY.

The Model 2275 Disk Drive User Manual documents the procedure used to format a PC diskette using the 2275. The necessary diskette header information needed by the Wang PC is also supposed to be recorded. I certainly will try this.

Some discussion has been raised as to the need for a utility to create a PC MS-DOS diskette. There is, at this time, NO plans to provide any further utilities. We will listen to suggestions but do not foresee a need.

2275 On the 2200T.

The ability of using a 2275 on a 2200T must be measured. Apparently, some 2200T users have determined that PC/BASIC-2 and the PC offers them a growth path. The $GIO sequences of the VP differ from the T however, the basic read and write commands may work.

Upgrading a 2275-10 (2275-30) To A 2275-20 (2275-60).

Some people, with a MVP-P package, have asked about upgrading from the 2275-10 to the -20, this could present a problem to customer engineering because all their diagnostics are available only on diskettes. In the event, the users winchester becomes disabled there is no way to BOOT the system nor exercise the diagnostics.

John Deutsch is to get in touch with customer engineering and iron out this issue.

Distribution

Dave Barrett

John Deutsch
Neil Aronson
Gene Mantoni

Mike Riley
Harvey Worthington

Loren Albright
Don Logan

2235

WANG

ECO

ECO NO. 32608

SHEET 1 OF 4

ORIGINATOR	Don Logan/Al Floyd	M/S 1439	EXT. 76302/77468	DATE	05/07/84
WRITTEN BY	Valerie Goguen	M/S 1218B	EXT. 4313	DATE	05/07/84

PART NO.	210-8396-A
DWG NO.	8396
MODEL NO.	2200MVPA/C
CLASS	I II III
DESCRIPTION	5 1/4 Disk Contrl M/L
PEP #	PEP# H0197A
TYPE	☑ HARDWARE ☐ SOFTWARE

DESCRIPTION OF CHANGE

Change schematic, software loading chart and sample board as follows:

	FROM	TO
L100	379-2000	378-2000-R1

Change BOM 210-8396-A as follows:

	WLI#	DESCRIPTION	UM	COMP TYPE	QTY	QTY TYPE
DELETE:	379-2000	Prom	EA	5 P		1
ADD:	379-2000-R1	Prom	EA	5 P	1	1

Delete the Product Structure and Part Number from the Data Base for the following prom:

379-2000

WLN = 378-0522

REASON/SYMPTOM FOR CHANGE

SEE ATTACHED SHEET.

DOCUMENTS	REVISIONS F	REVISIONS T
BOM		
ARTWORK		
E-REV		
SAMPLE BD		
ASSY. DWG.		
DRILL DWG.		
SCHEM. DWG.	3	4
MECH. DWG.		
CBL ASSY. DWG.		
S.P.I.		
MECH ASSY. DWG.		
COMPONENT SPEC		

EFFECTIVITY	RE-MFG.	C.E.	DIST.	FINAL ASSY AREA	SUBY ASSY AREA	FUTR MFG.
TO CONFORM	X	X	X	X	X	
USE AS IS TO PREVIOUS REV.						

APPROVALS	DATE
ECO MGR.	
DES. ENGRG. Don Logan	5-9-84
CUST. ENGRG.	
MFG. ENGRG.	
ORIGINATOR Al Floyd	5-9-84

This general release of the 2275 Disk Processing Unit firmware obsoletes the original production version. This release corrects the following problems:

System hangs if board is powered up with no floppy diskette in the drive.

System hangs if a LIST DT command is issued when there is no media in the drive.

Alternate sector mapping function does not work.

Winchester hardware is not disabled when the Z80 attempts to access the alternate sector map, creating a memory contention.

Parity error hangs the system without returning an error message.

Cacheing problem causes invalid data to be returned to the 2200.

System status message does not properly identify the system configuration.

Additions have been made to the firmware as follows:

Firmware now supports 30 M-byte Quantum Winchester drives.

PC diskettes can be read, written, and formatted by the 2275.

Winchester format is now a one pass operation that writes a DB6, B6D, 6DB pattern onto the disk to help locate marginal sectors. ECC is disabled during format, and only two retries are allowed for header errors. In addition to mapping out a bad sector, the sectors immediately preceding and following the bad sector are mapped out.

A hook is provided so that Customer Engineering can perform radial-track and index-to-data head alignment.

A hook is provided to allow Customer Engineering to read the alternate sector map.

32608

SH 2

WANG

ENGINEERING CHANGE ORDER
CUSTOMER ENGINEERING EFFECTIVITY AND DISPOSITION

ECO 32608

SHEET 4 OF 4

LL UNITS ☐	EFFECTIVITY COMMENTS:		DOMESTIC	INTER-NATIONAL
ROB. ONLY ☐		EST. UNIT POP.		
IFO ☐		EST. SPARE POP.		
CO REQUIRED ☐		TOTAL		
KP.				
SC REWORK ☐				
KP.				

APPROVALS	DATE	PROJECTED COST IMPACT		PART #	QTY	EXT. COST
CH OPS		MATERIAL				
OGISTICS		LABOR				
NAL		TOTAL				
THER				TOTAL		

Projected Part Requirement: PART # / QTY / EXT. COST

ENERAL COMMENTS:

14-7101 Printed in U S A. 1 82

2260 DISK DRIVES

TYPES: 2260C/2260BC (MULTIPLEXED)

5MB - 2260C 1/4 / 2260BC 1/4	4,799(4,800) SECTORS	
10MB - 2260C 1/2 / 2260BC 1/2	9,599(9,600) "	
20MB - 2260C / 2260BC	19,583(19,584) "	
20MB - 2260C-2 / 2260BC-2	18,583(19,584) "	**DAISY CHAIN

DISK CONTROLLERS: 2260C--22C12 DISK CONTROLER MODUEL
2260BC--22C13 DISK CONTROLLER MODUEL & CONTROLLER
22C13 & 2230MXA -- MULTIPLEXER REQUIRES 2 SLOTS

2260BC-2/2260-2 "DAISY CHAINED"

****

2270 DISKETTE DRIVE

TYPE OF DISKETTES: 2270----8" SSSD HARD SECTOR WHITE LABEL ONLY by WANG
#177-0063-1 (262,144 BYTES)
2270A---8" SSSD SOFT SECTOR GREEN LABEL (IBM COMPATIBLE)
#177-0074-1

SECTORS:
2270 END = 1,023(1,024) (ONLY 64 OF THE 77 TRACKS USED)
2270A END = 1,231(1,232)

360 RPM

****

2275 DISKETTE DRIVE

MODEL	REMOVABLE	FIXED
2275-10	1-320KB	1-5MB
2275-20	0	2-5MB
2275-30	1-320KB	1-30MB
2275-60	0	2-30MB

5 1/4 320KB (2200 FORMAT) OR 360KB (PC FORMAT) DSDD DISKETTES

END=
DSDD FLOPPY END = 1279(1280)
10MB FIXED END = 38911
30MB FIXED END = 64023/15MB ADDRESS

22C03 DISK CONTROLLER
22C11 DUAL CONTROLLER
22C32 TRIPLE CONTROLLER

****

2280 PHOENIX DRIVE

ALL	13.4MB	REMOVABLE	END = 52,607(52,608)
2280-1/2280N-1	26.8MB	1R/1F	END = 52,607(52,608)
2280-2/2280N-2	53.6MB	1R/3F	END = 52,607(52,608)
2280-3/2280N-3	80.4MB	1R/5F	END = 52,607(52,608)

DPU: INTELLIGENT DPU COMES WITH EACH 2280
2280MUX: MULTIPLEXING (MAX OF 3 CPU"S) (MAX OF 1000' CPU TO DPU
22C80 CONTROLLER BOARD

****

2200 L V P

WINCHESTER DISK DRIVE (B10/D11):
2-MEG......END=8127(8128)
4-MEG......END=16319(16320)
8-MEG......END=32639(32640)
16-MB......END=65407(65408)
32-MB END=65407/16MB ADDRESS

DSDD RED LABEL DISKETTE (310/D10): PART # 177-0070-1
1-MEG......END=3873(3874)

****

FOURTH DISK DRIVE ON SYSTEM

ADDRESSES: 326/B26
MUST MODIFY UTILITIES MAT SEARCH STATEMENT TO ACCEPT THE NEW ADDRESS 320->326
B20->B26 ETC.
@MOVE...LINE 5030 // @BACKUP...LINE 0780 // @RECOVER...LINE 0770 // @INSTALL
LINE 5030 NOTE: ISS UTILITIES ARE ALSO AFFECTED !!!!

****

DISKETTES

8" SSSD HS (WHITE) #177-0063-1 (TYPE 3)
8" DSDD SS (RED) #177-0070-1 (TYPE 5)
5 1/4" DSDD SS (RED)#177-0080-1 (TYPE 9)
8" SSSD SS (GREEN) #177-0074-1
5 1/4 SSSD HS (YELLOW)#177-0064-1
**** OLD PCS II ****

To: Paul Morin
Subject: 2275 ECO

From: Paul Morin
Date: 08/14/85

As you know the 2275 R3 PROM will soon be available, and in addition to the PROM a jumper cable is needed. Hence the ECO's for both software and hardware are as follows.

PROM	37694
hardware	36892

Compliments of R&D (DB)

2275-10/20/30/60 ADDRESSING
5 1/4 DISK DRIVE

ON THE 10 MEG DRIVE	THE FLOPPY IS D10 (B10) WINCHESTER IS D11 (310)(5/5 MEG)
ON THE 20 MEG DRIVE	THE 1ST WINCHESTER IS D10 (B10) (5/5 MEG) THE 2ND WINCHESTER IS D11 (310) (5/5 MEG)
ON THE 30 MEG DRIVE	THE FLOPPY IS D10 (B10) WINCHESTER IS D11 (310) (15 MEG) D12 (15 MEG)
ON THE 60 MEG DRIVE	THE 1ST WINCHESTER IS D10 (B10) (15 MEG) D13 (310) (15 MEG) THE 2ND WINCHESTER IS D11 (15 MEG) D12 (15 MEG)

NO INFO ON THE SECTOR ADDRESS. BUT IT WILL PROBABLY BE AROUND 58 OF 59 THOUSAND ABOVE INFO COURTESY OF GEORGE COLELLA (TAC).

NOTE 326,327,325: If you try to plug the 2275 disk into a board that has device address 326,327,325 or any other oddball address, you will get a P48 error whenever you try to access the disk, no matter what you have specified in the OS device table. Apparently our newest storage peripheral can"t handle the unsupported device addresses on the 2200.....

2200 QUESTIONS AND ANSWERS
by Technical Support Center

The Wang Technical Support Center (TSC) provides the following answers to some commonly asked questions about the 2200 product line.

Q1. On page 2200/2-6 of the May 1, 1984 Wang Pricing Manual, I noticed that the descriptions of the MVP-P2 and MVP-P4, as well as the MVP-P3 and MVP-P5, were identical, with the exceptions of disk storage space and the number of available I/O slots. Why do the MVP-P4 and MVP-P5 have two less available I/O slots than the MVP-P2 and MVP-P3?

A1. The descriptions of the MVP-P4 and MVP-P5 on page 2200/2-6 of the May 1, 1984 Wang Pricing Manual are incorrect. For the correct descriptions, refer to page 843 of the May 15, 1984 issue of FOCUS. (The correct descriptions also appear in the August 1, 1984 issue of the Wang Pricing Manual.) In reference to the correct descriptions, note that the 2236MXE Terminal Processor and the 2229IOP Controller, each of which uses an I/O slot, are the reason for the MVP-P4 and the MVP-P5 having two less available I/O slots than the MVP-P1, MVP-P2 and MVP-P3.

2275

Q2. The 2275 disk device on my 2200 computer does not format the fixed disk to accommodate the format requirements of the Wang 2200/PC Interchange. I have followed the instructions in the Model 2275 Disk Drive User Manual. What is wrong?

A2. This is a known and correctable 2275 disk device anomaly. On the 210-8396 controller board, you must replace PROM 278-90000 (the original PROM) or PROM 278-92000 (the first update of the 278-90000) with PROM 379-2000-R1. Also note that the more recently distributed 210-8396-A controller boards may have PROM 379-2000. You must replace this PROM with PROM 379-2000-R1. PROM 379-2000-R1 is compatible with the microcode necessary for the 30MB Disk Drive and with the disk formatting requirements of the 2200/PC Interchange.

Q3. I just purchased a 2275-60 disk device for my MVP, and I cannot seem to access the fixed disks. What is the addressing scheme for this peripheral?

A3. The addressing scheme for the 2275-60 is slightly different than that of the 2275-20 because the 2275-60 contains four separately addressable disk surfaces. Address the four surfaces as follows:

Winchester 1	D10	(B10)
	D13	(310)
Winchester 2	D11	
	D12	

Each surface may be scratched from sector 0 through sector 64,023.

CUSTOMER ENGINEERING
TECHNICAL ASSISTANCE CENTER
NEWSLETTER

#40724

3110

PERIPHERALS-DISK DRIVES-5 1/4 DISK DRIVES

TOPIC: FCO 1100

FCO 1100, which has the new R1 PROM for the model 2275-10 and 2275-20, will be replaced by FCO 1107 due to D81 and D88 errors. R&D is planning to release the R2 PROM by the end of this month. The R2 PROM will take care of the D81 and D88 errors. Sixty R2 pre-release PROMS will be distributed to the Domestic Areas and ten to Europe for Beta site support on July 18. FCO 1107 will be available in August.

ANNOUNCING ADDITIONS TO THE 2200 FAMILY
by 2200 Product Marketing

Wang Laboratories is pleased to announce two additions to the family of 2200 system packages. The MVP-P4 and -P5 packages include the latest in 5 1/4" disk storage technology. Combined with aggressive pricing and increased flexibility in hardware configurations, the 2200 family continues to excel in price/performance.

The family of 2200 products now includes five MVP packaged configurations as well as four versions of the standalone 5 1/4" disk storage peripheral. The new MVP-P4 and -P5 packages with their unique complement of components define the high performance small business computer market.

HIGHLIGHTS

PRODUCT OVERVIEW

The Wang 2200MVP-P1 through -P5 systems represent five exciting small business computer solutions. The MVP-P1 through MVP-P3 packaged systems address the low-cost entry level market. The MVP-P4 and -P5 systems with increased disk storage capacity and intelligent backup capability provide the growth path from the earlier P-packages while appealing to the needs of the large end-user.

The 2275-30 and -60 provide for low cost add-on storage for existing users. Its compact design will be most attractive to the space conscious buyer. Measuring approximately 14.9" x 6.5" x 16" (38cm. x 16cm. x 40cm.), the unit will utilize an enclosure similar in design to the Wang PC electronics unit and house the disk drives, one printed circuit logic board, one Regulator board, a 60 watt linear power supply and one 50 cfm fan.

Because sector size and sectors per track are compatible, the 2275 can represent the media bridge for users who wish to migrate software and data files to the Wang PC. The units can be used with any 2200VP, LVP, LVPC, MVP or MVPC system. When combined with a low cost MVP it makes a very price competitive entry level system. Since this device incorporates the latest in 5 1/4" micro-Winchester technology, it will be helpful in selling new accounts. With the addition of the 2275 unit, the price/performance ratio of our low to middle range 2200 systems has been significantly improved.

SOLUTIONS

The new offerings provide a highly expandable progression from the earlier systems at prices competitive in the small business arena. The 5 1/4" package prices will take advantage of the new storage peripheral for the 2200. The pricing of the 5 1/4" storage device is competitive with that of other offerings in the personal computer marketplace.

New Accounts

For the small account, the packaged systems represent a low cost solution to their data processing needs. For the large account, these systems are a strong solution to their distributed processing needs.

PRODUCT DESCRIPTION
DETAILED DESCRIPTION

Two new offerings are being introduced. These are:

1. Two additions to the family of 2200 packaged systems -- the MVP-P4, and MVP-P5

2. An add-on disk peripheral unit available in two models -- 2275-30 and 2275-60

The 2275 Disk peripheral units must be used with either a 22C03, 22C11, or a 22C32 Controller.

Diskette formats supported by the 2275 disk peripheral unit are:

256 byte/sector (Standard format used for 2200 disk operations)

Sector size:	256 bytes
Sectors/track:	16
Tracks/side:	40
Sides:	2
Total Capacity:	320k bytes

512 byte/sector (Standard Wang PC format)

Sector size:	512 bytes
Sectors/track:	9
Tracks/side:	40
Sides:	2
Total capacity:	360k bytes

The 512 byte/sector format is the standard Wang PC format and is supported by the 2200 for interchange purposes. The 512 byte sectoring is transparent to the 2200 Operating System. The Disk Processing Unit (DPU) maps two 2200 256 byte logical sectors into one 512 byte physical sector.

BASIC-2 accesses diskettes with this format as if the platter were formatted with 256 byte sectors. All BASIC-2 disk operations can be performed.

Disk Peripherals

2275-10	Disk device containing one 10MB Winchester drive and one 320K floppy drive
2275-20	Disk device containing dual 10MB Winchester drives
2275-30	New disk device containing one 30MB Winchester drive and one 320K floppy drive
2275-60	New disk device containing dual 30MB Winchester drives

Disk Upgrades

UJ-5054	Upgrades 2275-10 to a 2275-30
UJ-5055	Upgrades 2275-20 to a 2275-60

Tape Device

2229	Intelligent start/stop tape device suitable for disk backup and archiving

AVAILABILITY

	Domestic	International
Effective:	April, 1984	May, 1984
First Customer Ship:	May, 1984	June, 1984

2200 QUESTIONS AND ANSWERS

by Technical Support Center

The Wang Technical Support Center (TSC) provides the following answers to some commonly asked questions about the 2200 product line.

SCRATCH 2
15MB = 64,023
10MB = 39,783

Q1. What are the formatting and scratching (initialization) procedures on a 2275-60 disk drive?

A1. The 2275-60 disk drive has four addresses that you must format and scratch individually. You must format each address with the following type of statement:

```
$FORMAT DISK T/D12
```

where D12 is a disk address. You must scratch each address with the following type of statement:

```
SCRATCH DISK T/D12, LS=25, END=64023
```

The LS statement specifies the number of sectors you wish to set aside for the Catalog Index. (Acceptable values are 1 through 255.) The END statement specifies the last sector you wish to assign to the Catalog Area. (The maximum value is 64023.) For additional information on scratching procedures, refer to the 2200 BASIC-2 Disk Reference Manual (700-4081G).

Q2. What are the disk addresses on the 2275-30 and 2275-60 disk drives?

A2. The disk addresses are as follows:

2275-30:

Diskette (DSDD)	D10 / D20 / D30	
Winchester	D11 / D21 / D31	(1st 15MB)
	D12 / D22 / D32	(2nd 15MB)

2275-60:

1st Winchester	D10 / D20 / D30	(1st 15MB)
	D13 / D23 / D33	(2nd 15MB)
2nd Winchester	D11 / D21 / D31	(3rd 15MB)
	D12 / D22 / D32	(4th 15MB)

If you have the disk drive positioned horizontally (vent on the bottom), the diskette drive or 1st Winchester drive is on the right. If you have the disk drive positioned vertically (vent on the right), the diskette drive or 1st Winchester drive is on the top.

Q3. <u>I recently received some red-labeled, 5 1/4-inch diskettes for my 2200 PCSII from Wang ExpressProducts. The new diskettes do not work. Has there been a diskette format change?</u>

A3. There has been no change in the diskette formats for the 2200 PCSII. Wang now has two types of 5 1/4-inch diskettes available for the 2200 series computers:

- Single-sided, single-density (SSDD), hard-sectored (Wang yellow label). (177-0064-1)
- Double-sided, double-density (DSDD), soft-sectored (Wang red label). (177-0080-1)

For the 2200 PCSII and PCSIIA, you must use the SSSD, hard-sectored (yellow label) diskettes. The DSDD, soft-sectored (red label) diskettes are for use with the 2275 disk drives. When you order diskettes for the 2200 PCSII from ExpressProducts, be certain to specify "yellow label" and provide the part number indicated above.

Q4. <u>I have a 2200 MVP and am using Word Processing (WP) Release 2.2. Sometimes when I try to use the Move or Copy function, I get the error message "work file in use." What causes this to happen?</u>

A4. This error usually occurs because you or another user exited WP before completing a Move or Copy function (or any editing function that uses a work file). As the error message suggests, this causes the work file to remain in use. To clear the in use condition, access the Supervisory Functions menu and select the Clear In Use Condition option (first option). Enter the document ID, use the TAB key to select the Work File option, and press EXECUTE. When you access the document now, you can use the editing function. This clearing procedure also applies to WP Release 2.1; however, the Work File option reads "Reference File."

WORKSTATION 118 - USER PEM - PAUL MORIN X-60350

10:20:41 AM MONDAY MAY 20, 1985

```
*********************************************************************************************
****         1         2         3         4         5         6         7         8 ****
**** 1234567890123456789012345678901234567890123456789012345678901234567890123456789O ****
*********************************************************************************************
*  *                                                                                    *  *
* 1*  VS EXPRESS:  DISPLAY A MEMO                          (1/15)   SAVE/PRINT          * 1*
* 2*                                                       (^2)     REROUTE             * 2*
* 3*  TO:       PAUL MORIN VS100-1                         (4/5)    PREV/NEXT MEMO      * 3*
* 4*  FROM:     KEN MAILLOUX VS100-1                       (8)      DELETE MEMO         * 4*
* 5*  SENT:      1/15/85     9 25                          (^13)    SPECIAL OPTIONS     * 5*
* 6*  SUBJECT:  2275-10 AND UPGRADES____                   (14/^14) INS/DEL LINE        * 6*
* 7*                                                       (16/^16) RETURN/EXIT         * 7*
* 8*     ------------------------------------------------------------------------       * 8*
* 9*                                                                                    * 9*
*10*      THE 2275-10   5.25 DSDD AND 5.25 10 MB WINCHESTER CANNOT BE FIELD UPGRAD      *10*
* 1*    ED TO A 2275-20   DUAL 5.25 10 MB WINCHESTERS WITH NO REMOVABLE.                * 1*
* 2*                                                                                    * 2*
* 3*      HOWEVER, ON OR AROUND APRIL 16TH WANG LABS. WILL PROUDLY ANNOUNCE  THE        * 3*
* 4*                                                                                    * 4*
* 5*          2275-30   5.25 DSDD AND A 30 MB WINCHESTER.                               * 5*
* 6*                                                                                    * 6*
* 7*      UPGRADES WILL BE IN THE NEXT PRICE LIST FOR 2275-10 AND 2275-20 TO -30S.      * 7*
* 8*                                                                                    * 8*
* 9*         THE WINCHESTER IS SWAPPED AND PROMS CHANGED.                               * 9*
*20*         THIS INVALUABLE INFORMATION WAS OBTAINED FROM THAT WIZARD OF HARDWARE      *20*
* 1*           MARKETING HIMSELF--GENE MANTONI.                                         * 1*
* 2*                                              KENNETH D.                            * 2*
* 3*     --------------- MESSAGE REPLY  FROM (KEN MAILLOUX VS100-1     ) -------------  * 3*
* 4*     ------------------------------------------------------------------------       * 4*
*  *                                                                                    *  *
*********************************************************************************************
****         1         2         3         4         5         6         7         8 ****
**** 12345678901234567890123456789012345678901234567890123456789012345678901234567890 ****
*********************************************************************************************
```

WORKSTATION 118 - USER PEM - PAUL MORIN X-60850

10:19:43 AM MONDAY MAY 20, 1985

```
************************************************************************************
****         1         2         3         4         5         6         7         8 ****
**** 1234567890123456789012345678901234567890123456789012345678901234567890123456789O ****
************************************************************************************
*  *                                                                            *  *
* 1* VS EXPRESS:  DISPLAY A MEMO                        (1/15)    SAVE/PRINT     * 1*
* 2*                                                    (^2)      REROUTE        * 2*
* 3* TO:      PAUL MORIN VS100-1                        (4/5)     PREV/NEXT MEMO * 3*
* 4* FROM:    KEN MAILLOUX VS100-1                      (8)       DELETE MEMO    * 4*
* 5* SENT:     1/15/85     9 28                         (^13)     SPECIAL OPTIONS* 5*
* 6* SUBJECT: WINCHESTER_FORMAT_ERRORS                  (14/^14)  INS/DEL LINE   * 6*
* 7*                                                    (16/^16)  RETURN/EXIT    * 7*
* 8*  ---------------------------------------------------------------------------* 8*
* 9*                                                                             * 9*
*10*                                                                             *10*
* 1*                                                                             * 1*
* 2*   I GOT A CALL FRIDAY AFTERNOON FROM SHEENA EENEY (DENVER OFFICE) CONCERNING  * 2*
* 3*   FORMAT ERRORS ON WINCHESTER DRIVES. SHE BELIEVES THAT RELEASE 2.4 OF THE    * 3*
* 4*   MVP OPERATING SYSTEM WILL CAUSE I93'S AND I96'S ON WINCHESTER DRIVES UNLESS * 4*
* 5*   LEVEL E9 PROMS ARE INSTALLED IN THE LVP. IF ANYONE HEARS OR HAS HEARD       * 5*
* 6*   ANYTHING ABOUT THIS, PLEASE CONTACT ME. SHEENA IS CHECKING THIS OUT WITH    * 6*
* 7*   THE HARDWARE PEOPLE.                                                        * 7*
* 8*                                                                             * 8*
* 9*                                                                             * 9*
*20*                                           -GREG-                            *20*
* 1*   NEEDS 2607 - 3 BOARD IN WINCHESTERS ON LVP WITH LESS THAN 16 MEG DRIVES.    * 1*
* 2*          PROM LEVEL SHOULD BE 4,5, OR 6                                      * 2*
* 3*   --------------- MESSAGE REPLY  FROM (LEE COLLETTE EXT 3594   ) ------------ * 3*
* 4*   ------------------------------------------------------------------------- * 4*
*  *                                                                             *  *
************************************************************************************
****         1         2         3         4         5         6         7         8 ****
**** 12345678901234567890123456789012345678901234567890123456789012345678901234567890 ****
************************************************************************************
```

WORKSTATION 118 - USER PEM - PAUL MORIN X-60350

10:19:29 AM   MONDAY   MAY 20, 1985

```
*****************************************************************************************
****         1         2         3         4         5         6         7         8 ****
**** 1234567890123456789012345678901234567890123456789012345678901234567890123456789 ****
*****************************************************************************************
*  *                                                                                *  *
* 1*  VS EXPRESS:  DISPLAY A MEMO                            (1/15)    SAVE/PRINT   * 1*
* 2*                                                         (^2)      REROUTE      * 2*
* 3*  TO:        PAUL MORIN VS100-1                          (4/5)     PREV/NEXT MEMO * 3*
* 4*  FROM:      KEN MAILLOUX VS100-1                        (8)       DELETE MEMO  * 4*
* 5*  SENT:       1/15/85      9 28                          (^13)     SPECIAL OPTIONS * 5*
* 6*  SUBJECT: 2275 DISK UNIT___________                     (14/^14)  INS/DEL LINE * 6*
* 7*                                                         (16/^16)  RETURN/EXIT  * 7*
* 8*   --------------------------------------------------------------------------   * 8*
* 9*                                                                                * 9*
*10*                                                                                *10*
* 1*       IF YOU TRY TO PLUG THE 2275 DISK INTO A BOARD THAT HAS DEVICE ADDRESS    * 1*
* 2*   326, 327, 325 OR ANY OTHER ODDBALL ADDRESS, YOU WILL GET A P48 WHENEVER      * 2*
* 3*   YOU TRY TO ACCESS THE DISK, NO MATTER WHAT YOU HAVE SPECIFIED IN THE OS      * 3*
* 4*   DEVICE TABLE. APPARENTLY OUR NEWEST STORAGE PERIPHERAL CAN'T HANDLE THE      * 4*
* 5*   UNSUPPORTED DEVICE ADDRESSES ON THE 2200.........                            * 5*
* 6*   -------------- MESSAGE REPLY  FROM (KEN MAILLOUX VS100-1     ) ------------  * 6*
* 7*                                                                                * 7*
* 8*                                                                                * 8*
* 9*                                                                                * 9*
*20*                                                                                *20*
* 1*                                                                                * 1*
* 2*                                                                                * 2*
* 3*                                                                                * 3*
* 4*   --------------------------------------------------------------------------   * 4*
*  *                                                                                *  *
*****************************************************************************************
****         1         2         3         4         5         6         7         8 ****
**** 1234567890123456789012345678901234567890123456789012345678901234567890123456789 ****
*****************************************************************************************
```

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL: 617/459-5000, TWX 710-343-6769, TELEX 94-7421

WANG

TECHNICAL SERVICE BULLETIN

SECTION: HardWare Technical

NUMBER: HWT 5033 REPLACES: N/A DATE: 02/26/85 PAGE 01 OF 01

MATRIX ID. 3110 PRODUCT/RELEASE# 2275

TITLE: 2275 Problems

PURPOSE:
To inform the field of the problems that have been identified and will be resolved on the 2275.

1. The '40' bit of the drive address is not checked. The 2275 responds to 350 as if it were 310, 360 as if 320, and 370 as if 330.

2. The $GIO "read PROM Rev. level" command does not work.

3. The non-sequential multi-sector write problem is from an incomplete implementation of multi-sector write as implemented in the Phoenix and the SVP/LVP.

4. The head alignment routine requires changes to the read routine.

5. The power-up diagnostic does not flag floppy drive stepper motor failures.

6. Formatting a drive does not reset the read/write caches.
   NOTE: Notify all customers to use the reset key in the keyboard after formatting the drive as a circumvention.

7. In the Winchester back-up subroutine the 2275 does not recognize unformatted diskette and hangs the operation.
   NOTE: Notify all customers to use formatted diskettes only as a circumvention.

8. The diagnostic gets lost if it gets an error on the Winchester.

All these problems will be corrected in the R3 PROM. Estimated completion date for the PROM is the end of April.

GROUP: Peripheral Hardware Support Group MAIL STOP: 0125

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL: 617/459-5000, TWX 710-343-6769, TELEX 94-7421

WANG

TECHNICAL SERVICE BULLETIN

SECTION: HardWare Technical

NUMBER: HWT 5233 REPLACES: ________ DATE: 11/05/85 PAGE 1 OF 1

MATRIX ID. 8108 PRODUCT/RELEASE# Professional Computer

TITLE: 5 1/4" Floppy Diskette Problem

PURPOSE:

To inform the field of a possible media (5 1/4" DSDD) problem, when using P.C. as an Archiving WS on an OIS or VS system.

EXPLANATION:

This problem occurs when a P.C. is attached to an OIS or VS system as an "Archiving Workstation" and the user attempts to "initialize" a diskette for use as an archive diskette. The problem occurs with some, but not all, new (un-formatted) Wang diskettes when you try to prepare or initialize a new disk. Various error messages are given which indicate that the system does not recognize the floppy diskette in the drive. The resolution for this problem is to format the problem diskettes with PC FORMAT and then initialize them for archiving.

This problem may also cause a "Drive Not Ready" error when trying to run the disk copy utility. The same circumvention applies to this symptom also. Customers with 4250 workstations should be told to contact the Customer Action Line (CAL), at 1-800-323-WANG, for help in circumventing the problem.

GROUP: Office Automation Hardware Support Group MAIL STOP: 0114

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL: 617/459-5000, TWX 710-343-6769, TELEX 94-7421

WANG

TECHNICAL SERVICE BULLETIN
SECTION: HardWare Technical

NUMBER: HWT 5238 REPLACES: ________ DATE: 10/22/85 PAGE 1 OF 1

MATRIX ID. 3110 PRODUCT/RELEASE# 5 1/4 Disk Drive

TITLE: FCO 1111A, 2275-10, 20, 30 Disk Peripheral

PURPOSE:

To inform the field that FCO 1111A has been released.

EXPLANATION:

FCO 1111A, released October 1, 1985, documents ECO 37694 and replaces FCO 1111. One EPROM on the 210-8396-A Disk Controller PCB is changed. The reason for the change is to correct the following problems.

1. The '40' bit of the drive address is not checked. The 2275 responds to 350 as if it were 310, 360 as if 320, 370 as if 330.
2. An "I92" error occurs when the '$GIO' "Read Prom Rev Level" command is given.
3. The power-up diagnostic does not flag floppy drive stepper motor failures.
4. Formatting a drive does not reset the read/write caches.
5. In the Winchester back-up subroutine, the 2275 does not recognize an unformatted diskette and hangs the system.
6. The diagnostic gets lost if there is an error on the Winchester.
7. Will not report the number of sectors available.

FCO 1175 must be done in conjunction with FCO 1111A. (An "I91" error may occur if FCO 1175 is not installed.)

FCO kit #728-0129A is available and can be obtained by placing a routine order through the Logistics order processing system.

GROUP: ECO Support Group MAIL STOP: 0139

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL: 617/459-5000, TWX 710-343-6769, TELEX 94-7421

WANG

TECHNICAL SERVICE BULLETIN
SECTION: HardWare Technical

NUMBER: HWT 5239 REPLACES: ________ DATE: 10/22/85 PAGE 1 OF 1

MATRIX ID. 3110 PRODUCT/RELEASE# 5 1/4 Disk Drive

TITLE: FCO 1175, 2275-10, 20, 30 Disk Peripheral

PURPOSE:
To inform the field that FCO 1175 has been released.

EXPLANATION:
FCO 1175, released October 1, 1985, documents ECO 36892. Three jumper wires and one resistor are added to the 210-8396A Disk Controller PCB. The reason for the change is to support the R3 EPROM revision. Without this change an "I91" error may occur when the R3 EPROM is installed.

FCO kit #728-0191 is available and can be obtained by placing a routine order through the Logistics order processing system.

GROUP: ECO Support Group MAIL STOP: 0139

WANG LABORATORIES, INC.
ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL: 617/459-5000, TWX 710-343-6769, TELEX 94-7421

MEMORANDUM

TO : Joe Scaglione MS/001-260

FROM : Harvey A. Worthington

SUBJECT: Floppy Media For the 2275 (WPN 177-0080)

DATE : April 15, 1987

PROBLEM- Field escalated the inability to format new diskettes from Wang Direct on the Wang model 2275.

