Xylogics

HOME OF THE PHOENIX

XYLOGICS Inc.
42 Third Ave.
Burlington Ma. 01803

2852

Xylogics, Inc.

Phoenix 211 Disk Controller

Operator's Manual

Volume 1 of 11

Dwg. No. 900-006-300

Revision A

Date OCT 1978

Xylogics

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT'D)

PHOENIX 211 CDC DISK SUBSYSTEM UNPACKING AND INSTALLATION INSTRUCTIONS

1.0 Unpacking Instructions

The Phoenix 211 Disk Controller physically consists of five printed circuit boards and one systems unit. The boards and systems unit are individually packed in one common shipping container, along with the appropriate documentation.

Inspect the shipping container for obvious shipping damage and notify the carrier immediately in the event of any damage.

Carefully unpack and set aside each printed circuit board and systems unit assembly.

Refer to Phoenix 211 Installation Instructions.

2.0 CDC SMD Disk Drive Unpacking Instructions

The CDC documentation manuals are located in the storage cabinet at the base of the drive, or in the shipping box.

Follow the detailed unpacking and installation instructions outlined in Section 1.0, Volume 1 of 2 of the CDC Hardware Maintenance Manual.

WARNING — The CDC disk drive has been shipped with a restraint installed to prohibit head carriage movement. This restraint must be removed prior to applying power to the disk drive. Failure to remove this shipping pin prior to power application may damage the disk drive.

2.1 Installation Procedure for Storage Module Drive Index/Sector on "B" Cable, 75 Pin Amp Connector Interface

Note 1) This procedure is applicable for units serial number 713 and above.

Note 2) Skip Section 2.1 for Flat Cable Interface

2.1.1 Power down unit.

2.1.2 Open case assembly

2.1.3 Loosen logic chassis hold-down bar. Raise logic chassis, and support with logic support arm.

2.1.4 Remove connector (PA6) from the logic backpanel.

2.1.5 Remove the following pins from the P6 connector, PA6-05A, PA6-05B, PA6-06A and PA6-06B. Insulate these wires to prevent any shorting.

2.1.6 Locate spare wires in the existing harness which are presently capped. Remove the heatshrink caps from the wires which go to P2-EE, HH, FF, and JJ.

2.1.7 Determine the continuity of each wire and install them in connector PA6 as follows:

J2-HH to PA6-06B	White Wire	Bottom Cable
J2-EE to PA6-06A	Black Wire	
J2-JJ to PA6-05B	White Wire	Top Cable
J2-FF to PA6-05A	Black Wire	

2.1.8 Reinstall connector PA6 onto backpanel.

2.1.9 Remove logic chassis card cover.

2.1.10 Remove transmitter card (BTVV) in Location A06.

2.1.11 Rework the transmitter card by cutting the foil to Pin 10 of I.C. A3. Then add a jumper from Pin 10 to Pin 5 of I.C. A3. (see Figure 1 attached).

2.1.12 Reinstall BTVV in location A06.

2.1.13 Change manual No. 83308400 Pg. 4.12 as shown in Figure 2.

2.2 Sector Selection Procedure # 1 (For all CDC Drives)

2.2.1 Remove FLTV logic card from the card cage. (See table for location)

2.2.2 Using Figure 3 set the switches to the desired sector quantity. If the desired sector number is not shown on Figure 3 see section 2.3 for explanation of dibits and the equation required to select switch settings.

LOGIC CARD LOCATION

DRIVE	FLTV
9760/62, BJXX	AØ3
9764/66, BJXX	
9760/62, BK4, BK5	BØ8
9764/66, BK6, BK7	AØ6

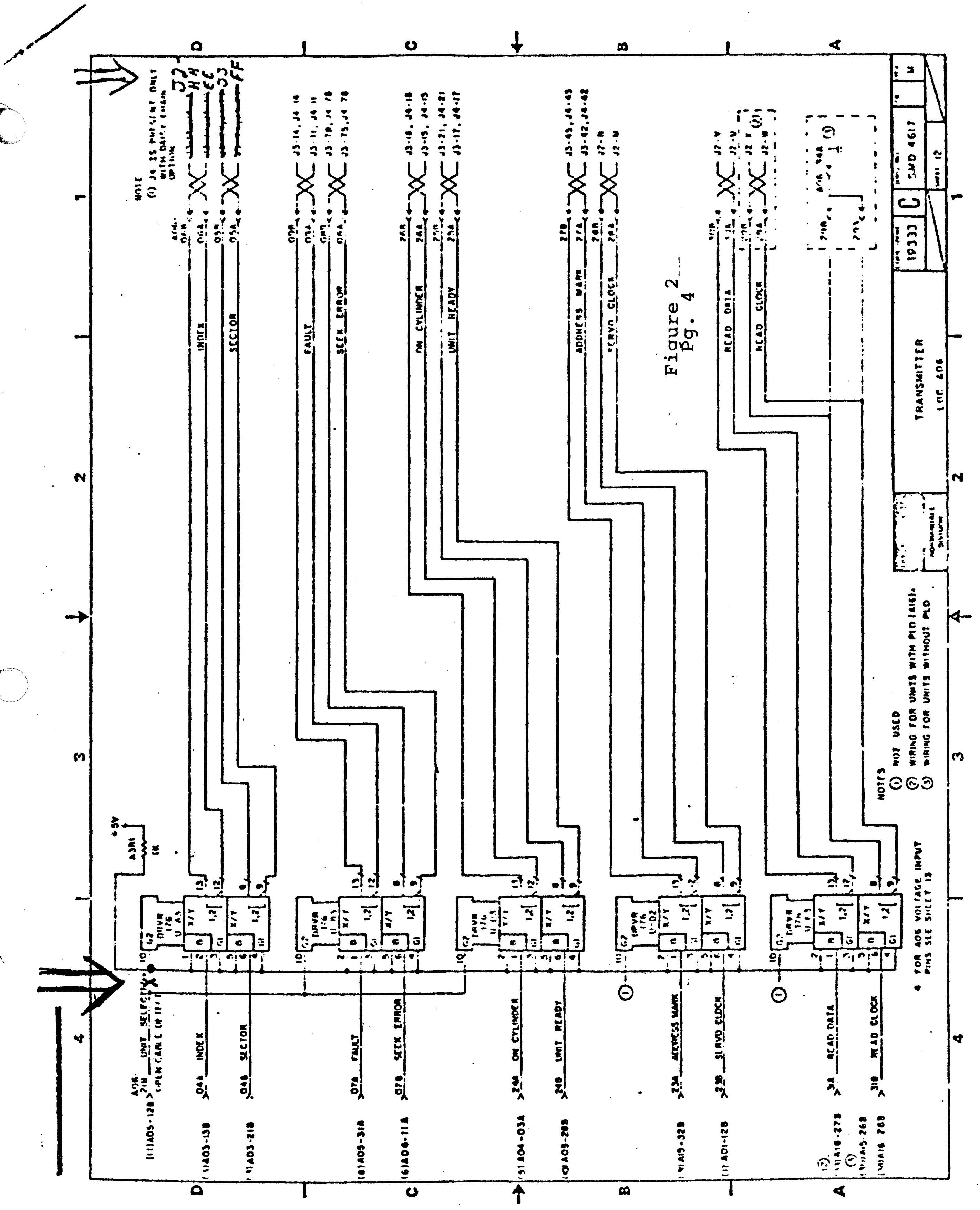


Figure 2
Pg. 4
INDEX
SECTOR
FAULT
SEEK ERROR
ON CYLINDER
UNIT READY
ADDRESS MARK
SERVO CLOCK
READ DATA
READ CLOCK
UNIT SELECT
+5V
NOTES
NOT USED
WIRING FOR UNITS WITH PLO (A16)
WIRING FOR UNITS WITHOUT PLO
FOR A06 VOLTAGE INPUT PINS SEE SHEET 13
TRANSMITTER
LOG 406
19333
SMD 4617

FIGURE 1

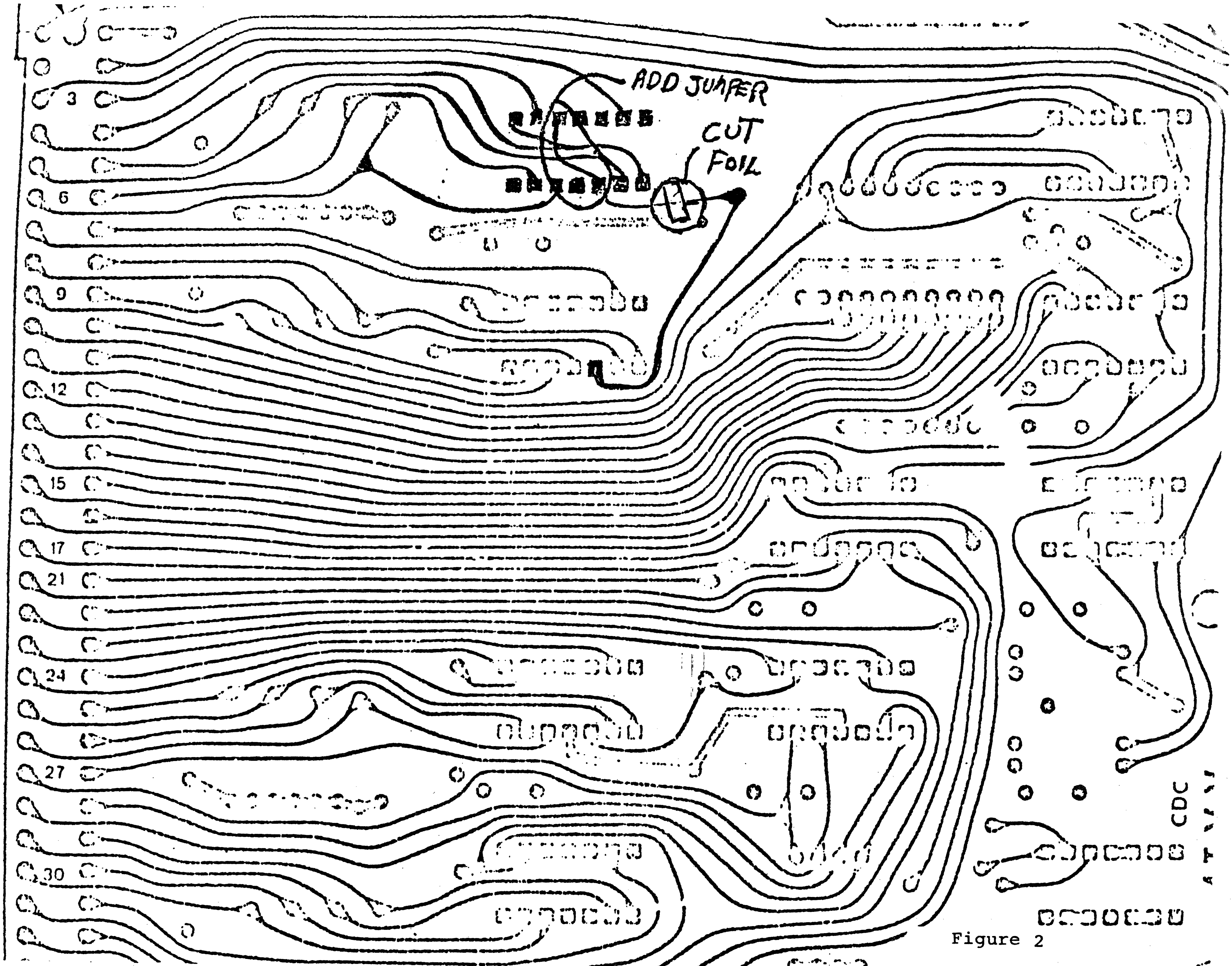


Figure 2

TABLE 2
SW POSITION FOR EVEN LENGTH SECTORS

SECTORS	0	1	2	3	4	5	6	7	8	9	10	11
4	0	0	0	0	0	+	+	+	0	+	0	0
5	0	0	0	0	0	0	0	+	+	0	+	0
6	0	0	0	0	0	0	+	0	+	+	+	0
7	0	0	0	0	0	0	0	+	0	0	0	+
8	0	0	0	0	+	+	+	0	+	0	0	+
10	0	0	0	0	0	0	+	+	0	+	0	+
12	0	0	0	0	0	+	0	+	+	+	0	+
14	0	0	0	0	0	0	+	0	0	0	+	+
15	0	0	0	0	0	0	0	+	0	0	+	+
16	0	0	0	+	+	+	0	+	0	0	+	+
20	0	0	0	0	0	+	+	0	+	0	+	+
21	0	0	0	0	0	0	0	+	+	0	+	+
24	0	0	0	0	+	0	+	+	+	0	+	+
28	0	0	0	+	+	+	0	+	+	+	+	+
30	0	0	0	0	0	0	+	0	0	+	+	+
32	0	0	+	+	+	0	+	0	0	+	+	+
35	0	0	0	0	0	0	0	+	0	+	+	+
40	0	0	0	+	0	+	+	0	0	+	+	+
42	0	0	0	0	0	0	+	+	0	+	+	+
48	0	0	0	+	0	+	+	+	0	+	+	+
56	0	0	0	0	+	0	0	0	+	+	+	+
60	0	0	0	0	0	+	0	0	+	+	+	+
64	0	+	+	+	0	+	0	0	+	+	+	+
70	0	0	0	0	0	0	+	0	+	+	+	+
80	0	0	0	+	+	0	+	0	+	+	+	+
84	0	0	0	0	0	+	+	0	+	+	+	+
96	0	0	+	0	+	+	+	0	+	+	+	+
105	0	0	0	0	0	0	0	+	+	+	+	+
112	0	0	0	+	0	0	0	+	+	+	+	+
120	0	0	0	0	+	0	0	+	+	+	+	+
128	+	+	+	0	+	0	0	+	+	+	+	+
22	0	+	+	+	+	0	0	+	+	0	+	+
36	0	+	0	+	0	0	0	+	0	+	+	+

NOTE: + Open

0 Closed

Fig. 3

Note: The switch positions referred to in Figure 3 do not apply to the numbers physically on the switch module but rather to the etched numbers, Ø - 11, on the logic card itself.

2.2.3 Reinstall the FLTV card in the proper location. Replace logic chassis card cover. Lower and secure logic chassis.

2.3 <u>Sector Selection Procedure # 2 (Omit This Section if Sector Selection Completed in Section 2.2)</u>

2.3.1 Remove FLTV logic card from Position AØ3 or B08 in the card cage.

2.3.2 Follow the explanation below and select the proper switch setting.

The FLTV card has the capability of selecting sector quantity. The number of sectors is determined by counting dibits, and each switch position gives a fixed number of dibits when closed (table 1).

<u>TABLE 1</u>

SW POSITION	0	1	2	3	4	5	6	7	8	9	10	11
NO. DIBITS	1	2	4	8	16	32	64	128	256	512	1024	2048

One dibit is used in resetting the counters and has to be added to each sector. To calculate the correct switch position for the number of sectors required use the following formula.

Total no. dibits <u>13,440</u> + no. of sectors equals the maximum number dibits per sector.

To calculate the correct no. of sectors, close the switches, which is the closest total without going over the maximum dibits per sector.

EXAMPLE: WANT 12 Sectors

$$\frac{13,440}{12} = 1,120$$

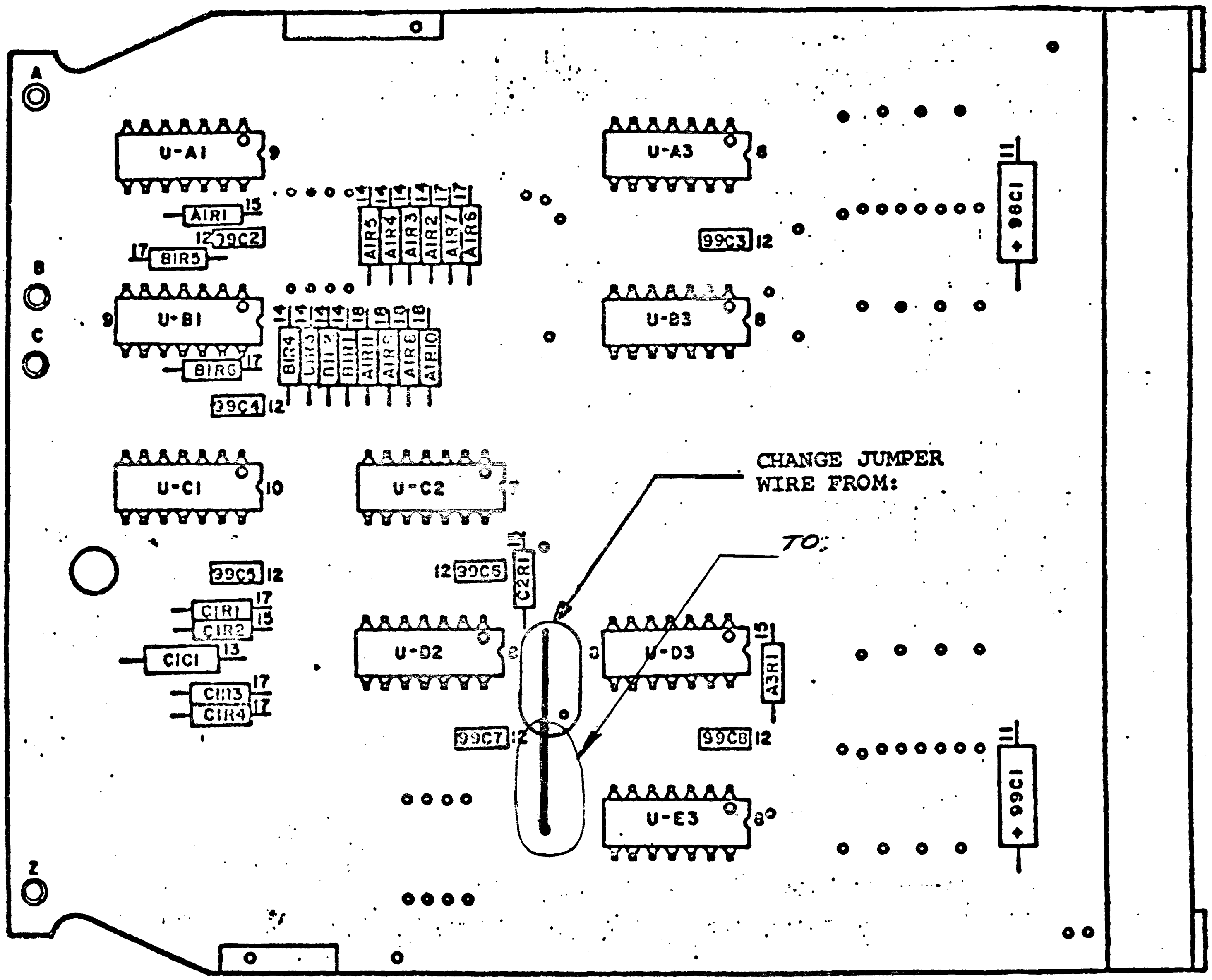

CHANGE JUMPER
WIRE FROM:
TO:
U-A1
U-B1
U-C1
U-C2
U-D2
U-A3
U-B3
U-D3
U-E3
A1R1
B1R5
B1R6
99C2
99C3
99C4
99C5
99C6
99C7
99C8
C1R1
C1R2
C1C1
C1R3
C1R4
C2R1
A3R1
98C1
99C1
A1R5
A1R4
A1R3
A1R2
A1R7
A1R6
B1R4
B1R1
A1R11
A1R10
A
B
C
Z

FIGURE 3A

Close SW 10	=	1024 dibits
Close SW 6	=	64 dibits
Close SW 4	=	16 dibits
Close SW 3	=	8 dibits
Close SW 2	=	4 dibits
Close SW 1	=	2 dibits
Close SW 0	=	1 dibit
One dibit for reset	=	1 dibit
		1120 dibits

Note: each dibit is equivalent to 12 data bits.

2.3.3 Reinstall the FLTV card in the proper location. Replace logic chassis card cover. Lower and secure logic chassis.

2.3.4 Close case assembly.

2.4 Installation Procedure for Storage Module Drive, CDC #9760 or 9762 Flat Cable

Index, Sector and Unit Select Modification(Refer to CDC SPO 68542-4)

NOTE: Skip this Section for any drive other than 40-80 MB Flat cable drive.

2.4.1 Power down unit and unplug power cable.

2.4.2 Open top case assembly and swing open logic chassis.

2.4.3 Remove the following wires from the wire wrap logic panel as follows:

From	To	Z	From	To	Z
B01-06B	JA82-18B	1	B03-06B	JA83-18B	1
B01-06A	JA82-18A	1	B03-06A	JA83-18A	1
B01-05B	JA82-25B	1	B03-05B	JA83-25B	1
B01-05A	JA82-25A	1	B03-05A	JA83-25A	1

2.4.4 Add the following wires to the wire wrap logic panel as follows:

From	To	Z	From	To	Z
B01-06B	JA82-43B	1	B03-06B	JA83-43B	1
B01-06A	JA82-44A	1	B03-06A	JA83-44A	1
B01-05B	JA82-45B	1	B03-05B	JA83-45B	1
B01-05A	JA82-45A	1	B03-05A	JA83-45A	1

2.4.5.1 Remove wire from JA82-41A.

2.4.5.2 Remove wire from JA82-41Band install this wire on JA82-41A.

2.4.5.3 Install wire removed from JA82-41A on JA82-41B.

2.4.6 Repeat 2.4.5 for JA83-41A and JA83-41B.

2.4.7 Secure logic chassis in closed position.

2.4.8 Locate and remove the transmitter card FTVV in location B01. For dual channel units the second transmitter card is found in Location B03.

2.4.9 Rework the transmitter FTVV as shown in Figure 3A. Remove the letter "F" from the card type designation FTVV and mark a "G" in its place so that the card type becomes GTVV.

2.4.10 Install the GTVV into Location B01. For dual channels units, install the second GTVV into Location B03. Close top case assembly.

2.4.11 Update unit FCO log, add this option as an entry on the units FCO log.

2.5 <u>Installation Procedure for Storage Module Drive. CDC 9764 or 9766 Flat Cable Interface Index, Sector and Unit Select Modification</u>

NOTE: Skip this section for any drive other than 150 - 300 MB flat cable drive.

2.5.1 Power down unit and unplug power cable.

2.5.2 Open top case and remove side panel to access I/O connectors.

2.5.3 Move the following wires:

A Connector (60 Pin)	B Connector (26 Pin)
25	13
55	26
18	12
48	24

2.5.4 Remove and insulate the following wires from the other A Connector, 25,55 18, and 48.

2.5.5.1 Remove wire from B Connector Pin 9 .

2.5.5.2 Remove wire from B Connector Pin 22 and install in Pin Location 9.

2.5.5.3 Install other wire in Location 22.

2.5.6 Remove FTVV Logic Card from cage assembly location AØ 1 (and AØ3 for dual port drives.)

2.5.7 Rework card as shown in Fig. 3A. Remove the letter "F" and add letter "G".

2.5.8 Install GTVV card(s) in proper location(s).

2.5.9 Update FCO Log, add this option as an entry on the FCO Log.

3.0 Phoenix 211 Installation Instructions

3.1 General

The Phoenix 211 Disk Controller is designed to mount directly in any 10½" PDP11 processor chassis or BA11-K Expansion Box. Installation consists of:

a.) Physically mounting the Phoenix 211 systems unit in the PDP11 computer chassis or expansion box.

b.) Plugging the Phoenix 211 systems unit power harness to the computer or expansion box power distribution panel.

c.) Plugging in the four Phoenix 211 printed circuit boards supplied into the systems unit.

d.) Plugging in the Phoenix 211 interface board and cable into PDP11 SPC slot.

e.) Connecting the Disk Drive to the Phoenix 211 Controller using the supplied cables.

f.) Running diagnostics to verify the integrity of the disk subsystem.

Detailed Phoenix 211 installation instructions are contained in the following paragraphs.

3.2 Phoenix 211 Systems Unit Mounting Instructions

3.2.1 The Phoenix 211 Disk Controller Systems Unit will mount in any available systems unit slot in any PDP11 computer or BA11-K expansion box which allows insertion of full hex boards.

3.2.2 PDP11/05-NC, PDP11/05-SC, BA11-K Instructions

1.) Remove top and bottom covers of computer or expansion box to gain access to system unit mounting area.

Carefully save all screws removed for later installation.

2.) Tilt computer or expansion box up on slides to expose system unit wirewrap field.

Caution: Insure that existing cables have adequate slack before tilting unit.

Note: The PDP11/05-NC chassis requires a 9 position power connector, all other configurations are 15 position.

3.) Physically place the Phoenix 211 Systems Unit in the next available systems unit slot.

Caution: Insure that the system unit is physically oriented such that the systems unit power harness extends over the computer or expansion box power distribution panel.

4.) Fasten the systems unit to the computer or expansion box chassis by engaging the two captive screws located at each end of the systems unit with matching threaded holes in the computer or expansion box chassis.

Tighten the captive screws with a screwdriver.

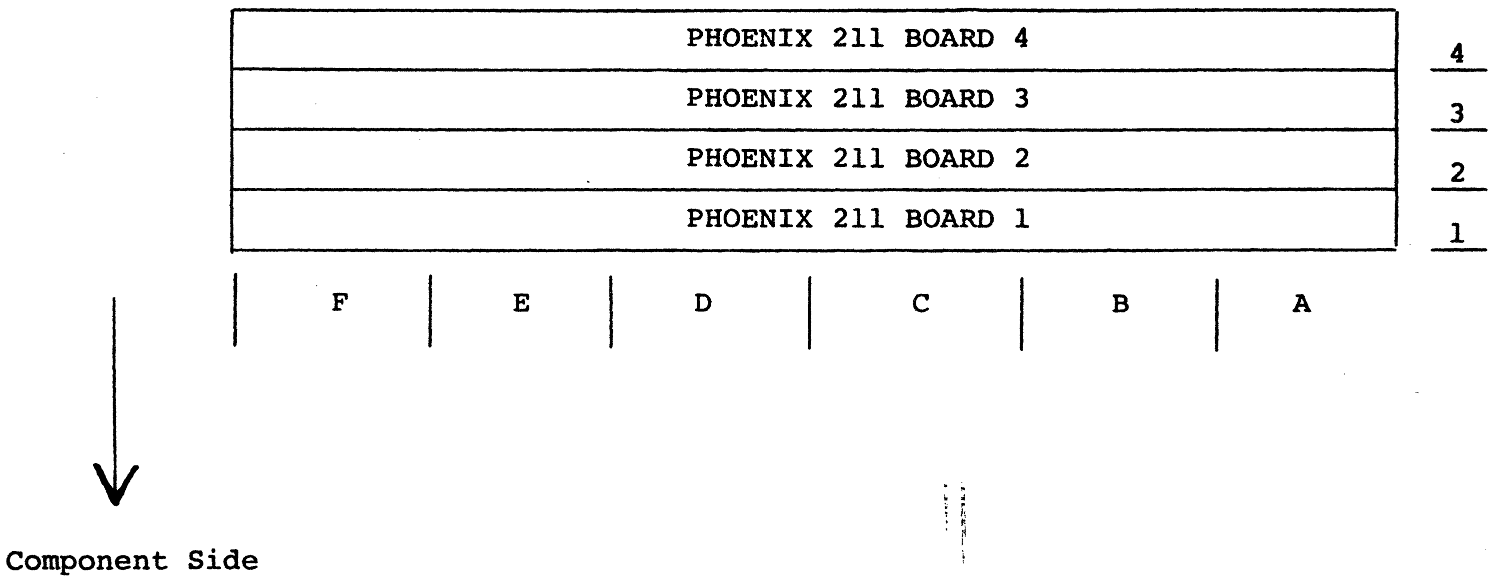
5.) Connect the systems unit power harness to the computer or expansion box power distribution panel by plugging the power harness connector into a matching distribution panel receptacle.

6.) With the computer in the "Halt" turn power on. Insure that +5v DC is present on the systems unit backplane printed circuit board between the power harness red wire pads (+5v) and the black wire pads (GND). Insure that -15VDC is present on the systems unit backplane between the power harness violet wire pad (-15) and the black wire pads (GND). Turn power off.

7.) Tilt computer and expansion box chassis to normal position.

PHOENIX 211 DISK CONTROLLER
SYSTEMS UNIT UTILIZATION

(Top of Board View)



4.0 Reference: Figure 4

Phoenix 211 Printed Circuit Board Installation

4.2 Carefully plug Phoenix 211 Board #1 into the system unit slot.

Caution: Insure that board is correctly oriented with respect to other boards before plugging in board.

4.3 Carefully plug Phoenix 211 Board #2 into the system unit slots A2-F2.

a.) Slide board carefully along computer or expansion box card guides until connector fingers engage the systems unit connector receptacles.

b.) Start board to seat into systems unit by firmly "rocking" the board into place by pushing down alternately on left and right handles of board.

c.) Seat board in final position by pushing down evenly on the board handles.

4.4 Carefully plug Phoenix 211 Board #3 and #4 into the system unit slots A3-F3 and A4-F4 respectively by following the procedure outlined in 4.3 for board #2.

4.5 Insure that all boards are seated properly by checking the relative height of the board handles.

4.6 Carefully plug the Phoenix 211 PDP11 coupler Board into the available standard small peripheral controller (SPC) slot by following the procedure outlined in section 4.3 and 4.5.

4.7 Install the interconnect cable between the Phoenix 211 Board #1 and the Phoenix 211 PDP11 interface Board.

IMPORTANT - Note the pin #1 designations on the cable and on the Boards.

5.0 NPR Grant Jumper Removal Instructions

1.) The Non-Processor Request grant is normally jumpered across in all standard peripheral slots. This line must be removed in order to operate the Phoenix 211 Disk Controller as in all direct memory access devices. This jumper is located between CA1→CB1.

2.) Reference Fig. 6 - Locate the C connector field of the SPC slot containing the Phoenix 211 PDP11 coupler card.

3.) Reference Fig. 6 - Remove the wirewrap wire connecting pins CA1 to CB1 of the slot containing the Phoenix 211 PDP11 coupler card and only of that slot.

Note: a.) Connector "C" is the third connector from the Unibus end of the backplane.

b.) The CA1 and CB1 pins are the first positions of the "C" connector (See Fig. 6).

c.) If the Phoenix 211 PDP11 coupler is ever relocated or removed, reconnect CA1 to CB1.

6.0 Phoenix 211 Cabling Instructions

The Phoenix 211 can have from 2 to 5 cables connected from Board #4 going to from one up to four disk drives.

6.1 A 60 conductor ribbon "A" cable and a 26 conductor ribbon "B" cable per drive is used to connect the Phoenix 211 formatter to the disk(s).

6.1.1 Remove the screws on the top rear of the computer or expansion box holding the cable clamp.

6.1.2 Plug the "A" cable (60 pin) into the connector on Board #4 making sure the pin #1 designations of both cable and connector are correctly aligned.

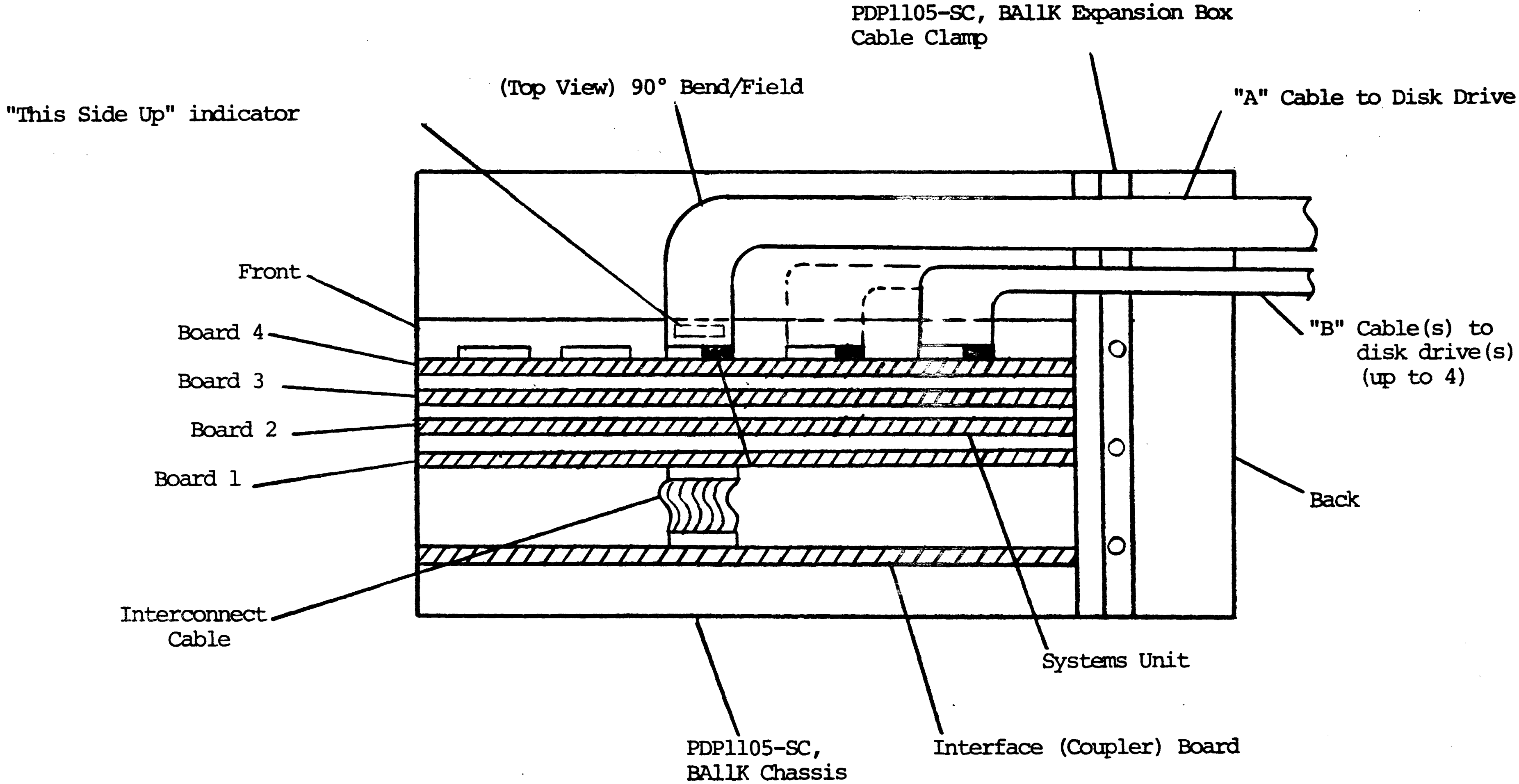
6.1.3 Plug the "B" cable(s) into the drive port connectors on Board #4.