The diskettes that you gave me are from Sentinel. This vendor by mistake shipped approximately 20000 diskettes (2000 boxes) to the supply division. Wang Direct has none of these left in stock at this time. This vendor error was corrected and new procedures have been implemented at Wang Direct to include a verification of erasure at incoming.

The reason why the 2275 failed to format the diskettes is that the media was tested by the vendor, at the end of the test the vendor had all one's pattern on all tracks on both sides. Usually after this test we force the vendor to degausse the media.

Recommendations-

1. The 2275 should format any 177-0080 diskette without problems with left over data, different format or testing information. In order to prevent future media problems I recommend to have the Software group in R&D verify their format routine and see if they can correct this difficulty.

2. You can recommend the field to use another Wang System (i.e. PC), once you format the diskettes they will be able to format the diskettes in the 2275.

cc. Robert Beaudette MS/014-490
Ray Peltzman MS/001-140
Kim Thompson MS/001-140

DS

2275F/R DISK UNIT

FOR THE WANG 2200

SOFTWARE DESIGN SPECIFICATION

Document Number: MVP-07

Revision 1.2

May 14, 1986

AUTHOR
David M. Barrett
X-74561
M/S 1489

CONTENTS

i. REVISION HISTORY

i.1 Revision 0.0	December 9, 1985	Rough Draft - Rev 0.
i.2 Revision 0.1	December 24, 1985	Rough Draft - Rev 1.
Section 3	All modules were renumbered and the following sections were added: 3.10 2200 Interface Module, 3.11 DMA Module, 3.12 Status Port Module, 3.13 Drive Table Module, 3.14 DPU Ready/Busy Module, 3.16 Power-On Module.	
Section 4	Section 4.6, PROM File Creation, was added.	
Appendix B	Source module and usage information was added to the data dictionary.	
Appendix D	An appendix specifying the contents of the batch files used to build the system was added.	
Appendix E	An appendix listing the names of all source files was added.	
Appendix F	An Appendix specifying the contents of all drive table was added	
i.3 Revision 1.0	January 22, 1986	Working Draft - Rev 1.
Section 2.1	PROM size and mapping strategy changed.	
Section 3.2	Platter Table Module updated.	
Section 3.3	Bit map of platter formattability added.	
Section 3.4	Sector zero caches moved to low memory. Routine to set the number of valid sectors in a cache was added.	
Section 3.7	PROM mapping scheme was changed.	
Section 3.12	Floppy media disturbed and write protect routines added.	
Section 3.13	The number of cylinders to reserve for alternate sectoring was added to the Drive Table format.	
Section 3.16	Modified to reflect the new alternate sector map format and the new reserved cylinders strategy.	
Section 4.7	Prom Image File Creation section added.	
Appendix B	Data Dictionary updated.	
Appendix C	Memory Map updated.	
Appendix D	Batch file "prom" corrected.	
Appendix F	Drive tables corrected.	
Appendix G	I/O Port Assignments - added.	
Appendix H	Status Port Bit Assignments - added.	
i.4 Revision 1.1	April 16, 1986	Working Draft - Rev. 1.1
All Sections	PROM size increased to 32Kb.	
Section 3.1	Task File routine definitions updated.	
Section 3.3	Alternate Sector Map data structure corrected and routine definitions updated.	
Section 3.4	Caching routine definitions updated.	
Section 3.7	Memory Bank Selections parameters modified.	
Section 3.8	Switch Setting routines updated.	
Section 3.10	2200 Interface routines updated.	
Section 3.15	2010 Command routines updated.	

Section 3.16	Power-On Module routines redefined.
Section 3.17	High Level Command routine redefined.
Section 3.18	Winchester Command module added.
Section 5.1	Code Size Restrictions updated.
Section 5.2	Data Size Restrictions updated.
Appendix B	Data Dictionary radically changed, see note at beginning of Appendix.
Appendix C	Memory Usage Map updated to include an ECC correction buffer,
Appendix E	Source file names updated.
Appendix G	I/O port assignment updated to include port address labels.
Appendix I	Microcode/Power Up Diagnostics Interface description added.

i.4 Revision 1.2 — May 14,1986 — Working Draft - Rev. 1.2

All Sections	An optional Streaming Cartridge Tape Drive (SCTD) was added to the system configuration.
All Section	The ROM/RAM memory structure was redefined.
Section 3.1	The Task File Module was merged into the 2010 Command Module to provide decreased Winchester access times.
All Sections	Section 3.1 thru 3.18 were renumbered to reflect the deletion of section 3.1, all references use the new numbers.
Section 3.1	The Platter Table data structure and its access routines were redefined to reduce run time overhead.
Section 3.3	The Cache Module was redefined by Andrew Gruber.
Section 3.6	Select Temporary Bank and Restore Bank routines were added to the Memory Bank Module. The select Prom routine was deleted.
Section 3.7	The switch setting were redefined to allow for future expansion to a 1.2 Mb diskette drive.
Section 3.12	The Drive Table format was redefined to refelect the new platter table structure.
Section 3.14	The Abort a 2010 Operation routine was deleted.
Section 5.1	The code size restrictions were updated.
Section 5.2	The data size restrictions were updated.
Appendix C	The memory usage map was redefined and updated.
Appendix F	The Drive Tables were updated.
Appendix I	The Microcode/Diagnostics interface through the Device Validity table was updated.

ii. LIST OF FIGURES

iii. LIST OF TABLES

1. OVERVIEW

1.1 Summary - This software is the 2275F/R Disk Unit resident software and is the interface between the 2200 BASIC-2 Operating System and the disk drives of the 2275F/R. The software can also control the optional 2275S Disk Slave Unit and the optional 2275T Streaming Cartridge Tape Drive. Services compatible with other type 'D-Zero' 2200 disk units are provided. The 2275F/R is a peripheral for the Wang 2200 product family and can control one 5 1/4" 360KB floppy diskette drive, one 10Mb removable cartridge Winchester disk drive, and one fixed Winchester disk drive. If the optional 2275S is attached, two additional fixed Winchester disk drives may also be used. Fixed Winchesters can ranging in size from 10Mb to 112Mb can be accommodated by the hardware and the software. Tables to drive the 10Mb, 20Mb, 33Mb and 67Mb Winchesters are provided. Other tables can be added as needed in the future.

1.2 Objectives - The objective of the 2275F/R software is to:

- Provide a 2200 type 'D-Zero' disk interface for the 2200 BASIC-2 Operating System.
- Provide a large, high speed, intelligent cache to decrease response time and increase throughput.
- Provide support for all commands offered by the 2280 Disk Processing Unit.
- Provide flexibility in types of Winchesters and system configurations can be supported to allow changes in the requirements of the market place to be easily accommodated.

2. ARCHITECTURE

2.1 HARDWARE ARCHITECTURE

There are seven main components in the 2275F/R hardware: a Z80 CPU, an AM9517 DMA controller, a 765A floppy controller, a WD2010 Winchester controller, a QIC-02 parallel interface, a 36Kb PROM, and a 256K SIMM memory module. The function of the Z80 CPU is to control the actions and interactions of the other major components and to control communications between the 2275F/R and the 2200. The Z80 decodes requests from the 2200 and directs the floppy, Winchester and/or DMA controllers to take appropriate actions.

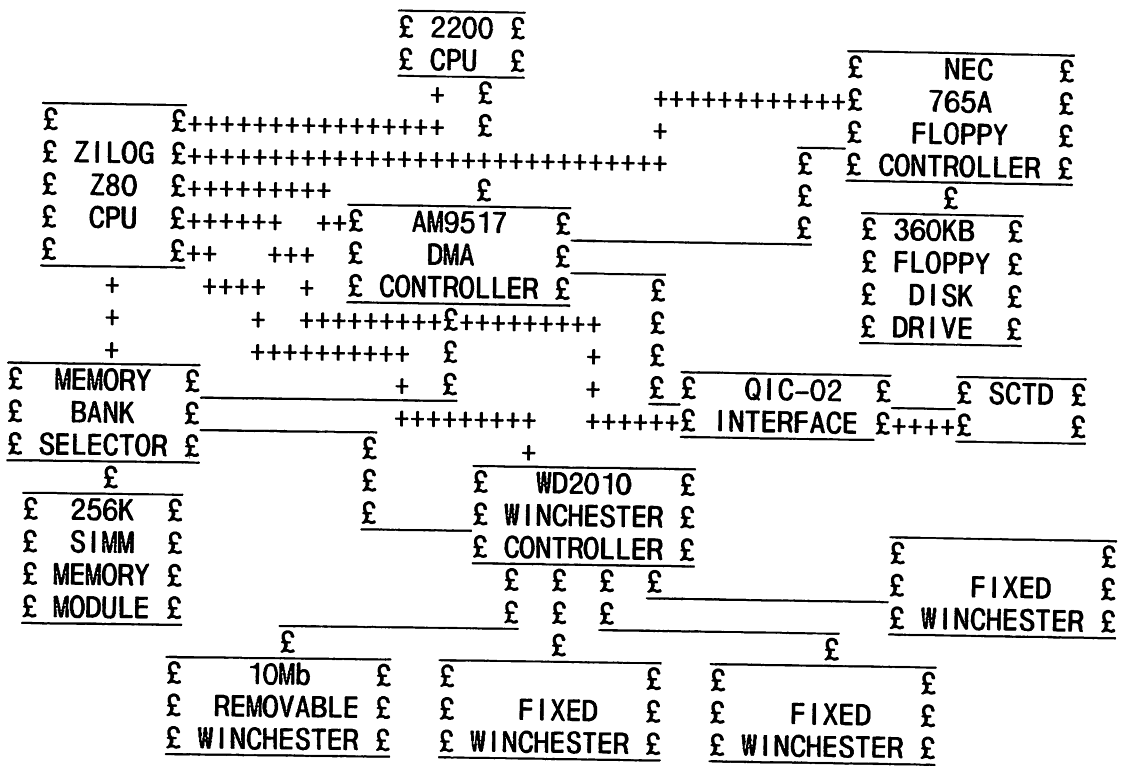
The function of the AM9517 is to move data between the 2200 and memory, and between the 765A floppy controller and memory, under the direction of the Z80.

The WD2010 Winchester controller is used to control up to four Winchesters, two of of which are located in the 2275S Disk Slave

Unit. The WD2010 performs all Winchester reads, writes, seeks, and formats as directed by the Z80, and is responsible for all data transfers between the Winchesters and memory. It also computes the ECC syndrome bytes needed by the Z80 to correct ECC errors.

The 256K SIMM memory module is divided into 16 banks of 16K each. The banks are numbered 0 through 15 and are interfaced to the rest of the system through a memory bank selection circuit. This circuit is controlled by the Z80 and is used to select which one of the upper 15 banks is mapped into the high quarter of the Z80's address space. Bank number 0 is permanently mapped into locations 32 through 48k of the Z80's address space. The 32Kb EPROM occupies locations 0 through 32k.

The interrelationships of the major hardware components are given in terms of the flow of control information and data in Figure 2.1.



KEY

++++++ control information paths
——— data paths

FIGURE 2.1
HARDWARE ARCHITECTURE

2.2 SOFTWARE ARCHITECTURE

The software to be used in the 2275F/R Disk Unit can be broken into three separate parts: the first part is the code used to interface with the floppy drive and DMA controllers, this code is being taken from the existing 2275 microcode. The only changes will be that a $GIO sequence will be provided to enable radial head alinements instead of a switch setting and provisions will be made to accommodate the memory structure of the 2275F/R. The second part of the software involves the Winchester drive controller interface, the platter addressing, the caching, and the commands available to the 2200 CPU. This code will be unique to this product. The third part is the software used to control the QIC-02 interface and support the Streaming Cartridge Tape Drive. Due to the size and complexity of this software it will be documented in its own design specification.

Data encapsulation will be used to hide implementation details from the higher levels of the software. Both the major hardware components and the data structures will be encapsulated in software modules. For example, a set of service routines will be built around the WD2010 so that the details of how to program and communicate with the chip are hidden from the rest of the system. This will make both programming and debugging easier. Simular software modules will be used around data structures such as the Alternate Sector Map and the caches.

Another type of software module is also present in the system. This second type of module is not built around any single data structure or object, but is a collection of logically related routines that use the services of other modules. An example of a Service Module is the 2010 Command Module which provides Winchester services such as read, write, seek, and format track. A diagram showing the flow of data between the major modules is given in Figure 2.2.

All routines in both types of modules will have the following properties unless explicitly stated otherwise in a routine's description:

- o Any registers, other than the accumulator, that are used and are not defined as exporting results will be restored to their original state before control is returned to the calling code segment.
- o The only entry point to a routine will be that defined by the routine's name.
- o When an error is detected, the carry will be set on return to the calling segment. If there was no error the carry will be reset.

The major Data Encapsulation Modules are:

a. The Platter Table Module - The Platter Table, Drive Tables, and Switch Settings provide the system with the drive dependant information needed to control Winchester operations, logical to physical sector mapping, and system

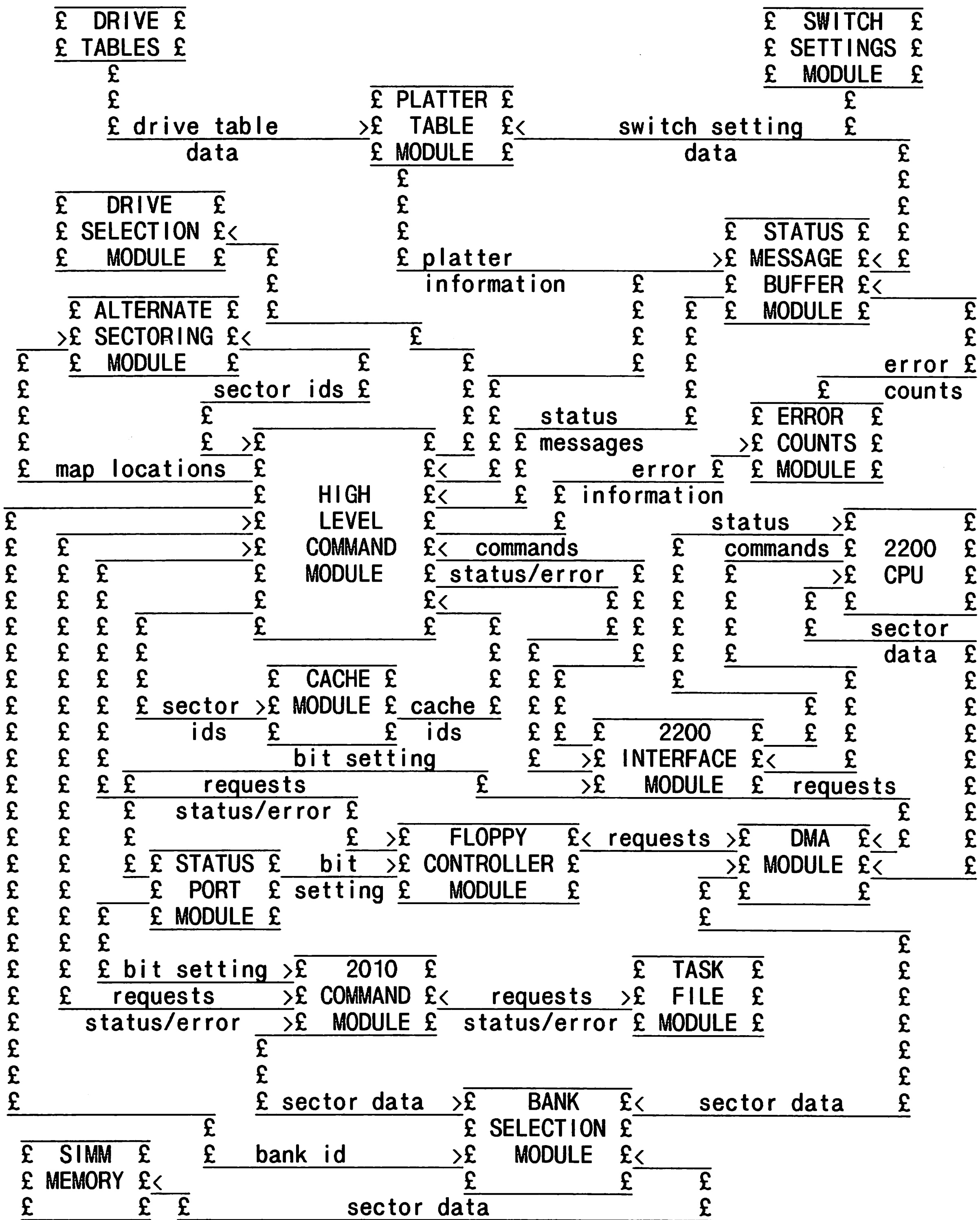


FIGURE 2.2
DATA FLOW BETWEEN MODULES

configuration information. The Platter Table is built based on the switch settings and the information in the Drive Tables.

b. The Alternate Sectoring Module - The Alternate Sector Map is used to track the reassignment of relocated sectors. Routines are provided that initialize, add to, remove from, and search the Alternate Sector Map. The services of the Platter Table Module are used.

c. The Cache Module - This module is responsible for managing the caches. Services are provided for initializing caches, searching for sectors within cache, invalidating sector entries in cache, and assigning cache usage.

d. The Status Message Buffer Module - This module prepares status messages for the $GIO Status Request command. The Platter Table and Error Count Modules are used to gather the necessary information.

e. The Error Count Module - The Error Count Module manages error counts for all of the disk drives on a per platter basis. The module is used by the high level commands to log error counts.

f. The Memory Bank Module - The Memory Bank Module is used to select which bank of memory is mapped into the high half of the Z80's address space. It is used by the high level commands to select the proper memory bank for other low level modules.

g. The Switch Setting Module - This module provides services which allow the rest of the system to discover what type of Winchester drive is in any of the four Winchester location based on the switch settings

h. The Drive Selection Module - Winchester selection is handled by this module. It hides how the software configures the hardware to communicate with the various Winchesters.

i. The 2200 Interface Module - This module is responsible for exchanging command and error data with the 2200. It is also responsible for decoding commands and transferring control to the appropriate High Level Command.

j. The DMA Module - The routines for programming the AM9517 and handling its interrupts are located in this module.

k. The Status Port Module - This module controls access to the status port of the DPU

l. The Drive Table Module - The Drive Table Module holds the tables that define the physical characteristics of each of the drives in the system.

m. The DPU Ready/Busy Selection Module - This module provides a way of setting the boards status as seen by the 2200 to READY or BUSY.

The major Service Modules are:

a. The 2010 Command Module - The WD2010 Command Set is used to request to, and to control the operation of, the WD2010. It supports logical operations such as Seek, Read, Write, Format Track, and other WD2010 specific functions. The services provided by the Task-File Module are used.

b. The Floppy Controller Module - With the exception of the memory interface, this module is being taken intact from the 2275 microcode and is not described in this document

c. The Power-On Module - This module initializes the hardware and the software data structures. The CTC interrupt vectors and the memory parity error handler are also defined here.

d. The Winchester Command Module - This module provides the high level Winchester operations such as read, write and format.

e. The High Level Command Module - This module implements the commands available to the 2200. It uses the services provided by most of the other modules. The error codes and command protocols used by these commands are given in Appendix B of the 2275F/R Disk Unit Software Functional Specification.

3.1 PLATTER TABLE MODULE

a. Data Structure Description - The Platter-Table holds the information needed to: 1) validate platter addresses and sizes, 2) map logical platter/sector addresses to physical drive/cylinder/head/sector addresses, and 3) control the physical operation of the disk drives. Information on the state of the drive is also kept. Note that the term platter as used means a logical surface as seen by the 2200 (such as D24) and not a physical surface within a disk drive. The contents of the Platter Table depend on the switch settings, the contents of the Drive Tables, and any drive related defects that prevent operation.

b. Data Structure Format - The Platter-Table, located at memory location PlatTabl, has 32 entries, one for each platter that may be present within the disk unit sub-system. The first entry is used for the 2275F/R Floppy drive and the second through fifteenth entries are used for the fixed Winchester platters of the 2275F/R. The sixteenth entry is used by the 2275F/R's removable Winchester. The seventeenth entry is undefined, while the eighteenth through thirty first entries are used by the fixed Winchesters of the 2275S. The thirty second entry is used for the streaming cartridge tape drive. Each entry contains the following information: the number of heads per platter - 1, the number of cylinders per platter - 1, the starting cylinder for Reduced Write Current, the number of sectors per platter, the number of platters in the drive, the number of cylinders to be reserved for alternate sector mapping, and a map of unformattable platters. The format of each entry is specified in Table 3.2.

c. Associated Data Structures - One Drive-Table is assembled into the ROM along with the microcode for each of the different types of disk drives supported by the 2275F/R and 2275S. Each drive table contains most of the same information as the platter entries and uses the same general format. There are tables for the 360Kb floppy drive, 10Mb removable, 10Mb fixed, 20Mb fixed, 33Mb fixed, and 67Mb fixed Winchesters. The names of the various Drive-Tables are given in Table 3.1. PlatNum is a variable that is used to hold the current logical platter number. PlatNum has a range of 0 to 31.

Table Name	Drive Type
FlopTab	360Kb floppy
RWinch10	10Mb removable Winchester
FWinch10	10Mb fixed Winchester
FWinch20	20Mb fixed Winchester
FWinch32	33Mb fixed Winchester
FWinch64	67Mb fixed Winchester

TABLE 3.1
DRIVE-TABLE NAMES

Byte	Name	Use
0	PlatVal	Platter Validity Flag. 00 = invalid platter, FFh = defective platter, FEh = unformattable platter. If equal to the number of heads per platter, then the platter is valid.
1	HeadPlat	Number of heads per platter.
2	StrtHead	Number of first head for this platter.
3	RWCCyl	Starting cylinder for Reduced Write Current divided by 4 for the WD2010.
4	NAltCyl	Number of cylinders reserved for alternate sectoring.
5	DrvNum	Number of drive containing this platter. 0 = floppy, 1 = F/R fixed Winchester, 2 = F/R removable Winchester, 3 = 1st slave drive, 4 = 2nd slave drive, and 5 = streaming tape drive.
6	NumSect	High byte of number of sectors per platter, the low byte is an implied 00h.
7		Reserved.

TABLE 3.2
PLATTER TABLE ENTRY DEFINITION

d. Access Routines

d.1 InitPTE   Initialize Platter-Table entry

Imports: Drive number in A

Exports: The updated Platter-Table entry, CurntDrv is modified.

Function: Copy the Drive-Table pointed to by the contents of HL into the Platter-Table for the drive indicated by CurntDrv. Set the validity flag based on the code passed by the Power On diagnostics, assume that the platter is formattable.

d.2 DrFrmtB   Mark Drive as Unformattable or Formattable

Imports: platter number in (IX) and a flag in A (00h = formattable, ffh = unformattable).

Exports: The Platter-Table with the bit corresponding to the logical platter in the bit map set as indicated by the flag.

Function: To record formattability of the indicated drive in the Platter-Table entry bit map.

d.3 ChkPlat Check Platter Address

Imports: The platter address in (IX)

Exports: The carry will be set in case of an error and an error code returned in the A register.
Error codes or platter number in (IX+1): 00h - illegal platter, FFh = drive defective, FEh = platter unformattable.

Function: To verify that the platter address exists and that the drive is not defective in some way. If the drive is valid, then initialize PlatNum.

d.4 CalWSect Calculate Winchester Sector Address

Imports: A = platter number, the logical sector address pointed to by IY (3 byte ordered high to low).

Exports: The physical cylinder, head, and sector in NewCylH, NewCyl, NewHead, and NewSect.

Function: Compute the physical address for the indicated sector, taking Alternate Sector Mapping cylinders and Alternate Sector Map Reading into account. Set the carry if the address is illegal.

d.5 ChkFSect Check and Calculate Floppy Sector Address

Imports: The logical 3 byte sector address pointed to by IY.

Exports: The physical cylinder, head, and sector in NewCylH, NewCyl, NewHead, and NewSect. Carry set if sector address is illegal.

Function: Verify the logical sector address. Compute the physical address for the indicated sector. Set the carry if the address is illegal.

d.6 ChkSect Check Sector Address

Imports: The logical sector address pointed to by IY, platter number in A

Exports: Carry set if sector address is illegal.

Function: Verify the logical sector address as being legal.

3.2 ALTERNATE SECTORING MODULE

a. Data Structure Description - The Alternate Sector Map, referenced using AltSectM along with CurntDrv, is used to keep track of alternate sectors assigned to sectors that have been relocated due to format failures. All logical platters within a physical disk share the same map. This is done to provide more usable space by reducing repetition. The map is located starting at cylinder 0, head 0, sector 0 and will use as many consecutive sectors as is needed. It is implied that physical sector 0 must be free of defects as it cannot be relocated. This should not cause any problems as Wang's Winchester Disk Drive Specifications state that cylinder 0 is to be defect free. The length of the map is dependant on the number of cylinders reserved for alternate sectoring but limited to a maximum of eight sectors. This allows up to 681 sectors to be relocated. The actual length of the map is determined by:

sectors in map = [# heads per platter x # platters x 32 x
  # cylinders for alternate sectoring x 3] / 256

rounded up to the nearest whole integer or to 8 which ever is smaller; where 32 is the number of sectors per track and 3 is the number of words per map entry. The number of entries that can be placed in the map is equal to the number of words in the map minus 5 (there are 5 words of overhead), divided by 3 or:

# of map entries = [(# sectors in map x 256) - 5] / 3

The cylinders reserved for alternate sectoring may be read and or written by the 2200 by issuing the Enable Alternate Cylinders command.

b. Data Structure Format - Each relocated sector has one 3 word entry in the map. The entries contains information specifying which sectors were relocated and what head they reside on. What sector is assigned as an alternate is implied by their position in the map; the first entry is mapped to the first alternate sector, the second entry is mapped to the second alternate sector and so forth. Any defects found in the area reserved for alternate sectoring will be mapped to themselves to prevent double relocations. Since holes may appear in the map due to the reformatting of platters, bit 7 of word 2 of each entry also functions as a flag. If this bit is equal to 1 then the entry is a hole and can be reused. The format of each entry is shown in Table 3.3.

The first three words of the map are used to hold an identifier that is used to distinguish real maps from random data. The contents of these three words are the ASCII characters "WLI". The fourth word is a bit map of the formattability of the logical platters contained in this drive. The bit will be set if the corresponding platter is unformattable. The next two words hold information describing the map itself. The first of these, locate at offset RoomLeft, is a flag that indicates if the map is full; a value of 00h means that there is room in the map, and value of FFh means that the map is full. The sixth word, offset specified by AltStart, gives the sector number (cylinder and track are implied to be 0) of the first alternate sector. The format of the map is shown below in Table 3.4.

The location of any of the alternate sector maps in memory can be found using the following formula.

((CurntDrv-1) * 8 * 256) + 2000h

WORD	BITS	USE
0	7 - 0	High byte of sector number of sector to relocate
1	7 - 0	Low byte of sector number of sector to relocate
2	7	Entry in-use flag
2	6	Not Used
2	5 - 0	Head number of the sector to relocate

TABLE 3.3
ALTERNATE SECTOR MAP ENTRY FORMAT

WORD(S)	NAME	DESCRIPTION
0 - 2	Identifier	the ASCII characters "WLI"
3	BitMap	bit map of logical platter formattability
4	RoomLeft	flag indicating if map is full
5	AltStart	sector address of first alternate sector
6 - 8	map entry	the first map entry
9 - 11	map entry	the second map entry
12 - ...	map entries	the rest of the map

TABLE 3.4
ALTERNATE SECTOR MAP FORMAT

c. Associated Data Structures - None.

d. Access Routines

d.1 LoadMap — Load Alternate Sector Map

Imports: The drive number in CurntDrv.

Exports: The map in memory, NextAltL and NextAltH initialized. Carry set with the A register set to FFh if the read operation fails, carry set and the A register set to 00h if the map is not found.

Function: To load the Alternate Sector Map into memory. The carry will be set and A set to FFh if a read error occurs. If the first three characters of the first sector read are not "WLI", then it will be assumed that no map exists and the carry will be set with A being set to 00h.

d.2 <u>InitMap</u> Initialize Alternate Sector Map

Imports: CurntDrv containing the drive number.

Exports: The Alternate Sector Map in cache with the identifier, RoomLeft and AltStart initialized.

Function: To initialize the in-use flag of all entries of the Alternate Sector Map to 1, to place the identifier "WLI" in the first three bytes of the map, to initialize RoomLeft to 00h, and to place the sector address of the first alternate sector into AltStart.

d.3 <u>AddToMap</u> Add entry to Alternate Sector Map

Imports: The sector number (in LowSect and MidSect) and the platter number (in PlatNum) of sector to be relocated. The drive number in CurntDrv. It is assumed that a valid map is already in memory or that InitMap has been called to set up one.

Exports: The updated Alternate Sector Map with an alternate sector assigned. The carry is if the entry could not be added to the map.

Function: The sector to be relocated is assigned to the next available alternate sector using logical sector number and physical head number as the map entries. The in-use bit is also cleared. In the case where the sector to be relocated is within the area reserved for alternate sectoring, the sector will be relocated to itself. Physical location sector 0, head 0 can never be relocated, trying to do so will cause an error. If for any reason the sector to be relocated could not be added to the map, the carry will be set and no other action taken. If the sector is already in the map no action shall be taken.

d.4 <u>SrchMap</u> Search Alternate Sector Map

Imports: The sector number specified by LowSect and MidSect, the head number (NewHead) and the drive number (CurntDrv).

Exports: LowSect and MidSect contain the location of the alternate sector. The carry is set if no alternate sector could be found.

Function: Make sure that the Alternate Sector Map exists in memory, read the map in to memory if it does not. Search the Alternate Sector Map for the sector. If the sector is found then change LowSect and MidSect to point to the alternate sector. If the search, or the map read, fails then set the carry on return.

d.5 SaveMap Save Alternate Sector Map

Imports: The drive number in CurntDrv. It is assumed that a valid map exists in memory.

Exports: The map saved on disk, the carry is set on an error.

Function: To save the Alternate Sector Map starting at physical address cylinder 0, head 0, sector 0. If the write of the map fails, then the carry will be set on return.

d.6 ClearMap Clear Map of Platter Entries

Imports: The drive number in CurntDrv, the head number in NewHead.

Exports: The map with all entries from the specified platter removed from memory. NextAltL and NextAltH are pointed to the first possibly available entry in the map.

Function: To remove all previous entries for the specified platter from the map, to point NextAltL and NextAltH to the first hole in the map, and to set RoomLeft to its proper state depending on whether or not a hole was found.

d.7 GBitMap Get Bit Map

Imports: The drive number in CurntDrv.

Exports: The bit map of platter formattability for the indicated drive in the A register.

Function: Insure that a valid map exists in memory; read it from the disk if it does not and return with the carry set if the read fails. Copy the bit map for the specified drive into the A register.

d.8 WBitMap Write Bit Map

Imports: The new bit map of platter formattability in the A register.

Exports: Carry set on error.

Function: Verify that the alternate sector map exists in memory. Set carry and return if it does not, else replace the bit map in the alternate sector map with the contents of the A register.

3.3 CACHE MODULE

Cache Software Specification
May 12, 1986   A. Gruber

This document will describe the cache module of the 2275 F/R disk storage unit. The purpose of the cache is to speed up disk operations in two ways. The first is to keep copies of frequently used sectors in memory rather than on disk. The second is to try to guess what sector the cpu will ask for next and prefetch that into memory. This will take the form of fetching the sectors following the one requested. This saves rotational latency time on the following access.

The cache operates on fixed size (256 byte) buffers. These buffers are allocated and reclaimed by the storage allocator portion of the cache module. While any number of buffers may be requested, the storage allocator cannot guarantee that the buffers will be in contiguous memory. This implies that disk operations must be done in single sector chunks with any data pointers being relocated to the next buffer in between operations. In situations where this is not possible (such as 512 byte floppy sectors), the data must first be gathered into a suitably sized chunk of contiguous memory prior to the actual data transfer.

The Buffer Entry Data Structure

To manage the 256 byte buffers, the storage allocator keeps in low memory a pool of buffer entries. These 11 byte data structures are used to keep track of the buffers.

The buffer entries are used in three different situations:

1) an entry for a cached sector
2) an entry for general purpose allocated storage
3) as entry for unallocated storage

The use of the fields of the structure in each of these situations will be described

Buffer Bank (Byte 1) - holds the four bit bank number of the allocated buffer for all three situations. In addition in the case of cache storage, if Bit 7 is set, the entry is part of the hash table. Bits 4 - 6 are used to keep track of how many previous contiguous sectors have been used. This information is used by the prefetch algorithm to decide how many sectors to read from disk.

Bit 6	Bit 5	Bit 4	
0	0	0	The sector has not been used
0	0	1	The sector has been used. No previous sect. used
0	1	0	The sector and one previous sector was used
0	1	1	The sector and two previous sectors were used
1	0	0	The sector and three previous sectors were used
1	0	1	The sector and four previous sectors were used
1	1	0	The sector and five previous sectors were used
1	1	1	The sector and more than five previous sectors were used

In non-cache situations these bits are cleared.

Buffer Address High (Byte 2) - holds the pointer to the high byte of the address within the bank of the allocated buffer for all three situations. The low byte is assumed to be 0.

Next Entry Address (Byte 3 low, Byte 4 high) - holds a pointer to the next entry. A pointer of 0 (or any number less than 32K) indicates there is no next entry. In a cache this points to the next entry representing a sector that hashed to the same value. In general allocated storage this points to the next entry for a multiple buffer request. In unallocated storage this points to the next free entry.

Platter No. (Byte 5) - used only in a cache entry, this holds the logical platter number of the cached sector. If Bit 7 is set, the entry is invalid.

Sector No. (Byte 6 high, Byte 7 low) - used only in a cache entry, this holds the logical sector number of the cached sector.

More Recent Ptr (Byte 8 low, Byte 9 high) - used only in a cache entry, this points to the first entry that has had a cache hit more recently than the current entry. Used to implement a least recently used cache replacement algorithm.

Less Recent Ptr. (Byte 10 low, Byte 11 high) - used only in a cache entry, this points to the first entry that has had a cache hit less recently than the current entry. Used to implement a least recently used cache replacement algorithm.

Other Data Structures

Hash Table - a contiguous section of memory containing 256 buffer entries. These entries are sorted in order of the lowest 8 bits of their logical sector number. This allows a simple index mechanism to find the cache buffer corresponding to a given logical sector.