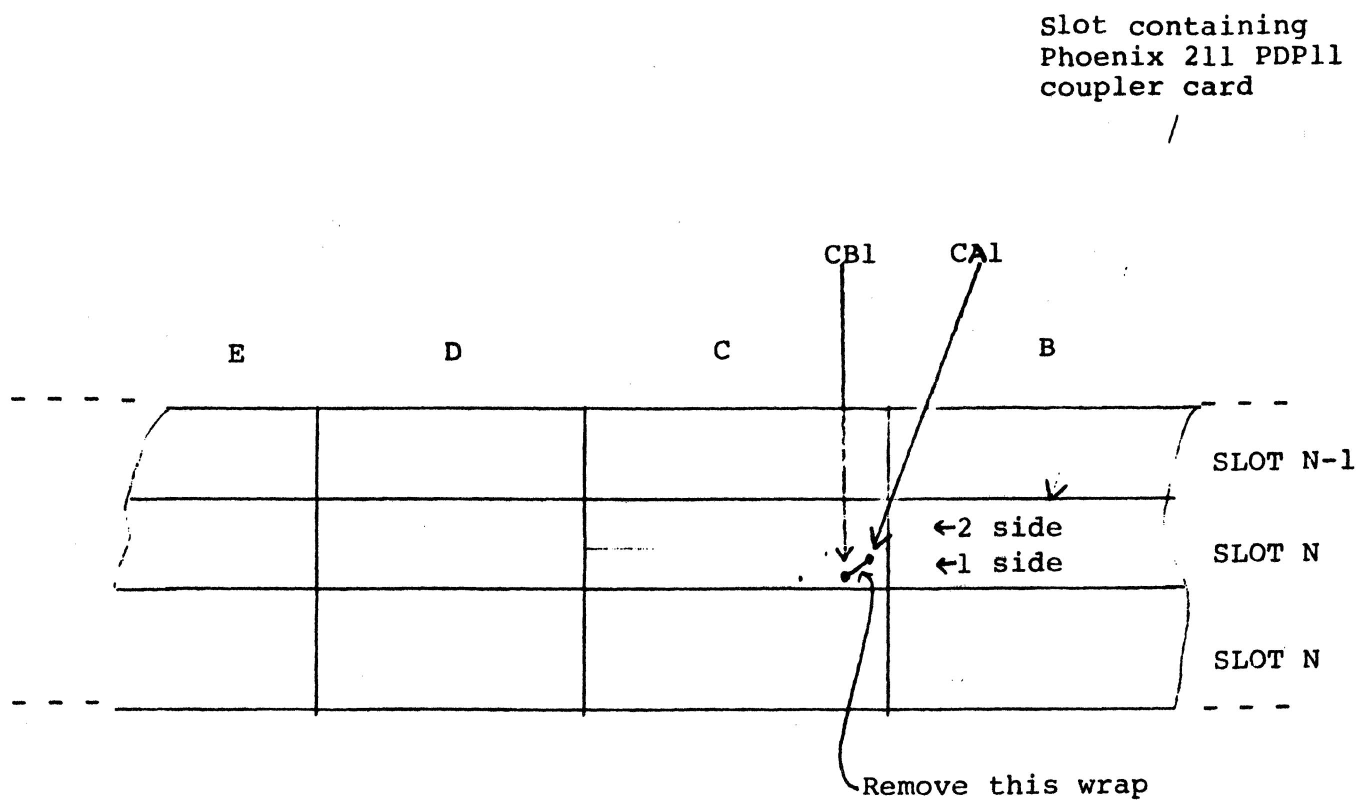
Note: The port designations do not correspond to the drive identification numbers, i.e. drive number Ø can be plugged into any port Ø-3 and it will remain drive Ø.

FIGURE 5

PHOENIX 211 DISK CONTROLLER

DISK DRIVE CABLE INSTALLATION





PHOENIX 211 DISK CONTROLLER
REQUIRED DMA GRANT BACKPLANE CONNECTIONS

FIGURE 6

NOTE: Backplane connector "A" is normally at the top, when the backplane is vertically mounted or at the front of the computer or expansion box when the backplane is horizontally mounted.

Caution: Insure that the cable connections are oriented such that pin #1 to the cable matches pin #1 of the receiving connector.

6.1.4 Dress the cables to obtain minimum strain on the board connectors. (this may require a 90 degree bend). Check that when the computer or expansion box cover is reinstalled all cables are clear if being pinched or cut. (See Fig. 5)

6.1.5 Reinstall the cable clamp and screws

7.0 Disk Drive Connection Instructions

For Flat Cable Interface skip to 7.4

Reference: Control Data Hardware

Maintenance Manual vol. 1 of 2 Pg. 1-10, 11

The disk drive cable is terminated at the disk drive end with a drive terminator plug (CDC P/N 40067207) which plugs into an "A" cable position, J4, at the rear of the drive.

7.1 Attach the "A" Cable to J3 at the rear of the disk drive.

7.2 Attach the "B" cable (s) to J2 at the rear of the disk drive(s).

7.3 Attach the terminator plug to J4 at the rear of the drive.

(See Note end of Section 7)

Continue on Section 8

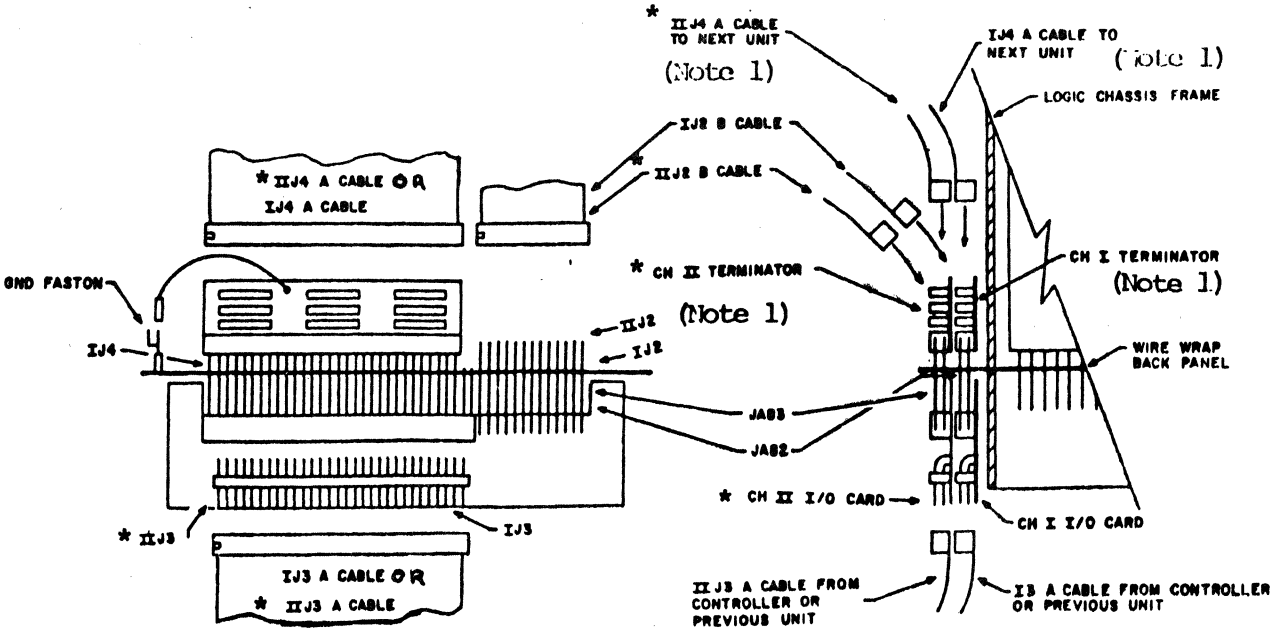
7.4 Attach the "A" Cable to IJ3 (IIJ3 for Port II). (See Fig. 7 for CDC9762)

7.5 Attach the "B" cable to IJ2 (IIJ2 for Port II).

7.6 Attach Terminator (s) to IJ4 (IIJ4 for Port II). See Note at end of Section 7.

Caution: Insure that the cable connections are oriented such that pin #1 to the cable matches pin #1 of the receiving connector.

Note: Units interfaced with more than one disk drive require a daisy chain cable. This cable is connected in place of the terminator plug. At the end of the chain a terminator is still required. (See Hardware Maintenance Manual Vol. 1 of 2 Fig. 1-7, pg. 1-11).



*All CHII Designations refer to Channel II on a dual port drive.

Notes 1.) If there is only one drive or if the drive being cabled is the last daisy chained drive a terminator must be installed. If the drive is to be in the daisy chain, the "A" cable from the previous unit must connect to J3 and the "A" cable to the next unit must connect to J4.

FIGURE 7

8.0 Initial Power Turn-On Instructions

8.1 Disk Drive Power Turn-On Instructions

Remove all shipping hardware and go through initial power turn-on procedure outlined by the disk drive manufacturer in his respective reference manual.

Warning: Failure to remove all disk drive shipping constraints before applying power may cause permanent damage to the disk drive.

8.2 Phoenix 211 Disk Controller Power Turn-On Instructions

8.2.1 With the computer in "Halt" mode turn power on to the system. Observe that the computer "run" indicator remains out.

8.2.2 Manual Console Control Check

Insure that the computer bus is normal by checking that all console switch register functions operate properly.

Check specifically for proper operation of "Load Address", Deposit, and Examine functions in multiple mode.

If console functions do not operate properly, check for proper Unibus connections, and termination.

8.2.3 Manual Disk Controller Status Check

8.2.3.1 With the disk drive stopped insert 764000 (Control and Status Reg.) Address into Console Switch Register.

8.2.3.2 With the Halt/Enable Switch in Halt Position, momentarily depress the "Load Address" and "Start" Console switches in succession.

8.2.3.3 Successively momentarily depress the "Examine" console switch. Observe that each of the Disk Controller Registers contains the proper content as specified in Table 1.

8.2.3.4 Run the Phoenix 211 Disk Controller Diagnostic per section 9.

TABLE 1

PHOENIX 211 DISK CONTROLLER

INITIAL POWER ON REG STATUS

REGISTER	ADDRESS	PROPER CONTENTS
Control & Status	164ØØØ	2ØØ
Unit, Sector, Head	164ØØ2	Ø
Buss Address	164ØØ4	Ø
Word Count	164ØØ6	Ø
Cylinder Address	164Ø1Ø	Ø
Disk Status	164Ø12	Ø
Disk Error	164Ø14	Ø

9.0 Diagnostic Program Loading & Operating Instructions

9.1 General

The Phoenix 211 Disk Controller Diagnostic is supplied as a paper tape load module. The diagnostic load module is in absolute format and can be loaded by the standard DEC Absolute Loader.

9.2 User Considerations -

.1 Diagnostic will destroy data on any disk cartridge contained in the disk drive when it is run.

User should put a scratch disk cartridge in the disk drive before running the disk diagnostic.

The Diagnostic Program requires a formatted scratch disk cartridge for proper operation. Disk cartridges may be formatted using the format program located at the end of the diagnostic (Test 49).

9.3 Detailed Operating Instructions -

Detailed operating instructions are contained at the beginning of the disk diagnostic.

10.0 Phoenix 211 Parameter Selection

The Phoenix 211 Disk ontroller may be configured to varied system application requirements by appropriately adjusting strappable control parameters. This section is devoted to defining all Phoenix 211 adjustable parameters and available settings.

10.1 Base Register Address Selection (Reference Drawing D-1025-08, Sht 3)

The base Unibus register address of the standard Phoenix 211 Disk Controller is selectable MOD 20_8 and the Command Queue version of the Phoenix 211 Disk Controller is selectable MOD 40_8.

10.1.2 The standard base address of the Phoenix 211 Disk Controller is 764000_8, and staples are provided on the Phoenix 211 Coupler Board to enable other base addresses to be utilized.

10.1.3 The Address selection logic is shown on the upper lefthand corner of sheet 4 of D-1025-08.

10.1.4 The following Table shows the selectable address bits and the corresponding control staples.

TABLE 1

Base Register Address Selection

Address Bit	Control Staple Point
A12	B15-13
A11	B15-15
A10	C15-02
A09	C15-04
A08	C15-06
A07	C15-11
A06	C15-13
A05	C15-15

Note: 1.) For an address bit to be a logic "1", the control staple point must be wired high to the pull up string associated with J5-15.

2.) For an address bit to be a logic "Ø", the control staple point must be wired low to any convenient ground point.

10.2 Interrupt Vector Address Selection (Reference D-1025-08, Sht 1)

10.2.1 The standard interrupt vector value for the Phoenix 211 Disk Controller is 270_8.

10.2.2 The interrupt vector control logic is shown on the lower right section of sheet 1 of D-1025-08.

10.2.3 Other interrupt vector values are selectable by wiring appropriate vector control signals directly to the corresponding Unibus Data Signals.

10.2.4 Table 2 below shows the interrupt vector Address bits and the corresponding required wiring.

TABLE 2

Interrupt Vector Address Selection

Vector Address Bit	Required Control Wiring
A02	A08-01 to CU2
A03	A08-04 to CT2
A04	A08-10 to CN2
A05	A08-13 to CP2
A06	C08-01 to CV2
A07	C08-04 to CM2

Note: 1.) Logic 1 = control wire in.

2.) Logic Ø = control wire removed.

10.3 Interrupt Priority Level Selection (Reference D-1025098, Sht 1)

10.3.1 The standard interrupt priority level for the Phoenix 211 Disk Controller is BR5.

10.3.2 The interrupt priority level of the Phoenix 211 Disk Controller is determined by staples at component location A9 on the Phoenix 211 Coupler Board as shown on the lower right hand corner of sheet 1 of D-1025-08.

10.3.3 Required BR4 Stapling

Required stapling for BR4 interrupt priority stapling is as follows:

1.) A09-01 to A09-05
2.) A09-02 to A09-15
3.) A09-03 to A09-14
4.) A09-04 to A09-13
5.) A09-07 to A09-11
6.) A09-12 to A09-16

10.3.4 Required BR5 Stapling

Required stapling for BR5 interrupt priority is as follows:

1.) A09-01 to A09-16
2.) A09-02 to A09-05
3.) A09-03 to A09-14
4.) A09-04 to A09-13
5.) A09-07 to A09-10
6.) A09-12 to A09-15

10.3.5 Required BR6 Stapling

Required stapling for BR6 interrupt priority is as follows:

1.) A09-01 to A09-16
2.) A09-02 to A09-15
3.) A09-03 to A09-05
4.) A09-04 to A09-13
5.) A09-07 to A09-09
6.) A09-12 to A09-14

10.3.6 Required BR7 Stapling

Required stapling for BR7 interrupt priority is as follows:

1.) A09-01 to A09-16
2.) A09-02 to A09-15
3.) A09-03 to A09-14
4.) A09-04 to A09-05
5.) A09-07 to A09-08
6.) A09-12 to A09-13

10.4 DMA Throttle Value Selection (Reference D-1025-08, Sht 2)

On the Phoenix 211 Disk Controller the number of consecutive DMA Memory cycles that are to be used by the disk controller for data transfers is uniquely selectable from 1 to 256 words via the DMA "Throttle" Control Logic and associated control staples are shown on the lower right hand corner of sheet 2 of D-1025-08.

10.4.1 The normal factory setting for the DMA Throttle is 8.

10.4.2 The DMA Throttle setting enables the user to adapt the Phoenix 211 Disk Controller to the Unibus environment and system configuration of his application via limiting the maximum number of consecutive DMA cycles utilized by the disk controller for data transfers.

10.4.3 Throttle Selection

DMA throttle value selection is accomplished via staples located on connector J1.

(J1 is the rightmost connector position at the top of the Phoenix 211 Interface Board.) Table 3 below defines the control staples and corresponding binary weighted values associated with the DMA Throttle Logic.

TABLE 3

DMA THROTTLE SELECTION

Staple No.	Binary Weight	Required Connection Points
1	1	J1-19 to J1-20
2	2	J1-21 to J1-22
3	4	J1-23 to J1-24
4	8	J1-25 to J1-26
5	16	J1-27 to J1-28
6	32	J1-29 to J1-30
7	64	J1-31 to J1-32
8	128	J1-33 to J1-34

Note: The DMA Throttle value selected is equal to the sum of the binary weights of the control staples inserted.

10.5 Maximum Head Value Selection (Reference Board 3 Logic Drawing D-1034-01, Sheet 5)

The Phoenix 200 Disk Controller may be used with a large number of disk drives of varying capacities and number of heads.

The disk controller will automatically abort on any attempt to utilize a non-existent head of any disk drive connected to it, provided that the maximum number of allowable heads to be utilized is defined to it via control staples.

10.5.1 The maximum head decode logic is shown on the lower right hand corner of sheet 5 of logic drawing D-1034-01.

10.5.2 Normal Maximum Head Selection. Normal or not mixed drive head configurations are provided by connecting or not connecting pins 1-5 of location E11 of Phoenix 200 Board 3 to ground as appropriate to select the desired maximum number of heads.

10.5.2.1 The interconnections on the logic drawing select a maximum of five heads (Ø-4).

10.5.2.2 Table 4 below shows the binary weight of each control pin at location E11.

TABLE 4

Maximum Head Value Selection

Board 3, Location E11 Control Pin No.	Binary Weight
1	1
2	2
3	4
4	8
5	16

Note: 1.) For a logic 1, leave pin open
2.) For a logic Ø, tie pin to ground at E11-15 or E11-8.
3.) The maximum head value selected is equal to the sum of all logic "1" or open E11 select pins plus 1.

10.5.3 Mixed Disk Drive Maximum Head Selection

When drives of different numbers of heads are connected to the disk controller at the same time, two alternatives are possible for head selection.

10.5.3.1 Mixed Drive Head Select Option 1

The user could select per above procedure the maximum number of heads allowable for the disk drive of the largest capacity.

The only possible disadvantage to this approach is that this would permit the programmer to attempt to overrun any disks of lesser capacity and could result in putting the disk drive in the Fault state.

10.5.3.2 Mixed Drive Head Select Option 2

Location E11 is already preconnected to accomodate a ROM. A ROM could be optionally inserted into location E11 (at extra cost) which would provide the correct maximum number of head value to the selection logic as a function of what disk drive is selected.

This solution thus will not allow the programmer to select a not existent head on any disk drive connected to the disk controller no matter what number of heads is associated with what disk drive.

The only restriction is that the system cabling configuration, once defined, must remain constant with respect to what capacity disk drive is connected to what radial port of board 4 of the disk controller.