Hash List - a list of buffer entries in the cache that hash to the same value. The first entry in this list is contained in the hash table.

Free Buffer List - a linked list of buffer entries that are not currently being used

Free Entry List - a linked list of buffer entries that are not currently being used. This differs from the Free Buffer List in that the entries have no valid pointers to buffers. This occurs when the buffer is contained in a hash table entry instead.

Basic Algorithms

Cache Lookup - The low byte of the sector number will be used to index into a hash table of cached sectors. If no entry is present this signifies a cache miss. If an entry is present the platter number and high byte of the sector is checked. If these match, this signifies a cache hit. Otherwise the next entry pointer is followed. Reaching the end of the list with no hit constitutes a cache miss. Note that logical rather than physical addresses are cached. This allows faster look up as in the case of cache hits, the logical to physical translation need not be done. However, if anything strange goes on with the addressing (such as a change in the logical to physical mapping as happens when alternate sectors are enabled) the cache will be fooled and should be turned off.

Cache Replacement - New buffers are allocated from the free pool of buffers. If this pool is empty then the buffers are taken out of the cache on a modified least recently used basis. Cache entries are ordered according to time of last use. When a cache entry is used it moves up the list -- either to the head of the list if it has been used at least once before or to the middle of the list if it has never been used. This allows frequently used sectors to "fall" to the bottom slower than sectors that are only used once (say for program load). Newly read sectors from the disk are also put into the middle of the list

Cache Prefetch - On detecting a cache miss, the data will be read from disk into cache. The sectors following the requested sector might also be read in. This is determined by the following rules:

1. If the previous sector is not present in the cache or has not been used or if it has been used but no sector previous to that has been used, then read one extra sector or to the end of the track, whichever is fewer.
2. If the previous sector is present and has been used and less than four sectors previous to that have been used, then read three extra sectors or to the end of the track, whichever is fewer.
3. If the previous sector is present and has been used and four or more sectors previous to that have been used, then read to the end of the track.

General Storage Allocation - buffer allocation for purposes other than cache buffers is handled in the same manner as cache allocation. That is if there are no more free buffers, buffers are taken from the cache on a least recently used basis. Storage allocated for general storage is reclaimed only when it is explicitly freed by a CFree call.

Write Strategy - No writes are cached. Sectors that are already in cache are invalidated if written to.

Description of Routines

CAlloc

Imports:
A - Desired number of buffers
Exports:
A - Number of buffers provided
HL - Points to buffer entry list
Function:
Allocates dynamic storage either for cache or other purposes

CFree

Imports:
HL - Points to buffer entry list
Exports:
None
Function:
Explicit reclaimation of storage. Note that the list should end with a null pointer.

ChkCache

Imports:
A - Platter Number
HL - Sector Number (bytes swapped)
Exports:
Carry Set if not found
HL - Cache entry if found

Function:
Does cache lookup to see if a given sector is cached.

CHitUpdate

Imports:
A - Platter Number
HL - Sector Number (bytes swapped)
DE - Pointer to Cache Entry
Exports:
The Cache Entry has its use status updated. Its position in the least recently used and hash lists is changed.
Function:
The entry for the previous sector is looked up and used to modify the use status of the current entry. The current entry is also moved to the top or middle of the least recently used list. It is also moved up in the hash list.

CMissUpdate

Imports:

A - Platter Number
HL - Sector Number
DE - Pointer to Buffer List
IX + MaxSCOff - holds number of valid sectors in Buffer List

Exports:

The cache is updated and the read sectors added.

Function:

The newly read sectors are put into the cache. The first sector is marked as being used once, while the following ones are marked unused. The unused buffers are reclaimed. Prefetched sectors are checked to see if they are already in cache.

CInvalSec

Imports:

A - Platter Number
HL - Sector Number (bytes swapped)

Exports:

The cache entry for the specified sector is removed

Function:

Invalidates cache entry for the sector (if present) and reclaims the storage.

CInvalPlat

Imports:

A - Platter Number

Exports:

All cache entries for the specified platter are removed.

Function:

Invalidates the whole platter and reclaims the storage. This functions by actually scanning all the cache entries. Hence it is rather slow.

CacheStrat

Imports:

A - Platter Number
HL - Sector Number (bytes swapped)

Exports:

A - Maximum Number of Sectors to Read

Function

Implements prefetch strategy. The previous entry is read from cache to determine what to do. If the returned value is greater than the number of sectors left on the track, then just read to end of track.

GetCPtr

Imports:

HL - Pointer to Buffer Entry

Exports:

DE - Pointer to Buffer
The selected memory bank is changed

Function:

Extracts buffer pointer from buffer entry. The memory bank containing the buffer is swapped in.

CLink

Imports:
HL - Pointer to Buffer Entry
DE - Pointer to Buffer Entry
Exports:
None.
Function:
Buffer entry in HL is linked to buffer entry in DE (i.e. DE is head of list and HL is tail of list).

CInit

Imports:
None.
Exports:
Cache and storage allocator are initialized. All available storage is reclaimed.
Function:
Initializes the cache and storage allocator.

TurnCOff

Imports:
None.
Exports:
The variable CacheOn is set false.
Function:
Turns off cache but leaves the data that is currently cached valid. The storage allocator is left functioning. Data will simply always be fetched from the disk rather than from the cache.

TurnCOn

Imports:
None.
Exports:
The variable CacheOn is made True.
Function:
The cache is turned on. The data currently in the cache is assumed valid. If data has been written to the user area of the disk while the cache has been turned off, this assumption is not valid and the cache should be invalidated before turning it back on.

3.4 Status Message Buffer Module

a. Data Structure Description - The Status Message Buffer is used as a work space in which to build status messages to send to the 2200 CPU in response to the $GIO Status Request command. The message specifies the Disk Unit Type, Protocol Level, PROM Revision Number, Error Counts and Defect Byte for the requested platter.

b. Data Structure Format - The format of the Status Message Buffer (StatMess) is the same as the format specified for the status message itself in the 2275F/R Disk Unit Software Functional Specification. This document should be referenced for further information.

c. Associated Data Structures - The Platter Table and the Error Counts are used to build the status message. The Disk Unit Type is determined by the switch settings, the Protocol Level and the Prom Revision Level are constants.

d. Access Routines

d.1 MakeStat Build Status Message

Imports: The logical platter number in PlatNum, NumSect, Error Counts, switch settings, Protocol, PromRevL, and PromRevH.

Exports: A Status Message for the requested platter.

Function: To build a status message in the format specified for the requested platter from the imported data.

3.5 Error Count Module

a. Data Structure Description - The Error Count Module keeps track of the errors experienced by the disk drives on a per platter basis in the structure ErrCount. The following type of errors are counted: error corrected by ECC, error not correctable by ECC, errors corrected by retry, and errors not correctable by retry.

b. Data Structure Format - There is one 8 word entry for each possible platter in the system. Each word is stored low byte first, high byte second. Each of the possible 32 entries has the format specified in Table 3.8.

c. Associated Data Structures - PlatNum.

ERROR TYPE	LENGTH
corrected by ECC	2
uncorrectable by ECC	2
corrected by retry	2
uncorrectable by retry	2

TABLE 3.5
ERROR COUNT ENTRY FORMAT

d. Access Routines

d.1 IncECnt   Increment Error Count
Imports: Logical platter number in location PlatNum, the error count select value in the A register. The possible values of the error count select value are:
0 - corrected by ECC
1 - uncorrectable by ECC
2 - corrected by retry
3 - uncorrectable by retry
Exports: Updated error counts, carry set if error count select value out of bounds.
Function: Increment the selected error count by 1. If the value in the A register is less than 0 or greater than 3 then set the carry and return taking no other action.

d.2 ClrECnt   Clear Error Count
Imports: Logical platter number PlatNum.
Exports: Updated error counts.
Function: Set the error counts for the specified platter to 0.

d.3 ClrAECnt   Clear All Error Counts
Imports: None.
Exports: Updated error counts.
Function: Set all error counts for all platters to 0.

d.3 GetECnt   Get Error Count

Imports: Logical platter number in location PlatNum and the location to which to copy the error counts in HL.

Exports: Error counts copied.

Function: Copy the error counts for the specified platter to the memory location pointed to by HL.

3.6 MEMORY BANK MODULE

a. Data Structure Description - There is a total addressable space of 256KB of RAM memory. Since the Z80 only uses 16 bit addresses , this memory is being divided into 16 16KB banks. The bank with the lowest address is permanently mapped into memory locations 32Kb - 48Kb of the Z80's address space. Any of the other 15 banks can be mapped into the 16Kb of the Z80's address space by writing the number of the desired bank out to I/O port 70h. Whenever the Z80 uses an address above 48Kb, the contents of the port are used to select which bank of RAM is addressed
In addition to the RAM, there is 32Kb of PROM. This PROM resides in the lower 32Kb of the Z80's address space

b. Data Structure Format - The upper memory bank selection is done by writing the number of the desired bank out to a port.

c. Associated Data Structures - BSelPort is a constant which is equal to the address of the I/O port used in bank selection and BankSave is a variable that is used to store the number of the currently selected bank.

d. Access Routines

d.1 SelBank Select Memory Bank
Imports: A one byte hexadecimal number indicating the requested memory bank in the A register.
Exports: The requested bank of memory selected. Carry set on error.
Function: The indicated bank of memory is mapped into the high quarter of the Z80's address space by writing the contents of the A register out to the port specified by BSelPort. The number of the selected bank is also saved in the location SaveBank.

d.2 SelTBank Select Temporary Memory Bank
Imports: A one byte hexadecimal number indicating the requested memory bank in the A register.
Exports: The requested bank of memory selected. Carry set on error.
Function: The indicated bank of memory is mapped into the high quarter of the Z80's address space by writing the contents of the A register out to the port specified by BSelPort. The number of the bank is not saved.

d.3 RestBank Restore Memory Bank
Imports: Variable SaveBank.
Exports: The previously selected bank reselected.
Function: To load I/O port BSelPort with the contents of the memory location SaveBank.

3.7 SWITCH SETTING MODULE

a. Data Structure Description - There are two eight bit DIP switches on the 2275F/R PC board. These switches are used to define the 2275F/R system configuration by indicating what types and sizes of Winchesters are present. They are referenced as Switch1 and Switch2.

b. Data Structure Format - Each bank of switches is divided into two four bit groups, with one group being used to define the size of each of the Winchester drives. The various switch settings that are used are defined in the 2275F/R Disk Unit Software Functional Specification Appendix C.

c. Associated Data Structures - None.

d. Access Routines

d.1 GDrvType Get Drive Type

Imports: The number of the drive whose size is desired in A where: 0 = floppy, 1 = 2275F/R fixed Winchester, 2 = removable Winchester, 3 = 1st 2275S Winchester, 4 = 2nd 2275S Winchester.

Exports: A code defining the drive size in the A register. The codes used are the same as those used in the switch settings but will be right justified. The carry is set if the drive number is out of bounds.

Function: To read the setting of the appropriate switch and strip the desired drive identification code from it placing it in the A register.

3.8 DRIVE SELECTION MODULE

a. Data Structure Description - Winchester drive selection is done through a hardware port. Physically, because all drives used by Wang are hardwired to be drive number 1 in a daisy chained configuration, two separate daisy chains are used and the 2275F/R hardware bank switches them as needed. To select a drive, the logical number of the drive is set in the WD2010's SDH register, a non-zero value is then written to port WinchPrt. To deselect all Winchesters, a value of 00h is written to the port.

b. Data Structure Format - One eight bit port is used to select the Winchesters. The port address is given by the constant WinchPrt.

c. Associated Data Structures - None.

d. Access Routines

d.1 WinchSel   Winchester Selection

Imports: FFh,to select the current Winchester, or 00h, to deselect all Winchesters, in the B register. The SDH register of the 2010 already set up.

Exports: The current drive selected, or all Winchesters deselected.

Function: To select the current drive, or to deselect all Winchester drives by writing the contents of the B register to the I/O port WinchPrt.

3.9 2200 INTERFACE MODULE

a. Data Structure Description - The 2200 Interface Module's major responsibility is to receive and verify commands from the 2200. This module has the ability to send messages to the 2200 as well.

b. Data Structure Format - The primary data structure used in this module is a five word section of memory used to receive commands from the 2200. Each word in the section has its own label and they are (in order from first to last); Comm2200, Platter, DumSect, MidSect, and LowSect. Comm2200 holds the platter address and the command byte. Platter holds the command byte for extended commands and is undefined for normal commands. DumSect is a meaningless dummy variable used to receive the high byte of the sector address which is always guaranteed to be zero since the 2200 only uses two byte addresses anyway. MidSect receives the middle byte of the sector address and LowSect receives the low byte.

c. Associated Data Structures - The read and write caches. If a mulitsector write is interrupted with a non-write command, it is the responsibility of this module to insure that the contents of the write cache be written to disk and the appropriate read caches flushed, before the new command is processed.

d. Access Routines

d.1 Request — Receive a Command from the 2200
Imports: A Re-init has been received from the 2200.
Exports: The platter address and command byte if its an extended command.
Function: To receive the first two words from the 2200 and if an extended command has been specified, transfer control to the extended command decoder, otherwise transfer control to the normal command decoder.

d.2 Extended — Extended Command Decoder
Imports: Comm2200 and Platter
Exports: An error message sent to the 2200 if needed.
Function: If its an illegal extended command an error is returned to the 2200. Otherwise control is transferred to the appropriate High Level Command routine.

d.3 Decode — Normal Command Decoder
Imports: Comm2200
Exports: An error message sent to the 2200 if needed.
Function: If its an illegal command an error is returned to the 2200. Otherwise control is transferred to the appropriate High Level Command routine.

d.4 Illegal — Notify the 2200 of an Error

Imports: The word containing the error code in the E register.
Exports: The error code sent to the 2200.
Function: To send the error code to the 2200.

d.5 MSWPrtct — Multisector Write Protect

Imports: Comm2200 and Platter.
Exports: The contents of the write cache written to disk if needed.
Function: If the multisector write in progress flag is on; and the currently requested operation is not an End Multisector Write nor a write, or if the requested operation is a write but the sector to be written is not sequential with the last sector in the write cache (current sector number <> WCacheSM/WCacheSL + WCacheN + 1), then call EndMSW.

d.6 GetByte — Get A Byte from the 2200

Imports: An OBS strobe interrupt was received from the 2200.
Exports: The byte received from the 2200 in the A register.
Function: Read a byte from the port In2200 into the A register.

d.7 SendByte — Send a Byte to the 2200

Imports: The E register contains the byte to send.
Exports: The byte is sent to the 2200.
Function: Wait for CPB (2200 ready) from the 2200, then set the DPU status to busy and write the contents of the E register to the port Out2200.

d.8 LrcByte — Calculate LRC

Imports: The sector data, pointed to by DE.
Exports: The LRC byte in location Lrc.
Function: To compute the LRC by doing a binary add without carry of all 256 bytes of sector data, and to store the results in location Lrc.

d.9 GetData — Receive 257 Bytes from the 2200

Imports: The 9517 programmed to receive 257 bytes into the location indicated by the DE register.
Exports: The data in memory, or the 2200 notified of a transmission error.
Function: Receive 256 bytes of data, and calculate its LRC. Receive a 1 byte LRC from the 2200, compare the two LRCs and send a 04h to the 2200 if they don't match.

d.10 SendData — Send 257 bytes to the 2200

Imports: The DMA chip programmed for a 257 byte transfer to the 2200. The DE register contains the starting address for the transfer.
Exports: 256 bytes of data and a 1 byte LRC sent to the 2200.
Function: Calculate the LRC on the data to be sent. Send the data and the LRC to the 2200.

d.11 Idle — Wait for a request from the 2200

Imports: None:

Exports: Stack pointer reset.

Function: The stack pointer reset to the beginning of the stack space. The floppy motor timer enabled to turn off floppy if it times out. The DPU Status set to READY.

d.12 Reinit — Receive a Reinit from the 2200

Imports: Reinit received from the 2200.

Exports: DPU id sent to the 2200, retries enabled.

Function: Reset the floppy motor timer and disable floppy interrupts. Send a D0h to the 2200. Set the retry enable flag (RetryAll) to hex FFh, clear the CTCs and transfer control to Request. Disable retries and alternate cylinder accessing.

d.13 I95ErrT — I95 Error Trap

Imports: None.

Exports: An I95 error code is sent to the 2200.

Function: A 01h is sent to the 2200 to signal that the drive is defective.

d.14 GetBytes — Get 1 or more bytes from the 2200

Imports: A register non-zero = echo bytes to the 2200, A = 0 means don't echo nothing. B = number of bytes to receive, HL = location to read bytes into.

Exports: Bytes received.

Function: Receive the requested number of bytes from the 2200 and store them in the indicated location. Echo bytes if requested to do so.

3.10 DMA MODULE

a. Data Structure Description - NOTE: This module is being borrowed intact from the microcode for the 2275. The addresses of the various ports may change with the new hardware. The AM9517 is normally programmed by writing three locations. First the starting address of the transfer is specified, then the number of bytes to transfer, and finally the transfer type.

b. Data Structure Format - The format of the data structure used by the 9517 is implied in the routine and not externally visible to the user.

c. Associated Data Structures - None.

d. Access Routines

d.1 LoadDma Program DMA Controller

Imports: The A register contains the data to be written into the Mode register, the C register contains the channel port address, DE contains the starting address of the transfer, and HL contains the number of bytes to transfer.

Exports: AM9517 programmed as specified.

Function: Write the contents of the A register to the port specified by DmaMode. Write the E and then the D register to the port indicated by the contents of the C register. Increment the C register and write the contents of the L and H registers to that port.

d.2 DmaEop DMA Interrupt Handler

Imports: None.

Exports: The status of the DPU is set to BUSY.

Function: Set the status of the DPU to busy for the 2200.

3.11 STATUS PORT MODULE

a. Data Structure Description - The status port is an 8 bit port used to hold the status of various events. CPU busy, DN3, floppy drive index mark, floppy drive media disturbed, floppy drive write protect, removable Winchester media disturbed, and removable Winchester write protect are all represented in the port.

b. Data Structure Format - The status port is pointed to by the constant StatPort. The assignment of logical events to individual bits within the port, and meanings of the values of these bits, will not be known until the hardware design is further advanced.

c. Associated Data Structures - None.

d. Access Routines

d.1 RWrtPrtc Removable Winchester Write Protect

Imports: None.

Exports: The zero flag cleared if the removable Winchester Write Protect bit is true, and it is set if it is false (equal to 0).

Function: Read the status port bit 4 and set the flag to match the setting of the removable Winchester Write Protect bit.

d.2 RWinDoor Removable Winchester Door Open

Imports: None.

Exports: The zero flag cleared if the removable Winchester Media Disturbed bit is true, and the flag is set if it is false (equal to 0). The alternate sector maps for the removable Winchester invalidated if the media has been disturbed.

Function: Read the status port bit 3 and set the flag based on the setting of the removable Winchester Media Disturbed bit. Invalidate the alternate sector maps if need be.

d.3 DN3Test 2275S Selected

Imports: None.

Exports: The zero flag set if the DN3 false, and it cleared if it is true (equal to 1).

Function: Read the status port bit 6 and set the zero flag to match the complemented setting of the Dn3 bit.

d.4 FlpDoor Floppy Door Open

Imports: None.

Exports: The zero flag cleared if the Floppy Drive Media Disturbed bit is true, and the zero flag is set if it is false (equal to 0).

Function: Read the status port bit 1 and set the zero to match the setting of the Floppy Drive Media Disturbed bit.

d.5 RB2200 2200 Ready/Busy

Imports: None.

Exports: The zero flag set if the 2200 is ready.

Function: Read the status port bit 5 and set the zero flag to match the 2200 R/B bit (Ready is active low).

3.12 DRIVE TABLE MODULE

a. Data Structure Description - There are 6 drive tables: one each for the 360Kb floppy drive, the 10MB removable Winchester, the 10Mb Winchester, the 20Mb Winchester, the 32Mb Winchester, and the 64Mb Winchester. These tables are used to hold the information that physically describes the drives and is used to control their operations.

b. Data Structure Format - Each table is eight bytes long and holds the following information as shown below: the number of heads per platter, the number of cylinders per platter, the number of the cylinder at which to enable Reduced-Write-Current, the high byte of the number of sectors per platter, and the number of platters per drive. The drive table contents are given in Appendix F, Drive Tables.

BYTE	USE
0	# of heads/platter
1	low byte of # of cylinders/platter - 1
2	high byte of # of cylinders/platter - 1
3	RWC cylinder address/4
4	# of sectors/platter, high byte
5	# of logical platters/drive
6	# of cylinders reserved for alternate sectoring

TABLE 3.6
DRIVE TABLE FORMAT

c. Associated Data Structures - DrvTabls points to the location of the drive tables in PROM.

d. Access Routines - None.

3.13 DPU READY/BUSY SELECTION MODULE

a. Data Structure Description - The Ready/Busy port is used to set the status of the DPU as seen by the 2200. Writing a non-zero value to this port sets the board to READY, writing the port with 00h sets the board to BUSY.

b. Data Structure Format - The constant BRdyPort is the address of the port used to set READY/BUSY. The port is an 8 bit wide I/O port.

c. Associated Data Structures - None.

d. Access Routines

d.1 SetReady Set the board's status to READY
Imports: None.
Exports: The board's status set to READY.
Function: Write FFh to port BRdyPort.

d.1 SetBusy Set the board's status to BUSY
Imports: None.
Exports: The board's status set to BUSY.
Function: Write 00h to port BRdyPort.

3.14 2010 COMMAND MODULE

a. Services Provided - The 2010 Command Module provides the system a way of issuing commands to the WD2010. Routines are provided that implement the functions read sector(s), write sector(s), format track, restore Winchester, compute ECC correction, set ECC correction span, and initialize the WD2010.

b. Modules Accessed - None.

c. Access Routines

c.1 ReadSec Read Sector(s)

Imports: The Winchester drive and memory bank selected, the starting memory address for the data in HL, the starting physical location in NewSect, NewHead, NewCyl, and NewCylH, and the number of sectors to transfer in MaxSectC.

Exports: The requested sectors transferred into memory, MaxSectC is modified to be equal to the number of sectors actually read. In the event of an error, the carry is set and the contents of the WD2010 Error Register are returned in the A register.

Function: To transfer the requested sector(s) from disk into memory. The transfer is halted on any error.

c.2 WriteSec Write Sector(s)

Imports: The Winchester drive and memory bank selected, the starting memory address for the data in HL, the starting physical location in NewSect, NewHead, NewCyl, and NewCylH, and the number of sectors to transfer in MaxSectC.

Exports: The requested sector(s) transferred to disk, MaxSectC is modified to be equal to the number of sectors actually transferred. In the event of an error, the carry is set and the contents of the WD2010 Error Register are returned in the A register.

Function: To transfer the requested sector(s) from disk into memory. The transfer is halted on any error.

c.3 FrmtTrk Format Track

Imports: The proper Winchester selected, a pointer to the sector buffer containing the format data (as explained in the Western Digital handbook) in HL, the cylinder to format in NewCyl and NewCylH, and the head to format in NewHead.

Exports: The requested track formatted. The carry is set on error.

Function: Format the requested track using the data in the sector buffer. If, after three attempts, the format operation fails, set the carry and return.

c.4 SeekCyl — Seek to Cylinder

Imports: The proper Winchester selected, the cylinder to seek to specified in NewCyl and NewCylH

Exports: The Seek In Progress Flag, SeekProg, set to FFh.

Function: Set the Seek In Progress flag and start a seek to the requested cylinder using a step rate of 1.6 usec.

c.5 CompCorr — Compute ECC syndrome Correction Bytes

Imports: No command has been issued to the 2010 since the read command that experienced the ECC error. The data read copied to location EccBlock and the memory counters pointing to the block immediately following.

Exports: Four syndrome bytes followed by a two byte offset specifying the location of the error and a three byte error correcting pattern to be XOR'ed to the specified error location. The carry is set if correction fails.

Function: Instruct the WD2010 to compute the correction pattern for the defective data. If the error is uncorrectable, set the carry and return.

c.6 RstWinch — Restore Winchester

Imports: The proper drive selected.

Exports: The drive restored with the read/write heads located over physical cylinder 0, carry set on error.

Function: Position the heads over cylinder 0 without knowing their current position.

c.7 SetCorr — Set ECC Correction Span

Imports: None.

Exports: 5 bit correction span selected.

Function: To select 5 bit ECC correction span.

c.8 Int2010 — WD2010 Interrupt Request Routine

Imports: None.

Exports: The WD2010's status register contents in Stat2010.

Function: Read the 2010's status register into Stat2010 and execute an IRET.

PORT ADDRESS	REGISTER READ ONLY	REGISTER WRITE ONLY
10	BUS TRI-STATED	
11	Error Register	Write Precomp. Cylinder
12	Sector Count Register	
13	Sector Number Register	
14	Cylinder Low Register	
15	Cylinder High Register	
16	SDH Register	
17	Status Register	Command Register

TABLE 3.7
WD2010 TASK FILE REGISTERS

HEX CODE	COMMAND
00	Select 5-bit ECC correction width
01	Select 11-bit ECC correction width
08	Compute correction
1F	Restore with 1.6usec step pulse rate
21	Single sector read
2D	Multiple sector read
31	Single sector write
35	Multiple sector write
41	Scan id
50	Format track
7F	Seek with 1.6usec step pulse rate

TABLE 3.8
WD2010 COMMANDS

3.15 POWER-ON MODULE

a. Services Provided - The Power-On Module initializes all hardware components such as the CTCs, the 9517 DMA controller, the WD2010, and memory. It also initializes all of the data structures used by the firmware. The parity error handler is included in this module, to allow for the fact the memory may not have the correct parity when power is first applied.

b. Modules Accessed - The following modules are accessed to assist in the initialization of the hardware: the Task File module, the Memory Bank module, the Drive Selection module, the 2010 Command module, the DMA module, and the Floppy module. The software initialization makes use of the services of the Platter Table module, the Cache module, the Status Message Buffer module, the Error Count module, the Memory Bank module, and the Switch Setting module.

c. Access Routines

c.1 CtcVctr CTC Interrupt Vector
Imports: None:
Exports: The address of the specified CTC interrupt handling routine.
Function: To provide a jump table for CTC interrupts.

c.2 ParErr Parity Error Handler
Imports: A memory parity error has occurred.
Exports:
Function: If the error has occurred while servicing a request from the 2200, send a 01h to the 2200 to flag the error. Reset the Relnit interrupt vector to point to an IRET (so as to attempt to insure that access cannot be made) and attempt to flash the lights on the floppy and fixed Winchester in the 2275F/R.

c.3 Start Hardware and Software Initialization
Imports: None.
Exports: System initialized.
Function: Using the services of other modules whenever possible, initialize all hardware components including memory and all software data structures.

3.16 HIGH LEVEL COMMAND MODULE

a. Services Provided - The High Level Command Module implements the commands available to the 2200. These commands are Read, Write, Compare, Format Platter, Start Multisector Write, End Multisector Write, Verify, Copy, $GIO Disable Retries, $GIO Enable Radial Head Alignments $GIO Enable Retries, $GIO Disable Radial Head Alignments, $GIO Enable Alternate Cylinders, $GIO Disable Alternate Cylinders, and $GIO Status Request. This module also contains two other routines that help to insure the integrity of the command protocol by allowing requests to defective disks and nonterminated multisector writes to be handled properly.

b. Modules Accessed - The services of the following modules are used by the High Level Command Module. The Platter Table Module, the Status Message Buffer Module, the Winchester Command Module, the Floppy Controller Module, and the DMA Module.

c. Access Routines

c.1 Read — Read Sector(s)

Imports: The logical platter and sector address from the 2200 in locations Comm2200, Platter, DumSect, MidSect, LowSect.

Exports: The requested sector data or an appropriate error message.

Function: To verify and acknowledge the platter and sector addresses. If the requested platter is a floppy then call ReadOpF, else call WRead to obtain the data. If no errors occurred send the requested sector to the 2200. In case of an error notify the 2200 and return. If it was an ECC or a CRC error allow the 2200 to receive the data if desired.

c.2 Write — Write Sector

Imports: The logical platter and sector address from the 2200 in locations Comm2200, Platter, DumSect, MidSect, LowSect.

Exports: The requested sector data written to disk or an appropriate error message.

Function: Verify and acknowledge the platter and sector addresses. If the requested platter is the floppy then call WriteOpF, else call WWrite. Notify the 2200 of any errors or of success, which ever is appropriate.

c.3 Compare — Compare

Imports: The logical platter and sector address from the 2200 in locations Comm2200, Platter, DumSect, MidSect, LowSect.

Exports: An appropriate error message.

Function: Verify and acknowledge the platter and sector addresses. If the platter is the floppy, call ReadOpF to obtain the desired sector data, else use WRead. Notify the 2200 as to the success of the read. Receive into the Write Cache 256 bytes of data and a one byte LCR from the 2200 Compare the data received from the 2200 against that read from the disk. If the data match send 00h to the 2200 as the acknowledgement, else send 08h to flag the error. Mark the Write cache contents as invalid.

c.4 Format Format Platter

Imports: The logical platter address and command from the 2200 in locations Comm2200 and Platter.

Exports: The requested platter formatted.

Function: Receive the go code from the 2200 saving it in location FormWhat. Verify the platter address. If the address is invalid, send an I93 error to the 2200. If the address a Winchester address then call WFormat, else call FormatFl. Send the completion code to the 2200 to signal the end of the operation.

c.5 StartMSW Start Multisector Write

Imports: The logical platter address and command from the 2200 in locations Comm2200 and Platter.

Exports: The multisector write in progress flag (Multi) set.

Function: Receive the go code from the 2200, set the multisector write in progress flag to FFh, and set WCacheN to 00h.

c.6 EndMSW End Multisector Write

Imports: The logical platter address and command from the 2200 in locations Comm2200 and Platter.

Exports: The multisector write in progress flag (Multi) cleared and the write cache contents written to disk.

Function: Receive the go code from the 2200. Write the contents of the write cache to disk by calling WriteCache, notify the 2200 of the results of the cache dump. Then set Multi and WCacheN to 00h.

c.7 Verify Verify Sectors

Imports: The logical platter and starting sector address from the 2200 in locations Comm2200, Platter, DumSect, MidSect, LowSect. The ending sector address in DumSectE, MidSectE, and LowSectE.

Exports: The logical address of the last successfully verified sector and an acknowledgement of the verify.

Function: Receive the starting sector address. Verify the starting sector and platter addresses. Receive and verify the ending sector address. Receive the go code from the 2200. Read all sectors from the starting address thru and including the ending address using the write/copy cache so that a track at a time can be read. On any read error, or after the ending sector has been read, send the logical address of the last successfully read sector to the 2200 along with the acknowledgement. Invalidate the contents of the Write cache.

c.8 RetryNo   Disable Retries
Imports: The retry enable flag (RetryAll).
Exports: The retry enable flag cleared.
Function: Receive the go code from the 2200 and set RetryAll equal to 1.

c.9 WCopy   Winchester Copy
Imports: The logical platter source address and command in locations Comm2200 and Platter.
Exports: The requested sectors copied to the destination specified.
Function: Receive the starting source sector address. Verify the starting sector and platter addresses of the source. Receive and verify the ending sector address of the source using DumSectE, MidSectE, and LowSectE for storage. Receive and verify the destination platter and starting sector addresses using locations PlatterD, DumSectD, MidSectD, and LowSectD for storage. Make sure that the full range of the sectors to be copied will fit on the destination platter. Receive the go code from the 2200. Copy the requested sectors to the destination using the write/copy cache so as to move a track at a time. If the source and destination platters are the same then move no more than starting sector destination minus starting sector source at a time. Send the acknowledgement of the operation to the 2200. Invalidate the contents of the Write/Copy cache.

c.10 EnabHead   Enable Radial Head Alignments
Imports: None.
Exports: The radial head alignment enable flag set to FFh.
Function: Receive the go code from the 2200 and set the flag HeadAlin to FFh.

c.11 SendStat   Send Status Message to 2200
Imports: The logical platter number in PlatNum.
Exports: The status message sent to the 2200.
Function: Receive the go code. Build the status message for the requested platter and send it to the 2200.

c.12 EAltCyl   Enable Alternate Cylinders
Imports: None.
Exports: ReadAlt set to 00h.
Function: Receive the go code and set ReadAlt to 00h.

c.13 RetryYes Enable Retries and ECC correction

Imports: None.

Exports: RetryAll set to 0Ah.

Function: Receive the go code and set RetryAll to 0Ah.

c.14 DisAHead Disable Radial Head Alignments

Imports: None.

Exports: HeadAlign set to 00h.

Function: Receive the go code and set HeadAlign to 00h.

c.15 DAltCyl Disable Alternate Cylinders

Imports: None.

Exports: ReadAlt set to FFh.

Function: Receive the go code from the 2200 and set ReadAlt to FFh.

3.17 WINCHESTER COMMAND MODULE

a. Services Provided - The Winchester Command Module implements the basic Winchester services. These are Read, Write and Format.

b. Modules Accessed - The services of the following modules are used by the Winchester Command Module. The Platter Table Module, the Alternate Sectoring Module, the Cache Module, the Error Count Module, the Memory Bank Module, the Drive Selection Module, and the Task-File Module.

c. Access Routines

c.1 WRead   Winchester Read

Imports: The logical platter and sector address from the 2200 in locations Comm2200, Platter, DumSect, MidSect, LowSect.

Exports: HL pointing to the requested sector data or the carry set and an appropriate error code in the A register.

Function: Select the proper drive and start a seek to the requested cylinder. While the seek is in progress, call the cache routines to search for the requested sector. If the sector is found in cache, abort the seek, point HL at the data in memory and return. If the sector is not in cache, read the sector and the indicated trailing sectors into cache. How many sectors to read is set by the caching routines and is found in location MaxSectC. This number may be adjusted to prevent reading beyond the end of the track. In the case of an ECC error when retries are enabled, request the correction information from the WD2010, correct the data, and end the read. If a sector can not be found; search the Alternate Sector Map for the sector (load in the Alternate sector Map if need be), retrieve the data from the alternate sector, and return. If an alternate sector can not be found return an 193 error code. If an error occurs on a sector other than the requested one, end the read. Adjust the number of sectors in cache and return

c.2 WWrite   Winchester Write

Imports: The logical platter and sector address from the 2200 in locations Comm2200, Platter, DumSect, MidSect, LowSect.

Exports: The requested sector data written to disk or an appropriate error message. If the multisector write flag is on, the sector data may be in left in the write cache. WCacheSM, WCacheSL, WCacheN, and WCachPlt update to reflect the number of the first sector in the write cache, the number of sectors in the write cache, and the logical platter to which they belong.

Function: To verify and acknowledge the platter and sector addresses. Select the proper drive and start a seek to the requested cylinder. Accept the sector data from 2200 into the Write Cache while the seek is in progress. Search the cache for the requested sector and remove it if found. If the Write Cache is full or the end of the track has been reached (last sector number was 31), write the entire cache to disk. Write the sector data to disk if the Multisector Write In Progress Flag is not ON (not equal to FFh). Update WCacehSM, WCacheSL, WCacheN, and WCachPlt to reflect the current contents of the write cache. If WCacheN equals 00h then the write cache is empty.

c.4 <u>WFormat</u> Format Winchester Platter

Imports: The logical platter address and command from the 2200 in locations Comm2200 and Platter.

Exports: The requested platter formatted.

Function: Read the Alternate Sector Map into memory for the requested drive. If the map exists remove all entries for the platter to be formatted. If the map does not exist, create one by initializing it in memory. Format the platter a track at a time, adding any defective sectors to the map. If the map did not exist, format the area reserved for alternate sectoring before formatting the platter. If a error is found in physical sector 0, or the map overflows, return an I93 error and set the defect byte for this drive to FFh. Update the bit map of platter formattability appropriately after the format completes. Write the new alternate sector map to disk starting at physical sector 0.

4. MAINTENANCE

4.1 Assembly Procedures - This code is assembled on an OIS using the Z80 assembler. Note that the OIS assembler can not provide a complete cross reference listing for an object as large as this code. To assemble the code, select the "Z80 Assembler" from the Merged 100 Menus and enter the following command:

perform "pathname.asm"

"pathname" as used in the above statement is a substitution for the system name, drive name and directory that specifies where the file resides. A valid example of a pathname is "2200DEV;z:Dave" where 2200DEV is the system name, z is the drive name, and Dave is the directory. Using this pathname the above command becomes:

perform "2200DEV;z:Dave.asm"

When the utility asks for the pathname, enter the pathname used in the PERFORM command. The contents of the asm batch file are specified in Appendix D.

4.2 Linking Procedure - The linking of this code is also done on the OIS in the same manner as the assembly. Select the Linkage Editor from the Merged 100 Menus and enter the following command:

perform "pathname.lnk"

When the utility asks for the pathname, enter the same pathname as used above. For an explanation of "pathname" see section 4.1. The contents of the lnk batch file are specified in Appendix D.

4.3 Debug Aids - A ZEBUG terminal is used as a debugger for Z80 products. To provide additional functionality, a 2200 with a 2250 option board running the program ZEMON may be used to control the ZEBUG unit.

4.4 Source Files - The names of all source files and a brief description of their contents is given in Appendix E.

4.5 Batch Files - The contents of all batch files are specified in Appendix D. The names of the batch files are as follows.

File Name	Use
asm	controls the Z80 Assembler
build	list of files to assemble
lnk	controls the Linkage Editor
prom	controls the creation of PROM image files

4.6 PROM Object Creation - PROM Object files are created from the assembled and linked object code by using the PROM File Creation Utility on Alliance system "PAM". To create an object file, place a formatted and scratched 2200 8" single sided white label floppy in drive 01 of "PAM". Then select the creation utility from the Program Development Menu and enter the following command:

perform "pathname.prom"

When the utility asks for the pathname, enter "pathname.cim". For an explanation of "pathname" see section 4.1. When prompted for the destination file name, fill in the revision number and press the return key. After the utility has completed creation, exit back to the main system menu and select WP. Once in WP press the "COMMAND" key followed by the "INDEX" (up arrow) key. This will unlock the floppy drive door allowing the diskette to be removed.

4.7 PROM Image File Creation - To convert the PROM Object file into a Chip Image file use the Chip Image File Utilities found under the main menu of the Production Chip Maker diskette version 05. The output will be a PROM image file suitable for creating PROMs.

5. PERFORMANCE

5.1 Code Size Restrictions - The code size is limited by the size of the PROM which is 32Kb, 4Kb of this space is reserved for the Power On Diagnostics. The rest is usable for system microcode.

5.2 Data Size Restrictions - The resident data must be co-resident with the microcode. Non-resident data will be stored in the 1st 16Kb bank of RAM. That leaves 240Kb of RAM that is reserved for Winchester read caches and alternate sector maps.

5.3 Interrupts - Interrupts issued by various hardware components will be disabled when those components are not in use. Parity errors in memory cause a non-maskable interrupt to occur.

5.4 Timing Restrictions - Each byte received from the 2200 during command specification must be echoed back within .3 ms. During data transfers bytes the DPU must be able to send or receive a byte every .3 ms. For performance reasons, all sector address validation must occur within 5 ms to keep the MVP from breakpointing. Simularly, if cache searches can be done in less than 5 ms, breakpoints during read operations will be minimized.

5.5 Operational Performance - Since this product is supposed to replace the 2280 Phoenix it should perform at comparable speeds. The goal is to have the speed of the 2275F/R be at least 90% to 95% of that of the Phoenix when using 64Mb Winchester drives. When 10Mb, 20Mb, or 32Mb Winchester drives are used the 2275F/R performance will be less due to the physical limitations of the drives themselves. To achieve a high level of performance the 2275F/R will have:

- Read and write routines that are optimized to best take advantage of the capabilities available in the hardware and to execute in the smallest amount of time.
- 1 to 1 sector interleaving.
- Intelligent high speed caching that is fine tuned to match capabilities and requirements of the 2200.

APPENDIX A
REFERENCES

Internal Documentation

Wang 2200 BASIC-2 Language Reference Manual
Wang 2200 BASIC-2 Disk Reference Manual
2200 Disk Command Sequences for 2200 LVP
    Max Blomme, Nov. 10, 1980
2200 I/O Bus
    Roger M. Kirk Jr., Aug. 1, 1985
2200 LVP Caching Routines
    Scott Tagen, 1985
2275F/R Software Functional Specification - Document Number MVP-06
    David M. Barrett, Nov. 19, 1985
5 1/4 Inch, Half Height, 40 Track (48 TPI) Double Sided Floppy Disk Drive Product Specification
    John Strazdes, Aug. 27, 1984
5 1/4 Inch 10 Mb Winchester Disk Drive Product Specification
    John Strazdes, Aug. 27, 1984
5 1/4 Inch 10 Mb Removable Winchester Disk Drive Product Specification
    John Strazdes, Nov. 29, 1984
3 1/2 Inch 20 Mb Winchester Disk Drive Product Specification
    John Strazdes, June 19, 1985
5 1/4 Inch 33 Mb Winchester Disk Drive Product Specification
    John Strazdes, Feb. 6, 1984
5 1/4 Inch 85 Mb Winchester Disk Drive Product Specification
    John Strazdes, Sept. 10, 1985

External Documentation

Advanced Micro Devices
    MOS Microprocessors and Peripherals Data Book
NEC Electronics
    Microcomputer Products Data Book
Western Digital
    Storage Management Products Handbook
    Storage Management Products Handbook Addendum January, 1985
Zilog
    Z80 Assembly Language Reference Manual
    Z80-CPU Technical Manual
    Z80-CTC Technical Manual
BYTE Magazine
    "Maximizing Hard-Disk Performance"
        Roy Chaney and Brian Johnson, May 1984
    "An Algorithm for Disk Caching with Limited Memory"
        Brian McKeon, September 1985

APPENDIX B
DATA DICTIONARY

The purpose of the Data Dictionary is to provide the developers with a cross reference of defined variables, constants, and routines. For each entry a name, type and description are given.

NOTE: Due to the shear volume of material the author has been unable to keep this dictionary up-to-date. For a complete list of all variable and constants see the source code file "sysequ" (system equates).

NAME	TYPE	DESCRIPTION
AddToMap	routine	Add an entry to the alternate sector map.
Addr2010	constant	Address of the memory mapped I/O port used to select registers of the WD2010 Task File.
AddrErrs	constant	Points to the number of addressing errors in the Status message Buffer.
AltSectM	constant	The base address of all alternate sector maps.
AltStart	constant	The offset into an alternate sector map to the pointer to the first alternate sector.
AnyGood	routine	Retrieves the results of the last floppy operation.
BitMap	constant	Offset into an alternate sector map to the bit map of platter formattability.
BRdyPort	constant	Pointer to the port used to set DPU Ready/Busy.
BSelPort	constant	The address of the port used to select which bank of memory is mapped into the high half of the Z80's address space.
BackSecF	variable	Stores the last floppy sector read/written.
BadECC	constant	Pointer to the number of uncorrectable ECC errors in the Status Message Buffer.
BitMask	routine	Creates a bit mask based on input.
CacheMap	constant	Points to the start of the cache map.
CacheNum	variable	Stores the number of the currently active Winchester read cache.
CAge	constant	The offset within a cache map entry to the age of the cache.
CalFSect	routine	Calculates physical floppy addresses from the logical sector address.
CalWSect	routine	Converts logical Winchester sector addresses into physical cylinder/head/sector addresses
CCheck	routine	Search for the specified sector in cache.
CClrPlat	routine	Removes all entries for the specified platter from cache.
CEmpty	variable	A flag indicating if an empty cache exists.
ChkFSect	routine	Calculates floppy drive sector addresses.
ChkPlat	routine	Validates platter addresses, and initializes PlatNum.
ChkSect	routine	Validates logical sector addresses.
ChkWSect	routine	Validates logical Winchester sector addresses and converts them into physical cylinder/head/sector addresses
CInit	routine	Initializes a cache.

NAME	TYPE	DESCRIPTION
CInitAll	routine	Initializes all caches.
ClearMap	routine	Removes all entries for the specified logical platter from the alternate sector map.
ClearReti	routine	A dummy interrupt handler.
ClrAECnt	routine	Sets all error counts for all platters to 00h.
ClrECnt	routine	Sets all error counts for the specified platter to 00h.
Comm2200	variable	Holds the command and platter bytes of requests from the 2200.
Compare	routine	Compares a sector on disk with data from the 2200.
CompCorr	routine	Causes the WD2010 to compute the ECC correction bytes after an ECC error has occurred.
Copy	routine	Copies a range of sectors from one location to another.
CPlatter	constant	The offset within a cache map entry to the logical platter number of the cache.
CPtHigh	constant	The offset within a cache map entry to the high byte of the pointer to the data for the cache.
CPtLow	constant	The offset within a cache map entry to the low byte of the pointer to the data for the cache.
CRemove	routine	Removes the entry for the specified sector from cache.
CSize	constant	The offset within a cache map entry to the number of sectors in the cache.
CStHigh	constant	The offset within a cache map entry to the high byte of the starting sector address in the cache.
CStLow	constant	The offset within a cache map entry to the low byte of the starting sector address in the cache.
CtcVctr	constant	A pointer to the CTC interrupt vector jump table.
CurntDrv	variable	Holds an integer number ranging from 0 to 4 indicating which physical drive is being accessed.
CValid	constant	The offset within a cache map entry to a flag indicating if the contents of the cache are valid.
Decode	routine	Decodes non-extended commands.
DefectDr	routine	Modifies the Platter-Table entry pointed to by CurntDrv to indicate that the drive is defective.
DmaEop	routine	The AM9715 interrupt handler.
DmaMode	constant	I/O port address for DMA.
DN3Test	routine	Tests DN3 to see if the '40' bit of the platter address is on.
DrFrmtB	routine	Modifies the Platter-Table bit map entry pointed to by PlatNum to indicate that a format attempt failed.
DumSect	variable	A dummy variable used to receive the third (high) byte of sector addresses during requests from the 2200.
DumSectD	variable	A dummy variable used to receive the third (high) byte of the destination sector address during copy requests from the 2200.
DumSectE	variable	A dummy variable used to receive the third (high) byte of ending sector addresses during verify and copy requests from the 2200.
EAltCyl	routine	Enables reading and writing of the reserved cylinders.
EnabHead	routine	Sets the radial head alignment flag to true to allow radial head alignments.
EndMSW	routine	Clears the multisector write in progress flag and writes the contents of the write cache to disk.

NAME	TYPE	DESCRIPTION
Err2010	variable	Stores the contents of the WD2010's error register.
ErrCount	constant	A pointer to the error counts array.
Extended	routine	Decodes extended commands.
False	constant	Equals 00h.
FlpDoor	routine	Detects if the floppy media has been disturbed.
FlopTab	constant	Pointer to the drive table for the floppy drive.
FrmtTrck	routine	Formats the indicated physical track.
FWinch10	constant	The pointer to the start of the 10Mb fixed Winchester Drive Table in ROM.
FWinch20	constant	The pointer to the start of the 20Mb fixed Winchester Drive Table in ROM.
FWinch32	constant	The pointer to the start of the 32Mb fixed Winchester Drive Table in ROM.
FWinch64	constant	The pointer to the start of the 64Mb fixed Winchester Drive Table in ROM.
GBitMap	routine	Returns the bit map of platter formattability.
GDrvType	routine	Places the switch setting for the requested drive in the A register.
GetByte	routine	Receives 1 byte from the 2200.
GetData	routine	Receives 256 bytes and a 1 byte LRC from the 2200.
GetECnt	routine	Returns the error counts for the specified platter.
GetPlatP	routine	Initializes the platter parameters with the platter table information indicated by PlatNum.
GetStat	routine	Places the contents of the WD2010's status register into the B register.
HeadAlin	variable	A flag used to indicate when radial head alignments have been enabled.
HardInit	routine	Initializes all of the major hardware components.
HeadPlat	variable	The number of the heads for the current logical platter.
I95ErrT	routine	Produces I95 error messages for the 2200 when requests are made to defective drives.
Idle	routine	Waits for re-init from the 2200.
Illegal	routine	Notifies the 2200 of illegal command requests.
IncECnt	routine	Increments the indicated error count for the specified platter.
InitMap	routine	Initializes the alternate sector map for the drive pointed to by CurntDrv.
InitPTE	routine	Initializes the specified platter table entry with the information from the indicated Drive Table.
Int2010	routine	The WD2010 interrupt handler, resets SeekProg to indicate that the event has occurred.
GetBytes	routine	Gets one or more bytes from the 2200.
LastHead	variable	The number of the last physical head of the current logical platter.
LoadDma	routine	Programs the AM9715 with the specified command
LoadMap	routine	Loads the alternate sector map from the drive indicated by CurntDrv.
LowSect	variable	Stores the low byte of the current logical sector number.
LowSectD	variable	Stores the low byte of the destination sector address during copy commands.

NAME	TYPE	DESCRIPTION
LowSectE	variable	Stores the low byte of the ending sector address during copy and verify commands.
Lrc	variable	Holds the LRC calculated by the routine LcrByte.
LrcByte	routine	Calculates the LRC for the specified sector.
MakeStat	routine	Compiles a status message for the requested platter
MaxSectC	variable	Contains the number of sectors that may be transferred to and from the presently active cache.
MidSect	variable	Stores the high byte of the current logical sector number.
MidSectD	variable	Stores the high byte of the destination sector address during copy commands.
MidSectE	variable	Stores the high byte of the ending sector address during copy and verify commands.
MSWPrtct	routine	Corrects interrupted multisector writes by writing the contents of the write cache to disk.
Multi	variable	A flag that indicates that a multisector write is in progress when its contents are FFh.
NAltCyl	variable	Stores the number of cylinders reserved for alternate sectoring on the current logical platter.
NewCyl	variable	Used to store the low order word of the physical cylinder number for Winchester accesses.
NewCylH	variable	Used to store the high order word of the physical cylinder number for Winchester accesses.
NewHead	variable	Used to store the head number for Winchester accesses.
NewSect	variable	Used to store the physical sector number during Winchester accesses.
NextAltH	variable	Holds the high byte of the next possibly available alternate sector map entry.
NextAltL	variable	Holds the low byte of the next possibly available alternate sector map entry.
NumCyl	variable	Stores the low byte of the number of cylinders on the current logical platter.
NumCylH	variable	Stores the high byte of the number of cylinders on the current logical platter.
NumSect	variable	Stores the high byte of the number of sectors on the current logical platter.
OldAgeC	variable	Stores the age of the oldest valid cache.
OldestC	variable	Stores the number of the oldest valid cache.
OpInProg	variable	Store a flag indicating if an operation is in progress.
ParErr	routine	Non-maskable interrupt handler for parity errors in memory.
PlatNum	variable	Holds the number of the current logical platter.
PlatTabl	constant	The pointer to the start of the Platter-Table.
Platter	variable	Holds the second command/platter byte for extended command requests.
PlatterD	variable	Holds the destination platter address for copy commands.
PromPort	constant	The address of the PROM mapping port.
PromRevH	constant	The high byte of the PROM revision level in ASCII.
PromRevL	constant	The low byte of the PROM revision level in ASCII.
Protocol	constant	The disk protocol level used by this device. Will always be equal to an ASCII "1" for 'DO' type devices.

NAME	TYPE	DESCRIPTION
ReadAlt	variable	Flag indicating whether or not reserved cylinder accessing has been enabled.
ReadSec	routine	Reads 1 to 32 sectors from the Winchester into the specified memory location.
Reinit	routine	Receives 'reinit' from the 2200, acknowledges it by sending back D0h.
RErrReg	routine	Reads and returns the WD2010 Error Register contents in the B register.
Request	routine	Receives and verifies commands from the 2200.
RetryAll	variable	A flag used to indicate if retries are enabled or disabled.
RetryNo	routine	Sets RetryAll so as to disable retries.
ReuseC	variable	A flag indicating if the current cache should be reused.
RoomLeft	constant	A pointer into an alternate sector map to a flag used to indicate if space is available in the map.
RstWinch	routine	Causes the currently selected Winchester to place its heads over physical cylinder 0.
RWCCyl	variable	Holds the low byte of the starting cylinder address for Reduced Write Current, or precompensation, for the current drive. A value of FFh indicates that no precompensation is to be done.
RWCCylH	variable	Holds the high byte of the starting cylinder address for Reduced Write Current.
RWinch10	constant	The pointer to the start of the 10Mb removable Winchester Drive Table in ROM.
RWinDoor	routine	Detects if the removable Winchester media has been disturbed (door opened).
RWrtPrtc	routine	Detects if the removable Winchester media is write protected.
SaveMap	routine	Writes the alternate sector map for the drive pointed to by CurntDrv out to disk.
SCylNumL	routine	Sets the low order word of the cylinder number to be used by the WD2010 to be the contents of NewSect.
SCylNumH	routine	Sets the high order word of the cylinder number to be used by the WD2010 to be the contents of NewSectH.
SecPCach	constant	The maximum number of sectors per cache.
Sect0Map	constant	A pointer to the cache map for all logical sectors numbered 0.
SeekCyl	routine	Causes the specified Winchester to seek to the requested cylinder.
SeekProg	variable	A flag indicating if a Winchester Seek operation is in progress.
SelBank	routine	Maps the requested memory bank into the high half of the Z80's address space.
SelProm	routine	Maps and unmaps the PROM on to the RAM.
SendStat	routine	Builds and sends a status message for the requested platter to the 2200.
SendByte	routine	Sends one byte to the 2200.
SendData	routine	Sends 256 bytes and a one byte LRC to the 2200.
SetBusy	routine	Sets the DPU status to busy.

NAME	TYPE	DESCRIPTION
SetCommR	routine	Places the contents of the B register into the WD2010's command register.
SetCorr	routine	Causes the WD2010 to use either 5-bit or 11-bit ECC correction spans.
SetCSize	routine	Sets the number of valid sectors in a cache.
SetReady	routine	Sets the DPU status to ready.
SetSDH	routine	Sets the SDH register of the WD2010 to specify the drive number, head number and to enable ECC. The contents of CurntDrv and NewHead are used.
SoftInit	routine	Initializes all data structures.
SPrecomp	routine	Sets the address used by the WD2010 to start Reduced Write Current based on the value of RWCCyl.
SrchMap	routine	Searches an alternate sector map for an entry corresponding to the given logical sector.
SSectCnt	routine	Sets the number of sectors to be transferred by the WD2010 according to the contents of MaxSectC.
SSectNum	routine	Sets the starting sector number of transfers by the WD2010 according to the contents of NewSect.
StartMSW	routine	Sets the multisector write flag to indicate that one is in progress.
Stat2010	variable	Holds the contents of the WD2010's status register.
StatPort	constant	The address of the status port for the DPU's hardware.
StrtHead	variable	The number of the first physical head of the current logical platter.
Switch1	constant	The address of the memory mapped I/O port containing switch number 1.
Switch2	constant	The address of the memory mapped I/O port containing switch number 2.
WCache	constant	The address of the write/copy cache in memory.
WCacheN	variable	Holds the number of sector in the write cache.
WCacheSL	variable	Holds the low byte of the sector number of the last sector in the write cache.
WCacheSM	variable	Holds the high byte of the sector number of the last sector in the write cache.
WCachPlt	variable	Holds the platter address of the sectors in the write cache.
WFormat	routine	Formats the indicated platter and updates the alternate sector map.
WinchPrt	constant	The address of the port used in Winchester drive selection.
WinchSel	routine	Enables the selection of the requested Winchester drive.
WRead	routine	Retrieves the requested logical sector from disk or cache and sends it to the 2200.
WriteSec	routine	Writes 1 to 32 sectors from the specified memory location onto the currently selected Winchester.
WWrite	routine	The high level write routine, handles all write logic coordination.

APPENDIX C

MEMORY USAGE MAP

	STARTING ADDRESS	ENDING ADDRESS	USE
Bank 0			
	8000h	81FFh	Variable Space
	8200h	83FFh	Floppy Pad 1 / ECC Correction Buffer
	8400h	85FFh	Floppy Pad 2 / Status Message Buffer
	8600h	BCFFh	Cache Tables
	BD00h	BDFFh	Platter Table Storage
	BE00h	BEFFh	Error Counts Storage
	BF00h	BFFFh	System Stack
Bank 1			
	C000h	DFFFh	Alternate Sector Maps and Storage
	E000h	FFFFh	Cache Space
Banks 2 - 15			
	C000h	FFFFh	Cache Space

APPENDIX D
BATCH FILES

This appendix specifies the contents of each of the batch files needed to assemble and link the source code, and to produce a PROM data file. "pathname" as used here is explained and defined in section 4.1.

D.1 FILE "asm" - This file controls the assembling of the source code. It is input to the Z80 OIS Assembler and uses the file "build" to identify the individual source files to be used.

```
%Filefix &,"pathname?"
Source /&.build/
Object /&.rel/
List /&.lst/
Assemble Macro,Xref,Fullref,Symbmax
End
```

D.2 FILE "build" - This file is used by the file "asm" to specify the source files and the order in which they are to be assembled.

```
2275F/R   MODULE
          LIST

* microcode
     TITLE     "System Equates"
     INSERT    "&.sysequ"

     TITLE     "Power-On Routines"
     INSERT    "&.poweron"

     TITLE     "2200 Interface"
     INSERT    "&.2200intf"

     TITLE     "Floppy Module 1"
     INSERT    "&.floppy1"
     TITLE     "FLoppy Module 2"
     INSERT    "&.floppy2"

     TITLE     "Alternate Sectoring"
     INSERT    "&.altsect"

     TITLE     "Winchester Caching"
     INSERT    "&.cache"
```

```
        TITLE     "Status Message Buffer"
        INSERT    "&.statmess

        TITLE     "Status Port"
        INSERT    "&.statport"

        TITLE     "Platter Table"
        INSERT    "&.pltrtabl"

        TITLE     "Drive Tables"
        INSERT    "&.drvtable"

        TITLE     "DPU Ready/Busy Selection"
        INSERT    "&.dpu"

        TITLE     "DMA Control"
        INSERT    "&.dma"

        TITLE     "Error Counts"
        INSERT    "&.errcnts"

        TITLE     "Memory Bank Selection"
        INSERT    "&.membank"

        TITLE     "System Switches"
        INSERT    "&.switch"

        TITLE     "Winchester Drive Selection"
        INSERT    "&.drvsel"

        TITLE     "WD2010 Commands"
        INSERT    "&.2010cmd"

        TITLE     "High Level Commands"
        INSERT    "&.highcmd1"
        INSERT    "&.highcmd2"
        INSERT    "&.highcmd3"
        INSERT    "&.highcmd4"

    * diagnostic code
        currently unknown

    END
```

D.3 FILE "lnk" - This file is used as input to, and controls the actions of, the OIS Linkage Editor. It produces a fully linked object code module.

```
%Map /pathname? .map/
%Fill /pathname? .cim/,0
%With /pathname? .rel/
End
```

D.4 FILE "prom" - This file controls the actions of the Alliance PROM Object file creation utility. It prompts the user for the name and location of the source file to be used. The name of the output file produced is "2275FR01".

```
Disk 01
%Filefix &,"pathname?"
Source "&.cim"
Object "2275FR01"
Execute
End
```

APPENDIX E
SOURCE FILE NAMES

Listed below are the names of the source files for each of the modules and a brief note as to their contents.

Source File name	CONTENTS
mods	Modification and revision history
sysequ	System equates
poweron	Power On Module
2200intf	2200 Interface Module
floppy1	Floppy Controller Module (format, read, and write)
floppy2	Floppy Controller Module (765 FDC subroutines)
altsect	Alternate sectoring Module
cache	Cache Module
statmess	Status Message Module
statport	Status Port Module
platrtabl	Platter Table Module
drvtable	Drive Table Module
dpu	DPU Ready/Busy Selection Module
dma	DMA Control Module
errcounts	Error Count Module
membank	Memory Bank Module
switch	Switch Setting Module
drvsel	Drive Selection Module
2010cmd	2010 Command Module
highcmd1	High Level Command Module (Winchester read/write)
highcmd2	High Level Command Module (Winchester format)
highcmd3	High Level Command Module (Copy, Compare & Verify)
highcmd4	High Level Command Module ($GIO commands)
misc	Miscellaneous General Purpose Routine Module
diag	The names of the source files for the diagnostic routines are unknown as of this time.

APPENDIX F
DRIVE TABLES

F.1 FlopTab - 360Kb 5 1/4" dual sided floppy drive

DB	2	# of heads per platter
DW	40	# of cylinders per platter
DB	32	Cylinder number of Reduced-Write-Current/4
DB	05	# of sectors per platter
DB	01	# of platters per drive
DB	00	# of cylinders for alternate sectoring

F.2 RWinch10 - 10Mb removable cartridge Winchester Drive

DB	2	# of heads per platter
DW	610 - 1	# of cylinders per platter
DB	32	Cylinder number of Reduced-Write-Current/4
DB	152	# of sectors per platter
DB	01	# of platters per drive
DB	02	# of cylinders for alternate sectoring

F.3 FWinch10 - 10Mb Winchester Drive

DB	4	# of heads per platter
DW	306 - 1	# of cylinders per platter
DB	32	Cylinder number of Reduced-Write-Current/4
DB	152	# of sectors per platter
DB	01	# of platters per drive
DB	02	# of cylinders for alternate sectoring

F.4 FWinch20 - 20Mb Winchester Drive

DB	2	# of heads per platter
DW	612 - 1	# of cylinders per platter
DB	32	Cylinder number of Reduced-Write-Current/4
DB	152	# of sectors per platter
DB	02	# of platters per drive
DB	02	# of cylinders for alternate sectoring

F.5 FWinch32 - 32Mb Winchester Drive

DB	4	# of heads per platter
DW	512 - 1	# of cylinders per platter
DB	32	Cylinder number of Reduced-Write-Current/4
DB	254	# of sectors per platter
DB	02	# of platters per drive
DB	04	# of cylinders for alternate sectoring

F.6 FWinch64 - 64Mb Winchester Drive

DB	2	# of heads per platter
DW	1024 - 1	# of cylinders per platter
DB	32	Cylinder number of Reduced-Write-Current/4
DB	254	# of sectors per platter
DB	04	# of platters per drive
DB	08	# of cylinders for alternate sectoring

APPENDIX G
I/O PORT ASSIGNMENTS

HEX ADDRESS	NAME	USE
00	DmaChan0	DMA Channel 0
02	DmaChan1	DMA Channel 1
04	DmaChan2	DMA Channel 2
06	DmaChan3	DMA Channel 3
08	DmaCoSt	DMA Command and Status Register
09	DmaReq	DMA Request Register
0A	DmaMask	DMA Mask Register (Independent Channels)
0B	DmaMode	DMA Mode Register
0C	DmaClear	DMA Clear F/L
0D	DmaReset	DMA Reset Register
0F	DmaMskAl	DMA Mask Register (All Channels)
10	Addr2010	WD2010 Bus Tri-State
11	WDErrReg/WDPreCmp	WD2010 Error/Write Precomp. Register
12	WDSectCt	WD2010 Sector Count Register
13	WDSectN	WD2010 Sector Number Register
14	WDCylLow	WD2010 Cylinder Number Low Register
15	WDCylHi	WD2010 Cylinder Number High Register
16	WDSDHReg	WD2010 SDH Register
17	WDStatR/WDCmdReg	WD2010 Status/Command Register
18	Winchprt	Winchester Drive Select Enable
20	CtcA0	CTC A Channel 0 (WD2010 Operation Complete)
21	CtcA1	CTC A Channel 1 (Unused)
22	CtcA2	CTC A Channel 2 (Floppy Operation Complete)
23	CtcA3	CTC A Channel 3 (2200 OBS)
30	CtcB0	CTC B Channel 0 (DMA Operation Complete)
31	CtcB1	CTC B Channel 1 (2200 Re-Init)
32	CtcB2	CTC B Channel 2
33	CtcB3	CTC B Channel 3 (General Purpose Timer)
40	FlpMstat	Floppy Controller Main Status Port
41	FlopComm	FLoppy Controller Control Register
50	FlpReset	Floppy Controller Reset
52	FlopCtrl	Floppy Drive Control Register
54	BRdyPort	2275F/R Ready/Busy Flag
56	In2200	2200 Write Data Register
60	StatPort	2275F/R Status
62	Switch1	2275F/R Configuration Switch (Switch 1)
64	Switch2	2275S Configuration Switch (Switch 2)
66	Out2200	2200 Read Data Register
70	BSelPort	Memory Bank Select / PROM Mapping Register
72		Enable/Disable Parity Register
74		Parity Error Reset
76	FlopTc	Floppy Output Terminal Count

APPENDIX H
2275F/S STATUS PORT BIT ASSIGNMENTS

BIT	USE
0	Floppy Index
1	Floppy Media Disturbed
2	Floppy Write Protect
3	Winchester Media Disturbed
4	Winchester Write Protect
5	CPB (CPU Ready/Busy)
6	DN3 (40 bit of platter address)
7	Unused

APPENDIX I
MICROCODE/POWER UP DIAGNOSTICS INTERFACE

I.1 Flow of Control

Whenever a power-on reset occurs the microcode will first execute a jump to location 7000h in PROM. It is expected that the Power Up Diagnostics entry point will reside at this point. Prom addresses 7000h through 7FFFh are reserved for diagnostics. During execution of the diagnostics the Z80 interrupt register I will contain 1Ch, this will allow the parity error handler to return control to the diagnostics during memory initialization or other times when this might be desirable. When they have completed testing the system the diagnostics shall return control to the microcode by jumping to the label "PowerUp1" (location 0003h). In the event of a catastrophic system component failure, the diagnostics jump to the label "ParErrb". The code that resides at that location will cause the LEDs to flash on and off with interrupts disabled preventing use of the unit.

I.2 Device Validity Table

DrvValMap points to a 6 byte array that is used by the diagnostics to notify the microcode of defective drives. The drive numbers of each drive is used as the array subscript. See Table 3.1 for the drive number definitions. If the diagnostics find that a particular drive is defective then the entry in the table for that drive shall be set to FFh, otherwise the entry shall be set to 00h. This will allow the microcode to let the rest of the unit be used while effectively removing the defective drive from the system.

Alternate Sectoring Specification V1.0
Aug. 8, 1986 A. Gruber

This document will describe the operation of the alternate sector module of the 2275 F/R disk storage unit. Alternate sectors are used to replace any bad sectors found during formatting. A format operation is the only operation that can create alternate sectors. A single alternate sector map will be used for the all the platters on a single drive.

The Alternate Sector Map

To manage the alternate sectors, a structure is kept on the first track of the disk called the alternate sector map. This structure is described herein.

The first sector of the disk (sector 0, head 0, cylinder 0):

Byte 0 - 2 - contains the ASCII code "WLI". The presence of this code indicates a valid alternate sector map.

Byte 3 - unused

Byte 4 - 5 contains a count of the total number of alternate sectors whether used or not (byte 4 low, byte 5 high). This number is derived from the number of cylinders reserved for alternate sectoring.

bytes 6 - 255 - unused.

The rest of the first track is composed of 4 byte entries representing the identity of the bad sector:

Byte 0 - ~~Low~~ High byte of bad cylinder address. Bit 7 is set if the entry is valid.

Byte 1 - ~~High~~ Low byte of the bad cylinder address

Byte 2 - Bad head number

Byte 3 - Bad sector number

The replacement sector is known by the location of the entry in the sector map. That is the first entry slot corresponds to the first available alternate sector. Alternate sectors start on the track following the alternate sector map and extend for the number of cylinders reserved for alternate sectoring. The maximum number of alternate sectors allowed is limited by the size of the alternate sector map. This is 31 sectors * 64 entries per sector or 1984 alternate sectors.

Basic Algorithms

Initial Alternate Sector Map Creation - During a format, the first sector of the disk is checked for "WLI" as the first 3 bytes. If they are there, the map is assumed valid and formatted. If they are not there, the first track of the disk is formatted and written with 0's (clearing all the entries). The first sector is written with "WLI" and the number of available entries in the map. Then the alternate sector area itself is formatted. Any bad sectors found are mapped to themselves.

Alternate Sector Creation - If during a format, a sector is found bad, the track is reformatted with the sector and its neighbors marked with bad block marks. The three sectors are then added to the alternate sector map. If the sector is already present in the map, the alternate sector is reused. Otherwise the first available alternate sector is used.