10.6 Maximum Sector Value Selection (Reference Board 3 Logic Drawing D-1034-01, Sheet 5 and Board 3 Assembly Drawing D-1034-03)

10.6.1 The Phoenix 200 Disk Controller may be set up to accommodate a varying number of sectors per track by selection staples provided on Board 3 of the Formatter at Component location E11.

10.6.2 The standard factory setting for the number of sectors/track is 32.

10.6.3 The disk controller will automatically abort when any attempt is made to utilize a non existent sector of any disk drive connected to it.

10.6.4 The disk controller maximum sector selection and comparison logic is shown at the upper right hand corner of Sheet 5 of logic drawing D-1034-01.

10.6.5 The selection reference value established by the user by tying pins 1-7 of component location E10 to ground or leaving them open is compared with the sector address contained in the DUSH register.

A non existent sector error is generated whenever the contents of the sector field of the DUSH Register is greater than that specified by the strapping of location E10.

10.6.6 The strapping shown on the logic drawing is for the standard 32 sectors per track configuration.

10.6.7 Table 5 below shows the binary maximum sector selection value of each control pin at location E10.

Table 5

Maximum Sector Value Selection

Board 3 Location E10 Control Pin No.	Sector Select Binary Weight
1	1
2	2
3	4
4	8
5	16
6	32
7	64

Note:
1.) For a logic 1, leave pin open.
2.) For a logic Ø, tie the pin to ground at E10-15 or E10-08.
3.) The maximum sector value selected is equal to the sum of all logic "1" or open E10 select pins plus 1

10.7 Defective Sector Identification and Skip Option Selection (Reference Sheet 1 of Board 3 Logic Drawing D-1034-01 and Assembly Drawing D-1034-03.)

If any sectors on a disk medium utilized are defective, in terms of any bit cells being unable to properly contain both logic 1 and logic data states, as determined by a disk pack initialization utility after the medium is formatted, the disk controller may be optionally set up to skip over such defective sectors whenever they are encountered during normal data transfers.

10.7.1 Normal Handling of "Hard" Defective Sectors

Hard defective sectors are normally located by disk utility programs such as "BAD" under RSX11-M.

The utility program rotates a worst case data pattern through every character of every sector data field on the disk medium.

Any defective sectors found are cataloged by the utility program and allocated to a file created on the medium called "BAD".

Hence, all "hard" defective sectors are then prevented from being allocated to normal user programs by the file management and no special action is required by the disk controller.

10.7.2 Normal Disk Controller Staple Setting

The proper factory setting for the disk controller for normal handling of any "hard" defective sectors is for staple W1 on Board 3 to be removed.

10.7.3 Optional Skip on Detection of Defective Sector

Optionally the disk controller may be set up to automatically skip over any hard defective sectors referenced in any data transfer operation.

10.7.3.1 In this mode of operation the header of any defective sectors found by the disk initialization utility program would be reformatted to set the two "Bad Sector Flag" bits contained in the sector header.

10.7.3.2 The disk controller would then automatically skip over any such sector when the Bad Sector Flag bits were accessed along with the header during sector position verification.

10.7.3.3 This mode of operation permits the user to perform multiple sector transfers involving defective sectors without having the data transfers to be delimited by defective sectors.

10.7.3.4 This mode of operation is selected by connecting staple W1 on board 3 as shown on the lower left hand corner of Sheet 1 of logic drawing D-1034-01.

10.7.4 Optional Abort on Detection of Defective Sector

10.7.4.1 The disk controller may be optionally set up to abort any operation involving a flagged defective sector in any data transfer on the basis that the file management system should not allocate any defective sectors for transfers.

10.7.4.2 In this mode of operation defective sectors are determined by the disk utility and the Bad Sector Flag bits are set in the headers of all defective sectors as above.

10.7.4.3 However, whenever the disk controller detects that a defective sector is being accessed, as determined by the presence of the "Bad Sector Flag" header bits, the disk controller will immediately abort whatever data transfer is being conducted.

10.7.4.4 This abort on detection of defective sector mode is selected by removing staple W1 from board 3 as shown on the lower right hand corner of Sheet 1 of logic drawing D-1034-01.

10.7.5 W1 Control Staple Location (Reference Board 3 Assembly Drawing D-1034-03)

The defective sector control staple W1 is etched onto the etch side of the printed circuit board just above and parallel to resistor R9.

10.8 CDC Disk Radial Port Selection (Reference Sheet 2 of CDC Board 4 Logic Drawing D-1035-01 and CDC Board 4 Assembly Drawing D-1035-03)

10.8.1 The Phoenix 200 Disk Formatter can control up to four disk drives.

Each disk drive requires connection to the disk formatter via the so called "B" or "Radial" Cable.

10.8.2 Four 26 pin Berg connectors are provided on Board 4 of the formatter to accomodate up to four radial disk drive cables. These connectors are labeled J1, J2, J4 and J5, with J1 being the rightmost connector on the board and J5 being the leftmost connector.

10.8.3 Any disk drive radial cable may be connected to any physical radial connector provided on the board.

However, any radial connector ports not being used should be disabled by staple selection to prevent introduction of noise into the disk formatter.

10.8.4 The radial cable interface logic for the four possible disk drives is contained on Sheet 4 of the CDC Board 4 logic drawing D-1035-01.

10.8.5 The presence of control staples W35, W6, W34 and W33 enable the formatter receiving logic for connectors J1, J2, J4 and J5 respectively. (These are located in the middle of each port logic section.)

10.8.6 The factory staple settings for various numbers of disk drives is shown below.

Number of Connected Disk Drives	Connectors Used	Staples Inserted
1	J1	W35
2	J1,J2	W35,W6
3	J1,J2,J4	W35,W6,W34
4	J1,J2,J4,J5	W35,W6,W34,W33

Note: User must use radial connectors corresponding to radial port enable staple configuration.

10.9 Rotational Positioning Sensing Selection (Reference Sheet 3 of CDC Board 4 Logic Drawing D-1035-01 and Board 4 Assembly Drawing D-1035-03)

10.9.1 Rotational Position Sensing (RPS) is a standard feature of the Phoenix 200 Formatter. Rotational Position Sensing enables a user to maximize disk subsystem throughput by considering sector position information as well as head cylinder position information in transfer allocation among multiple disk drives.

10.9.2 Seek Complete in the Phoenix 200 Formatter is defined as being both "ON CYLINDER" and "ON SECTOR".

To allow for operating system overhead the seek complete signal is normally generated four sectors before the actual target sector allowing a nominal two millisecond for software setup time. (This delay is programmable.)

10.9.3 The Rotational Positioning Sensing logic is shown on Sheet 3 of the CDC Board 4 logic drawing D-1035-01.

10.9.4 Rotational Position Sensing Enable/Disable Control Staple

10.9.4.1 Rotational Position Sensing is enabled if control staple W3 is inserted. When this staple is inserted, seek complete cannot be generated until the current actual and reference commanded sector are identical, as determined by the sector comparator logic.

When sector comparison occurs, signal LASECCOMPH is high or active.

10.9.4.2 When staple W3 is removed, rotational position sensing is disabled as LASECCOMPH is always active.

10.9.5 Sector Remapping ROM's

10.9.5.1 In order to generate the sector comparison signal four or more sectors ahead of the actual target sector, the actual target sector address is mapped via ROM's A7, C7, B7 & C8 to the address of the sector two after it.

10.9.5.2 ROM A7 is normally the only one supplied with a standard 32 sector formatter, since only one (32 X 8) ROM is required to remap 32 sectors.

10.9.5.3 ROM's C7,B7 & C8 are only required if the formatter is set up for more sectors as indicated on the logic drawings.

10.9.6 Temporary Storage of Reference Target Sector Address

10.9.6.1 Seek Commands for multiple drives may be overlapped by the user, and the formatter has only one sector command field in the DUSH Register.

10.9.6.2 To provide rotational position sensing in an overlapped seek environment, the commanded sector must be saved temporarily to provide a reference target sector address for each disk drive.

10.9.6.3 The commanded target sector for each disk drive is stored in the internal registers implemented by components A9 & A10 when any Seek operation is initiated. This reference sector value is then accessed and used by the seek sector comparison logic.

Note: To utilize rotational position sensing properly, the user must define the target sector at the time that a Seek Only command is initiated.

10.10 Optional Sector Interleaving Selection (Reference Sheet 5 of CDC Board 4 logic drawing D-1035-01 and Board 4 Assembly drawing D-1035-03.

10.10.1 The Phoenix 200 Formatter supports absolute contiguous sector addressing as standard and interleaved sector addressing as optional.

10.10.2 The Sector interleaving control logic is shown on the upper right hand corner of Sheet 5 of Board 4 logic drawing D-1035-01.

10.10.3 Sector Interleaving in the Phoenix 200 Formatter is achieved by ROMS C6,D6,A6 and B6 so that any desired interleaving can be readily programmed. One to four ROM's may be required for interleaving depending on how many sectors are used, as shown on the logic drawing.

10.10.4 Phoenix 200 Formatters are normally set up at the factory for absolute contiguous sector addressing so that at location C6, jumpers are shown directly connecting input sector address lines to corresponding output sector address lines.

10.10.5 A one time, non-recurring charge will be assessed for any controllers ordered with interleaving to recover programming and testing costs.

10.10.6 Control Staple Settings for Interleaving More Than 32 Sectors

When sectors are to be interleaved, control staples W1 and W2 must be removed.

Table 6

Sector Interleaving Required Control Staples and ROMS

No. of Sectors	Required Staple Deletes	Required ROMS
Ø - 31	W1,W2	C6
32 - 63	W1,W2	C6,D6
64 - 95	W1,W2	C6,D6,A6
96 - 127	W1,W2	C6,D6,A6,B6

10.11 Maximum Cylinder Selection Logic (Reference Sheet 4 of CDC Board 4 Logic Drawing D-1035-01 and CDC Board 4 Assembly Drawing D-1035-03

10.11.1 The Phoenix 200 Formatter may be used with a large number of disk drives with a varied number of cylinders.

The disk formatter will automatically abort on any attempt to utilize a non-existent cylinder of any disk drive connected to it, provided that the maximum number of allowable cylinders to be utilized is defined to it via control staples.

10.11.2 The maximum cylinder decode logic is shown on the lower right hand corner of Sheet 4 of the CDC Board 4 logic drawing D-1035-01.

10.11.3 This logic compares the cylinder address contained in the formatter Cylinder Address Register with a maximum cylinder value specified by the user via staples or ROM's at component locations E12 & F12.

10.11.4 Any time that the contents of the Cylinder Address Register is larger than the user defined maximum value, a Non-Existent Cylinder Error will be generated and cause any transfer operation to be aborted.

10.11.5 Normal Maximum Cylinder Selection

Normal maximum cylinder address selection is accomplished by connecting or not connecting pins 1-7 of location E12 and pins 1 & 2 of location F12 to ground as appropriate to select the desired maximum cylinder value.

10.11.5.1 The interconnections shown on the logic drawings select a maximum cylinder value of 823 appropriate to a CDC 9762 80 MB Disk Drive.

10.11.5.2 Table 8 shows the binary weight of each control pin involved in maximum cylinder selection.

Table 8

Maximum Cylinder Control Pins

Board 4 Control Location & Pin	Cylinder Select Binary Weight
E12-01	1
E12-02	2
E12-03	4
E12-04	8
E12-05	16
E12-06	32
E12-07	64
E12-09	128
F12-01	256
F12-02	512

Note: 1.) For a logic 1, leave pin open.
2.) For a logic Ø, tie pin to ground at pin 08, or 15 of component location.
3.) The maximum cylinder value selected is equal to the sum of all logic "1" or open select pins plus 1.

10.12 External -5VDC Supply Option Selection (Reference CDC Board 4 Logic Drawing D-1035-01 Sheet 1 (top center of page) and Assembly Drawing D-1035-03.

The Phoenix 200 Formatter requires -5VDC power as bias voltage for the differential drivers and receivers used to interface with the CDC disk drives.

The -5VDC is normally generated from -15VDC by a 7905 power regulator as shown on the logic diagram. Provision is made to enable the user to supply the -5VDC voltage directly from an external source by removing the supplied 7905 regulator and inserting staple W36.

Note: 1.) Staple W36 is a vertical staple just to the right of and below SIP component RP131 just above the "A" connector of Board 4.
2.) If the user chooses this option, -5VDC can be brought into the formatter via any unused pin on the power harness to backplane pin AN2.

10.13 Maintenance Error Abort Inhibit Option Selection (Reference Board 3 Logic Drawing Sheet 4 (top center of page) and Assembly Drawing D-1034-03.

10.13.1 For maintenance fault isolation purposes it may be desirable to temporarily inhibit the formatter from aborting any commanded operation at the detection of any error condition.

10.13.2 This maintenance only mode of operation can be entered by removing staple W9 on Board 3.

10.13.3 Staple W9 is etched diagonally onto the printed circuit board immediately above and to the left of component G8.

Warning!!

If staple W9 is temporarily removed for maintenance purposes, it must be replaced before putting the formatter back into normal service.

10.14 Maintenance Mode Data Complementation Option Selection (Reference CDC Board 4 Logic Drawing D-1035-01 Sheet 4 (top center) and Assembly Drawing D-1035-03

10.14.1 In Maintenance Mode the contents of the 32 bit header are used to determine the validity of all internal data paths back to computer memory.

The data words determined by the contents of the header words are used as received disk data and transmitted to the host computer via normal means.

10.14.2 By varying the contents of the Cylinder and Unit, Sector Head Registers, varying data patterns may be used to thoroughly exercise all formatter logic and data paths.

10.14.3 Staples W27 and W28 determine whether or not the header data will be continuously used as "data" for the number of words to be read as determined by the commanded transfer block length, or whether the data will be complemented every other transmission.

10.14.4 If staple W27 is in place, the contents of the header words will be used continuously as data words and the pattern will repeat every two words.

Note: Staple W27 is a vertical run etched onto the printed circuit board immediately above decoupling capacitor C85.

10.14.5 If staple W28 is in place, the header words will be complemented after each two words resulting in a four word repeating data pattern.

Note:
1.) The staple W28 common point is immediately above decoupling capacitor C85 and the other point is diagonally to the left at the same height as staple W27 connection point.
2.) Staple W27, which is etched onto board, must be cut is staple W28 option is to be selected.

10.15 Header Format Staple Selection (Reference Sheet 4 of CDC Board 4 Logic Drawing D-1035-01, Sheet 4 (right hand side) and CDC Board 4 Assembly Drawing D-1035-03)

10.15.1 The Phoenix 200 Disk Dormatter utilizes two basic header formats, one for the Control Data Corporation family of disk drives and one for Calcomp Disk Drives.

10.15.2 Header Cylinder Format Staple Selection

The 32 bit Header Shift Register and associated staples are shown on the lower right side of Sheet 4 of Logic Drawing D-1035-01.

Note: The logic drawing shows the proper staples for the normal Control Data Corporation disk drive family. Table 9 below shows the required staples for both Control Data Corporation and Calcomp Disk Drives.

10.15.3 Header Cylinder Comparison Enable Count Selection

A 4 stage counter at location A2 of Board 4 is used to determine when the Cylinder Bits of the header are to be compared with the current contents of the Cylinder Register during the position verification cycle performed by the formatter at the beginning of each sector.

This logic and associated control staples are shown at the upper center of Sheet 4 of CDC Board 4 logic drawing D-1035-01.

Table 9

Header Format Cylinder Staple Selection

(Reference D-1035-01 Sheet 4, D-1035-03)

Staple #	CDC Configuration	Optional Configuration	Staple Location
W7	OUT	IN	E9
W8	IN	OUT	E9
W9	OUT	IN	E9
W10	IN	OUT	E9
W11	OUT	IN	E9
W12	IN	OUT	E9
W13	OUT	IN	E10
W14	IN	OUT	E10
W15	OUT	IN	E9
W16	IN	OUT	E9
W17	OUT	IN	E9
W18	IN	OUT	E9
W19	IN	OUT	E10
W20	OUT	IN	E10
W21	IN	OUT	E10
W22	OUT	IN	E10
W23	IN	OUT	E10
W24	OUT	IN	E10
W25	IN	OUT	E10
W26	OUT	IN	E10

Note: All staples required for normal CDC configuration are etched on board and must be cut if header format is to change.

10.15.3.1 When the board is configured normally for CDC disk drives, the counter is prebiased with a count of Ø, enabling the cylinder comparison to begin during the second header word,

10.15.3.2 When the board is configured for Calcomp Disk Drive emulation, the counter is prebiased with a count of 14_8 enabling the cylinder comparison to begin at the beginning of bit 13 of the first header word.

10.15.3.3 The logic drawing shows proper staple selection for the normal CDC disk drive configuration. Table 10 below shows proper staple selection for both CDC and Calcomp Disk Drive configurations.

Table 10

Header Cylinder Comparison Enable Count Selection

(Reference D-1035-01 Sheet 4, D-1035-03)

Staple #	CDC Configuration	Calcomp Configuration	Staple Location
W29	IN	IN	Immed.Below A2
W30	IN	IN	Immed.Below A2
W31	IN	OUT	Immed.Below A2
W32	IN	OUT	Immed.Below A2

10.16 FIFO Memory Storage Capacity Selection (Reference Board 2 Logic Drawing D-1033-01, Sheet 5 and Assembly Drawing D-1033-03)

10.16.1 The Phoenix 200 Formatter may be equipped with 128, 256, or 512 words of FIFO memory. Standard memory factory FIFO capacity is 128 words.

10.16.2 The formatter FIFO storage is provided by two physical memory banks, each of which contains half of the storage provided.

10.16.3 The memory banks are alternately accessed to half the access time of one equivalent memory bank.

10.16.4 Both memory banks are shown on Sheet 5 of the logic drawing wired for a full 256 word storage capacity each.

10.16.5 In actual practice the FIFO memory banks are depopulated when less than 512 words of storage are to be utilized, and jumper wires are added from the data outputs of last used memory elements to the output of pin of the last corresponding memory element location in the matrix.