Alternate Sector Access - If during a read or a write, a bad block mark is encountered, the alternate sector map is read in and the map is searched for the desired sector. If it is present in the map, the actual alternate sector is read or written as appropriate. The alternate sector map is read in each time it is needed and is not permanently kept in memory.

Description of Routines

ChkAltMap

Imports:
IY - points to physical area
IX - points to command structure
HL - holds a buffer list (at least 1 sector long)
Exports:
Carry set if no valid map is found
Function:
Read in first sector of disk into buffer and check for "WLI"

CreatAltMap

Imports:
IY - points to physical area
IX - points to command structure
HL - holds a buffer list (at least a track long)
Exports:
Carry set if a valid map can't be created.
Function:
Initialize the alternate sector map and format the alternate sector area.

AddAltSec

Imports:
IY - points to physical area
IX - points to command structure
HL - holds a buffer list (at least a track long)
AddAltOn - if false, don't do the add
AddAltSelf - If true, map the sector to itself
Exports:
Carry set if we can't add to the map
Function:
Add a sector to the alternate sector map. The sector is initialized to zero.

ReadAltSec

Imports:
IY - points to physical area
IX - points to command structure
HL - holds a buffer list (at least one sector long)
A - holds sector number to relocate
Exports:
Carry set if we can't find the sector in the alternate sector map.
Function:
Read the alternate sector into the buffer list.

WriteAltSec

Imports:
IY - points to physical area
IX - points to command structure
HL - holds a buffer list (at least on sector long)
A - holds sector number to relocate
Exports:
Carry set if we can't find the sector in the alternate sector map

Function:
Write the buffer into the alternate sector.

FindAltSec

Imports:

HL - holds pointer to a buffer list (one sector long)
IX - points to command structure
IY - points to physical area
B - holds sector number to relocate
C - holds head number to relocate
DE - holds cylinder number to relocate
FlopBuf+256 is used to hold the alternate sector map info

Exports:

Carry Set if we can't find the desired alternate sector
Structure in IY wil be updated with the alternate cylinder, head, and sector.

Function:

Read in the alternate sector map and find the desired alternate sector.

RAM Disk Specification V1.0
Aug 14,1986 A. Gruber

This document will describe the operation of the RAM disk module of the 2275 F/R disk storage unit.

The RAM disk is in some ways treated like the cache. The sectors are accessed thru the same hash lookup mechanism as the cache. The RAM disk however may not be reclaimed when additional storage is needed. It is only reclaimed by explicit deallocation.

Reads from the RAM disk are done in two ways. If the high level routine normally checks the cache for the desired sector then the RAM disk sector will be found. Reads directed to the device result in the RAM disk sector being looked up and its data pointer copied into allocated storage. The allocated storage is marked to denote that the pointer has been copied. The cache mechanism knows to restore the original pointer when the sector is deallocated.

Writes to the RAM disk result in the RAM disk sector being looked up and its data pointer swapped with the pointer from allocated storage containing the data to be written. This works except when the data in allocated storage has previously been read out of the RAM disk. In this case the pointer is a "copied" one and swapping pointers would result in two sectors of the RAM disk pointing to the same data. The data in this case is physically moved, first into low memory and then to the RAM disk sector.

Some routines (such as copy) function on a track by track basis. A track on the RAM disk will be defined as extending 32 sectors from the current sector or to the end of the RAM disk, which ever is less.

Description of Routines

RAMDAlloc

Imports:

RAMDBufPtr holds start of allocated RAM disk buffer
E - holds command byte

Exports:

RAMDBufPtr holds new start of RAM Disk storage. The RAM disk is initialized and zeroed. The Platter table for the RAM disk is filled in.

Function:

High level routine to allocate a RAM disk. The previous RAM disk is deallocated. Must leave at least 32 sectors free for cache and working storage. If a RAM disk of 0 sectors is requested, the platter table entry for the RAM disk will be voided.

ReadRSec

Imports:

HL points to storage buffer list
IX points to command structure
IY points to physical parameter area

Exports:

The RAM disk sector is read into the buffer list.
A and carry are cleared

Function:

Read a sector (or sectors) from the RAM disk. The RAM disk pointer is copied into the buffer list.

WriteRSec

Imports:

HL points to storage buffer list
IX points to command structure
IY points to physical parameter area

Exports:

The sector is written to the RAM disk
A and carry are cleared

Function:

Write a sector (or sectors) to RAM disk. The RAM disk pointer and the buffer list pointer are swapped.

FrmtRTrk

Imports:

HL points to storage buffer list
IX points to command structure
IY points to physical parameter area

Exports

The sectors are written with 0's
A and carry are cleared

Function

Write a "track" of the RAM disk with 0's. A track is 32 sectors long or to the end of the RAM disk, which ever is less.

Cache Software Specification V1.4
August 8, 1986 A. Gruber

This document will describe the cache module of the 2275 F/R disk storage unit. The purpose of the cache is to speed up disk operations in two ways. The first is to keep copies of frequently used sectors in memory rather than on disk. The second is to try to guess what sector the cpu will ask for next and prefetch that into memory. This will take the form of fetching the sectors following the one requested. This saves rotational latency time on the following access.

The cache operates on fixed size (256 byte) buffers. These buffers are allocated and reclaimed by the storage allocator portion of the cache module. While any number of buffers may be requested, the storage allocator cannot guarantee that the buffers will be in contiguous memory. This implies that disk operations must be done in single sector chunks with any data pointers being relocated to the next buffer in between operations. In situations where this is not possible (such as 512 byte floppy sectors), the data must first be gathered into a suitably sized chunk of contiguous memory prior to the actual data transfer.

The Buffer Entry Data Structure

To manage the 256 byte buffers, the storage allocator keeps in low memory a pool of buffer entries. These 12 byte data structures are used to keep track of the buffers.

The buffer entries are used in three different situations:

1) an entry for a cached sector
2) an entry for general purpose allocated storage
3) as entry for unallocated storage

The use of the fields of the structure in each of these situations will be described

Flag (Byte 0) - contains status information about the entry. Bit 7 will be set if the entry is part of the cache. During reads from the RAM disk, rather than moving data, the pointer to the data will be copied into BufBank and BufAddr. Bit 5 is set if this has been done. If Bit 5 is set, the permanent contents of BufBank and BufAddr have been moved to NextL and NextH. These two slots would not otherwise have been used. If it is a cached entry, Bit 4 is set if the entry is in the top part of the cache. (i.e. above the point that unused sectors are added) Bit 3 is set if it has been used at least once. Bits 0 - 2 count how many previous sectors were present in the cache when the sector was first read into the cache. A count of 7 indicates that atleast 7 previous sectors were present. In non-cache situations this byte will be 0.

Buffer Bank (Byte 1) - holds the bank number of the allocated buffer for all three situations.

Buffer Address High (Byte 2) - holds the pointer to the high byte of the address within the bank of the allocated buffer for all three situations. The low byte is assumed to be 0.

Next Entry Address (Byte 3 low, Byte 4 high) - In a cache this points to the next entry representing a sector that hashed to the same value. In non cache situations these bytes are unused, except for RAM disk reads, where they are used to hold the previous contents of BufBank and BufAddr

Platter No. (Byte 5) - used only in a cache entry, this holds the logical platter number of the cached sector.

Sector No. (Byte 6 high, Byte 7 low) - used only in a cache entry, this holds the logical sector number of the cached sector.

More Recent Ptr (Byte 8 low, Byte 9 high) - In a cache entry, this points to the first entry that has had a cache hit more recently than the current entry. Used to implement a least recently used cache replacement algorithm. In non cache situations this points to the previous entry in the list.

Less Recent Ptr. (Byte 10 low, Byte 11 high) - In a cache entry, this points to the first entry that has had a cache hit less recently than the current entry. Used to implement a least recently used cache replacement algorithm. In non cache situations this points to the next entry in the list.

Other Data Structures

Hash Table - a contiguous section of memory containing 256 pointers to buffer entries. These entries are sorted in order of the lowest 8 bits of their logical sector number. This allows a simple index mechanism to find the cache buffer corresponding to a given logical sector. A pointer of 0 (or any number with 0 in the high byte) signifies that no sector that hashes to this entry is cached.

Hash List - a list of buffer entries in the cache that hash to the same value. The first entry in this list is pointed to from the hash table.

Used List - a doubly linked list of buffer entries ordered from most recently to least recently used. All cache entries are part of this list as well as unused entries. New buffers are allocated from the end of the list. Using a buffer more than once results in it being moved to the top of the list. New entries are put into the middle of the list. Unused and newly freed buffers are put at the end of the list.

Basic Algorithms

Cache Lookup - The low byte of the sector number will be used to index into a hash table of pointers to buffer entries for cached sectors. If no entry is present this signifies a cache miss. If an entry is present the platter number and high byte of the sector is checked. If these match, this signifies a cache hit. Otherwise the next entry pointer is followed. Reaching the end of the list with no hit constitutes a cache miss. Note that logical rather than physical addresses are cached. This allows faster look up as in the case of cache hits, the logical to physical translation need not be done. However, if anything strange goes on with the addressing (such as a change in the logical to physical mapping as happens when alternate sectors are enabled) the cache will be fooled and should be turned off.

Cache Replacement - New buffers are allocated from the cache on a modified least recently used basis. Cache entries are ordered according to time of last use. When a cache entry is used it moves up the list -- either to the head of the list if it has been used at least once before or to the middle of the list if it has never been used. This allows frequently used sectors to "fall" to the bottom slower than sectors that are only used once (say for program load). Newly read sectors from the disk are also put into the middle of the list

Cache Prefetch - On detecting a cache miss, the data will be read from disk into cache. The sectors following the requested sector might also be read in. This is determined by the following rules:

1. If the previous sector is not present in the cache or has not been used or if it has been used but no sector previous to that has been used, then read one extra sector or to the end of the track, whichever is fewer.

2. If the previous sector is present and has been used and less than four sectors previous to that have been used, then read three extra sectors or to the end of the track, whichever is fewer.

3. If the previous sector is present and has been used and four or more sectors previous to that have been used, then read to the end of the track.

General Storage Allocation - buffer allocation for purposes other than cache buffers is handled in the same manner as cache allocation. That is buffers are taken from the cache on a least recently used basis. Storage allocated for general storage is reclaimed only when it is explicitly freed by a CFree call.

Write Strategy - No writes are cached. Sectors that are already in cache are invalidated if written to.

Description of Routines

CAlloc

Imports:
A - Desired number of buffers (0=256)
Exports:
A - Number of buffers provided. Carry Set if less than requested.
HL - Points to buffer entry list
Function:
Allocates dynamic storage either for cache or other purposes

CFree

Imports:
HL - Points to buffer entry list
Exports:
None
Function:
Explicit reclaimation of storage. Note that the list should end with a null pointer. If Bit 5 of the flags are set, the permanent pointer to storage is restored to BufBank and BufAddr.

ChkCache

Imports:
A - Platter Number
HL - Sector Number (bytes swapped)
Exports:
Carry Set if not found
HL - Cache entry if found

Function:
Does cache lookup to see if a given sector is cached.

CHitUpdate

Imports:
HL - Pointer to Cache Entry
Exports:
The Cache Entry has its use status updated. Its positions in the least recently used list and hash list are changed.
Function:
The entry for the previous sector is looked up and used to modify the use status of the current entry. The current entry is moved to the top or middle of the least recently used list. It is also moved to the top of the hash list.

CMissUpdate

Imports:

A - Platter Number
HL - Sector Number
DE - Pointer to Buffer List
IY + MaxSC - holds number of valid sectors in Buffer List
PrevFlag - holds flag byte of previous sector number

Exports:

The cache is updated and the read sectors added.

Function:

The newly read sectors are put into the cache. The first sector is marked as being used once, while the following ones are marked unused. The unused buffers are reclaimed. The sectors are inserted at the top of the hash list.

CInvalSec

Imports:

A - Platter Number
HL - Sector Number (bytes swapped)

Exports:

The cache entry for the specified sector is removed

Function:

Invalidates cache entry for the sector (if present)and reclaims the storage.

CInvalPlat

Imports:

A - Platter Number

Exports:

All cache entries for the specified platter are removed.

Function:

Invalidates the whole platter and reclaims the storage. This functions by actually scanning all the cache entries. Hence it is rather slow.

CacheStrat

Imports:

A - Platter Number
HL - Sector Number (bytes swapped)
B - Number of sectors left to end of track

Exports:

A - Maximum Number of Sectors to Read
PrevFlag - holds flag byte of previous entry

Function

Implements prefetch strategy. The previous entry is read from cache to determine what to do. If the value is greater than the number of sectors left on the track, then just read to end of track. It also checks to see whether any prefetched sectors are already in cache. If so then the returned value will stop prefetch just prior to that sector.

GetCPtr

Imports:

HL - Pointer to Buffer Entry

Exports:

DE - Pointer to Buffer

The selected memory bank is changed

Function:

Extracts buffer pointer from buffer entry. The memory bank containing the buffer is swapped in.

CLink

Imports:

HL - Pointer to Buffer Entry

DE - Pointer to Buffer Entry

Exports:

None.

Function:

Buffer entry in HL is linked to buffer entry in DE (i.e. DE is head of list and HL is tail of list).

CInit

Imports:

None.

Exports:

Cache and storage allocator are initialized. All available storage is reclaimed.

Function:

Intializes the cache and storage allocator.

CSkip

Imports:

A - holds number of buffer entries to skip

HL - holds pointer to start of buffer entry list

Exports:

HL - holds pointer to start of new buffer entry list

Function:

The number of entries in A are removed from the front of the buffer entry list.

DCAS1700

PRODUCT SPECIFICATION
CASSETTE TAPE 17 Track (D/CAS)
WANG PART NO. 725-4893

APPROVAL OF THIS DOCUMENT

Ed Mac Neil	EDITOR
Ed MacNeil	TASK ENGINEERING
10/6/89	PROJECT MANAGER
10/6/89	ENGINEERING MANAGER
	MARKETING

SHEET NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
REV. LEVEL	A																												A
SHEET NO.	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
REV. LEVEL	A																				A								
SHEET NO.	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87
REV. LEVEL																													
SHEET NO.	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116
REV. LEVEL																													
SHEET NO.	117	118	119	120																									
REV. LEVEL																													

BY		
DATE		
REVISION	ORIGINAL RELEASE:	Ed. Mac Neil Ed. Mac Neil
LTR	A	

WANG LABORATORIES, INC.	DWN	DATE	TITLE	WLI NUMBER	
	CHK	DATE	PRIDUCT SPECIFICATION CASSETTE TAPE (D/CAS) 17 TRACK WANG PART NO. 725-4893	191-3468	
	APPR	DATE			REV.
	APPR	DATE		1 of 50	A

14-166 3

T A B L E o f C O N T E N T S

WANG LABORATORIES, INC.	DWN	DATE	TITLE	WLI NUMBER 191-3468
	CHK	DATE	PRIDUCT SPECIFICATION CASSETTE TAPE (D/CAS) 17 TRACK WANG PART NO. 725-4893	
	APPR	DATE		2 of 50
	APPR	DATE		REV. A

TABLE of CONTENTS

1.0 SCOPE

The purpose of this document is to describe the specifications of a specific product, including it's general, physical and operational characteristics. Although this device is implemented via the generic interface known as QIC-02, this specification, will treat each item on a device basis. Additional applications information is included for clarity. For reference, the Cassette devices evolved through a standard known as D/CAS, (Data Cassette Interchange), which is a sister to the QIC-02 interface.

All commands, features, protocol, and timing, specified herein are MANDITORY. Any reference to optional, preferred, or not recommended are for applications of the product, with respect to future interface compatibility and performance.

2.0 APPLICABLE DOCUMENTS

All product received by Wang Laboratories will conform to the latest revision of the following documents. All product shall be designed to meet the following Wang Laboratories documents and compliance standards. Wang will monitor incoming product and notify the vendor of any descrepencies. The documents listed may be obtained by contacting Wang Laboratories or the organization of origin.

In the case of conflict the order of precedance of the applicable documents is as follows:

1. This specification
2. Compliance Documents
2. Purchase Order
3. Vendor Product Specification.

2.1 COMPLIANCE DOCUMENTS

TITLE	DOCUMENT NUMBER
This Specification	191-3468
Wang Specificaiton for Lable, Model and Serial Number	615-2319
Electronic Data Processing Units and Systems	UL 478
Data Processing Equipment	CSA 22.2 154
Regulations for Electric Motor Operated Appliances	VDE 0806
Radio Frequency Interference	VDE 20870 (CLASS A)
General European Safety Standards for Office Equipment	IEC 380
Measurement and Designation of Noise emitted by Computer and Business Equipment	ANSI S12.10
Shipping Specification	10-521
Shock and Vibration Test Procedure	191-2505
Wang Computing Device Susceptibility	SPI-010-630
Wang Environment Specification (Mechanical)	SPI-10-708
General Vendor Packaging	VPI-1000 and VPI-1001

2.2 REFERENCE DOCUMENTS

TITLE	DOCUMENT NUMBER
Magnetic Tape Cassette 17 track Unrecorded Specification	191-3487
Magnetic Tape Cassette 17 track (Wang Part Number)	725-7548

3.0 WANG REQUIREMENTS

The information in this section identifies specific requirements which must be met by each drive received by Wang Laboratories. The vendor is expected to adjust or monitor processes as necessary to guarantee compliance. The product must meet all parts of this specificaiton when operating in either an offline diagnostic mode, utilizating vendor recommended test equipment or in any Wang system.

3.1 MODIFICATIONS AND CHANGES

Product modificatons or changes will require prior written approval by Wang Laboratories before implimentation and shipment, unless Wang Laboratories is negligent in responding to an ECN within the specified time period.
All critical engineering changes must be submitted to Wang Laboratories for approval, sixty (60) days prior to shipment. If Wang Laboratories does not disapprove of the ECN within the sixty (60) day period, the vendor may implement the ECN.
Any change which affects form, fit or function with respect to criterias defined in this mutually agreed appon specificaton or as it applies to the Wang applications.

These changes are more specifically:

1. Any change which affects the electrical interface definiton.
2. Any chabge to on- board processor micro- code.
3. Any artwork changes to printed circuit boards.
4. Circuit design changes, changes which improve performance or correct problems. Notification of vendor part substitions if that substitution corrects a specific problem or affects field spare requirements.
5. Any change that substantially alters the product burnin and test processes.
6. Any change to the mechanical configuration
7. Any mecahnical or electrical change internal to the drive mechanism.

For changes which require modeling, Wang Laboratories will be supplied a set(s) of all effected subassemblies and sufficient documentation to determine effectivity. These assemblies are to have all proposed changes incorporated. Approval or disapproval with findings, will be confirmed within sixty (60) days of receipt of assemblies and requested documentation. In emergancy situations, Wang Laboratories will approve or disapprove changes using best effort. Wang Laboratories shall have indicated, via placement of a vendor on the (AVL) Approved Vendor List, that a baseline configuration has been identified. When the vendor is placed on the AVL, it is required that a copy of all E.C.O's be sent to Wang Laboratories for information use only. When production orders ore placed, a new baseline configuration will be identified. All E.C.O's will now be approved by Wang Laboratories befor installation by the Vendor.

NOTE: A baseline configuration is defined as the REVISION LEVEL of the product. The Vendor will provide in a timely manner, to a duly authorized representative of Wang Laboratories the following informaiton:

1. Change Notification.
2. Reason for the change.
3. Affected assemblies.
4. Cut in date and serial number
5. Serial numbers affected.
6. Up grade kit availability.
7. Implementation plan.
8. Technical notes or revised training material.
9. Revised product or diagnostics, manuals or revision inserts.

Should a conflict arise as to whether a change will have an affect on the form, fit or functon of Wang Laboratories systems utilizing the product, additional material needed may include:

1. Schematics or mechanical drawings of the affected area.
2. Test data which identifies that a change is required.
3. Test data which substantiates the improvement.
4. Detailed explanation of firmware changes.

3.2 SAFETY STANDARDS

All drives received by Wang Laboratories must conform to whatever general, physical, electrical or safety requirements are imposed on this type of equipment, by UL, CSA and VDE organizations. The vendor warrents that equipment purchased for use in North America is "UL Recognized" and "CSA Certified" as component equipment under electronic data processing equipment. For world wide usage, the vendor shall state that the equipment purchased will meet the requirements of the international standards stated below.

USA	Manditory 478 File No. E77688 - U.L. Recognized U.L. 478 Forth Edition, Standard for Safety Electronic Data Processing Units and Systems (Underwriter Laboratories).
Canada	Manditory Agreement No. LR52307 - CSA Certified CSA 22.2 No. 154 - M1983 Second Edition Data Processing Equipment (Canadian Standards Association)
Foreign	Manditory DIN IEC 380/VDE 0806/8.81 50 Hz Units Safety of Offise Machines (Association of German Electrical Engineers IEC - 380)

3.3 WANG COMPUTING DEVICE SUSCEPTIBILITY

The drive will not produce any unrecoverable errors when subject to an Electrostatis transient discharge of 12.5 KV.per Wang Laboratories specification (SPI 10-630)

3.4 RADIO FREQUENCY INTERFERANCE

All drives received by Wang Laboratories will meet the radiated and conducted emission limits for equipment as required in the following specifications:

1. FCC rules, Part 15, Subpart J, 80-148, 27114, Docket 20780 (Class A)
2. VDE 0871 / 6.78 (50 Hz Units).
3. The vendor warrents that the equipment purchased shall meet all requirements outlined in FCC Docket No. 20780 for Class A computing devices.
4. Once the Drive is installed and is electrically and or physically attached to a Wang Laboratories system, a drive must not in itself, due to faulty design or assembly, prevent Wang Laboratories from gaining EMI or RFI compliance of said systems under FCC dockte No. 20780 for Class A computing devices.

3.5 SHIPPING SPECIFICATIONS

No degradation of performance as measuered by this spcification will be allowed during all packaging, handeling, assembly and shipping to Wang Laboratories. Packaging and shipping will be the responsibility of the vendor. Broken or damaged parts due to shipping shall be returned to the vendor for credit, per Wang Standard SPI 10-521.

4.0 PHYSICAL DESCRIPTION

This device is categorized in general as a "Half High 5 1/4 inch" form factor. It shall conform to the specific dimensions specified herein. The basic drive is a 3 1/2 inch form factor, with a modified beizel, filler frame, and power/ signal adaptor board.

4.1 DIMENSIONS

overall unit (5 1/4 inch formfactor) (also refer to FIGURE 20)

Height	41.3 +/- 0.5 mm	(1.626 +/- 0.020 inches)
Width	146 +/- 0.5 mm	(5.750 +/- 0.020 inches)
Depth	203 +/- 1.0 mm	(8.000 +/- 0.040 inches) (Excluding External Interface connector projection.)

Base unit (3 1/2 inch formfactor)

Height	41.3 + 0 - 1 mm
Width	101.6 + 0.2 -0.5 mm
Depth	146 + 0 - 1 mm

4.2 MOUNTING

The drive may be affixed using the mounting holes on the sides and bottom, reference FIGURE 20.

4.3 ORIENTATION

The Drive may be mounted in either of two (2) orientations, with the front bezel at right angle to a horizontal plain:

a. Ejection lever on the right hand side.
b. Ejection lever on the top.

The drive may be mounted as specified above with a declination angle of 15 degrees maximum, such that the front panel is at an equal or higher elevation than the rear of the unit.

4.4 COLOR

The front bezel and all of it's exposed parts, with the cassette removed shall be BLACK (Munsell N1). The front panel indicator shall be Amber (when iluminated).

4.5 WEIGHT

Overall unit	920 g	Approximately
Base unit	650 g	Approximately

5.0 POWER

5.1 REGULATION

The drive shall be independant of power sequencing, in any order, without damage to data recorded on the medium. The drive shall conform to this specification under the following power input conditions:

Nominal (Volts)	Regulation	Ripple [1]	Current [2]
+ 5.0	+/- 5%	100mV	0.8 Amp Max.
+12.0	+/- 5%	200mV	1.3 Amp Max.

NOTES: 1. Ripple includes spike noise.
2. Maximum current is source impedance dependent (reference FIGURE 12)

5.2 CONNECTOR (DRIVE)

Amp part number 172349-1 or equivalent.
A 4 pin inline keyed connector with pins labeled 1 thru 4 shall be mounted to the drive per FIGURE 12.

5.3 CONNECTOR (MATING)

Housing: Amp part number 1-480424-0 or equivalent
Contacts: Amp part number 60619-1 or equivalent

5.4 CABLE

The cable guage and length are to be selected with reguard to the maximum power consumption, and the required regualtion. Reference SECTION 14.1 and FIGURE 12. The pin assignments are as follows: (PIN 1 +12 V), (PIN 4 +5V) (PINS 2 & 3 0V DC-GROUND)

5.5 FRAME GROUND CONNECTIONS

Refer to figure 11 for frame ground conndction.

Isolation method: Disconnect jumper W1 (0 ohm resistior), mounted on the the drive PCBA MD.
Insulation resistance: (after isolation) greater than 10 M ohms, measured at DC 100 Volts.

6.0 ENVIRONMENTAL

The operating environmental conditions specified herein are with Natural Air Convection Cooling.

6.1 AMBIENT TEMPERATURE (Media dependant)

Operating: + 5 to +45 deg. C
Nonoperating: -25 to +60 deg. C

6.2 TEMPERATURE GRADIENT

Operating: less than 15 deg. C per hour (Non Condensing)
Nonoperating: less than 30 deg. C per hour (Non Condensing)

6.3 RELATIVE HUMIDITY

Operating: 20% - 80% (Non Condensing Max. Wet Bulb 26 deg. C)
Nonoperating: 10% - 90% (Non Condensing Max. Wet Bulb 26 deg. C)

6.4 VIBRATION (Tested per Wang Specification reference SECTION 2.0)

Operating:	0.5 G	5 to 100 Hz. 3 minutes sweep	
	0.25 G	100 to 500 Hz. 3 minutes sweep	
Nonoperating	2.0 G	5 to 50 Hz. 3 minutes sweep	

6.5 SHOCK (Tested per Wang Specification reference SECTION 2.0).

Operating:	5.0 G	10 mS.
Nonoperating:	50.0 G	10 mS.

7.0 RELIABILITY

7.1 MTBF 10,000. Hours Minimum (Operating with TAPE MOTION)

7.2 MTTR 30 Minutes Maximum (Field replaceable unit)

7.3 COMPONENT DESIGN LIFE 5 years (except for head wear)

7.4 HEAD WEAR 3,000, Hours minimum

7.5 PREVENTATIVE MAINTENANCE Head Cleaning (monthly) or 200 (BOT to EOT to BOT) motions.

7.6 ERROR RATE
a. Soft: 10^{-7} bits Read (D/CAS-85)
b. Hard: 10^{-12} bits Read (D/CAS-85)
a. Soft: 10^{-7} bits Read (D/CAS-25)
b. Hard: 10^{-10} bits Read (D/CAS-25)

8.0 SHIPPING AND PACKAGING

8.1 TRANSPORTATION PERFORMANCE
The level of packaging required at the individual unit level shall be to (as accelerated in test reference below) protect the product when it is transported by motor freight (other than air suspension).
Rail, parcel post, express type service, loose ocean freight or ocean containers in which freight is moved or added during transhipment

8.2 PACKAGING PERFORMANCE
Packaged product shall be subjected to applicable test per Wang Standard SPI 10-521 "Shipping Container Performance Test" for individual unit packages. At completion of tests, unit shall show no signs of mechanical damage nor shall the operational performance be impaired. Allignment or adjustment of operator accessible controls may be employed.

8.3 TEST REQUIREMENTS
Prior to the acceptance of vendor packaging, a representative, functional product and package shall be subjected to the tests specified. This testing may be performed by Wang Laboratories R&D package engineering, an approved independant package testing laboratory or other laboratory acceptable to Wang Laboratories Inc. The report of these tests will, in addition to the information called in SPI 10-521 rev. A, contain engineering drawings of the package tested so that they may be used to determine continuing compliance with performance requirements of this test standard. Should a change in package design be made, retesting and recertification are required.

9.0 OPERATIONAL CHARACTERISTICS

9.1 TAPE SPEED — 90 ips nominal (2286 mm/s)
9.2 LONG TERM SPEED VARIATION — 3% max.
9.3 INSTANTANEOUS SPEED VARIATION — 4% max.

9.4 AVERAGE DATA TRANSFER RATE (in streaming operation)
122.5 k bytes per second 17 track mode. (D/CAS-85)
86.3 k Bytes per second 9 track mode. (D/CAS-25)

9.5 BLOCK LENGTH — 512 Bytes (fixed)

9.6 BUFFERS — 54 at (512 Bytes each)

9.7 RETRY COUNT RECORDING — 15 Maximum

9.8 RETRY COUNT READING — 16 Maximum

9.9 REPOSITIONING TIME — Recording Track 0 2.5 seconds Approximate
Reading & Recording 1.0 seconds Approximate

9.10 REWIND TIME — 80 Second Approximate. (Media length and current position dependant)

10.0 RECORDING GEOMETRY

10.1 STANDARD D/CAS-85

10.2 METHOD GCR 4/5 conversion (0-2 RLL Code) reference TABLE 2

10.3 RECORDING FORM Single track serpentine serial

Record and Read	17 track D/CAS-85	16 & 1 track dual partition
Read only	9 track D/CAS-25)	8 & 1 track dual partition

10.4 AREA & SEQUENCE Reference FIGURE 14 - 16

10.5 TRACK LOCATIONS Reference FIGURE 15

10.6 NUMBER OF TRACKS 17 total Reference SECTION 15.15

10.7 TRACK PITCH
0.203 mm (nominal) 17 Track
0.406 mm (nominal) 9 Track

10.8 TRACK WIDTH
Write 0.150 +/- 10 um
Read 0.100 +/- 10 mm

11.0 RECORDING FORMAT

The Recording format describes the longitutional parsing of all the information recorded on the medium, it includes a detailed description of all fields within each of the types of blocks used.

Default mode: Records in a single partition D/CAS standard format employing the two types of recorded blocks: DATA BLOCKS, FILEMARK BLOCKS

Partition Mode: Records in a two partition format employing the two types of recorded blocks: USER DATA BLOCKS, FILEMARK BLOCKS.

T R A C K 0 L A Y O U T

```
B O T                                                                                  EOT
!Reference!Elongated! Frame  0 ! Frame  1 ! Frame  2 ! //  ! FRAME  N ! Elongated  !  45 inches of
! Burst   !Preamble !          !          !          !//   !          ! Post Amble !  Erased Media
```

11.1 REFERENCE BURST (Reference FIGURE 14)

A 16,000. frpi pattern, recorded at the beginning of TRACK 0, and extends from a point within 381 mm of the BOT Clear Leader to a point within 12 mm prior to, or 101 mm after the BOT HOLE.

11.2 FRAME FORMAT

Data Blocks are recorded in groups of 16 followed by two ECC blocks. A frame with less than 16 Data Blocks remains in cache until the balance of 16 data blocks are received, or a WRITE FILEMARK command is issued, or ONLINE SIGNAL is dropped. Writing a file mark or dropping the ONLINE SIGNAL will cause the remaining data blocks to be recorded in a short frame.

F R A M E L A Y O U T

```
          ! Block 0 ! Block 1 ! Block 2 ! Block 3 ! //  ! Block N ! FIRST ECC ! SECOND ECC !
          !         !         !         !         !//   !         !   BLOCK   !   BLOCK    !
```

11.3 BLOCK FORMAT

The Block Format specifies the structure for all recorded blocks. All fields specified in bytes are converted into 10 bit GCR Patterns, according to TABLE 2.

B L O C K L A Y O U T

```
!Preamble ! Block Sycn. !  DATA  FIELD  ! Control Field !  C R C   !Postamble!
!         !  (1 Byte)   !  (512 Bytes)  !  (5 Bytes)    ! (2 Bytes)!         !
```

The individual fields are described in the succeeding sections.

11.4 PREAMBLE AND POSTAMBLE

A pattern recorded at a fixed frequency (16,000. frpi) over fixed or variable lengths for the purpose maintaining read circuit synchronization between blocks. The lengths are normally kept to a minimum during streaming, and are extended to account for head spacing and velocity changes associated with repositioning, caused by underruns, and track (direction) changes. The lengths of and conditions under, which the various Preamble & Postambles are implemented, is tabulated as follows:

P R E A M B L E / P O S T A M B L E T A B L E

Type		field length in flux reversals at 16000 frpi Minimum	Nominal	Maximum	Condition when invoked
Normal	Preamble	120	140	300	All blocks (written normally)
Elongated	Preamble	3,500	5,000	7,000	First Block after repositioning
Long	Preamble	15,000	170,000	200,000	First Block on each track
Normal	Postamble	5	10	20	All blocks (written normally)
Elongated	Postamble	3,500	5,000	7,000	REWRITTEN Block (media defect) or Buffer Underrun or End of of Track

11.5 BLOCK SYNC. BYTE

A Unique 10 Bit pattern, (GCR Byte) following a preamble and preceeding a Data Field. In purpose this allows the data separator to synchronize at the beginning of a Data Block. It has the following GCR bit pattern, is never transmitted over the BUSS and has no equivalent in 8 bit BUSS pattern.

```
+-----------------------+
!  1 1 1 1 1  0 0 1 1 1  !
+-----------------------+
```

11.6 DATA FIELD

This field will always contain 512 Byte of data converted to and from GCR patterns per TABLE 2. This field may contain three types of data: "User Data", "File Mark Data" and "ECC Block Data". The file mark block data is not relevant.

11.7 CONTROL FIELD

This field contains control and addressing information used by the drive to manage data recovery and positioning control. The format is described as follows:

```
        I N F O R M A T I O N   B L O C K   C O N T R O L   F I E L D   (block type 0 or 1)
BYTE    +-------------------------------------------------------------+
        !   7   !   6   !   5   !   4   !   3   !   2   !   1   !   0   !
        +-------+-------+-------+-------+-------+-------+-------+-------+
00      +  Block Type  !           Frame       Address                  !
        +-------+-------+-------+-------+-------+-------+-------+-------+
01      +       Track Code      !       Physical    Block   Address     !
        +-------+-------+-------+-------+-------+-------+-------+-------+
02      +     CONTROL  NIBBLE           !                               !
        +-------+-------+-------+-------+       +       +       +       +
03      +           LOGOCAL BLOCK ADDRESS  (20 bits)                    !
        +       +       +       +       +       +       +       +       +
04      +                                                               !
        +-------+-------+-------+-------+-------+-------+-------+-------+

        E C C   B L O C K   C O N T R O L   F I E L D       (block type 3)
BYTE    +-------------------------------------------------------------+
        !   7   !   6   !   5   !   4   !   3   !   2   !   1   !   0   !
        +-------+-------+-------+-------+-------+-------+-------+-------+
00      +  Block Type  !           Frame       Address                  !
        +-------+-------+-------+-------+-------+-------+-------+-------+
01      +       Track Code      !       Physical    Block   Address     !
        +-------+-------+-------+-------+-------+-------+-------+-------+
02      +     CONTROL  NIBBLE           !       Rewrite  Count          !
        +-------+-------+-------+-------+-------+-------+-------+-------+
03      +                                                               !
        +       +   TAPE POSITION COUNTER  (16 bits)    +       +       +
04      +                                                               !
        +-------+-------+-------+-------+-------+-------+-------+-------+
```

11.7.1 Block Type: A block type is specified by a two bit value:

VALUE (decimal)	DESCRIPTION
3.	Indicates an ECC Block.
2.	RESERVED
1.	Information block immediately preceding an ECC block.
0.	Information block other than type 1.

11.7.2 Frame Address:
A cyclic address number of 0 thru 63 is used to specify the frame sequence. The beginning frame address on tape is 0. All the blocks in any one frame contain the same address.

11.7.3 Track code:
Contains the quotient obtained from the current track address divided by 4.

11.7.4 Physical block Address:
An address to indicate the block sequence within a frame, with a range of 0 thru 17 decimal.

11.7.5 Control Nibble: (Defination dependent on block type.)

Block Type 1 - 2: (Identifies type of Information Block)

- 0 Data Block
- 1 File Mark Block
- 2 -F RESERVED

Block Type 3 (ECC Block)

Contains the correction code for the ECC BLOCK, generated using same method as the data field.

11.7.6 Logical Block Number:
A sequence of consecutive numbers starting from 0, which numbers each block on medium, including DATA BLOCKS and FILEMARK BLOCKS. Any given block which is rewritten will contain the same logical block number.

11.7.7 Rewrite Count: Specifies the rewrite count of the "current ECC Block", The initial value for each of the ECC Blocks is 0. Successive values are incremented by 1 for each rewrite.

11.7.8 Tape Position: Specifies a time conversion value of the interval between the current frame and the beginning of the current track. The resolution is approximately 5 mS per count.

11.8 CRC

The Cyclic Redundancy Check (CRC), is a check sum of two bytes, or 20 GCR bits. The checking process covers the Data Bytes (512) and the Block Address bytes (4).
The generation polynominal is $X^{16} + X^{12} + X^5 + 1$ (initial value: all bits 1).

11.9 USER DATA BLOCKS

These blocks contain the recorded user data, in 512 byte blocks, and are consecutively numbered from the beginning of a partition.

11.10 FILE MARK BLOCKS

The format of a filemark block is identical to a data block. These blocks are included within the Logical Block addressing scheme.

11.11 ECC BLOCKS

Two Error Correction Code (ECC) Blocks are recorded at the end of each frame. These ECC blocks contain exclusive or immages of the even and odd block groups in the frame. The maximum number of valid blocks in any frame is 16.

11.12 END OF RECORDED DATA

This condition is defined as: any Valid Block followed by 45 inches of erased tape.

12.0 ERROR MANAGEMENT (WRITE)

The testing all written data employs a read after write verification, using a seperate read head immediately following the write head. This verification is a byte image compare, including the CRC.

With a spacing of 0.15 inches between the write and read heads, and a block length of approximately 0.33 inches, there are two geometric patterns that can exist during error detection.
First, an error in the current block being written and second, an error in the preceding block.

In the first condition, the current block is completed followed by that block re-written.
Further medium defects are anticipated to cover one or more block lengths, and for this reason the drive will switch postamble lengths, in order to avoid, re-writing consecutive block pairs.

In the second condition the writing of the current block is completed, but verification is omitted, as re-writing that block is necessary to maintain sequential order of all validated blocks.

All defective Blocks are re-written up to 15 times prior to reporting a HARD Write Error.

FIRST ERROR CONDITION

```
 !  V  !  E  !     !  E  !     !  E  !     !  V  !     !  V  !     !  V  !
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+---
 ! N-1 !  N  !  P  !  N  !  P  !  N  !  P  !  N  !  P  ! N+1 ! N+2 ! N+3 !
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+---

Note:  V   =  Block Verified                        !     !                 !
       E   =  Block  Failed                      T(N-1) T(N)             T(N+1)
       I   =  Verification Ignored                 t1     t2                t3
```

SECOND ERROR CONDITION

```
 !  V  !  E  !  I  !  E  !     !  E  !     !  E  !     !  V  !     !  V  ! V
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+---
 ! N-1 !  N  ! N+1 !  N  !  P  !  N  !  P  !  N  !  P  !  N  !  P  ! N+1 !N+2
---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+---

NOTE:  t1, t2, & t3 shows the READ HEAD             !     !                 !
       position when the indicated blocks        T(N-1) T(N)             T(N+1)
       are qualified for BUSS Transfer             t1     t2                t3
       during a READ operation.
```

13.0 ERROR MANAGEMENT (READ)

The drive checks CRC, Block Address, and control fields.
This includes a read ahead process, and automatic track/direction changes when necessary.

A block is qualified for transfer to the Host when:

a. The current block and it's subsiquent block have been CRC verified. (reference TABLE ??)
b. Current block is CRC veri ied followed by a validated FILE MARK BLOCK.
c. Current block is CRC verified followed by END OF RECORDED DATA detection.

Soft read errors are defined as any block failing the above qualification.
Automatic retries (up to 16) are performed prior to reporting a hard error.

Included within the 16 retry algorithm are fractional track offsets, in both directions.
When a successful offset is found it becomes the new track center for the remainder of the media, or until subsequent read errors lead to a new offset.
This process is limited to reads and a total of four offsets, two in each direction, about the initial center.

14.0 SIGNAL INTERFACE

14.1 CIRCUITS Input/output

Receivers	Type:	TTL 74LS14 or equivalent
	Termination	Split 220/330 ohm 5% (non removable)
Drivers	Type:	TTL 7438 or equivalent
	Termination	none
Bidirectional	Type:	Driver and receiver as specified above
	Termination	Split 220/330 ohm 5% (non removable)

The terminators are connected as follows: 220 ohm to +5 VCC and 330 to 5 volt ground.

14.2 ELECTRICAL CHARACTERISTICS Applicable at the drive end of the interface cable.

INPUT LEVEL	Low	(TRUE)	0.5 V Maximum
	High	(FALSE)	2.0 V Minimum to 5.25 Maximum
	LOAD	-0.4 mA (excluding terminator)	
OUTPUT LEVEL	Low	(TRUE)	0.5 V Maximum
	High	(FALSE)	2.4 V Minimum to 5.25 Maximum
	Sink Current	Capable of driving up to 48 mA. (excluding the bidirectional terminator)	

14.3 CONNECTOR

The drive is supplied with a 50 pin edge card connector, with a key slot located between pins 4 & 6. (reference FIGURE 13)

14.4 CABLE

Twisted pair or ribbon type, 3M part number 3415-0001, or equavilent. Maximum length: 3 meters.

14.5 SIGNAL DEFINITIONS

This section describes the Input/Output Control Signals. All signals are true "ACTIVE LOW".

RESET (RST) Type: Unidirectional Direction: Host to Target

Resets the drive to an INITIALIZED condition equal to a power on reset.

a. Resets: ACKNOWLEDGE, READY and DIRECTION.
b. Sets: EXCEPTION
c. Terminates: all commands.

ONLINE (ONL) Type: Unidirectional Direction: Host to Target

Used by the Host to set and or hold a DRIVE ONLINE.

Single Partition: (power on default) ONLINE must remain active during all WRITE or READ operations. Droping ONLINE at the completion of an operation will:

WRITE: Flush the buffers to medium, WRITE a FILEMARK, Rewind to BOT.
READ: Rewind to BOT.

Dual Partition Mode: ONLINE must remain active during any WRITE or READ operation which begins at BOT. Droping ONLINE at the completion of an operation will have no affect on current media position or buffer pointers. Subsequent READ CONTINUE, or WRITE operations are NOT dependent uppon ONLINE being ACTIVE. Write buffers are flushed by issuing a REWIND Command, or issuing a WRITE FILEMARK Command. In either case the Host allegance to error reporting, is OPEN until the medium successfully reaches BOT.

REQUEST (REQ) Type: Unidirectional Direction: Host to Target

Controls the transfer of Commands to the target, or Status from the target, in a handshake control with the READY (RDY) line.

Qualification: Ready (RDY) must be true for command transfer. EXCEPTION (EXC) true Read Status must be transfered.

Exclusion: When EXCEPTION (EXC) is True all commands apart from READ STATUS are rejected.

READY (RDY) Type: Unidirectional Direction: Target to Host

a. Controls the transfer of Commands to the target, or Status from the target, in a handshake control with REQUEST (REQ) line.
b. Informs the Host that one of the following commands have been completed.
REWIND
ERASE TAPE (The entire tape)
PREWIND
SELECT Nth TRACK (This command may not exist)
WRITE FILEMARK
WRITE FILEMARK without ERASE
Implicit REWIND (executed via host droping ONLINE during a WRITE operation).

c. Informs the Host that a 512 Byte DATA BLOCK was transfered to or from the Target Buffers. At this point Data transfer may continue or a New command may be issued.

WRITE
WRITE (w/o ERASE)
READ

Exclusion: When EXCEPTION (EXC) is True all commands apart from READ STATUS are rejected.

TRANSFER (XFR) Type: Unidirectional Direction: Host to Target

Controls the transfer of WRITE and READ DATA to and from the target, in a handshake control with the ACKNOWLEDGE (ACK) line.
During a WRITE operation it indicates that the host has data setup on the bus.
During a READ operation it indicates that the host has removed data from the bus.

ACKNOWLEDGE (ACK) Type: Unidirectional Direction: Target to HOST

Controls the transfer of WRITE and READ DATA to and from the target, in a handshake control with the TRANSFER (XFR) line.

During a READ operation it indicates that the drive has data is setup on the bus.
During a WRITE operation it indicates that the drive has removed data from the bus.

EXCEPTION (EXC) Type: Unidirectional Direction: Target to HOST

A request from the Drive that a status is pending.
This signal is reset by a READ STATUS command, all other commands will be rejected.

DIRECTION (DIR) Type: Unidirectional Direction: Target to HOST

Informs the Host as to the valid direction of data over the buss, per the states as described:

True: STATUS and READ data can be transfered to the HOST.
False: COMMAND and WRITE data can be transfered to the DRIVE.

DATA BUS Bit 7-0 (HB7 - HB0) Type: Bidirectional

A common bus for READ, WRITE, STATUS, and COMMAND data. HB7 is the Most Significant Bit (MSB), and HB0 is the Least Significant Bit (LSB).

BUS PARITY (HBP) Type: Bidirectional

Parity is ODD. The drive ALWAYS generates PARITY, for STATUS and READ DATA. The drive TEST PARITY, for COMMAND and WRITE DATA, only if TP7 and TP8 are shorted together, else it will ignore PARITY.

14.6 TIMING

All timing s specified within FIGURES 1 thru 10, located in APPENDIX A.

15.0 COMMANDS

15.1 OP CODE11 Select Drive (lock) Mnemonic SDL

This command enables the front panel light circuit, such that the light will be ON when ever a CASSETTE is in place. This state will remain TRUE until either an SD command, or a drive RESET is issued.

COMMAND FORMAT: Single byte OP Code.

RETURNED STATUS: Illegal command if not issued at BOT.

POWER UP DEFAULT: SD mode.

INTERFACE TIMING: Standard single byte command, per APPENDIX A

15.2 OP CODE 01 Select Drive (without lock) Mnemonic SD

This command turns OFF the condition set by the SDL command.

COMMAND FORMAT: Single byte OP Code.

RETURNED STATUS: Illegal command if not issued at BOT.

POWER UP DEFAULT: SD mode.

INTERFACE TIMING: Standard single byte command, per APPENDIX A.

NOTE: A drive set in the SD (or default) MODE will turn on the front panel lamp during all oprerations, which cause actual tape motion.
Reading operations which are successful through buffer transfers, without tape motion will NOT illuminate the front panel light.
This is also true for redundant BOT operations.
The OP codes and Mnemonics are carried over from an interface option which is NOT supported in this product.

15.3 OP CODE 21 Rewind (to BOT) Mnemonic BOT

This command moves the tape from current position to the physical beginning of tape and the logical beginning of track zero (0). If following a WRITE operation all filled buffers will be flushed to medium prior to rewinding the medium.

COMMAND FORMAT: Single byte OP Code.

RETURNED STATUS: Illegal command if drive not "READY". Appropriate Write status if buffers contained valid information from a previous WRITE operation.

POWER UP DEFAULT: Drive Automatically rewinds to BOT.

INTERFACE TIMING: Standard single byte command, per APPENDIX A.

NOTE: The host has an alligence to WRITE error status for several seconds after issuing this command. It has an extended alligence for media damage, until command completion, (at BEGINNING OF MEDIUM), drive "READY", and no "EXCEPTION".

15.4 OP CODE 22 Erase (entire tape) Mnemonic ERA

This command erases the entire tape, from BOM, including all tracks and partitions. It may be issued from any point on the medium, with a prerequisite of drive "READY".

COMMAND FORMAT: Single byte OP Code.

RETURNED STATUS: Illegal command if Tape not at BOM & "READY". Command complete drive "READY". Successful Status "BOM" 00 88 Hex in status bytes 0 & 1 respectively.

POWER UP DEFAULT: NO Affect.

INTERFACE TIMING: Standard single byte command, per APPENDIX A.

15.5 OP CODE 24 Prewind (retension) Mnemonic PRW

This command moves the tape from current position to the physical end of medium and back to the physical beginning of medium or logical beginning of track zero (0).

COMMAND FORMAT: Single byte OP Code.

RETURNED STATUS: Illegal command if drive not "READY". Appropriate Write status if buffers contained valid information from a previous WRITE operation.

POWER UP DEFAULT: NO Affect.

INTERFACE TIMING: Standard single byte command, per APPENDIX.

NOTE: This commands prime purpose is to prepare a medium for write and read operations. It will help to remove the affects from intermittent motions, and long term storage, both of which can cause abnormal speed variations.

15.6 OP CODE 40 Write Command (with erase) Mnemonic WRT

This command Writes from current position om the medium.
The single byte OP Code is transfered via a REQUEST/READY handshake, and followed by data transfer via Transfer (XFER)/Acknowledge (ACK) handshake.
The prerequisites are: Drive "READY", "ONLINE" true.

RETURNED STATUS:
The "EXCEPTION" line set if manditory status is pending. Directional or inquiry status may be accessed any time following "READY" true.
The Drive will make the appropriate changes in control lines and complete the status transfer prepared for another WRITE operation or valid command.
It is possible to collect STATUS or EXTENDED STATUS, while the drive is recording the current buffers on medium.
Approximately 18 mS is available if all three (3) buffers are full.
To avoid a soft or hard underuns the host must complete the READ STATUS, and issue the next WRITE transfer of 1 Block minimum, prior to the last buffer being transfered to medium.

POWER UP DEFAULT:
Will cause the drive to WRITE in the Industry standard (17 or 9 track) Mode. Will cause the drive to WRITE from Beginning of Medium (BOM).

CARTRIDGE INSERTION:
Will cause the drive to WRITE in the Previously SET Mode.
Will cause the drive to WRITE from Beginning of Medium (BOM).

ERASE FUNCTION:
The erase head will be ON while writing on track zero (0). Any partial track writes (less than 5 Megabytes approximately from BOM) will result in a minimum of 45 inches of erase media following tke last block recorded.

INTERFACE TIMING: Per APPENDIX A.

SUCCESSFUL COMPLETION:
The completion of modullo 512 byte transfers, and a returned "READY" status from the drive.

UNDERUN:
Any time the host fails to keep up with with the Drive and all 3 buffers of (512 bytes) have been transfered to medium the drive stop and reposition the medium in preperation to continue the transfer. Following an underun the next two (2) blocks transfered to buffers, without tape motion. A third block transfer will trigger tape motion, and begin the transfer to medium when the next logical block on medium is located.

NOTE:
1. It is the host responsibility to track partial blocks, via the READY signal.

2. The DATA ERROR & UNDERRUN COUNTERS, are useful mechanisms to monitor the host performance as well as the media quality. Caution both counters are resset by: Reading Status via RSTU command.

15.7 OP CODE 42 Write Command (WITHOUT ERASE) Mnemonic WRTE

This command is identical to the "Write command OP Code 40 with the following difference. The erase head is TURNED OFF on all tracks.

It's singular intention is to allow recording on a partitioned medium, with out erasing information within an adjacent partition.

APPLICATION NOTE: This command is to be used in the following manner:

1. Prior to Writing from BOM (partition 0), tke entire tape is to be erased.
2. The drive is to be set in the Dual Partition mode.
3. WRITE and WRITE appends may be performed within either partition, and in any order. Refer to "LOCATE" and "SEEK END OF RECORDED DATA" commands.

15.8 OP CODE A3 Seek end of Recorded Data Mnemonic SEOD

This command searches tape tape from current position for the "End of Recorded Medium". When found "EXCEPTION" goes true, and the status read, will indicate current track, block and Physical end of media if true.
On a full tape (9 tracks recorded) without FILEMARKS will take in excess of 10 minutes.

COMMAND FORMAT: The single byte OP Code is transfered via a REQUEST/READY handshake. Two prerequisites are: Drive "READY", "ONLINE" true.

APPLICATION NOTE: All data blocks encountered during this process are (CRC) Cyclic Redundancy Checked, standard read retries are performed as well as soft error logging and Hard error reporting.

15.9 OP CODE 60 Write FILEMARK Command Mnemonic WFM

This command Writes from current logical position om the medium. All buffers holding data from a previous write operation are recorded on medium prior to recording the FILEMARK BLOCK.

COMMAND FORMAT: The single byte OP Code is transfered via a REQUEST/READY handshake. The prerequisites are: Drive "READY", "ONLINE" true.

ERASE FUNCTION: The erase head will be ON while writing a FILEMARK on track zero (0). Any partial track writes from BOM will result in a minimum of 45 inches of erase media following tke last block recorded.

15.10 OP CODE 80 Read Command Mnemonic RD

This command Reads from current position om the medium.
The single byte OP Code is transfered via a REQUEST/READY handshake, and followed by data transfer via Transfer (XFER)/Acknowledge (ACK) handshake.
The prerequisites are: Drive "READY", "ONLINE" true.

COMMAND FORMAT: Single byte OP Code.

RETURNED STATUS: The "EXCEPTION" line set if manditory status is pending.
Directional or inquiry status may be accessed any time following "READY" true.
The Drive will make the appropriate changes in control lines and complete the status transfer prepared for another WRITE operation or valid command.
It is possible to collect STATUS or EXTEND D STATUS, while the drive is reading into the current buffers from medium.
Approximately 18 mS is available if all three (3) buffers are full.
To avoid hard underruns the host must complete the READ STATUS, and issue the next READ transfer of 1 Block minimum, prior to the last buffer being filled from a medium transfer.

POWER UP DEFAULT: Will cause the drive to READ in the Industry standard (17 or 9 track) Mode.
Will cause the drive to READ from Beginning of Medium (BOM).

CARTRIDGE INSERTION: Will cause the drive to READ in the Previously SET Mode.
Will cause the drive to READ from Beginning of Medium (BOM).

INTERFACE TIMING: Per APPENDIX A.

SUCCESSFUL COMPLETION: The completion of modullo 512 byte transfers, and a returned "READY" status from the drive.

OVERUN: Any time the host fails to keep up with with the Drive and all 3 buffers of (512 bytes) have been filled the drive stop and reposition the medium in preperation to continue the transfer.
Following an overun the next two (2) blocks transfered will come form buffers, without tape motion.
A third block transfer will trigger tape motion, and begin to fill the buffers when the next logical block on medium is located.

NOTE:
1. It is the host responsibility to track partial blocks, via the READY signal.
2. The DATA ERROR & UNDERRUN COUNTERS, are useful mechanisms to monitor the host performance as well as the media quality.
Caution both counters are resset by: Reading Status via RSTU command.

15.11 OP CODE CO Read Status Command Mnemonic RSTU

This command requests 6 bytes of status from the drive. Transfered via a REQUEST/READY handshake.

COMMAND FORMAT: Single byte OP Code. CO

DATA RETURNED:

BYTE	7	6	5	4	3	2	1	0
00	STO	CNI	res[0]	WRP	EOM	UDE	BNL	FIL
01	ST1	ILL	NDT	res[0]	BOM	BPE	res[0]	POR
02	DATA ERROR COUNTER (MSB)							
03	DATA ERROR COUNTER (LSB)							
04	UNDERRUN COUNTER (MSB)							
05	UNDERRUN COUNTER (LSB)							

POWER UP DEFAULT: POR Set to a 1
INTERFACE TIMING: per APPENDIX A.
Exception Status: Two byte field with bit weighted status information, per TABLE 1.

RES THESE BITS ARE RESERVED AND ALWAYS ZERO.
STO A status bit is set in byte 0
ST1 A status bit is set in byte 1
FIL File Mark Detected, during a Read or Read Rile Rark operation.
CNI Cassette not in Place
WRP Write Protected (Cassette)
EOM End of Media This bit will remain set as long as the condition persists.
Early warning hole detected during a RD, RFM, WRT, or WFM operation. Bit set after current command termination. Additional writing is prohibited, however the blocks in cache will be fixed to media. Additonal reading is allowed if information exists on tape.
UDE Unrecoverable Data Error Is set by the following, and may be cleared by executing RSTU command.
1. When mechanical retries exceed 16,
2. ECC required and not possible (during read only)
3. Repositioning error detected during a WRITE.
4. Gap of more than 35 inches detected during a READ.

BNL Bad Block Not Located Is set by the following, and may be cleared by executing RSTU command.
1. When mechanical retries exceed 16,
2. ECC required and not possible (during read only)
3. Repositioning error detected during a WRITE.

ILL Illegal Command.

An Illegal command was sent to the drive. May be cleared by issuing RSTU command. The known conditions are described as follows:

1. SD or SDL Command issued after a WRITE or READ operation and prior to completing a rewind.
2. WRT, RD, WFM or RFM issued while ONL signal is false
3. Command other than WRT, WFM, RSTU, ESTA, BOT or PRW is issued during the execution of a WRT command.
4. Command other than RD, RFM, ESTA, BOT or PRW is issued during the execution of a RD command.
5. Command not in the drives command table was issued.
6. Command other than RSTU or ESTU is issued while FAULT STATUS BITS are set.
7. When BOM bit = 0 and WRT or WRTFM command is issued prior to NO DATA DETECTED.
8. When WRT, WFM or ERA commands are issued with the WRP BIT = 1.
9. When cassette is loaded, BOM bit = 0 and the STM command is issued.
10. When the TEST or MAINTENANCE commands are issued in the normal mode.
11. When a normal command is issued in the TEST mode.

NDT No data detected Set when a gap of more than 35 inches is detected during a Read or Read File Mark operation. May be cleared by executing RSTU command.

BOM Beginning of Media Tape logically positioned at beginning of track 0.

BPE Buss Parity Error Enabled when S3 strap is shoretd. May be cleared by issuing RSTU command. The drive checks commands and data from the Host for odd parity. During a WRITE all blocks prior to the error are fixed to media. Commands issued with parity errors are not executed. The drive sets EXCEPTION TRUE and sets the BPE bit.

POR POWER ON or BUS RESET occured.

DATA ERROR COUNTER: RESET by a Write or Read operation from BOT.
An unsigned 16 bit value containing the total number error blocks, resulting from any READ or WRITE operation. This count is incremented by 1 for each error as defined:
Write: Any block rewritten up to 15 times, will increment by that number.
Read: Any read retry up to 16 times, will increment by that number.

UNDERRUN COUNTER: RESET by a Write or Read operation from BOT.
An unsigned 16 bit value containing the total number of underruns/overruns, resulting from any READ or WRITE operation. This count is incremented by 1 for each condition as defined:

Underrun: During a Write, any time the host fails to keep up with the Drive and all filled buffers (512 bytes) have been written and verified on medium.
overrun: During a Read,any time the host fails to keep up with with the Drive and all available buffers (512 bytes) have been filled.

NOTE: It is the host responsibility to track partial blocks, via the READY signal.

15.12 Read Extended Status Command   Mnemonic   ESTU   OP CODE   C8

This command requests 6 bytes of status from the drive. Transfered via a REQUEST/READY handshake.

COMMAND FORMAT: Single byte OP Code.

DATA RETURNED:

BYTE	7	6	5	4	3	2	1	0
00	TID	0	0	WID	0	Identity Code [7]		
01	FSB	RES	RES	MFAULT	STL	EOT/BOT	RES	RES
02	FMT	reserved [0]		CURRENT TRACK NUMBER				
03	Current Block address (MSB)							
04	Current Block Address							
05	Current Block Address (LSB)							

RETURNED STATUS: READY if command accepted else EXCEPTION.
POWER UP DEFAULT: Self tests are executed at the POWER UP, or after a drive RESET SIGNAL (PIN32).
TID Tape Identification Set for media type: 1 for 17 teack media, 0 for 9 track media.
WID Wang Identification A bit set for unique "WANG IDENTIFICATION"
Identy Code: A 3 bit Reserved field to identify the drive type. (value 7 Hex)
INTERFACE TIMING: Per APPENDIX A.

Fault Status: Byte 01: All values and the associated EXCEPTION "signal", may ONLY be reset by issuing a Drive RESET via the RESET Signal Pin 32. Reading Status or Extended Status will NOT clear these bits, or the associated EXCEPTION "signal".

FSB One or more bits set in Fault Status Byte.
RES Reserved Bit.
MFAULT Drive detected BUFFER fault, after POWER-UP or after a RESET.
STL STALL, Tape did not move within specification, after motion command issued to drive.
EOT/BOT Drive failed to detect; EOT, BOT, or Clear Leader.

MODE: Operational Mode of the Drive:
0 Single Partition (power up Default)
1 Dual Partition Mode see Select Format Command.

RESERVED: Three Bit field fixed value (0).

Track Number: Bits 3 and 0 correspond to msb and lsb respectvely of the current track address.

Current Block Address: A 24 Bit field specifying the current block for the next operation.

15.13 Read Extended Status Command Mnemonic B thru E

This command requests 6 bytes of status from the drive.
Transfered via a REQUEST/READY handshake.

COMMAND FORMAT: Single byte OP Code.

EXTENDED STATUS B OP CODE C9 hex

BYTE	7	6	5	4	3	2	1	0
00	ERROR BLOCK COUNTER (MSB)							
01	ERROR BLOCK COUNTER (LSB)							
02	TAPE POSITION COUNTER (MSB)							
03	TAPE POSITION COUNTER (LSB)							
04	CORRECTED ERROR COUNTER (MSB)							
05	CORRECTED ERROR COUNTER (LSB)							

ERROR BLOCK COUNTER

Indicates in accumulative count the number of logical blocks which were rewritten one or more times, during a Write or Read operation. A given logical block is counted once reguardless of how many times it is duplicated. The count is reset at BOM following a WRT, RD, WFM, or RFM OPERATION.

TAPE POSITION COUNTER:

A 16 bit unsigned value the current travel position from BOT on each track.
The time interval is approximately 6 mS per count.
The approximate distance from Beginning of current track to current position in inches is:
$L_r = 90 (0.05) (N)$, where N is the counter value.
If tape is run from BOT, measurement is done by all commands.
The BOT, ERA, PRW commands stop measurement at the EOT hole.
The error due to repositioning during a Write operation is +/- 1 count per reposition.

CORRECTED ERROR COUNTER:

In a read operation this counter indicates the number of blocks where error correction is performed.
The counter is reset following a WRT, WFM, RD or RFM command from BOM.

EXTENDED STATUS C   OP CODE CB hex

BYTE	7	6	5	4	3	2	1	0
00	TRACK 0 DATA ERROR COUNTER							
01	TRACK 1 DATA ERROR COUNTER							
02	TRACK 2 DATA ERROR COUNTER							
03	TRACK 3 DATA ERROR COUNTER							
04	TRACK 4 DATA ERROR COUNTER							
05	TRACK 5 DATA ERROR COUNTER							

DATA ERROR COUNTER:

The number of WRITE retries per track, or the number of READ retries per track, in an 8 bit unsigned value. The counters are reset following a WRT, WFM, RD or RFM command from BOM.

EXTENDED STATUS D   OP CODE CC hex

Identical to EXTENDED STATUS C command, except bytes 0 thru 5, contain the DATA ERROR COUNTERS for tracks 6 thru 11 respectively.

EXTENDED STATUS E   OP CODE CD hex

BYTE	7	6	5	4	3	2	1	0
00	TRACK 12 DATA ERROR COUNTER							
01	TRACK 13 DATA ERROR COUNTER							
02	TRACK 14 DATA ERROR COUNTER							
03	TRACK 15 DATA ERROR COUNTER							
04	TRACK 16 DATA ERROR COUNTER							
05	RESERVED							

15.14 Select Format Command   Mnemonic   FMT   OP CODE   Fn

This command allows the host to select the recording format modes. This command will only be accepted (valid) with tape rewound to BOT Track 0, or "NO CASSETTE in the drive".

Industry Mode: Records all information per the D/CAS-12 standard which is as described within this specification, with the following exceptions:

1. Exclusion of "LOCATE BLOCK n" command.
2. All recording is in a single partition 9 track mode.
3. When "ONLINE" signal is dropped (false), the drive will flush the data buffer, "WRITE a FILEMARK", and rewind to Beginning of tape (BOT).
4. When "ONLINE" signal is dropped (false), following a "WRITE a FILEMARK" Command, the drive will rewind to Beginning of tape (BOT).

Dual Partition Mode: There are modifications to the CONTROL AND DATA BLOCK STRUCTURES, to allow the implementation of "LOCATE BLOCK" command, and partitioned media features. There is also a modification to the "ONLINE PROTOCAL" to enable multi tasking operations.

1. All recording is in a two partition mode, refer to the "LOCATE COMMAND".
2. When "ONLINE" signal is dropped (false) following a: WRITE OPERATION: The drive will flush the WRITE BUFFER to medium and maintain the current LOGICAL POSITION on tape, allowing the host to process error status for the current transfer. Subsequent WRITE operation(s) will Append at the End of recorded Medium. READ OPERATION; The drive will maintain current logical position, for subsequent valid read, write or position commands.

COMMAND FORMAT: Single byte OP code Fn, with the value of n used to select the following modes:

FORMAT MODES: FMT   Value of 0 will select Industry recording mode.
Value of 1 will select dual partition mode
The drive will remain in the selected mode until reselected or a power up reset occures. The drive will perform all subsequent commands in the selected state. The current mode may be detected via EXTENDED STATUS "MODE" bit, see SECTION 15.13.

POWER UP DEFAULT MODE: Shall be Industry RECORDING Mode.

STATUS RETURNED: Successful: Command completition will return with drive Ready.
Un-successful: Any conflicts which occure between the drive and media format will result in an EXCEPTION set and return status:

Conflict State	Results	STATUS BYTES Byte 0	Byte 1
Tape not at BOT	"Illegal Command not at BOT"	xxxx 0000	1100 0000
Read	"Read Abort"	100x 0100	1000 1000
Locate	"Read Abort"	100x 0100	1000 1000
Seek end of Data	"Read Abort"	100x 0100	1000 1000
NO Cassette	"No Cassette"	110x 0000	0000 0000

NOTE: "x" may be 0 or 1 value.

15.15 Locate command   Mnemonic   LBN   OP CODE   AD   Functional in the Wang Mode ONLY

This command allows the host to position the medium to any specified logical block, which exists within a contiguous string starting from the beginning of any partition. Upon successful completion the next block to be read will be that LOGICAL block. The block is located by a series of successive approximations, in such a manner as to minimize the random block access time.

COMMAND FORMAT: Single byte opcode AD followed by a four (4) byte ADDRESS BLOCK, using a REQUEST/READY hand shake.

The format of the ADDRESS BLOCK is described as follows:

```
BYTE   +-----------------------------------------------------------+
       !   7   !   6   !   5   !   4   !   3   !   2   !   1   !   0   !
       +-------+-------+-------------------------------+-------+-------+
 00    !                     P A R T I T I O N                         !
       +-------+-------+-------------------------------+-------+-------+
 01    !           LOGICAL BLOCK ADDRESS  (MSB)                        !
       +-----------------------------------------------------------+
 02    !       +       +       +       +       +       +       +       !
       +-----------------------------------------------------------+
 03    !           LOGICAL BLOCK ADDRESS  (LSB)                        !
       +-----------------------------------------------------------+
```

PARTITION: An 8 Bit field specifying the track group (partition) to be selected, prior to searching for the logical block specified, by the LOGICAL BLOCK FIELD. Partition (0) will select tracks 0 through 15. Partition (1) will select track 16.

LOGICAL BLOCK ADDRESS: A 24 Bit field specifying the ADDRESS of the LOGICAL BLOCK to be LOCATED. The first Logical Block in any partition is 0_H in order to maintain interchangeability with future interfaces.

RANGE of ADDRESSABLE BLOCKS: 00 00 00 $_H$ – FF FF FF $_H$ Media dependent.

Status Returned: Successful command will return with drive READY. Un-successful command will return with EXCEPTION and the appropriate bits set in the status bytes:

Results	STATUS BYTES Byte 0		Byte 1	
"Read Abort"	100x	0100	1000	1000

APPLICATION NOTES:

1. PREREQUISITE to the successful use of this command: a means to assure that the logical target block exists within a contiguous string of blocks starting from BOT of the specified partition.
   This may be accomplished by:
   a. Erasing the entire tape prior to the first recording on the media.
   b. All WRITES from BOT, or WRITE APPENDS from end of recorded data with "WRITE without ERASE", OP Code 42h.
   c. NO FILEMARKS are recorded on the media.
   d. The host does NOT ACCESS a block outside the range of a contiguous string within a specified partition.
2. WRITE APPEND: A write append operation requires the medium to be logically positioned immediately after the last recorded block within a partition. This may be accomplished by:
   a. LOCATING a block at or near the end of recorded media, followed by a SEEK END OF DATA (SEOD Hex A3) Command. A return status of (88 00 H) in Status Bytes 0 & 1 will qualify a write append.
   b. Issuing a SEEK END OF DATA (SEOD Hex A3) from any valid position within the partition, with the returned of (88 00 H). This method can take up to 8 minutes to process.