10.16.6 Table 11 shows the components required for 128 and 256 memory storage capacity options while Table 12 shows the required jumper wires for these capacities.

<u>Caution</u>: The FIFO Request Delay Selection Logic must also be changed anytime that the FIFO memory capacity is changed. See Section 10.17.

Table 11

FIFO Memory Elements Required for 128 AND 256 Word Capacities

(Reference Board 2 Logic Drawing D-1033-01 Sheet 5 and Assembly Drawing D-1033-03)

1.) Memory Elements Required for 128 Word Storage Capacity

Memory Bank Ø

U87
U98
U1Ø9
U121

Memory Bank 1

U92
U1Ø3
U114
U126

2.) Memory Elements Required For 256 Word Storage Capacity

Memory Bank Ø

U87,U88
U98,U99
U1Ø9,U11Ø
U121,U122

Memory Bank 1

U92,U93
U1Ø3,U1Ø4
U114,U115
U126,U127

<u>Note</u>: The basis memory element is a 64 x 4 FIFO MOS memory device.

Table 12

Jumper Wire Connections Required for 128 & 256 Word Memories

(Reference Board 2 Logic Drawing D-1033-01 Sheet 5 and Assembly Drawing D-1033-03)

Memory Bank Ø Required Jumpers

Data Bit Number	128 Word Origin Point	256 Word Origin Point	Destination Point
Ø	U122-04	U123-04	U124-13
1	U122-05	U123-05	U124-12
2	U122-06	U123-06	U124-11
3	U122-07	U123-07	U124-10
R(0-3)	U122-03	U123-03	U124-14
SO(0-3)	U122-02	U123-02	U124-15
4	U110-04	U111-04	U112-13
5	U110-05	U111-05	U112-12
6	U110-06	U111-06	U112-11
7	U110-07	U111-07	U112-10
R(4-7)	U110-03	U111-03	U112-14
SO(4-7)	U110-02	U111-02	U112-15
8	U99-04	U100-04	U101-13
9	U99-05	U100-05	U101-12
10	U99-06	U100-06	U101-11
11	U99-07	U100-07	U101-10
R(8-11)	U99-03	U100-03	U101-14
SO(8-11)	U99-02	U100-02	U101-15
12	U88-04	U89-04	U90-13
13	U88-05	U89-05	U90-12
14	U88-06	U89-06	U90-11
15	U88-07	U89-07	U90-10
R(12-15)	U88-03	U89-03	U90-14
SO(12-15)	U88-02	U89-02	U90-15

Memory Bank 1 Required Jumpers

Data Bit Number	128 Word Origin Point	256 Word Origin Point	Destination Point
0	U127-04	U128-04	U129-13
1	U127-05	U128-05	U129-12
2	U127-06	U128-06	U129-11
3	U127-07	U128-07	U129-10
R(0-3)	U127-03	U128-03	U129-14
SO(0-3)	U127-02	U128-02	U129-15
4	U115-04	U116-04	U117-13
5	U115-05	U116-05	U117-12
6	U115-06	U116-06	U117-11
7	U115-07	U116-07	U117-10
R(4-7)	U115-03	U116-03	U117-14
SO(4-7)	U115-02	U116-02	U117-15
8	U104-04	U105-04	U106-13
9	U104-05	U105-05	U106-12

Table 12

Memory Bank 1 Required Jumpers (Cont'd)

Data Bit Number	128 Word Origin Point	256 Word Origin Point	Destination Point
10	U104-06	U105-06	U106-11
11	U104-07	U105-07	U106-10
R(8-11)	U104-03	U105-03	U106-14
SO(8-11)	U104-02	U105-02	U106-15
12	U93-04	U94-04	U95-13
13	U93-05	U94-05	U95-12
14	U93-06	U94-06	U95-11
15	U93-07	U94-07	U95-10
R(12-15)	U93-03	U94-03	U95-14
SO(12-15)	U93-02	U94-02	U95-15

10.17 FIFO Memory Request Delay Selection (Reference Board 2 Logic Drawing D-1033-01, Sheet 4 (bottom right hand corner and Assembly Drawing D-1033-03)

10.17.1 When a word is entered into the FIFO memories contained in the formatter the data word has to "fall through" all 64XN locations of the memory before it can be accessed and retrieved from the memory.

10.17.2 The amount of time to "fall through" the memory elements is a direct function of how large the memory is and how many FIFO elements the data has to serially fall through.

10.17.3 When a data word is entered into a FIFO memory, no formatter action can be taken on such word until the word is available for access after fall through time has elapsed.

10.17.4 All requests for formatter direct memory access based upon FIFO data word entry are delayed by a time equal to the data FIFO memory "fall through: time by the logic shown on Sheet 4 prior to being used by the formatter.

10.17.5 This is accomplished by entering a bit into the delay logic when data is entered into or retrieved from the FIFO memory. This bit is then shifted through the required number of stages at a known clock frequency. When the delayed request signal appears at the selected output point of the delay logic, it can be safely used to cause a DMA transfer operation to load or access the FIFO memory.

10.17.6 Staples W44, W45, and W46 adjust the request delay to 128, 256 or 512 word memory capacities respectively and only one of these staples must be inserted at any one time. Table 13 shows the staple locations as well as the associated memory capacities.

Table 13

FIFO Memory "Fall Through" Request Delay Selection

(Reference Board 2 Logic Drawing D-1033-01
Sheet 2 and Assembly Drawing D-1033-03)

FIFO Memory Capacity	Staple Required	Staple Location
128	W44	Immed.Below A1-06
256	W45	Immed.Below A1-07
512	W46	Immed.Below A1-08

Note: W44 is etched onto the etch side of the printed circuit board and must be cut if W45 or W46 are to be inserted.

10.18 Pack Change Seek Interrupt Selection (Reference Board 2 Logic Drawing D-1033-01 Sheet 3 (bottom left of page) and Assembly Drawing D-1033-03)

10.18.1 Whenever a disk drive is powered down and then up by an operator the transition is detected by the disk controller and the "Pack Change" Status bit is set in the Disk Status Register.

10.18.2 If staple W28 is inserted in board 2 of the formatter the transition can also generate a Seek Done Interrupt to flag the host computer of the occurrence of the change in the disk drive status.

10.18.3 The Pack Change Interrupt is normally not enabled at the factory (W28 is removed)

10.18.4 Staple W28 connection points are immediately below and to the left and right of U50-01 and U50-02 respectively.

10.19 Optional Data Port Interface Selection

10.19.1 General

The data bus of the Phoenix 200 Formatter may be uniquely isolated from the control bus via staples. This permits the user to utilize his computer to control and initiate data transfers between the selected disk drive(s) and user external equipment physically connected to the Optional Data Port Connector located on Board 2 of the formatter.

The following paragraphs of this section are devoted to defining all of the staple changes required to put the formatter in the Optional Data Port Configuration.

10.19.2 Required Data Bus Modifications (Reference Board 2 Logic Drawing D-1033-01 Sheet 1 and Assembly Drawing D-1033-03)

10.19.2.1 Data Bus Transceiver Insertion

10.19.2.1.1 The Optional Data Port Interface Logic is shown on the extreme right of Sheet 1 and is comprised of components U130, U118, U107 and U96.

10.19.2.1.2 The transceiver receiver outputs ($\overline{\text{DØH-D15H}}$) are hardwired to the input of the 2=1 data multiplexer and latch shown on the extreme left of Sheet 1.

10.19.2.1.3 DMA write data and disk read data are multiplexed and latched by the 74298's shown on extreme left of Sheet 1 prior to being written into one of the two formatter FIFO memories.

10.19.2.1.4 To prevent noise from entering the formatter through the unused transceivers in normal formatter configurations, the transceivers U130, U118, U107 and U96 are not normally provided.

10.19.2.1.4 Transceivers U130, U118, U107 and U96 must be physically inserted into board 2 of the formatter for the Optional Data Port Configuration.

10.19.2.2 Isolation of Control Bus and Optional Data Port Input Buses

10.19.2.2.1 With the Optional Data Port Transceivers in place, incoming data from the Optional Data Port Interface and host computer are both connected to same input of the data multiplexer and latch shown on the extrmem left of sheet 1.

10.19.2.2.2 To isolate the incoming computer and Optional Data Port data, staples W12-W27 must be removed.

Note: 1.) All of these staples are etched onto the printed circuit board and must be carefully cut with an XACTO knife.

2.) Staples W14, W15 and W16 are horizontal staples located immediately below U86.

3.) All other staples (W17-W27) are physically located immediately below U120. (Staples run vertically in two rows.)

10.19.2.2.3 Isolation of Outgoing Data from Internal Bus

10.19.2.2.3.1 To prevent data being sent to the Optional Data Port Interface (during any Read operation) from being gated onto the internal formatter bus staple W5 must be removed.

10.19.2.2.3.2 Staple W5 is etched onto the printed circuit board and must be carefully cut.

10.19.2.2.3.3 Staple W5 is the second vertical staple from the left of the lower row of staples occupying area directly above U65-15.

10.19.2.2.4 Gating of Data Onto Optional Data Port Interface

10.19.2.2.4.1 In order to enable the DMRDSTBL Signal from the Optional Data Port to enable data to be gated onto the Optional Data Port Interface, two staples must be removed and two more must be added as follows.

10.19.2.2.4.2 Staple Removals

Staple Number	Location
W6	Vertical staple above U65-16
W9	Vertical staple below U66-02

Note: W6 and W9 are etched onto the printed circuit board and must be cut carefully.

10.19.2.2.4.3 Staple Additions

Staple Number	Location
W7	Vertical staple above U65-13
W8	Vertical staple above U65-14

10.19.2.2.5 Clocking of Optional Data Port Data into Formatter Latch

10.19.2.2.5.1 Optional Data Port Interface data during Write operations is strobed into the 74298 latches prior to being written into one of the two formatter FIFO memories.

10.19.2.2.5.2 In order to allow the Optional Data Port DMLDSTB Signal to clock the data into the latches, staple W10 must be removed and staple W11 must be inserted.

10.19.2.2.5.3 Staple W10 is etched onto the printed circuit board and is located below U47. This staple runs horizontally below U47-05 to U47-07.

10.19.2.2.5.4 Staple W11 is located below U47 and is to run horizontally from connection points approximately at U47-01 and U47-03.

10.19.2.2.6 Gating of Optional Data Port DMACK Signal to DMA Request Logic (Reference Board 2 Logic Drawing D-1033-01 Sheet 4 (top center of sheet) and Assembly Drawing D-1033-03)

10.19.2.2.6.1 The Optional Data Port Interface Signal DMACK must be gated into the formatter DMA Request Logic shown on Sheet 4 in lieu of the normal computer supplied DMA acknowledge signal.

10.19.2.2.6.2 This is accomplished by removing staple W39 and inserting staple W38.

10.19.2.2.6.3 Staple W39 is a horizontal staple located below U60 and running approximately from U60-07.

10.19.2.2.6.4 Staple W38 should be located immediately to the left of the original W39 staple.

10.19.2.2.7 Gating of Optional Data Port Overflow Signal

(Reference Board 2 Logic Drawing D-1033-01, sheet 4 (left center of sheet) and Assembly Drawing D-1033-03)

10.19.2.2.7.1 The Optional Data Port Word Count Overflow Signal WCROVFL is enabled for internal formatter use by inserting staple W37 into Board 2.

10.19.2.2.7.2 Staple W37 is vertically installed between points immediately to the left of U119.

10.19.2.2.8 Disabling of Normal DMA Word Count Overflow Signal

(Reference Board 1 Logic Drawing D1032-01, Sheet 1 and Assembly Drawing D1032-04)

10.19.2.2.8.1 The normal DMA Word Count Register Overflow Signal (WCROVH) generated on the Phoenix 211 Interface Board must be disabled when using the Optional Data Port Interface, since it is to be supplied by user external equipment.

10.19.2.2.8.2 This is accomplished by physically removing staple W1 shown on the top center of sheet 1 of Board 1 Logic Drawing D1032-01.

10.19.2.2.8.3 (Reference Board 1 Assembly Drawing D1032-04) Staple W1 is a vertical staple etched onto the printed circuit board on the component side, immediately to the left of component F9 and must be carefully cut.

10.19.2.2.9 Disabling of Normal DMA DMDATIL Signal (Reference Board 1 Logic Drawing D1032-01, Sheet 1 and Assembly Drawing D1032-04).

10.19.2.2.9.1 The normal DMA Input Data Signal (DMDATIL) generated on the Phoenix 211 Interface Board must be disabled when using the Optional Data Port Interface.

10.19.2.2.9.2 This is accomplished by physically removing staple W2 shown on the center of Sheet 1 of Board 1 Logic Drawing D1032-01.

10.19.2.2.9.3 (Reference Board 1 Assembly Drawing D1032-04) Staple W2 is horizontal staple located immediately below and connected to pin 4 of component G8.

10.19.2.2.10 Disabling of Formatter DMA Request Signal (DMRL) To 211 Interface Board, (Reference Board 1 Logic Drawing D1032-01, Sheet 1 and Assembly Drawing D1032-04)

10.19.2.2.10.1 The normal DMA Request Signal generated by the formatter (DMRL) must be inhibited from making DMA requests to the Phoenix 211 Interface Board when the Optional Data Port Interface is used.

10.19.2.2.10.2 This is physically accomplished by removing staple W3 shown on the top right side of sheet 1 of Board 1 Logic Drawing D1032-01.

10.19.2.2.10.3 (Reference Board 1 Assembly Drawing D1032-04) Staple W3 is a horizontal staple physically etched onto the etch side of the printed circuit board, immediately above and connected to pin 5 of mini component J8B. (left most mini dip at location J8). The staple can be readily removed by carefully cutting the etch on the component side of the board connecting J8B-04 to the feed through pad immediately above J8B-04.

10.19.2.2.11 Disabling of Formatter DMA Directional Signal (DMDIRL) to 211 Interface Board (Reference Board 1 Logic Drawing D1032-01 Sheet 1 and Assembly Drawing D1032-04)

10.19.2.2.11.1 The normal DMA Direction Definition Signal (DMDIRL) generated by the formatter to the 211 Interface Board must be disabled when the Optional Data Port Interface is used.

10.19.2.2.11.2 This is physically accomplished by cutting the etch connecting the DMDIR driver output to the associated 01 connector pin as shown on the top of sheet 1 of Board 1 Logic Drawing D1032-01.

10.19.2.2.11.3 (Reference Board 1 Assembly Drawing D1032-04) The etch should be carefully cut on the component side of the printed circuit board immediately above Pin 5 of mini component J8A. (rightmost mini dip at location J8)

10.19.2.2.12 Disabling of Formatter Data Onto Data Bus to 211 Interface Board (Reference Board 1 Logic Drawing D1032-01, Sheet 1 and Assembly Drawing D1032-04)

10.19.2.2.12.1 The formatter must be inhibited from putting DMA data onto the data bus to the 211 Interface Board when the Optional Data Port Interface is used.

10.19.2.2.12.2 This is physically accomplished by cutting the etch connecting the DMDATIL bus gating signal to Pin 1 of the bus control gate E2 as shown on the bottom left of Sheet 1 of Logic Drawing D1032-01, and then adding a jumper wire connecting E2-01 to E2-02.

10.19.2.2.12.3 (Reference Board 1 Assembly Drawing D1032-04) The etch should be carefully cut on the etch side of the printed circuit board at Pin 1 of E2. A jumper wire should then be added on the etch side of the board connecting E2-01 to E2-02.

10.19.2.2.13 Termination Removal When Autoswitch Board Used (Reference Board 1 Logic Drawing D1032-01, Sheet 1 and Assembly Drawing D1032-04)

10.19.2.2.13.1 When the Autoswitch Board is used in conjunction with the Optional Data Port Interface the 211 Interface Board is physically connected to both Board 1 of the formatter and the Computer Port Connector of the Autoswitch.

10.19.2.2.13.2 To facilitate checkout of all ports of the Autoswitch, the normal termination resistors shown for the J1 211 Interface connector Board 1, on Sheet 1 Logic Drawings are physically removed from Board 1.

10.19.2.2.13.3 In this configuration the Autoswitch provides the proper termination for the 211 Interface Signals.

10.20 Abort Inhibit on Cylinder Header Error Parameter Selection (Reference Board 1 Logic Drawing D1032-01, Sheet 2 and Assembly Drawing D1032-04)

10.20 The Phoenix 211 Disk Controller will normally abort any operation in process upon detection of any error condition.

10.20.1 The user may optionally prevent the disk controller from aborting any operation in process whenever a cylinder header error is detected by removing staple W33, as shown on right center of Sheet 2 of Board 1 Logic Drawing D1032-01.

10.20.2 (Reference Board 1 Assembly Drawing D1032-04) Staple W33 is a horizontally oriented staple permanently etched onto the etch side of the printed circuit board, immediately below and connected to Pin 5 of component C3.

10.20.3 Staple W33 should be removed by carefully cutting the run on the etch side of the printed circuit board connecting C3-05 to the feed thru pad beneath C3-07.

10.21 Abort Inhibit on Sector Header Error Parameter Selection (Reference Board 1 Logic Drawing D1032-01, Sheet 2, and Assembly Drawing D1032-04)

10.21.1 The user may optionally prevent the disk controller from aborting any operation in process whenever a Sector Header Error is detected.

10.21.1 This is accomplished by physically removing staple W20 as shown on the bottom center of Sheet 2 of Logic Drawing D1032-01.

10.21.3 (Reference Board 1 Assembly Drawing D1032-04) Staple W20 is a vertical staple on the component side of the printed circuit board connecting Pin 5 of component D3 to the feed thru located immediately below D3-05, and should be carefully cut.