15.16 OP CODE A0 Read FILEMARK Command Mnemonic RFM

This command searches tape tape from current position searching for the first occurance of a FILEMARK. If one is found "EXCEPTION" goes true. and the filemark bit is set in status byte 0. If no FILEMARK is found the process will be terminated by the detection of END of RECORDED DATA (35 inches of erased tape), or PHYSICAL END of MEDIA (clear leader at the end of last track), which ever comes first. On a full tape without FILEMARKS will take in excess of 20 minutes.

COMMAND FORMAT: The single byte OP Code is transfered via a REQUEST/READY handshake. The prerequisites are: Drive "READY", "ONLINE" true.

APPLICATION NOTE: All data blocks encountered during this process are (CRC) Cyclic Redundancy Checked, standard read retries are performed as well as soft error logging and Hard error reporting.

15.17 OP CODE Bn Read "N" FILEMARKS Mnemonic RFM(N)

This command is identical to the READ FILEMARK COMMAND with the following difference. The OP Code has a total 16 values B0 through BF hexidecimal, specifying the number of filemarks to be spaced over. The OP Code value B0 specifies 16 decimal FILEMARKS. All data blocks encountered during this search are CRC verified.

TABLE 1
EXCEPITON STATUS BYTE SUMMARY

STATUS BYTE 0								STATUS BYTE 1									
STO	CNI		WRP	EOM	UDA	BNL	FIL	ST1	ILL	NDT		BOM	BPE	0	POR		DESCRIPTION
1	1		X														NO Cassette
1			1					X				X					Write Protected
1				1													End of Media
1			X		1			1				1					Read or Write Abort
1			X	X	1	1											Read Error, Filler Block XFERED.
1			X		1	1		1		1							Read Error No Data
1			X	1	1	1		1		1							Read Error No Data & EOM
1			X	X			1										File Mark detected
X	X		X					1	1			X					Illegal Command
X	X	X	X					1				X			1		Power ON / RESET
X	X		X					1				X	1				Bus Parity Error
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	F A U L T

NOTE: All BLANK BIT fields are zero (0) filled.
The value "X" may be zero or one.

TABLE 2

GROUP CODE RECORDING CONVERSION TABLE

Host Buss data				
B3 / B7	B2 / B6	B1 / B5	B0 / B4	Hex Value
				0
			1	1
		1		2
		1	1	3
	1			4
	1		1	5
	1	1		6
	1	1	1	7
1				8
1			1	9
1		1		A
1		1	1	B
1	1			C
1	1		1	D
1	1	1		E
1	1	1	1	F

RECORDED DATA (on tape)					
G4	G3	G2	G1	G0	Hex Value
1	1			1	19
1	1		1	1	1B
1			1		12
1			1	1	13
1	1	1		1	1D
1		1		1	15
1		1	1		16
1		1	1	1	17
1	1		1		1A
	1			1	09
	1		1		0A
	1		1	1	0B
1	1	1	1		1E
	1	1		1	0D
	1	1	1		0E
	1	1	1	1	0F

All data is converted on a nibble bound according to TABLE 2.
Please note that all zeros within the binary portion of the table have been omitted for readability.

This recording scheme is one of the common Group Code Recording (GCR) techniques use in the industry and is refered to as the 5 bit to 4 bit conversion, or 0-2 Run Length Limited (RLL) code. Recording in this method using 5 bits per nibble allows half the absolute binary combinations to be disgarded. The combinations chosen for recording are those which yield the minimum bit shift, and thus increase the reliability.

The (5-4) conversion relates to the range in number of bits recorded on the medium versus those transfered over the BUSS, for a given "nibble".

The (0-2) RLL description relates to the number of consecutive zeros recorded on medium, within any string of bytes or nibbles.

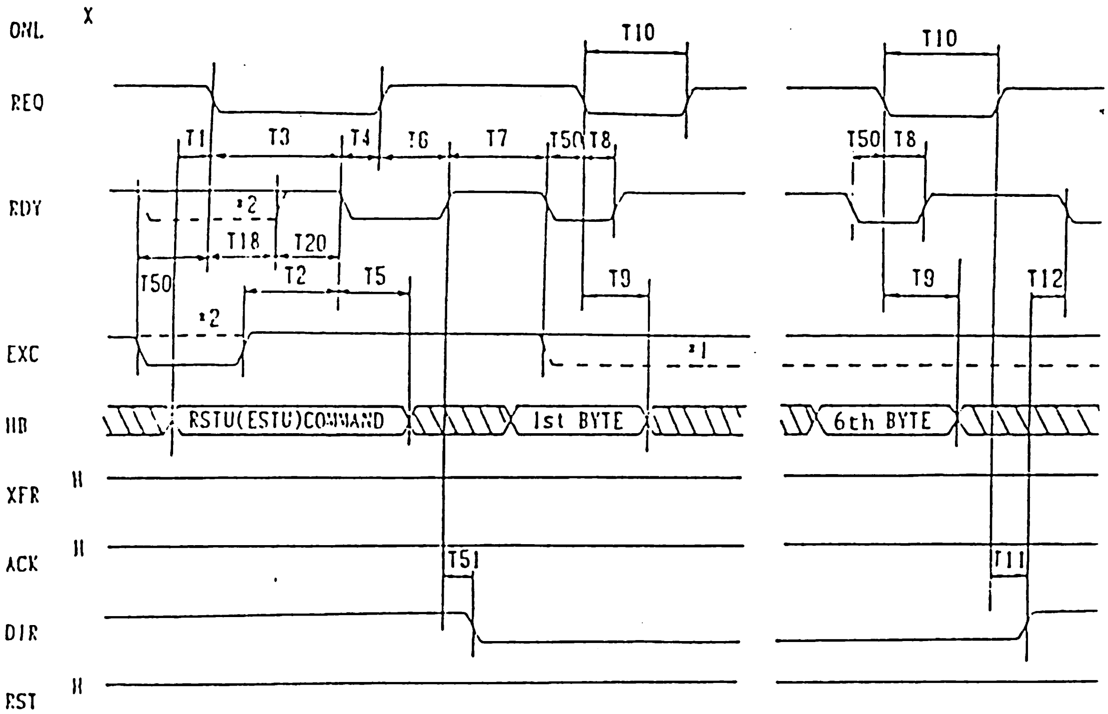
A deviation to the above is the unique synchronization "Data Block Marker" [11111 00111], which preceeds all recorded data field.

In the application used, they do not violate the above rules, as the data block marker is always 10 bits and bounded by 1's. and the filemark pattern is always 10 bits, preceeded by a "Data Block Marker".

TABLE 4

SIGNAL CABLE PIN ASSIGNMENTS

Pin Number		SIGNAL NAME	MNEMONIC	SOURCE
GND	Signal			
01	02			
03	04			
05	06			
07	08			
09	10	HOST BUSS (ODD PARITY)	HB-P	
11	12	HOST BUSS BIT-7 (MSB)	HB-7	
13	14	HOST BUSS BIT-6	HB-6	Bi-directional
15	16	HOST BUSS BIT-5	HB-5	
17	18	HOST BUSS BIT-4	HB-4	
09	20	HOST BUSS BIT-3	HB-3	
21	22	HOST BUSS BIT-2	HB-2	
23	24	HOST BUSS BIT-1	HB-1	
25	26	HOST BUSS BIT-0 (LSB)	HB-0	
27	28	ONLINE	ONL	HOST
29	30	REQUEST	REQ	
31	32	RESET	RST	
33	34	TRANSFER	XFR	
35	36	ACKNOWLEDGE	ACK	DRIVE
37	38	READY	RDY	
39	40	EXCEPTION	EXC	
41	42	DIRECTION (RENAMED I/O)	DIR	
43	44			
45	46			
47	48			
49	50			



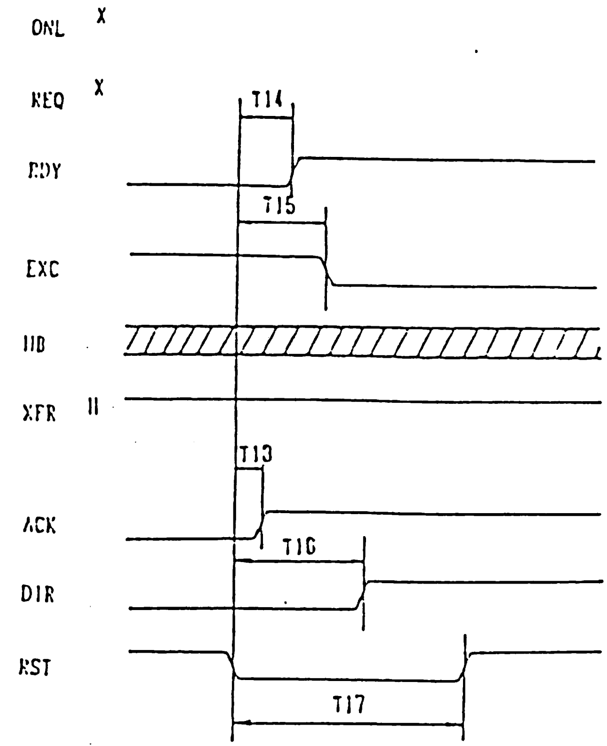
Sym-bol	MIN	NOM	MAX	Unit
T1	0			
T2	10			μS
T3	20	500		μS
T4	0			
T5	0			
T6	20		100	μS
T7	20			μS
T8			1	μS
T9	0			
T10	20			μS
T11	0			
T12	0			
T50	0			nS
T18			1	μS
T20	20	500		μS
T51	0	350		μS

Note: 1 The doted line *1 indicates the EXCEPTION signal timing following an OP CODE BUS PARITY error detected by the drive. In this case, the status information following the OP code is not transfered.

2 The dotted line *2 indicates the timing in the case where RSTU or ESTU commands are issued with READY active.

RSTU, ESTA thru ESTE COMMAND TIMING

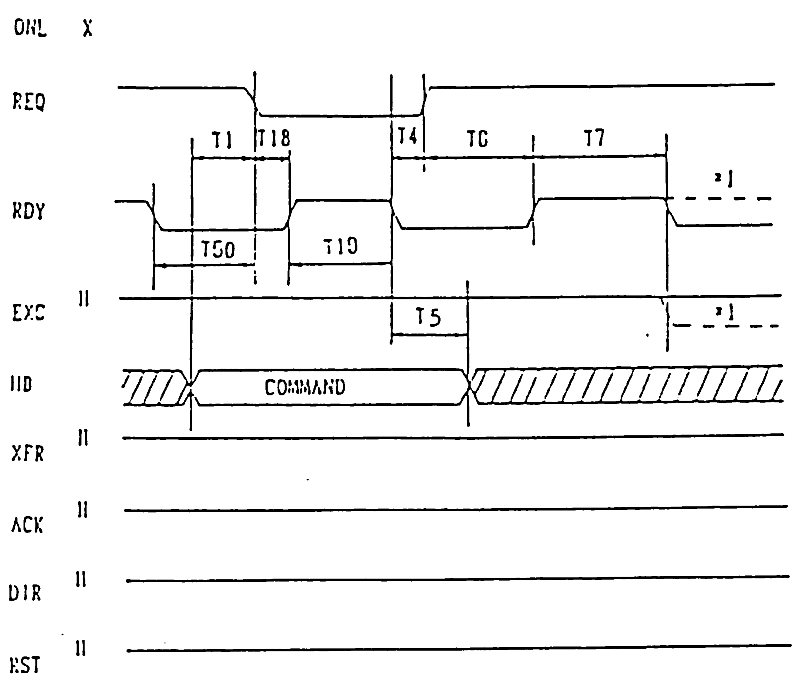
Figure 1



Symbol	MIN	NOM	MAX	unit
T13			1	μS
T14			1	μS
T15			3	μS
T16			3	μS
T17	25			μS

RESET TIMING

Figure 2

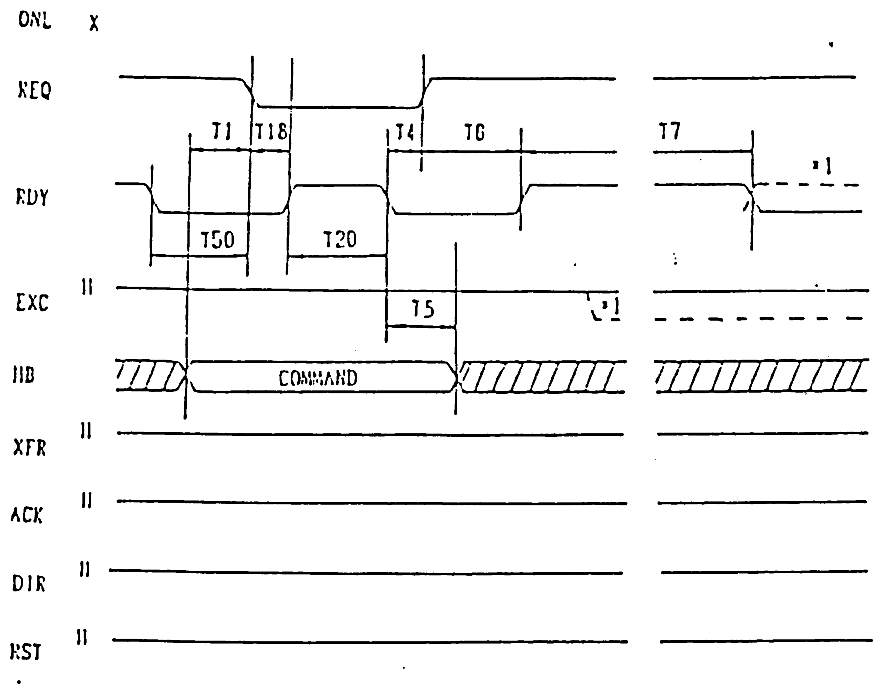


Symbol	MIN	NOM	MAX	Unit
T1	0			
T18			1	μS
T19	50	500		μS
T4	0			
T5	0			
T6	20		100	μS
T7	20			μS
T50	0			nS

Note: The dotted line *1 indicates the EXCEPTION signal active, following a rejected command.

SD, SDL, SFO, SFI, STM COMMAND TIMING

Figure 3

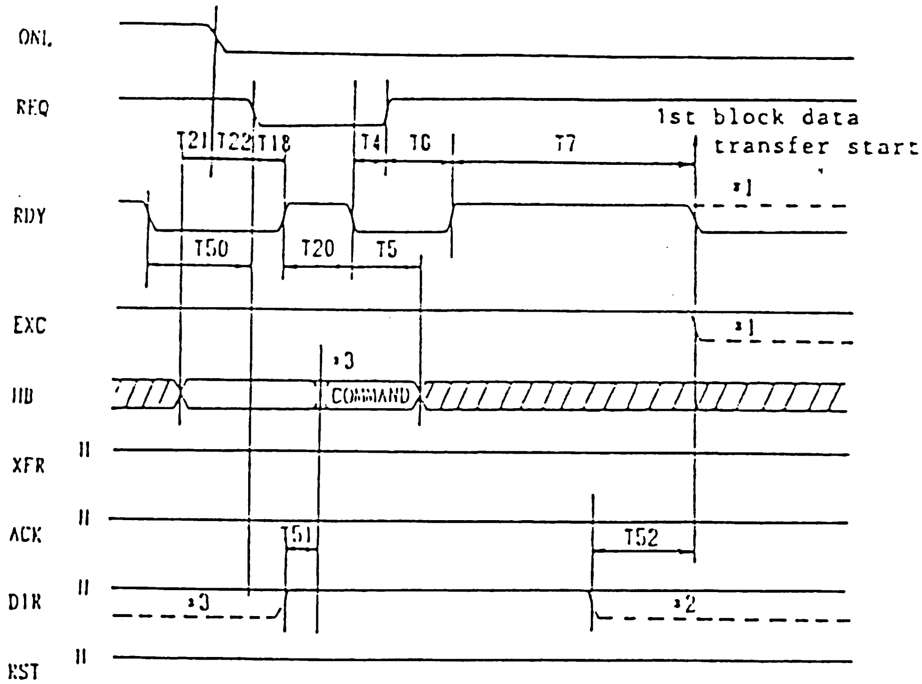


Sym-bol	MIN	NOM	MAX	unit
T1	0			
T18			1	μS
T20	20	500		μS
T4	0			
T5	0			
T6	20		100	μS
T7	20			μS
T50	0			nS

Note: The dotted line *1 indicates the EXCEPTION signal active, following a rejected command.

BOT, ERA, PRW COMMAND TIMING

Figure 4



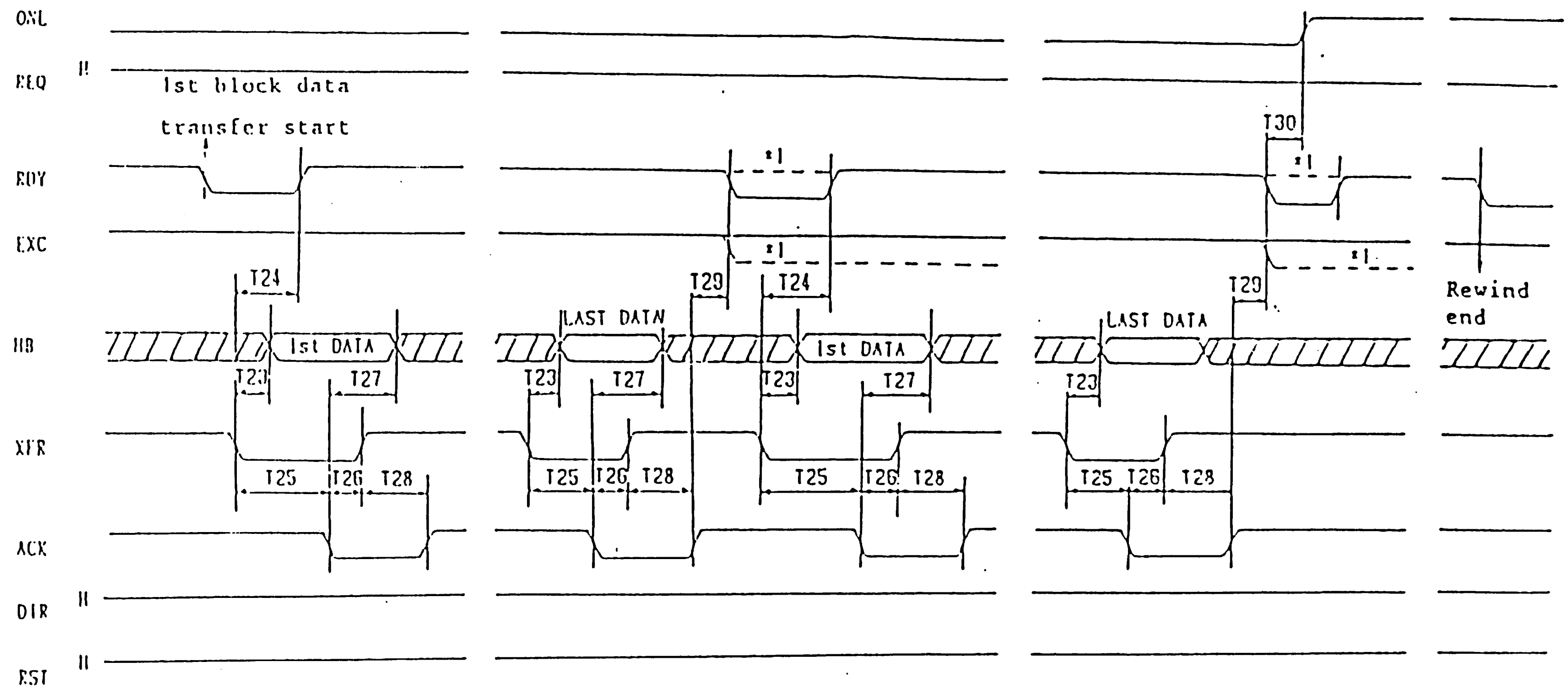
Symbol	MIN	NOM	MAX	Unit
T21	0			
T22	0			
T18			1	µS
T20	20	500		µS
T4	0			
T5	0			
T6	20		100	µS
T7	20			µS
T50	0			nS
T51	0		3	µS
T52	0			

Note:

1. The dotted line *1 indicates the EXCEPTION signal active, following a rejected command.
2. The dotted line *2 is valid following a READ OP CODE.
3. The dotted line *3 indicates DIR (DIRECTION) state when the previous command was a READ OPERATION.

WRT, WRTE, RD COMMAND TIMING

Figure 5



Sym-bol	MIN	NOM	MAX	unit
T23			40	ns
T24			1	μs
T25	0.5		100	μs
T26	0			μs
T27	0			μS

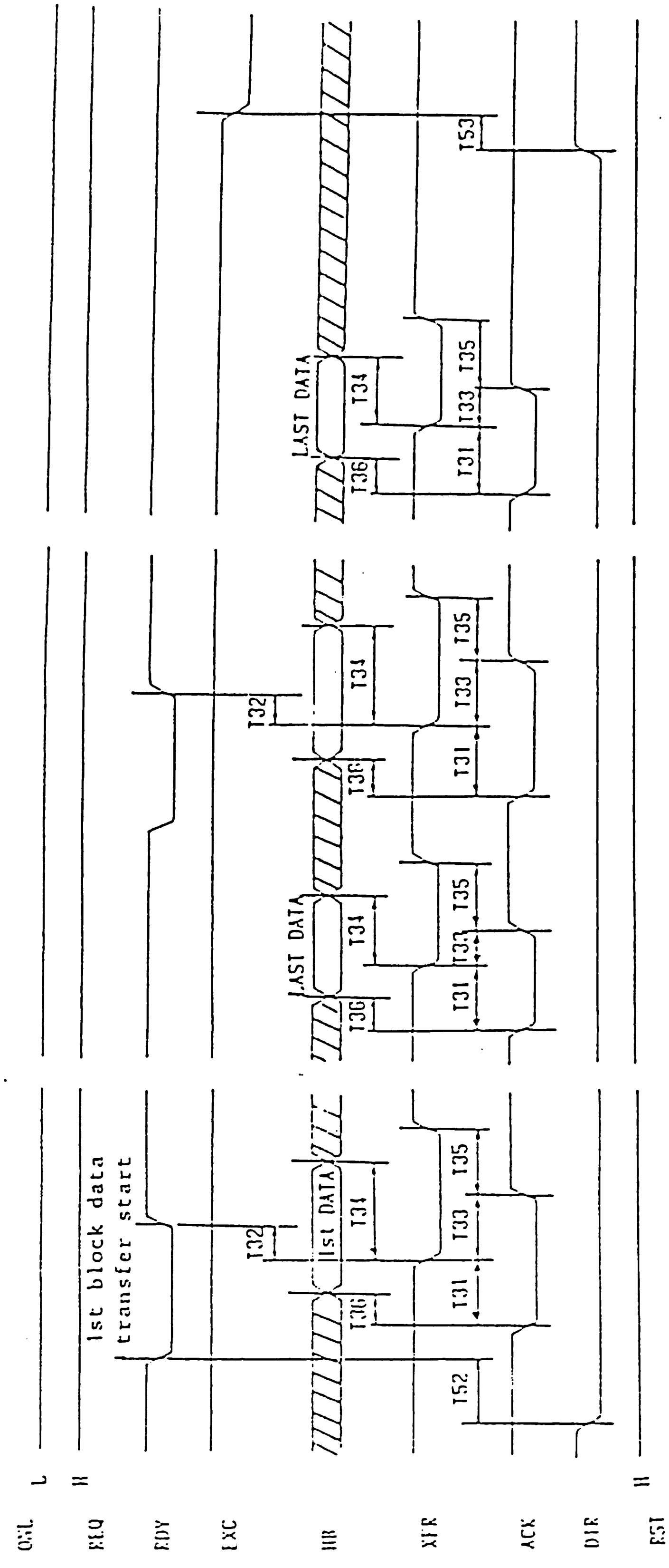
Sym-bol	MIN	NOM	MAX	Unit
T28	0			μS
T29	100			μS
T30	0			

Write data transfer timing

Figure 6

Note:

1. The dotted line *1 indicates the EXCEPTION signal active, following a drive detected BUS PARITY ERROR.
   The parity error may have occured in any byte of the 512 byte block.



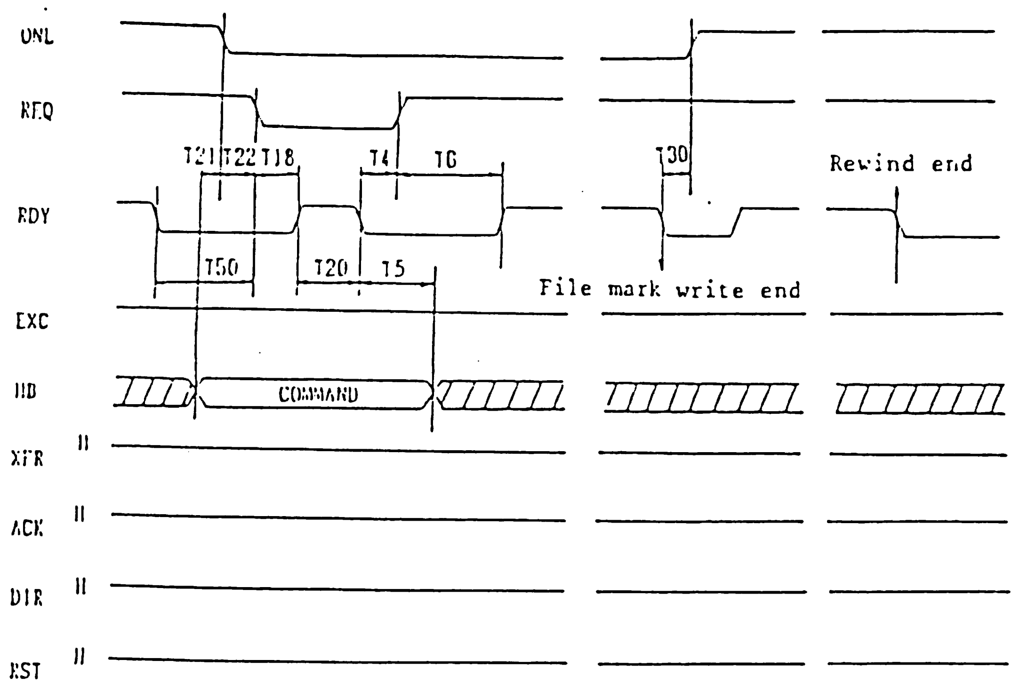
Symbol	MIN	NOM	MAX	unit
T36			40	ns
T31	0			
T32			1	μs
T33	0.1			μs

Symbol	MIN	NOM	MAX	Unit
T34	0			
T35	0			
T52	0			
T53	0			

Read data transfer timing

Figure 7

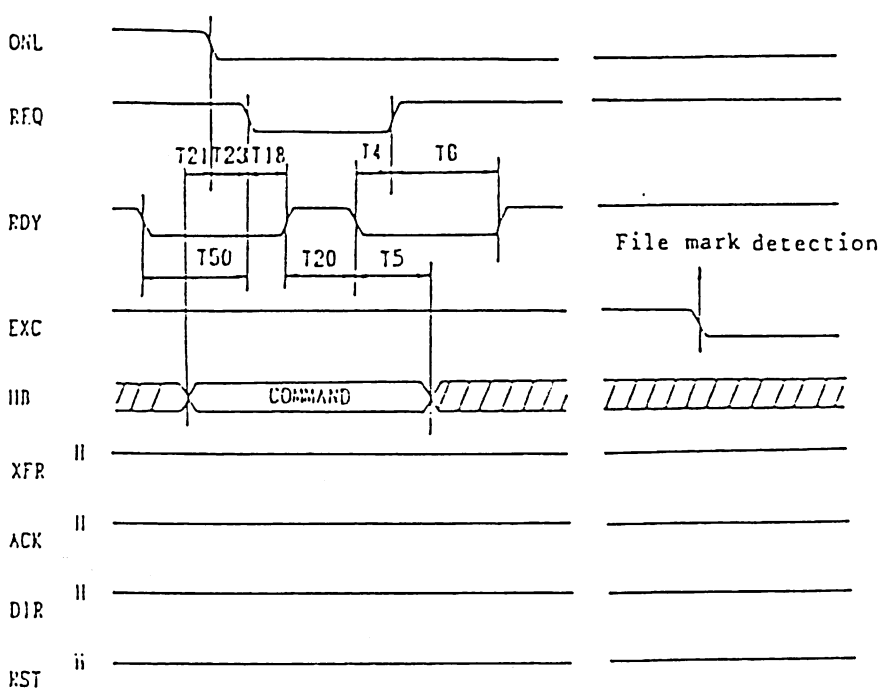
WANG LABORATORIES, INC.	DWN	DATE	TITLE PRODUCT SPECIFICATION CASSETTE TAPE (D/CAS) 17 TRACK P/N 725-4893	WLI NUMBER 191-3468	
	CHK	DATE			
	APPR	DATE		40 of 50	REV. A
	APPR	DATE			



Sym-bol	MIN	NOM	MAX	unit
T21	0			
T22	0			
T18			1	μS
T20	20	500		μS
T4	0			
T5	0			
T6	20		100	μS
T30	0			
T50	0			nS

WFM, WFME command timing

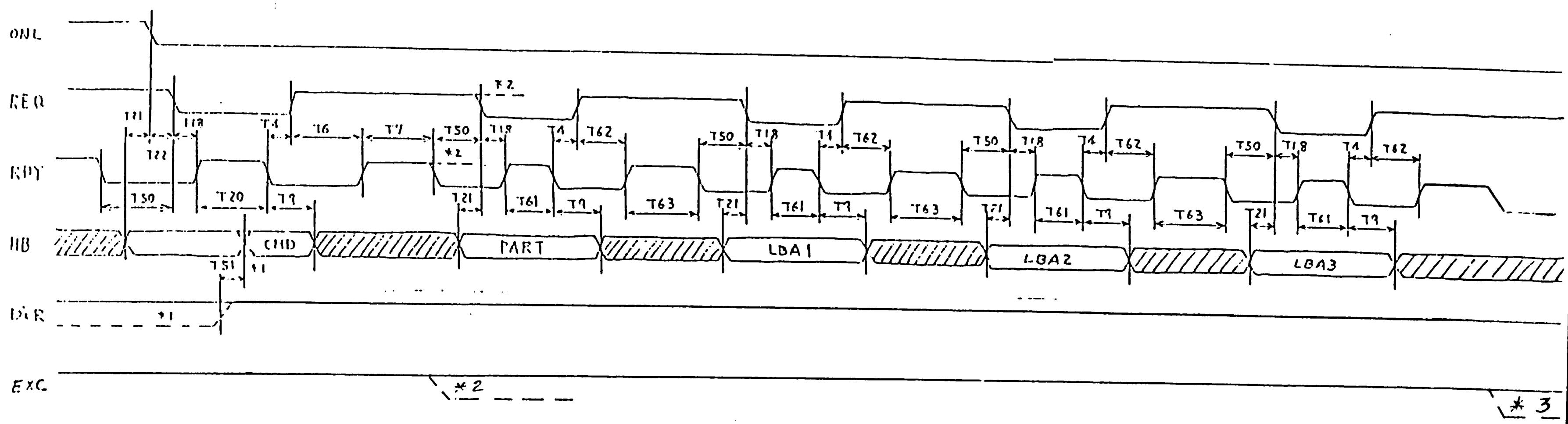
Figure 8



Sym-bol	MIN	NOM	MAX	Unit
T21	0			
T22	0			
T18			1	μS
T20	20	500		μS
T4	0			
T5	0			
T6	20		100	μS
T50	0			nS

RFM, RFM(N), SEOD command timing

Figure 9



Note:

1. The dotted line *1 indicates DIR (DIRECTION) state when the previous command was a READ OPERATION.

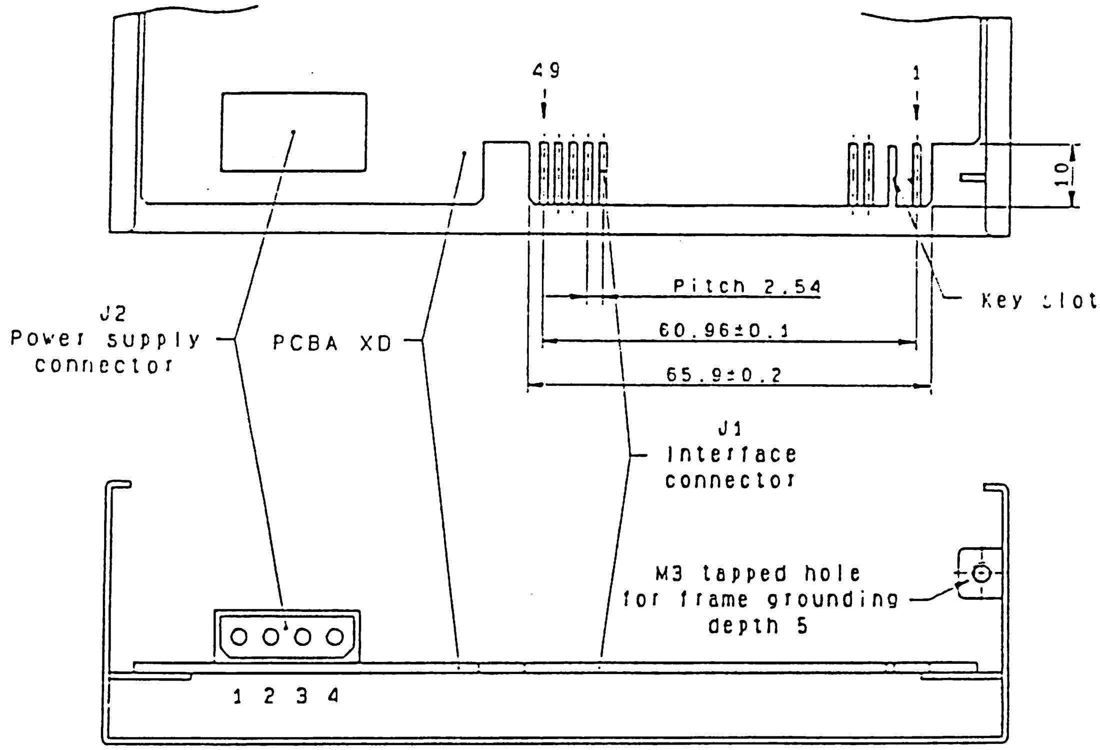
2. The dotted line *2 indicates the EXCEPTION signal active, following a drive detected BUS PARITY ERROR. This represents a parity error in the OP Code.

2. The dotted line *3 indicates the EXCEPTION signal active, following a drive detected BUS PARITY ERROR, in bytes 1 thru 4, or command execution not valid.

Symbol	MIN	NOM	MAX	Unit
T21	0			
T22	0			
T50	1			μs
T18			1	μs
T20	20	500		μs
T51	0		3	μs
T4	0			
T9	0			
T6	20		100	μs
T7	20			μs
T61	10			μs
T62	20	60	100	μs
T63	20			μs

LOCATE command timing

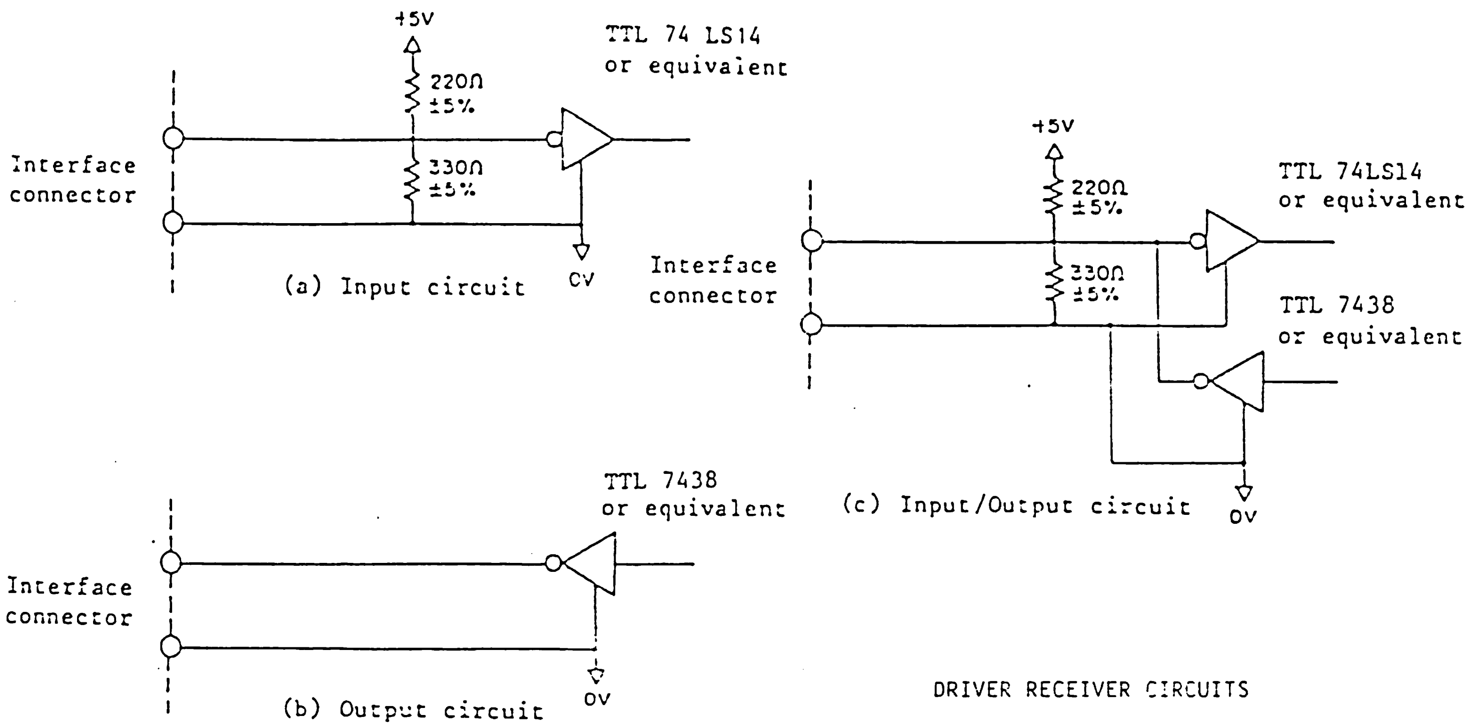
Figure 10



(Unit:mm)

POWER, SIGNAL, and GROUND connections

Figure 11

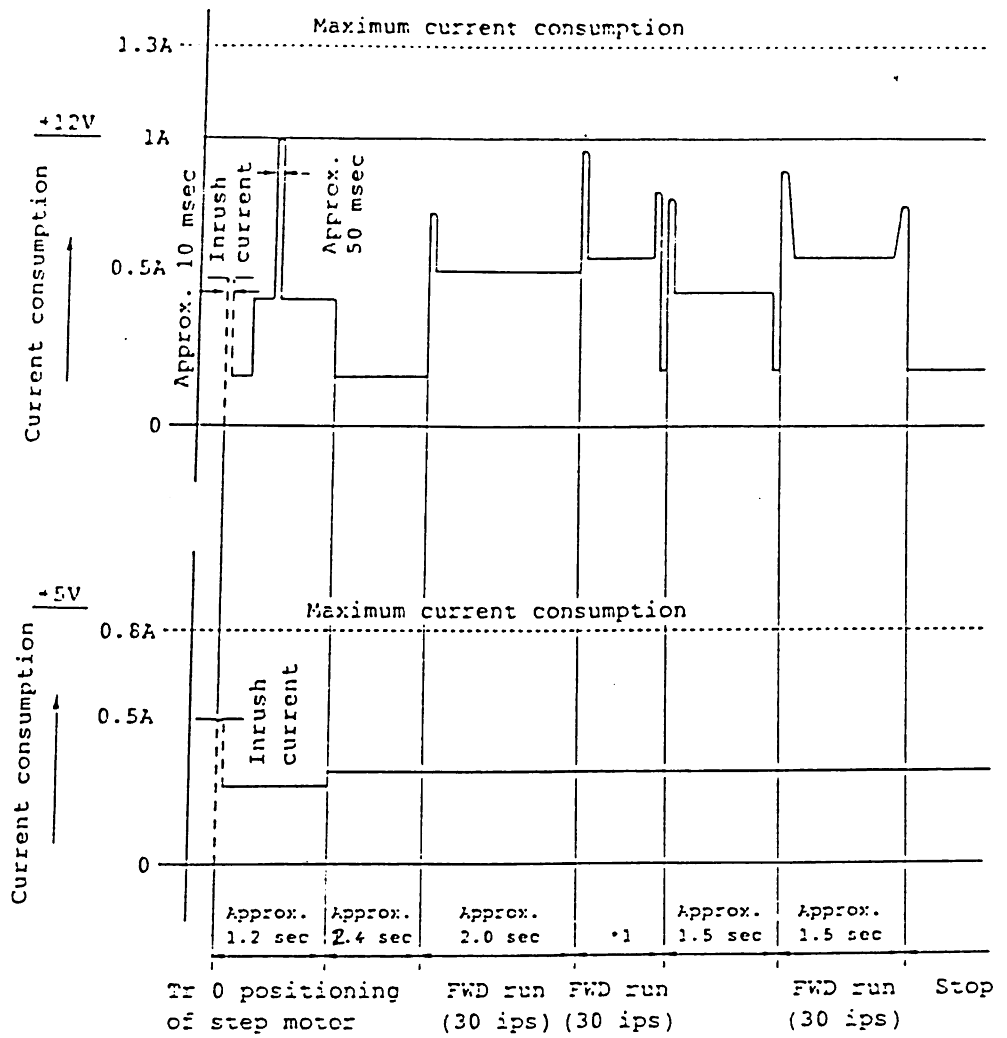


DRIVER RECEIVER CIRCUITS

Figure 12

WANG LABORATORIES, INC.	DWN	DATE	TITLE PRODUCT SPECIFICATION CASSETTE TAPE (D/CAS) 17 TRACK P/N 725-4893	WLI NUMBER 191-3468	
	CHK	DATE		43 of 50	REV. A
	APPR	DATE			
	APPR	DATE			

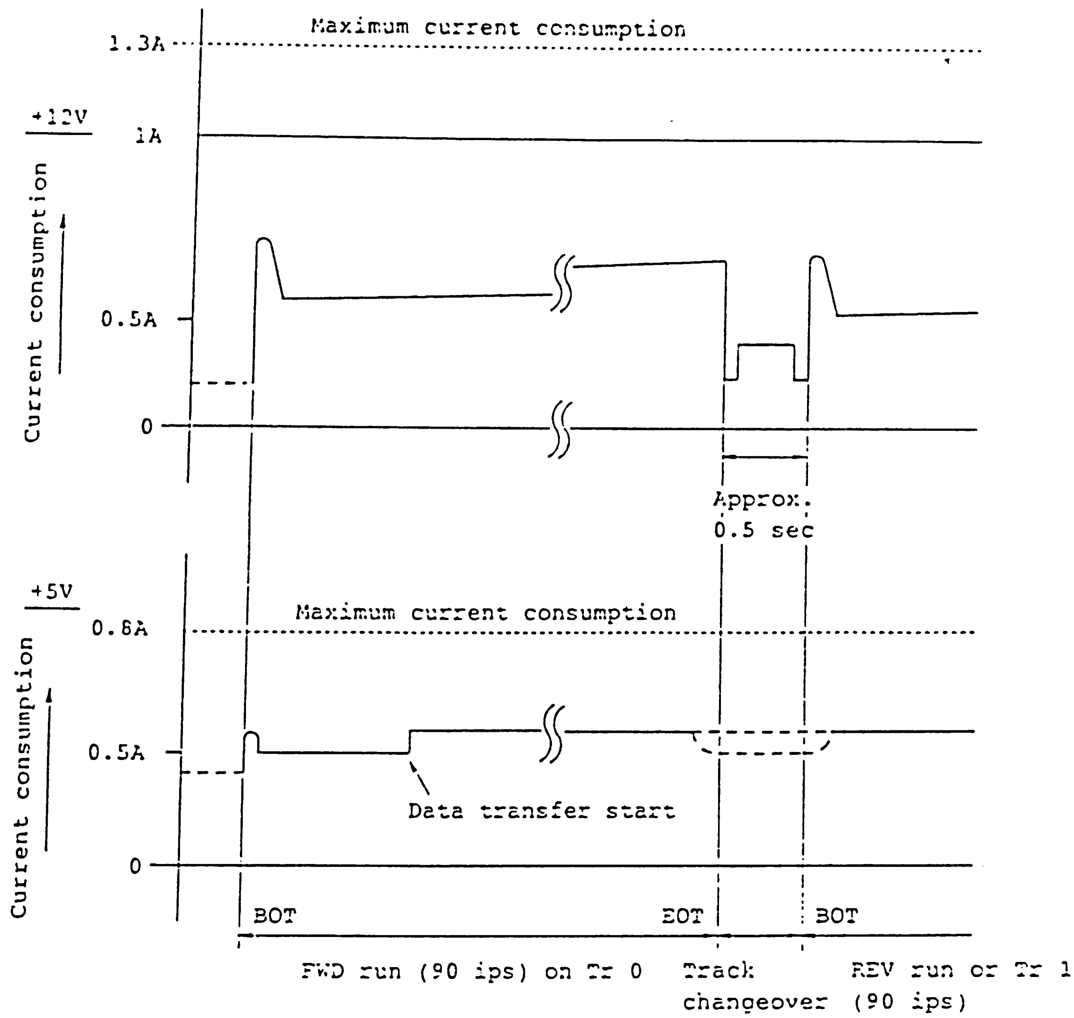
14-16698



POWER LOADING (power on)

Figure 13

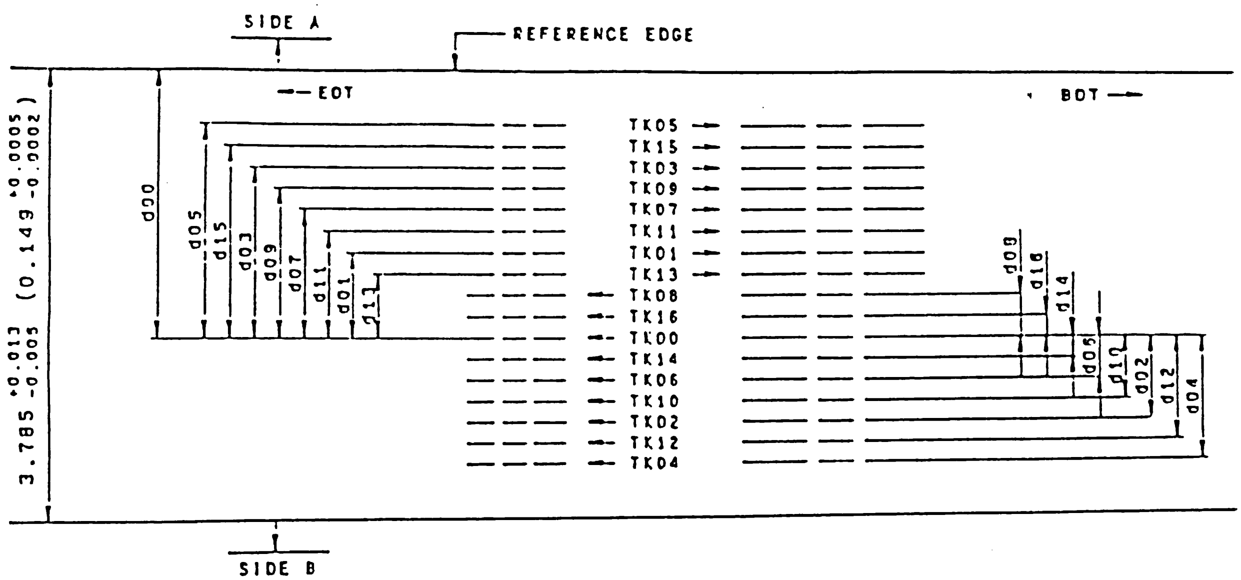
WANG LABORATORIES, INC.	DWN	DATE	TITLE	WLI NUMBER
	CHK	DATE	PRODUCT SPECIFICATION CASSETTE TAPE (D/CAS) 17 TRACK P/N 725-4893	191-3468
	APPR	DATE		44 of 50 — REV. A
	APPR	DATE		



POWER LOADING (Data transfer)

Figure 14

TAPE VIEWED LOOKING TOWARD RECORDED SURFACE



d00=2.286±0.076 (0.090±0.003)
d01=0.813±0.036 (0.032±0.0014)
d02=0.813±0.036 (0.032±0.0014)
d03=1.626±0.036 (0.064±0.0014)
d04=1.219±0.036 (0.048±0.0014)
d05=2.032±0.036 (0.080±0.0014)

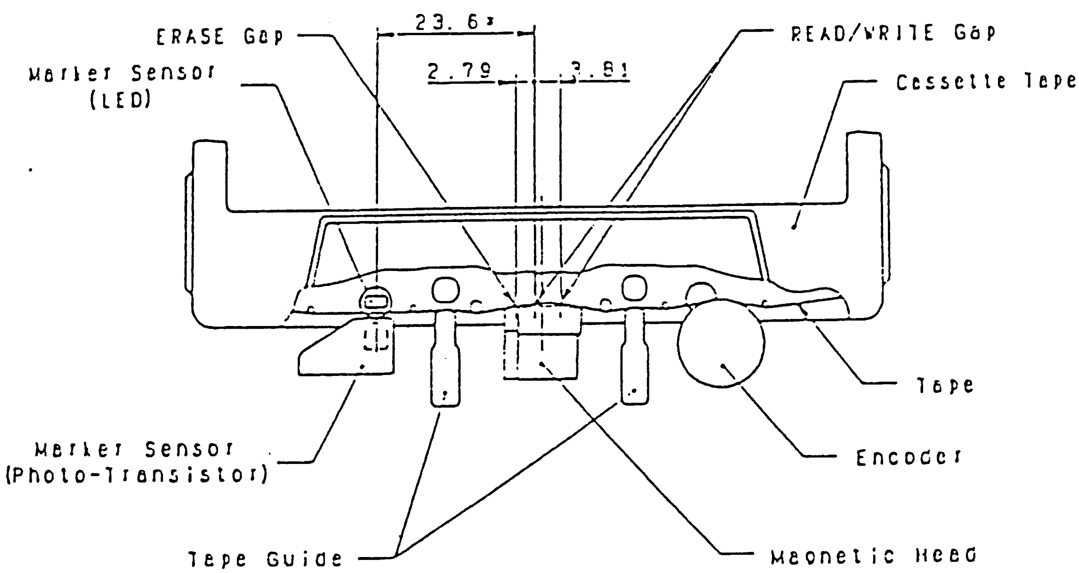
d06=0.406±0.036 (0.016±0.0014)
d07=1.219±0.036 (0.048±0.0014)
d08=0.406±0.036 (0.016±0.0014)
d09=1.422±0.036 (0.056±0.0014)
d10=0.610±0.036 (0.024±0.0014)
d11=1.016±0.036 (0.040±0.0014)

d12=1.016±0.036 (0.040±0.0014)
d13=0.610±0.036 (0.024±0.0014)
d14=0.203±0.036 (0.008±0.0014)
d15=1.829±0.036 (0.072±0.0014)
d16=0.203±0.036 (0.008±0.0014)

< UNIT: mm (inch) >

TRACK LOCATIONS

Figure 15

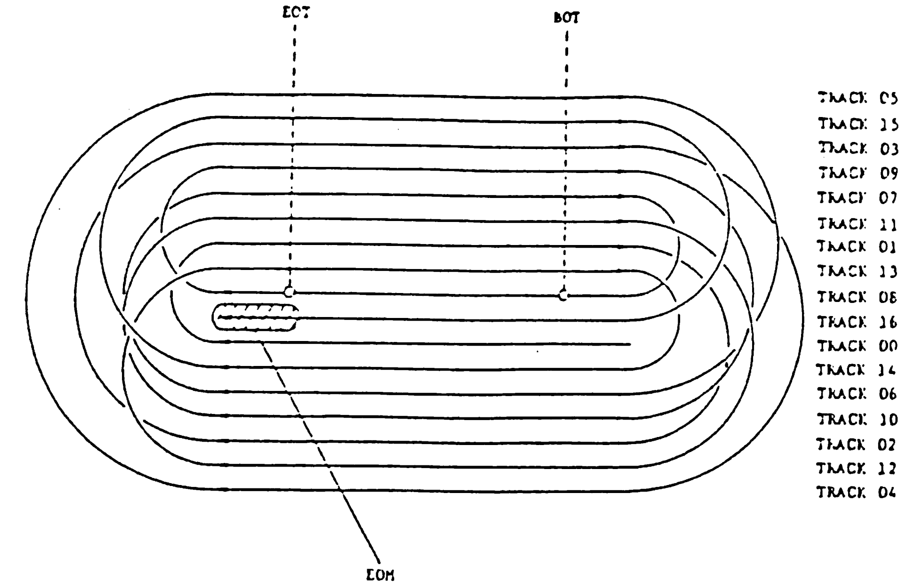


TAPE PATH

Figure 16 (unit:mm)

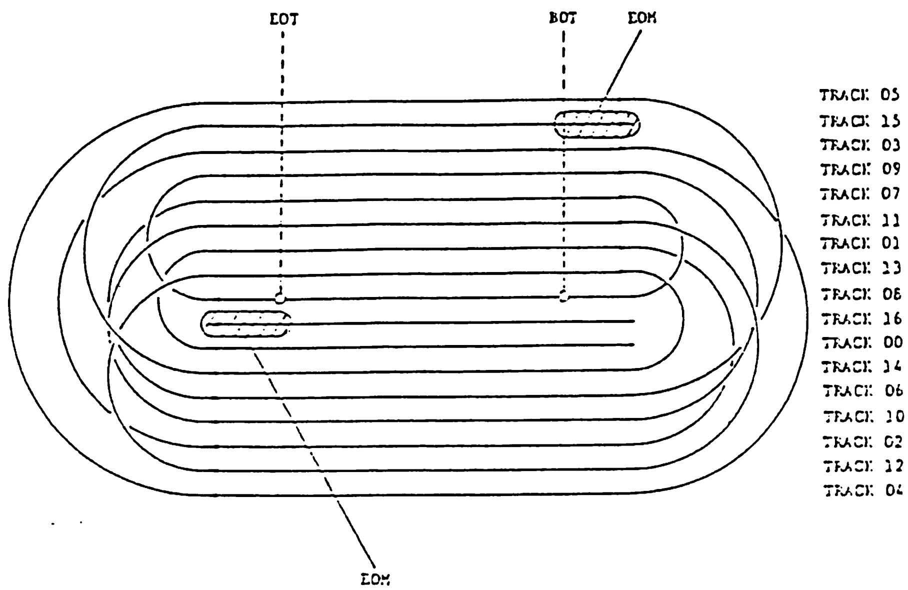
WANG LABORATORIES, INC.	DWN	DATE	TITLE PRODUCT SPECIFICATION CASSETTE TAPE (D/CAS) 17 TRACK P/N 725-4893	WLI NUMBER 191-3468	
	CHK	DATE			
	APPR	DATE		46 of 50	REV. A
	APPR	DATE			

14-16698



TRACK SEQUENCE (Single partition)

Figure 17



TRACK SEQUENCE (Dual partition)

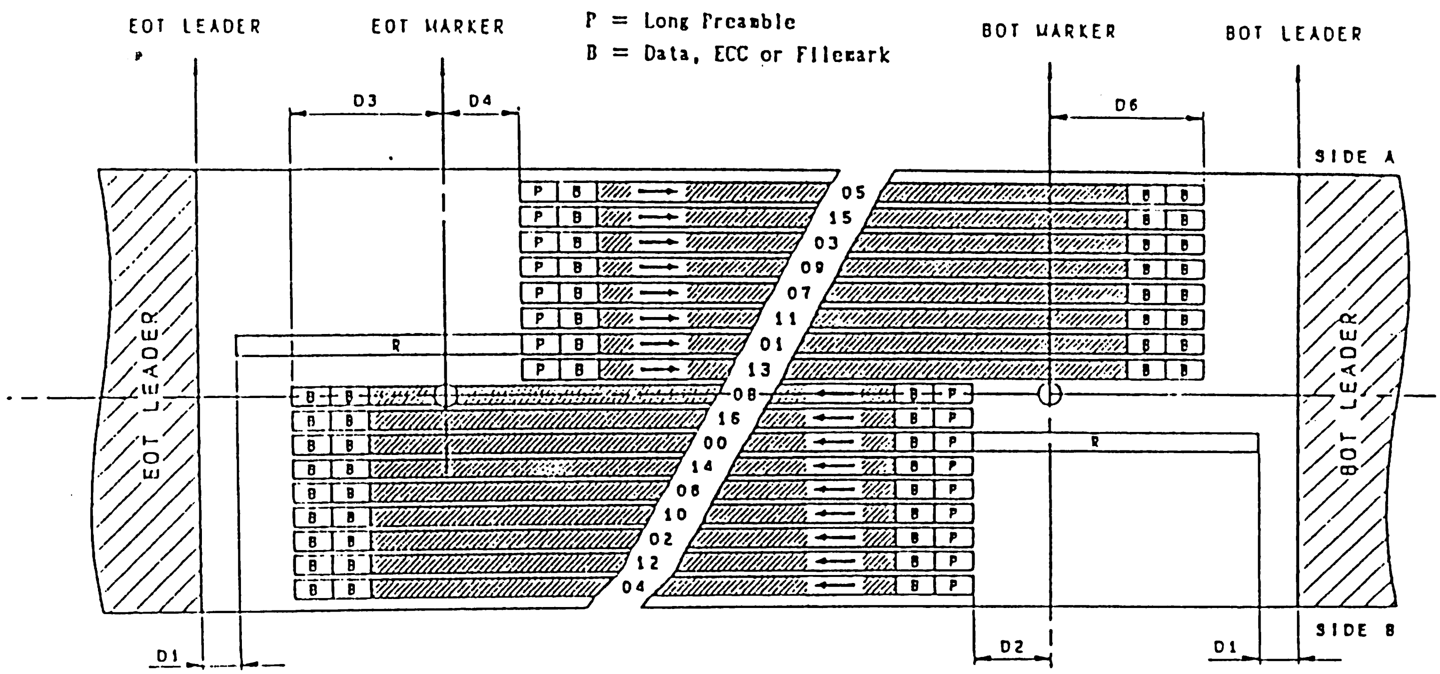
Figure 18

LEGEND:

R = Reference Burst

P = Long Preamble

B = Data, ECC or Filemark



DIMENSION	MIN (In)	MAX (In)	DESCRIPTION
D1	0	15	LEADER TO START OF TRACK REFERENCE BURST
D2	-.5	4	BOT MARKER TO END OF TRACK REFERENCE BURST AND START OF PREAMBLE ON EVEN TRACKS
D3	-	27	EOT MARKER TO END OF DATA ON EVEN TRACKS
D4	1	6	EOT MARKER TO START OF PREAMBLE ON ODD TRACKS
D6	-	27	BOT MARKER TO END OF DATA ON ODD TRACKS

Remark (1): This figure is as viewed from the magnetic face side of the tape.

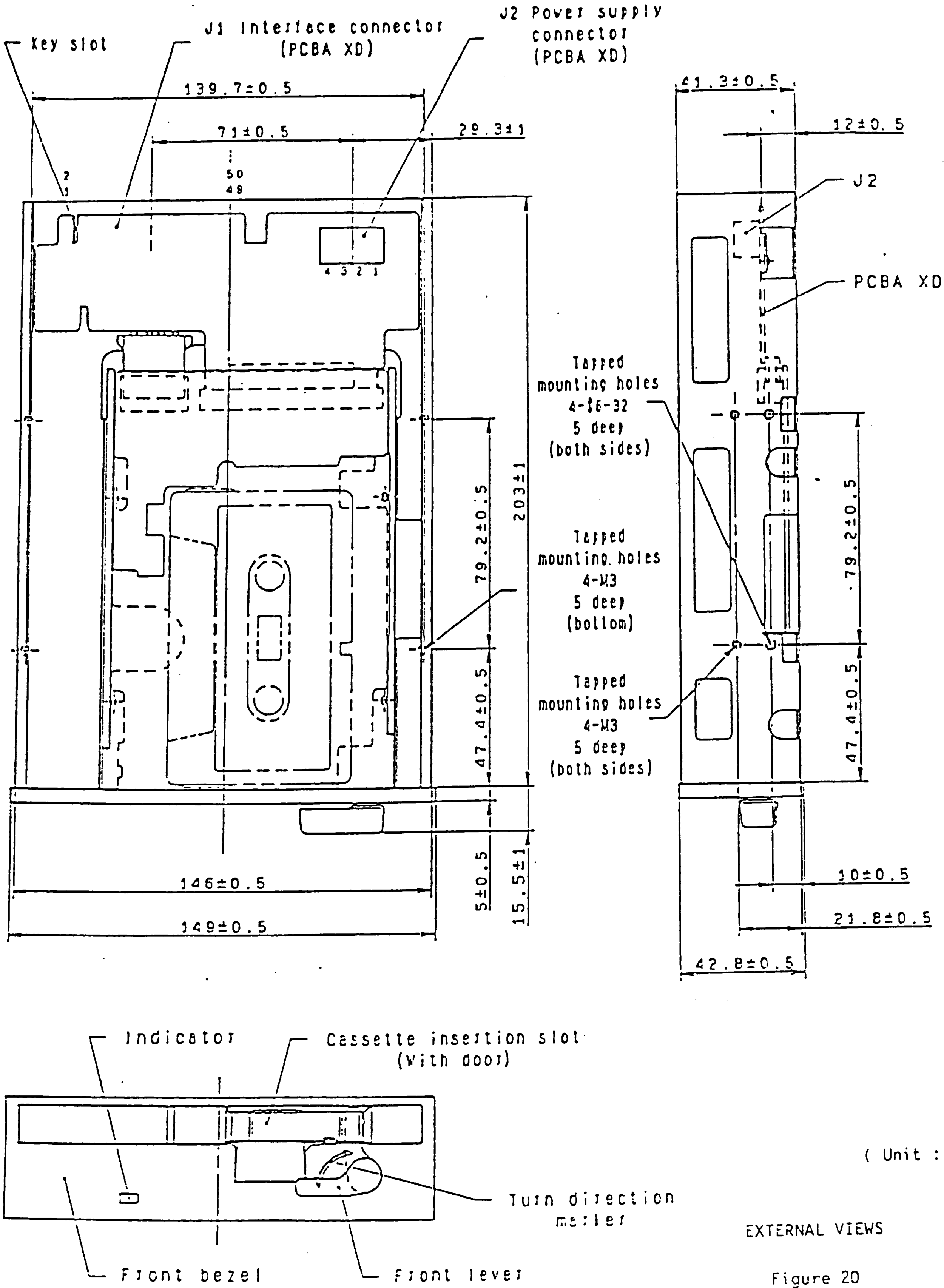
(2): The arrow indicates the recording direction of the relevant track.

Recording area on tape, and recording method for each track

RECORDED TRACK GEOMETRY

Figure 13

WANG LABORATORIES, INC.	DWN	DATE	TITLE: PRODUCT SPECIFICATION CASSETTE TAPE (D/CAS) 17 TRACK P/N 725-4893	WLI NUMBER 191-3468
	CHK	DATE		
	APPR	DATE		48 of 50 — REV. A
	APPR	DATE		



EXTERNAL VIEWS

Figure 20

WANG LABORATORIES, INC.	DWN	DATE	TITLE: PRODUCT SPECIFICATION CASSETTE TAPE (D/CAS) 17 TRACK P/N 725-4893	WLI NUMBER: 191-3468
	CHK	DATE		
	APPR	DATE		49 of 50 — REV. A
	APPR	DATE		

14-16698

16.0 QUALIFIED VENDORS LIST (FOR INTERNAL USE ONLY)

A list of current qualified vendors follows. The list includes manufacturer's addresses, product model number, and part number.

The following vendor is approved by: Vic D. McLane

VENDOR	PART NUMBER
Teac Corporation of America	Model Number MT-2ST/N65-73W Part Number 19305070-73
Wang Laboratories Inc.	Wang P/N 725-4893

DOC
ehm Sept 20, 1989

Notes on SCTD Software

August 26, 1986 S.Ramakrishnan

1. Introduction - SCTD Software is a part of 2275F/R Disk Unit resident software providing interface between the 2200 BASIC-2 Operating System and the optional 2275T Streaming Catridge Tape Drive. This software controls the tape drive through QIC-2 interface, interacts with disk unit software for data transfer between disk drive and tape drive, and implements 2200 command protocols.

1.1 Design Considerations - There are two sets of commands to the tape drive:

(a) non-data transfer commands
(b) data transfer commands

A command byte is transmitted to the tape drive to initiate any action; in case of data transfer commands, data transfer is set up once the command is accepted. The data transfer can be either DMA or non-DMA.

The command byte is transmitted to the tape drive using hand-shake signals, Ready and Request. The command byte is gated to the tape drive bus and Request is ascertained. The tape drive removes the Ready (or Exception) signal and ascertains Ready signal to indicate the acceptance of the command. This will not happen if the tape drive is not present. The Cipher tape drive manual indicates a nominal response time of 500 micro seconds. To take care of tape drives from other vendors, the timer is programmed to timeout in 16 milli seconds approximately to indicate the absence of tape drive.

Since the procedure to transmitt the command byte to the tape drive is same for all commands, a single routine (TransCMD) handles this function. Data transfer functions are handled by different routines depending on read or write command. DMA and non-DMA data transfers are handled by different routines. Non-DMA data transfers are either reading of status bytes or transferring the required block number for Block Search command.

Multiple entry routines are used to implement many non-data transfer commands. The command byte from 2200 is used to access the command table to get the tape drive command byte. A subroutine (TapeCMD) implements this procedure.

2. Software Architecture - There are four modules handling functions at different levels:

(a) SCTD Command Module - This higher level module interprets the 2200 commands for the tape drive and implements the necessary command protocols. The lower level modules, SCTD Interface Module and SCTD Disk Interface Module are invoked depending on the command from 2200.

(b) SCTD Interface Module - This lower level module interfaces with the tape drive to implement various commands. Both DMA and non-DMA data transfer functions for the tape drive are provided by this module.

(c) SCTD Disk Interface Module - This middle level module interfaces with the disk software to provide read and write functions for the disk drive. The type of disk drive (floppy or winchester) is transparent to this module.

(d) SCTD Interrupt Module - This lower level module is invoked by hardware exception or bus parity error. Six status bytes from the tape drive read and first two bytes are analysed to ascertain the error condition. In case of bus parity error, recovery is attempted by retrying read or write function upto four times.

The constants and storage definitions for all the above modules are defined in a separate file.

3. Points to remember:

(a)Timer delays are valid at 4MHz clock frequency.

(b)File mark is counted as one block; the length of this block is not 512 bytes.

(c)The variables, PhysArea, PlatTabl and Comm2200 are assigned values after assembling the disk software. Any change in the values of IdlCtcB2 and IdlCtcB3 in the disk software should be effected in SCTD software also.

List of files used in SCTD software appears in Appendix A.

List of source files in 2200DEV;source

```
        .constant
        .command
        .command2 (continuation of .command)
        .diskintr
        .tapeintr
        .interupt
```

```
source.build
2275tape    module
            INSERT        "&.constant"
            INSERT        "&.command"
            INSERT        "&.diskintr"
            INSERT        "&.tapeintr"
            INSERT        "&.interupt"
            END                                ;end module
```

```
source.asm
filefix     &,/source/
log         /&.log/
source      /&.build/
object      /&.rel/
list        /&.lst/
assemble    macro,xref,fullxref,symbmax
```

```
source.lnk
MAP         /source.map/
FILL        /source.cim/,0
WITH        /wp:andy.rel/
WITH        /source.rel/
% END
```

```
source.prom
Source   "2200DEV;source.cim"
%Object   "2295TR##"
Execute
End
```