10.22 Abort Inhibit on CRC Error Parameter Selection (Reference Board 1 Logic Drawing D1032-01, Sheet 2 and Assembly Drawing D1032-04)

10.22.1 The user may optionally prevent the disk controller from aborting any operation in process whenever a CRC Error is detected.

10.22.2 This is accomplished by removing staple W32 as shown on lower right center of Sheet 2 of Board 1 Logic Drawing D1032-01.

10.22.3 (Reference Board 1 Assembly Drawing D1032-04) Staple W32 is a permanently etched staple on the etch side of the printed circuit board connected to Pin 2 of C3 and running horizontally to the left of and below C3 to a point below and to the left of C3-01.

This staple should be removed by carfully cutting the run on the etch side of the printed circuit board.

10.23 Abort Inhibit On Sector Write Protect Error Parameter Selection (Reference Board 1 Logic Drawing D1032-01 Sheet 2 and Assembly Drawing D1032-04)

10.23.1 The user may optionally prevent the disk controller from aborting any operation in process whenever a Sector Write Protect Error is detected.

10.23.2 This is accomplished by physically removing Staple W31 as shown on the lower right center of Sheet 2 of Board 1 Logic Drawing D1032-01.

10.23.3 (Reference Board 1 Assembly Drawing D1032-04) Staple W31 is permanently etched onto the etch side of the printed circuit board and runs horizontally to connect feed thru pads located directly above C3-11 and C3-09.

The run should be carefully cut on the etch side of the printed circuit board.

10.24 Abort Inhibit on Data Late Error Parameter Selection (Reference Board 1 Logic Drawing D1032-01 Sheet 2, and Assembly Drawing D1032-04)

10.24.1 The user may optionally prevent the disk controller from aborting any operation in process whenever a Data Late Error is detected.

10.24.2 This is accomplished by physically removing staple W30 as shown on the lower right center of Sheet 2 of Board 1 Logic Drawing D1032-01.

10.24.3 (Reference Board 1 Assembly Drawing D1032-04) Staple W30 is permanently etched onto the etch side of the printed circuit board and runs horizontally to connect feed thru pads located directly above B3-14 and B3-12.

The run should be carefully cut on the etch side of the printed circuit board.

PHOENIX 211 9-SLOT SYSTEMS UNIT BOARD PLACEMENT CHART

SLOT NO.	A	B	C	D	E	F
1.	UNIBUS IN		SPC SLOT (QUAD)			
2.	SPC SLOT (HEX)					
3.	SPC SLOT (HEX)					
4.	SPC SLOT (PHOENIX 211 INTERFACE BOARD*)					
5.	UNIBUS OUT OR TERMINATOR		SPC SLOT (QUAD)			
6.	PHOENIX 211 BOARD NO. 1					
7.	PHOENIX 211 BOARD NO. 2					
8.	PHOENIX 211 BOARD NO. 3					
9.	PHOENIX 211 BOARD NO. 4					

PHOENIX 9-SLOT CHASSIS

LEFT SIDE VIEW

* NPR GRANT JUMPER IS REMOVED FROM THIS SLOT (CA1 → CB1)

CONNECTOR A

PIN	NAME	SOURCE BOARD-SHEET	DEST.	PIN	NAME	SOURCE BOARD-SHEET	DEST.
A1	NOT USED			A2	+5		
B1	"			B2	-15		
C1	"			C2	GND		
D1	"			D2	ACLO	BACKPLANE PC	4
E1	"			E2	NOSIPH	UREF 2-3	3
F1	"			F2	DRVONH	4-5	
H1	"			H2	RDESRL	1-1	1
J1	"			J2	NOT USED		
K1	"			K2	"		
L1	"			L2	"		
M1	"			M2	"		
N1	GOSEEKL	4-1	2	N2	-5	4-1	
P1	PORTREQL	3-2		P2	EOSH	3-3	1,3,4
R1	NOT USED			R2	PORTREL	3-2	4
S1	"			S2	WCKERR(Ø)	2-2	1
T1	GND			T2	NOT USED		
U1	NOT USED			U2	"		
V1	"			V2	"		

CONNECTOR B

PIN	NAME	SOURCE BOARD-SHEET	DEST.	PIN	NAME	SOURCE BOARD-SHEET	DEST.
A1	ROMTAG 3	3-3	4	A2	+5		
B1	HSHDRH	4-4	3	B2	-15		
C1	IHBSEEKL INBSEEKL	3-2	4	C2	GND		
D1	SPARE			D2	SEEKACKH	4-1	3
E1	SPARE			E2	GODLYH	1-2	2,3,4
F1	WORDL	3-4		F2	CLOCKH	1-2	2
H1	SET SEKH	3-4	2	H2	WCROVEØ	3-4	
J1	WCROVFH	(2-4) (1-1)		J2	FFORH	2-1	2
K1	SWPEP (1)	3-1	1	K2	CRCDATOUTH	3-1	2
L1	BAD SECT H	3-1	1	L2	CRC ERR(1)	3-1	1,3
M1	THEADERR(1	3-1	1	M2	SHEADERR(1)	3-1	1
N1	GENERRORL	1-2	3	N2	ENBDMRH	3-4	1,2
P1	SYSCLEARL	3-2	1	P2	DATAOUTL	2-2 4-2	3
R1	SEEKENDØH	4-2	2	R2	PUNSELØH	4-1	2,4
S1	SEEKEND1H	4-2	2	S2	PUNSEL1H		2,4
T1	GND			T2	NONSTRIPH	3-2	1
U1	WGH	3-3	2,3,4	U2	WRCHKH	3-2	
V1	RDGATEH	3-3	2,3,4	V2	SPARE		

CONNECTOR C

PIN	NAME	SOURCE BOARD-SHEET	DEST.	PIN	NAME	SOURCE BOARD-SHEET	DEST.
A1	ONCYH	4-1	1,3	A2	+5		
B1	BUSY (Ø)	3-4	1	B2	-15		
C1	MAXHEAD	3-5	1,4	C2	GND		
D1	DATAH	3-3	2,3	D2	SERVOMH	1-2	3,4
E1	WCOGH	3-3	2	E2	SERVOPH	1-2	3,4
F1	WSBH	3-3	2,3	F2	DSEH	1-2	4
H1	LASECCOMPH	4-3	2	H2	DSLH	1-2	4
J1	DONE(1)	3-4	1,3	J2	SECPLSI	4-4	4
K1	QUNABLE(1)	1-5	1,3	K2	DATLATE	2-4	1
L1	FLTCLRLL	3-2	4	L2	DMREQH	2-4	1,2
M1	RTZL	3-2	3,4	M2	QSEKINGL	2-3	1
N1	DMDATOH	1-1	2	N2	QSKDONEH	2-3	1
P1	RDDONE (1)	3-4	3	P2	SEKINTRL	2-3	1
R1	INTACKH	1-1		R2	LQSCAN2H	1-5	2
S1	CARRYH	3-4	2	S2	LQSCAN1H	1-5	2
T1	GND			T2	INCHDL	3-5	3,4
U1	SPARE			U2	LDUSHRL	1-1	3,4
V1	SEEKL	3-2	1,3,4	V2	DATAOUTH	2-2	4

CONNECTOR D

PIN	NAME	SOURCE BOARD-SHEET	DEST.	PIN	NAME	SOURCE BOARD-SHEET	DEST.
A1	RDCLKH	4-5	3	A2	+5		
B1	RDDATAH	4-5	3	B2	-15		
C1	WRTCK (H)	4-5	3	C2	GND		
D1	HDH	4-4	3	D2	GOINITL	1-2	1,2,3,4
E1	GOREL	3-2	3,4	E2	INITL	1-1	1,2,3,4
F1	READ DATAH	3-1	2,3	F2	COMØH	1-2	3
H1	READDATAH	4-5	4,3	H2	COM1H	1-2	3
J1	SECT1H	4-5	4,3	J2	COM2H	1-2	3
K1	SECT2H	4-5	4,3	K2	COM3H	1-2	3
L1	SEC3H	4-5	4,3	L2	RDCKH	3-1	2,3
M1	SECT4H	4-5	4,3	M2	UNSELØ(1)	3-5	2,3,4
N1	SECT5H	4-5	4,3	N2	UNSEL 1(1)	3-5	2,3,4
P1	SECT6H	4-5	4,3	P2	WRTCKH	3-1	2,3
R1	HFBDH	4-4	3,4	R2	DMACKH	1-1	2
S1	IOMHZH	3-1	1,2,3	S2	READ H	3-2	1,2,3
T1	GND			T2	WRITE H	3-2	1,2,3
U1	CKINTRL CLKINTL	1-3	1,3	U2	DISRCHNGH	2-3 3-4	4
V1	ENBHDRH	3-1	4	V2	DSKXFD(1)	3-4	1 (Data Port Opt.)

CONNECTOR E

PIN	NAME	SOURCE BOARD-SHEET	DEST.	PIN	NAME	SOURCE BOARD-SHEET	DEST.
A1	CONTCKH	3-3	3,4	A2	+5		
B1	SECADDØH	3-5	4	B2	-15		2
C1	SECADD1H	3-5	3,4	C2	GND		
D1	SECADD2H	3-5	3,4	D2	SEEKEND2H	4-2	2
E1	SECADD3H	3-5	3,4	E2	SEEKEND3H	4-2	2
F1	SECADD4H	3-5	3,4	F2	LASELØH	2-2	4
H1	SECADD5H	3-5	3,4	H2	LASEL1H	2-2	4
J1	SECADD6H	3-5	3,4	J2	ENBTAGL	3-5	1
K1	HEADØH	3-5	4	K2	SKINBL	1-2	4
L1	HEAD1H	3-5	4	L2	DONCYLH	4-1	1
M1	HEAD2H	3-5	4	M2	FLTH	4-1	1
N1	HEAD3H	3-5	4	N2	SKEERORH	4-1	1 2
P1	HEAD4H	3-5	4	P2	RDDSRL	1-1	4
R1	MAXCYLH	4-4	1,4	R2	DMDATIL	1-1	2
S1	LDCARL	1-1	4	S2	RDCARL	1-1	
T1	GND			T2	INCTARL	3-5	3
U1	MAXSECTH	3-5	1,3	U2	LOAD HDRL	3-1	4
V1	RDUSHRL	1-1		V2	LDWCRL	1-1	1,3

CONNECTOR F

PIN	NAME	SOURCE BOARD-SHEET	DEST.	PIN	NAME	SOURCE BOARD-SHEET	DEST.
A1	IDBØH	*	1	A2	+5		
B1	IDBIH	*	1	B2	-15		
C1	IDB2H	*	1	C2	GND		
D1	IDB3H	*	1	D2	IDB10H	*	1
E1	IDB4H	*	1	E2	IDB11H	*	1
F1	IDB5H	*	1	F2	IDB12H	*	1
H1	IDB6H	*	1	H2	IDB13H	*	1
J1	IDB7H	*	1	J2	IDB14H	*	1
K1	IDB8H	*	1	K2	IDB15H	*	1
L1	IDB9H	*	1	L2	DATA 15H	1-1	2, 3, 4
M1	DATA 14-H	1-1	1, 2, 3, 4	M2	DATA 7H	1-1	2, 3, 4
N1	DATA 13H	1-1	2, 3, 4	N2	DATA 6H	1-1	1, 2, 3, 4,
P1	DATA 12H	1-1	2,3,4	P2	DATA 5H	1-1	1, 2,3,4
R1	DATA 11H	1-1	1, 2,3,4	R2	DATA 4H	1-1	1,2,3,4
S1	DATA 10H	1-1	1,2,3,4	S2	DATA 3H	1-1	1,2,3,4
T1	GND			T2	DATA 2H	1-1	1,2,3,4
U1	DATA 9H	1-1	1,2,3,4	U2	DATA 1H	1-1	1,2,3,4
V1	DATA 8H	1-1	1,2,3,4	V2	DATA ØH	1-1	1,2,3,4

* SOURCES

Board	Sheet
1	1-5
2	1,4
3	5
4	1,4

XYLOGIC OEM COMPONENTS GROUP, INC.

42 Third Avenue

Burlington, Massachusetts 01803

617-272-8140

Phoenix 211 Disk Controller
Programming Reference Manual

Dwg. No. 1043-10

Revision F

Date March 30, 1977

TABLE OF CONTENTS

1.0 Introduction

The Phoenix 211 Disk Controller is a high performance, cost effective disk controller which will enable any PDP11 computer user to access up to 1.2 billion bytes of on-line storage.

The Phoenix 211 Disk Controller is a new generation disk controller employing microprocessor technology to enable improved performance at lower cost. The controller can be connected to up to four disk drives of the new "storage module" generation offered by Control Data Corp.,Calcomp and other manufacturers.

Two versions of the disk controller are currently available, the standard 211 model, and the Phoenix 215 Controller with Command Queue Capability. The standard Phoenix 211 Disk Controller is used under direct user program control, while the Command Queue equipped 215 unit is controlled by a additional microprocessor which automatically queues commands for up to four disk drives and performs all of the seek overlapping and data transfer initiation functions normally performed by a user operating system.

This manual is dedicated to defining the programming interface to the standard Phoenix 211 Disk Controller in terms of the capabilities provided and how they can be best utilized.

The programming interface to the Phoenix 211 Command Queue Disk Controller is covered in a separate supplement to this manual.

2.0 Overview

This section presents a overview of the controller programming interface and general capabilites.

2.1 Program Visible Registers (Loadable and Readable)

Register	Description	Standard Bus Address
DCSR	CONTROL AND STATUS	164000
DUSH	UNIT, SECTOR, HEAD	164002
DCAR	BUS ADDRESS REGISTER	164004
DWCNT	WORD COUNT	164006
DCYL	CYLINDER ADDRESS	164010
DSTAT	DISK STATUS REGISTER	164012
DERR	ERROR REGISTER	164014

Note: Base Bus Register Address is Strappable

2.2 Standard Interrupt Vector Address Assignment : 270_8-272_8 (Strappable)

2.3 Interrupt Priority: Level 5 (Strappable)

2.4 Number of 16 Bit Words Per Sector: 256 Standard - others available

2.5 Number of Sectors Per Track: 32 standard - others available

2.6 Disk Sector Addressing: Absolute Contiguous or optionally interlaced

2.7 Direct Memory Access Burst Duration ("Throttle") Control = 8 Strappable from 1 to 256 words

2.8 Number of Temporary Data Storage Buffers: Two

2.9 Temporary Data Storage Buffer Capacity: 128 words standard (Selectable to 512 in increments of 128)

2.10 Disk Unit Switching Time Latency: Zero*

*Separate sector counter is maintained for each disk drive

2.11 Disk Data Transfer Initiate Latency From Seek Done Interrupt: 1200 us Max. **

** Rotational position sensing is utilized. Seek done interrupt occurs when drive is on cylinder and 1200 μsec before selected sector. (Latency time is selectable)

2.12 Data Transfer Length: 1 to 65,556 words

Data transfer length controller by Word Count Register. Data Transfer may cross sector, surface and cylinder boundaries

2.13 Seek Control: Direct & implicit

All data transfer instructions contain an implicit seek command.

2.14 Functional Controller Commands

1. Seek
2. Read
3. Write
4. Format
5. Recalibrate
6. Fault Clear

7. System Clear
8. Read Header, Data, and CRC
9. Read Data, Ignore Position Verification Check
10. Write Header, Data, CRC
11. Write Over Write Protected Sector
12. Compare Disk Data, Memory Data
13. Port Release (Dual Port Disk Drives Only)

2.15 Simultaneous Seek Implementation

1. Supports true parallel simultaneous seek operations
2. Separate "seeking" and "seek done" status bits are provided for each drive.
3. Interrupt is generated at completion of seek only function.
4. Identity of disk drive terminating seek is presented to user as a word index.

2.16 Position Verification Technique

1. Uses separate header preamble for each sector.
2. Header contains sector, head, and cylinder positioning data
3. Header Words are followed by a header CRC value word
4. Header preamble automatically is checked 100% before any data transfer takes place on any accessed sector.

2.17 Disk Formatting Technique

1. Sector preamble header and header CRC words are generated once by Format program.
2. Once a disk pack has been formatted it is not normally formatted again for the life of the medium

2.18 Data Integrity Verification Technique

1. A CRC value is computed by controller during each write operation based upon actual data content written onto each sector
2. Computed CRC value is automatically appended to the end of each sector data field by controller
3. During read operations controller automatically recomputes the CRC value based on actual data read and compares resultant value with the reference value read from the disk
4. User may alternatively perform a actual data comparison with data written on disk and reference data contained in memory using a standard controller command.

2.19 Soft Error Recovery Technique

Recommended drive manufacturer's Retry Algorithem based on high resolution programmed offsets of both the head cylinder position and data strobe timing

2.20 Control Technique: Programmable special purpose microprocessor

2.21 Fault Isolation Capability

Can isolate fault to either disk drive or controller using self test mode to check integrity of the disk controller independently of the disk drive.

3.0 REGISTER ASSIGNMENTS -- SMD/PDP11 CONTROLLER

3.1 CONTROL AND STATUS REGISTER

3.1.1 FORMAT:

ERROR SUM.	SEEK IN-HIBIT	MEMORY EXTENSION		DSE	DSL	SV+	SV-	FOR-MATTER READY	INTER-RUPT ENABLE	Multi CPU Request	COMMAND				GO
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

3.1.2 FORMAT EXPLANATION:

BIT	NAME	FUNCTION
0	GO (R/W)	Starts the command specified by bits 1-4. Clears Formatter Ready. The command will not start until GO is set. This bit will always read as a logic Ø.
1-4	COMMAND (R/W) F_0 - F_3	Specifies what function is to be performed per the following: Detailed definitions of each command function performed are given in section 4.0.

F_3	F_2	F_1	F_0	COMMAND
0	0	0	0	SYSTEM CLEAR
0	0	0	1	SEEK ONLY
0	0	1	0	NORMAL READ
0	0	1	1	NORMAL WRITE
0	1	0	0	FORMAT
0	1	0	1	READ Header, Data, and CRC Word
0	1	1	0	WRITE Header, Data, and CRC Word
0	1	1	1	READ Data, Ignoring Header Check
1	0	0	0	WRITE OVER WRITE PROTECTED SECTOR
1	0	0	1	Drive FAULT Clear
1	0	1	0	RECALIBRATE (RTZ)
1	0	1	1	Write Check Disk Data
1	1	0	0	Port Release
1	1	0	1	Read No Strip (Special Option)
1	1	1	0	Port Request (Calcomp Disk Drives Only)
1	1	1	1	NOP

CONTROL AND STATUS REGISTER EXPLANATION CONTINUED

BIT	NAME	FUNCTION
5	MULTI CPU REQUEST (R/W)	This bit is used to request access to a 211 formatter equipped with the multiple cpu option. In 211's without this option this bit is always a zerø.
6	INTERRUPT ENABLE (R/W)	Allows Interrupts to occur if set to a '1'.
7	FORMATTER READY (READ ONLY)	Specifies that the Controller is ready to accept a new command.
8	SV- (R/W)	Servo Offset Minus. When this bit is true the drive head is offset from the normal on cylinder position away from the spindle (Read Error Recovery).
9	SV+ (R/W)	Servo Offset Plus. When this bit is true the drive head is offset from the normal on cylinder position towards the spindle. (Read Error Recovery).
10	DSL (R/W)	Data Strobe Late. When this bit is true the disk drive will strobe data at a time later than optimum. When false, normal strobe timing will be returned. (Error Recovery).
11	DSE (R/W)	Data Strobe Early. When this bit is true the disk drive will strobe data at a time earlier than optimum. When false, normal strobe timing will be returned. (Error Recovery).
12-13	MEMORY EXTENSION (R/W)	Upper extension bits of the Buss Address Register. Set under program control or incremented when an overflow of the Bus Address Register occurs.
14	SEEK INHIBIT (R/W)	Inhibits the implied seek during a Read or Write Command.
15	ERROR SUMMARY	'OR' condition of all error conditions in the Phoenix 211 SMD Controller System.

3.2 UNIT-SECTOR-HEAD REGISTER 164002 *

3.2.1 Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	UNIT		HEAD					SECTOR						

3.2.2 Format Explanation

BIT — **NAME** — **FUNCTION**

0-6 SECTOR O-SECTOR 6 (R/W)

1. Selects the initial sector to be written or read. (Standard format = 32 sectors/track for CDC Storage Module Disk Drives.
2. Sector field value is incremented by controller at the end of each sector during a data transfer operation.
3. When the last sector of a track is reached, sector counter overflows and causes incrementation of head field.
4. A separate sector counter is maintained for each disk drive connected to the controller to minimize disk latency associated with unit switching. There is no disk latency time associated with unit switching in the Phoenix 211 Disk Controller.

> Typically unit switching latency is ½ revolution to allow a single sector counter to be resynchronized with the index pulse of the newly selected disk drive.

7-11 HEAD (R/W)

1. Selects the initial Head (Surface) to be written or read.
2. Is incremented once for each overflow of the sector counter.
3. When the last surface on the current cylinder is processed, the head counter overflows and increments the Cylinder Register.
4. Incrementation of the Cylinder Register automatically initiates a seek to the cylinder specified by the new contents of the Cylinder Register.

* Typical - See Configuration Chart for Address used in your controller

12-13	UNIT NUMBER (R/W)	1. Selects the desired logical disk unit on which an operation is to be performed.
		2. Controller can interface to a maximum of four disk drives.
		3. Disk drive unit numbers can be assigned logically with the plug in elements supplied by the manufacturer with no re-cabling required.
14-15	NOT USED (READ ONLY)	1. Always Zeros.

3.3 BUSS ADDRESS REGISTER 164004

3.3.1 Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSS ADDRESS															

3.3.2 Format Explanation

BIT	NAME	FUNCTION
0-15	BAO-BA15 (READ/WRITE)	1. Loaded by the program to specify the starting memory address of a transfer. The BA register is incremented by two after each transfer of a word to or from memory.
		2. Overflow of Bus Address Register increments memory extension field (Bits 12 & 13) of Control and Status Register.

3.4 WORD COUNT REGISTER 164006

3.4.1 Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORD COUNT															

3.4.2 Format Explanation

BIT	NAME	FUNCTION
0-15	WCO-WC15 (READ/WRITE)	1. Set by the program to specify the number of words to be transferred in 2's complement form. The WC Register is incremented once for each word transferred to or from memory.

2. Overflow of Word Count Register terminates all word transfers to or from memory.

3.5 CYLINDER ADDRESS REGISTER 164010

3.5.1 Format

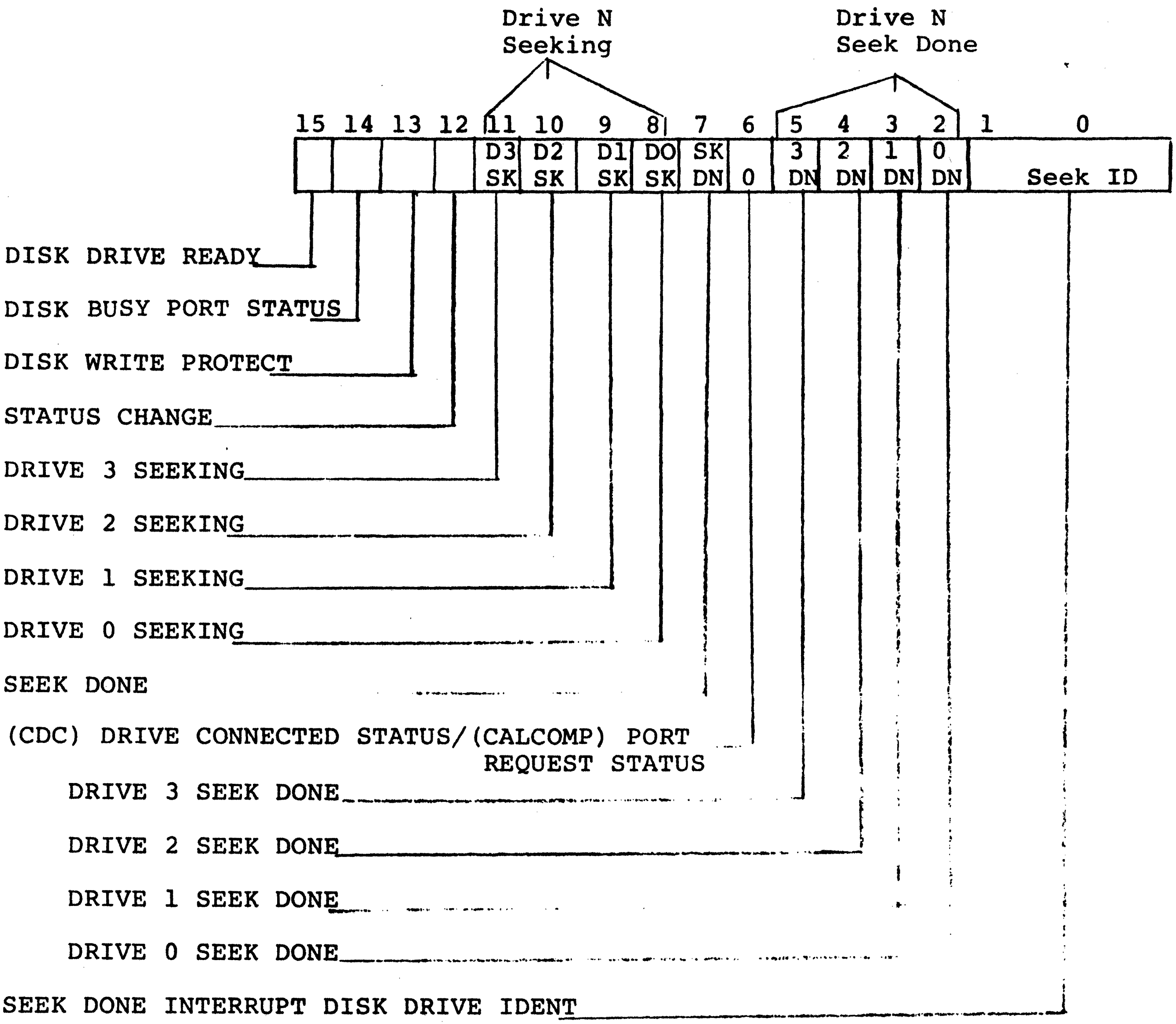
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CYLINDER ADDRESS									

3.5.2 Format Explanation

BIT	NAME	FUNCTION
0-9	CAO-CA15 (READ/WRITE)	1. Selected by the program to specify the initial cylinder on which a data transfer will take place.
		2. Is incremented by the overflow of the Head Counter.
		3. A seek is initiated at each incrementation to the cylinder specified by the new contents of the Cylinder Address Register.
		4. Incrementation of the Cylinder Address register beyond the last physical cylinder available on the disk will generate an Overrun Error and set bit 03 on the Error Register.
10-15	NOT USED (READ ONLY)	1. Always Zeros.

3.6 DISK STATUS REGISTER 164012* (Read Only)

3.6.1 Format



3.6.2 Format Explanation

BIT	NAME		FUNCTION/EXPLANATION
0-1	Seek Done Interrupt Drive Identification Field	1.	The contents of this field identifies the Logical disk drive generating a seek done interrupt. A seek done interrupt is generated whenever any one of the three seek done interrupt conditions occurs and the interrupt enable bit (06) of the Control and Status Register is set. The seek done interrupt conditions are defined in the Seek Done Interrupt Flag section.

*Typical - See Configuration Chart for Address of your controller

BIT	NAME	FUNCTION/EXPLANATION
		2. Whenever a seek done interrupt is generated bit 07 of the Disk Status Register will be set. Note: The Phoenix 211 Disk Controller utilizes Rotational Positioning Sensing (RPS) to maximize total system throughput. A Seek Done Interrupt is normally generated when the disk head is both on cylinder and on sector (2 sectors before specified target sector)
02-05	Seek Done Drive Status Bits	1. A seek done status bit is provided for each of the four disk drives serviced by the controller. 2. The seek done status bit for a given disk drive will be set whenever any one of the three seek interrupt conditions are detected by the controller for the associated disk drive. 3. When the Seek Done Status is set the corresponding Seeking Status bit for the associated disk drive is automatically reset by the controller. 4. The Seek Done status bits are mutually exclusive in that only one seek done status bit will ever be set any any one time. 5. The Seek Done status bit, once set will remain set until the associated disk drive is selected by loading the corresponding disk drive unit number into the Unit, Sector, Head Register and issuing any command to the disk controller. CAUTION: Every Seek Done Status Bit must be serviced by clearing the associated seek done status flag per the above procedure. No additional seek done status bits will be generated by the disk controller as long as any seek done status bit is set.

BIT	NAME	FUNCTION/EXPLANATION

6. Rotational Position Sensing
   The Phoenix 211 Disk Controller normally utilizes rotational position sensing to maximize total system throughput.

   In this technique both cylinder and sector head positions are combined to generate the "Seek Done" signal. A disk drive head must be both on the desired cylinder and over the "target" sector where the operation is to begin before the seek done signal and resulting interrupt are generated.

7. Sector Look-A-Head
   In order to provide the programmer with ample time to set up the disk controller with parameters for the data transfer to be performed after receipt of a Seek Done Interrupt, a programmable sector look-ahead feature is incorporated to generate the on sector signal one, two, or some other predetermined number of sectors before the actual physical sector on which the transfer is to be initiated. (Standard look-ahead factory setting is 2 sectors 1200 usec.)

8. Rotational Position Sensing Programming Considerations
   a. Rotational Position Sensing is intended for use in a multi drive system where disk latency is to be minimized by overlapping seeks under direct program control.

   b. The target sector must be specified at the time that each overlapped seek is to be initiated. This requirement is satisfied by loading the Unit, Sector, Head

BIT	NAME	FUNCTION/EXPLANATION
		Register with the required parameters. (Note that the unit # was required in any case.) c. An analysis of operating systems overhead and system interrupt priorities is necessary in order to determine the proper look-ahead time for a given system application. Rotational Position-Sensing cannot be used effectively in any system with variable unpredictable overhead interrupt response times. Note: The Command Queue Option automatically performs the functions of overlapping seeks and initiating data transfer operations on the basis of rotational position sensing in order to maximize system throughput while minimizing the disk driver software burden. This option should be seriously considered for users with high throughput objectives and unpredictable operating systems interrupt responses.
07	Seek Done Flag	The Seek Done Flag, when set indicates that: 1. A disk drive has completed a seek only or RTZ operation. 2. A disk drive has completed an unscheduled power down power up sequence. In this instance the programmer "pack change" status bit (12) of the Disk Status Register will also be set. 3. The port of the existing disk drive is now available for use. In this instance the port busy status bit (bit 14) of the Disk Status Register will be reset.

BIT NAME FUNCTION/EXPLANATION

Note that this condition is only associated with dual ported disk drives.

Note: Conditions Two and Three set bit 12, the Pack Change bit in the Disk Status Register.

Note that the Seek Done Interrupt flag may or may not be set when the controller completes a data transfer operation and generates a normal transfer operation complete interrupt.

When a disk drive completes a commanded seek only or RTZ operation:

1. The corresponding seek done drive status bit is set.
2. An interrupt is generated if the controller is not busy performing a data transfer operation.
3. The Seek Done Interrupt flag is set.
   If the controller is busy performing a data transfer operation when any seek operation completes:

   a. The corresponding seek done drive status bit is set.

   b. No seek done interrupt is generated.

   c. One interrupt is generated at the completion of the current data transfer in process.

   d. When this interrupt is generated the Seek Done Interrupt Flag is set indicating that a disk drive has completed a seek operation during the data transfer operation.

Disk Status Register Explanation Continued

BIT	NAME	FUNCTION/EXPLANATION
		The Seek Done Interrupt Flag once set, will remain set until a new command is issued to the disk controller.
08-11	Drive Seeking Status	One seeking status bit is provided for each of the four disk drives connected to the controller. This bit is set whenever the corresponding disk drive starts to execute a commanded seek only or RTZ operation. The bit is reset when the commanded seek only or RTZ operation has been completed and the associated disk drive seek done status bit has been set.
12	Disk Status (Pack?) Change	The disk Status Change bit, when set, indicates that a seek done condition has been detected on the selected drive which was <u>not</u> generated as a result of a commanded seek only or RTZ operation. The Status Change bit, once set, will remain set until a new command is issued to the selected drive. See the Seek Done Flag section for details on other Seek Done conditions.
13	Disk Write Protect Status	The Disk Write Protect Status Bit, when set, indicates that the selected disk drive is in a Write Protected State, as controlled by a manual switch on the front of some disk drives. If a drive has been so write-protected and an attempt is made to write on it, a Disk Fault Error will be generated and the write operation will be aborted by the controller.

BIT	NAME	FUNCTION/EXPLANATION
14	Dual Port Status	The Phoenix 211 Disk Controller supports the dual port option offer by some disk drive manufacturers. This bit is only applicable on systems configured with dual port disk drives. (Through this option two disk controller may "share" or access the same disk drive).

The meaning of this bit is different dependent upon which type of disk drive is connected to the controller.

CDC Storage Module Disk Drive Interpretation

1. The Disk Port Busy Status Bit, when set indicates that another controller is currently accessing the disk drive through the other port, and that the user will have to wait until the drive is available for use, at which time, the port busy status bit will be in the reset state.
2. The bit, once set, can be reset by the execution of a "Release Command" by the controller using the other port, or by the timeout feature provided by the disk drive manufacturer.

   When this bit is reset the selected disk drive is available for use by the controller.

BIT	NAME	FUNCTION/EXPLANATION
06	Disk Drive Connected/ Port Request Status Bit	Bit 06 of the Disk Status Register has different meaning depending upon which type of disk drive is physically connected to the disk controller.

CDC Storage Module Disk Drive Interpretation
Disk Drive Connected

If a CDC disk drive is used with the disk controller bit 06 will be set whenever the

Disk Status Register Explanation Continued

BIT	NAME	FUNCTION/EXPLANATION
		selected disk drive is physically connected to the disk controller and has power applied to it. Bit 06 will be reset if the selected drive is nonexistent or does not have power applied to it. This status bit is provided to enable user software to determine how many disk drives are connected to the disk controller. Calcomp Trident Disk Drive Interpretation Port Request Status If Calcomp dual port disk drives are used with the disk controller bit 06 is used to indicate the status of the port request signal issued by the disk controller. Bit 06 will be set if the Port Request Signal for the selected disk drive is currently active, and will be reset if it is not active. Notes: 1. This status signal is applicable only to dual ported Calcomp disk drives. 2. The Port Request signal for a given disk drive is set by selecting the disk drive and issuing a Port Request Command to the Calcomp Disk Drive. 3. If the controller interrupt is enabled a Port Available interrupt will be generated when the Disk Port Busy signal is reset after having been set, to facilitate systems utilization of this drive option.
15	Disk Drive Ready	The disk drive ready bit, when set indicates that the selected disk drive is up to speed, the heads are loaded, and no fault condition exists. This bit must be set before a Seek, Recalibrate, or Data Transfer Command is issued.

3.7 ERROR REGISTER 164014*(Read Only)

3.7.1 Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSS TIME OUT	WCK ERR	BAD SCT ID	DSK NOT RDY	FLT	SK ERR	TIM OUT	CRC ERR	DAT LTE	NXT SCT	NXT CYL	NXT HD	OVR	WRT PRT	HS ERR	CYL ADDR ERR HDR

BUS TIMEOUT ERROR

WRITE CHECK ERROR

BAD SECTOR FLAG DETECTED

SELECTED DRIVE NOT READY

SELECTED DRIVE FAULT

HARD SEEK ERROR

OPERATION TIME OUT ERROR

CYCLIC REDUNDANCY ERROR

DATA LATE ERROR

NON EXISTENT SECTOR PROGRAMMING ERROR

NON EXISTENT CYLINDER PROGRAMMING ERROR

NON EXISTENT HEAD PROGRAMMING ERROR

DISK OVERRUN ERROR

WRITE PROTECT ERROR

HEAD OR SECTOR ADDRESS COMPARISON ERROR

CYLINDER ADDRESS COMPARISON ERROR

* Typical - See Configuration Chart for Address used in your controller

3.7.2 Error Register Format Explanation

BIT	NAME	FUNCTION/EXPLANATION
0	Cylinder Address Comparison Error (Soft Seek Error)	Set whenever the commanded cylinder address and actual cylinder address read from the sector header word in sector being accessed do not agree during position verification check performed by controller prior to performing any normal data transfer operation. Also set whenever a difference is detected between the computed header CRC value read from the disk. If such a header CRC error is detected bit 8 of the Error Register will also be set. Once set, it is reset when the next functional command is issued to the disk controller.
01	Sector/Head Address Comparison Error (Soft Seek Error)	Set whenever the commanded head and sector address and actual head and sector address read from sector header in sector being accessed do not agree during position verification check performed by controller prior to performing any normal data transfer operation. Once set, it is reset when the next functional command is issued to the disk controller.
02	Write Protect Error	Set whenever an attempt is made to write on a sector in which the write protect header bit has been set using the normal write commands. In such an instance the write operation is aborted and the protected data is not over written. This error bit is <u>not</u> set if a write protected sector is rewritten using the Write Protect Override Command. Once set, the error bit will remain set until a new functional command is issued to the controller. Note: Writing on a write protected disk drive does not set this error bit.
03	Disk Overrun Error	The Disk Overrun Error bit will be set anytime an attempt is made to utilize a cylinder larger than the physical capacity of the disk drive itself. Once set, the error bit will remain set until a new functional command is issued to the controller.

BIT	NAME	FUNCTION/EXPLANATION
		The maximum number of cylinders allowed before the overrun error will be set is strappable to accomodate various disk capacities.
04	Non Existent Head Error	Set whenever an attempt is made to utilize a non existent head by loading the head field of the DUSH Register with a value beyond the range of the disk drives connect to the controller. The maximum number of allowable heads is strappable to accomodate disk drives of varying capacities. Once set, the error bit is cleared by issuing any functional command to the controller. The maximum number of allowable head is optionally physically programmed (Via ROM) for <u>each</u> physical cable port.
05	Non Existent Cylinder Error	Set whenever an attempt is made to utilize a non existent cylinder by loading the DCYL Register with a value beyond the physical capacity of the disk drives connected to the disk controller. The maxiumum number of allowable cylinders is strappable to facilitate disk drives of varying cylinder capacities. Once set, the error bit is cleared when a new functional command is issued to the controller. The maximum number of allowable cylinders is optionally physically programmed (Via ROM) for <u>each</u> physical cable port.

Error Register Explanation Continued

BIT	NAME	FUNCTION/EXPLANATION
06	Non Existent Sector Error	Set whenever an attempt is made to utilize a non existent sector by loading the sector field of the DUSH Register with a value beyond that of the capacity of the disk drives connected to the controller. The maximum allowable sector value is strappable to facilitate varying sector formats. Once set, the error bit may be cleared by issuing any functional command to the disk controller. The maximum number of allowable sectors is optionally physically programmed (Via ROM) for each cable port.
		NOTE: All nonexistent error conditions are extablished every time a disk controller command is executed by setting the "go" bit in the Control Status Register. All error conditions once set, remain set until a new command or Bus INIT is generated.
07	Data Late Error	This bit when set indicates that the effective direct memory access transfer rate between memory and the disk controller was less than that required to maintain proper synchronized data transfers with the disk drive. a. During a write operation this indicates that data transfers from memory to the controller FIFO memory buffer were not able to keep up with the rate at which data was taken from the FIFO memory to be written on the disk. The Data Late Error Bit was set when the FIFO memory was empty and another word was needed to write on the disk. At this point the operation was aborted.

Error Register Explanation Continued

BIT	NAME	FUNCTION/EXPLANATION

CAUTION: If a "data Late" error is detected during a write operation, the operation must be repeated to preclude generation of a unreadable sector.

b. During a read operation the presence of a Data Late Error Bit indicates that the data transfers from the controller FIFO memory to CPU memory was not able to keep up with the data transfers from the disk drive to the FIFO memory.

   The Data Late Error bit is set when the FIFO controller memory is full and there is no place to put the next word obtained from the disk drive, at which point the operation is aborted.

c. The frequent occurrence of Data Late Errors indicates that the controller DMA Burst Throttle is not properly set for the system environment in which the controller is being used.

   To eliminate Data Late Error occurrence the duration of the maximum DMA burst should be increased. The DMA burst duration is strappable from one to 64 words.

Error Register Explanation Continued

BIT	NAME	FUNCTION/EXPLANATION
		This error, once set, is cleared whenever a new command is issued to the controller.
08	Cyclic Redundancy Check Error	During every write operation a cyclic redundancy check (CRC) word is computed within the controller from the data being written on the disk. This CRC value is appended to the end of the data field at the end of each sector. During every read operation a new CRC value is computed from the actual data read, and compared with the original value generated during the write operation. The presence of the error indicates that these two computed values did not agree and that a data error occurred during the write or read operation. In any case the read operation should be repeated to verify the error as being "hard" per the error recovery algorithem. The CRC error bit is also set if a header CRC error was detected. In this case bit Ø of the Error Register will also be active. Once set, this bit may be cleared by issuing a new command to the controller.
09	Command Timeout Error	This bit is set whenever a commanded controller function exceeds the maximum time allowed by the controller for any commanded operation. (4 Seconds) The presence of this error indicates that the controller and/or disk drive hung up during the commanded operation. The error bit, once set, may be cleared by the issuance of any command to the controller.
10	Hard Seek Error	This bit is set whenever the selected disk drive is unable to complete a move within 500ms, or the carriage has moved to a position outside of the recording field, or that a cylinder address greater than the maximum physical number allowed has been issued to the disk drive. Once set this may be cleared only by commanding the selected drive to perform a RTZ operation.

BIT	NAME	FUNCTION/EXPLANATION
11	Drive Fault Error	This bit, when set indicates the presence of a fault condition in the selected disk drive rendering it unfit for use without operator remedial action. Please refer to the associated Disk Drive Manual. If the fault condition is temporary it may be cleared by the issuance of the FAULT CLEAR Command.
12	Drive Not Ready Error	This bit is set if an attempt was made to initiate a command on a disk drive that was not in the "Ready" state. This bit is reset when a new functional command is issued to the controller.
13	Bad Sector Detected	This error/status bit is set if an attempt is made to utilize a sector for data transfer operations which has the optional Sector Integrity Bit Set. In such a case the controller will optionally (strappable) automatically skip over the sector thus flagged as defective to the next available sector in which the Sector Integrity Bit is reset. The error bit is provided more as a status flag to alert the programmer than at least one intended sector did contain an active integrity bit, and that the controller did skip over at least one sector during the commanded operation. This bit is reset whenever a new command is issued to the disk controller.
14	Write Check Data Error	This bit is set during the Write Check Command execution whenever data read from computer memory does not agree exactly with data read from the disk. This bit is reset whenever a new command is issued to the controller.

Error Register Explanation Continued

15	Buss Timeout Error	Set when the controller is performing a DMA Transfer and the memory address specified in the Buss Address Register is nonexistent.

CAUTION: The disk controller will abort any operation when a bus timeout error is detected.

If a disk write operation was being performed it must be repeated to preclude the possibility of generating a "unreadable" sector.

4.0 PHOENIX 211 COMMAND DESCRIPTIONS

Note: All commands are initiated by a "GO" bit.

4.1 System Clear ($\emptyset\emptyset_8$)

When selected, this command does not cause the formatter to go busy. All formatter and interface flip flops are cleared to the reset state. This command is similar to a computer's I/Q reset command except it applies only to the formatter and interface logic. It does not clear a seek error or drive fault. No interrupt is generated by the execution of this command.

4.2 Seek Only (01_8)

When selected, this command causes the heads to be positioned to the cylinder specified by the Cylinder Address Register. The Formatter goes Not Ready while the disk drive accepts the cylinder address. Formatter Ready is set when the disk drive accepts the new cylinder address and starts seeking. Therefore, no other command may be given until the drive accepts the cylinder address, approximately 10.0 μsecs. If an error occurs while the drive is seeking, an interrupt is generated. Seek done is set when the seek completes, and an interrupt is generated. To determine which drive caused the interrupt, the disk status register bits 0-1 identify that drive which caused the interrupt and bit 7 will be set to signify that the seek has been completed. The program should check to see if an error occurred during a Seek Only command by selecting the drive and interrogating the error summary bit of the Control & Status Register. Once the disk drive accepts the cylinder address, the program is free to select another drive and perform an operation on it, i.e. Seek, Write, Read, etc. The Seek Inhibit Bit has no effect on this command.

4.3.0 Normal Read (02_8)

4.3.1 This command causes the controller to become busy for the duration of the operation.

4.3.2 A seek is initiated automatically (unless inhibited by bit 14 of the Control and Status Register , CSR) on the disk drive specified by the Unit Select Field of the Unit, Sector, Head Register (DUSH) to the Cylinder and Sector specified by the contents of the Cylinder Address Register (DCYL) and Unit, Sector Head Register.

4.3.3 At seek complete data is transferred from the accessed disk sector(s) to consecutive computer memory locations specified by the contents of the Bus Address Register (DBAR).

4.3.4 Once initiated the data transfer continues until the word count register overflows. The Word Count Register is initially loaded with the two's complement of the number of words to be tranferred.

4.3.5 Transfers from one to 65, 556 words may be transferred in one operation.

4.3.6 During the course of the transfer the controller will automatically increment across sector, head, and cylinder boundaries, performing automatically additional seeks as necessary to complete the specified transfer.

4.3.7 Read Positioning Verification

Prior to performing a read operation on any sector the header preamble is accessed to verify that the proper sector is being accessed. A 100% position verification technique verifies sector, head, and cylinder disk addressing information.

A cyclic redundancy check value is computed for the header data and is also recomputed and verified as part of the positioning verification procedure performed on every sector involved in a data transfer operation.

4.3.8 Data Field CRC Verification

During the data transfer for each sector a cyclic redundancy check value is computed by the controller based on the actual data read. This computed value is then compared with the reference value appended to the data field and based on the data when written. Any differences in the two CRC values will generate an error and cause the read operation to terminate.

4.3.9 Odd Length ReadData Transfers

If the number of words to be read, as defined by the Word Count Register, does not fall exactly on a sector boundary, direct memory access transfers to memory terminate when the Word Count Register Overflows. However, the controller will continue to read the remaining words in the sector in order to verify the integrity of the data by computing a CRC value for comparison with the reference value at the end of the sector data field.

4.3.10 The disk controller becomes ready when the Word Count Register has overflowed and the controller has completed the CRC data verification check on the last sector accessed, if no errors are encountered. If an error is encountered, the operation is terminated when the error is detected.

4.3.11 An interrupt is generated when the controller becomes ready if bit 06 of the Control & Status Register is set.

4.4.0 <u>Normal Write (03_8)</u>

4.4.1 This command causes the controller to go busy for the duration of the operation.

4.4.2 A seek is initiated automatically (unless inhibited by bit 14 of the Control and Status Register) on the disk drive specified by the Unit Select Field of the Unit, Sector, Head Register to the cylinder and sector specified by the contents of the Cylinder Address Register and Unit, Sector, Head Register.

4.4.3 At seek complete, data is transferred from consecutive memory locations specified by the Bus Address Register to consecutive disk sectors addressed by Unit, Sector, Head Register and Cylinder Address Register.

4.4.4 The duration or length of the data transfer in 16 bit words is defined by the Word Count Register, which is initially loaded with the two's complement of the number of words to be transferred. (From 1 to 65,556 words may be transferred in one operation).

4.4.5 The controller will automatically increment across sector, head, and cylinder boundaries, initiating seeks as appropriate to enable the total number of words specified to be automatically, successfully transferred to the disk.

4.4.6 <u>Write Positioning Verification</u>

Before any write operation is performed on any sector the header preamble of each sector is automatically read by the controller and verified for proper sector, head, and cylinder addressing information. Further, the header field itself is verified by a cyclic redundancy check (CRC) value that is computed by the controller and verified each time the header data is accessed.

4.4.7 <u>Data Field CRC Value Generation</u>

During the write operation a CRC value is computed by the controller for the data being written onto each sector. The computed CRC value is automatically written on to the disk at the end of the data field for use as a reference value by the controller during read operations.

The use of the CRC values during read and write insures the user that the integrity of all transferred data is preserved.

4.4.8 Odd Lot Write Transfers

If the number of words to be written, as defined by the Word Count Register, does not fall on an even sector boundary, direct memory access transfers from memory terminate when the Word Count Register Overflows. However, the controller will continue to repeatedly write the contents of the last two data words received from memory onto the disk until the end of the sector is reached. A checksum computed by the controller from the total sector data written is then written onto the end of the sector data field. This checksum is used to verify data integrity during all subsequent read operations.

4.4.9 Write Complete Interrupt

The controller will become ready when the Word Count Register overflows and the checksum has been written onto the last sector accessed, or when an error is detected.

When the controller becomes ready, an interrupt will be generated if bit 06 of the Control & Status Register is set.

4.5 Format Command (04_8)

This command is utilized to format the disk medium, and physically writes Sector, Head, and Cylinder position verification data as well as write protect and integrity status information onto the disk at the beginning of each disk sector as a header data preamble. This information is followed by a 16 bit CRC value unique to each sector which is generated and verified automatically by the controller.

Note: All "VIRGIN" disk packs must be "formatted" with this command before they can be read or written by using any of the data transfer commands.

This command specifically writes the absolute disk address of the current sector, in terms of sector, head, and cylinder, as well as write protect and sector information at the beginning of the sector being addressed. This positional information is then followed by a 16 bit CRC value derived from the two header words and written automatically by the controller. The above header information is contained in two 16 bit words of the format given below. When executed this command causes the Formatter to go Not Ready and starts an automatic seek. Upon completion of the seek normally a two word data transfer takes place which transfers the two header words containing the absolute disk address of the sector being addressed from memory to the header of the addressed sector. The Bus Address Register is initially loaded with the memory address of the first header word while the Unit, Sector, Head and Cylinder Registers are initialized to address the sector to be formatted. The two header words must be formatted as indicated below.

Note: 1.) It is up to the program to construct the header words correctly.

2.) A given disk pack is normally only formatted once to generate the proper header data preamble at the beginning of each sector. The positioning information contained in the header preamble is automatically accessed by the controller and used to verify that the proper sector(s) are being accessed prior to initiating any data transfer operation.

Failure of the accessed actual header position information to agree with the commanded disk position or address will cause the operation to abort and any one or more of the following "soft seek" error to be generated.

4.5 Format Command (04_8) Continued

- Cylinder Address Comparison Error
- Sector/Head Address Comparison Error
- Header CRC Error

4.5.1.1 Required Format For Header Word

Standard:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Ø
Ø	Ø	Ø	Ø	Head #					Sector #						

Optional:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Ø
Ø	Ø	Head #					Sector #							a	a

Bad Sector Flag Field

4.5.1.2 Header Word 1 Format Explanation

Sector Field

Should contain sector address of the current sector being formatted. Information should be the same as the sector field of the Unit, Sector, Head Register.

Head Field

Should contain head address of the current sector being formatted. Information should be the same as the head field of the Unit, Sector, Head Register.

4.5.1.3 Required Format For Header Word 2

Note: Header word two must be stored in the next consecutive memory location following header word 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a	a	b	b	CYLINDER ADDRESS											

a/a: BAD SECTOR FLAG FIELD

b/b: SECTOR WRITE PROTECT FIELD

4.5.1.4 Header Word 2 Format Explanation

4.5.1.4.1 Cylinder Address Field

Should contain the cylinder address of the current sector being formatted. Information should be the same as contained in the Cylinder Address Register.

4.5.1.4.2 Sector Write Protect Field

The Phoenix 200 Disk Formatter provides the user with the capability of selectively write protecting data on the disk storage medium to the individual sector level by means of the Write Protect bits in header word 2.

If the write protect bits are set by the user the sector will be write protected. Any subsequent attempt to write on the sector will cause the controller to abort the operation and generate a Write Protect Error.

4.5.1.4.3 Bad Sector Flag Field (Automatic Sector Skip Flags)

The Bad Sector Flag Field is used to identify known defective sectors to be controller.

Normally, the Bad Sector Flag Field bits should be reset. If the bits are set by the user, indicating that the corresponding sector is "defective" the controller will automatically skip over the sector during any disk data transfer and resume operation with the next available sector not flagged as being "defective".

4.5.1.4.4 Bad Sector Flag Utilization

The Bad Sector Flag is intended to enable multiple sector transfers to be conducted on a disk medium with known defective sectors. Such known defective sectors would normally be found and flagged by an Initialization Program which attempted to write and read "1" and "0" data into every bit cell of every sector on the disk. Once the defective sectors had been catalogued and flagged, multiple sector transfers could be conducted "around" a defective sector using the bad sector flag and skip feature without having to make the transfer operations begin and/or end on defective sector boundaries.

4.5.2 Manual Format Enable Switch

Since this command alters the head position verfication data or "header" stored on the disk, the "Format" switch on the optional Formatter Control Panel or closure must be manually enabled when this command is executed. This insures that the Format command will not be erroneously executed and result in any loss of data.

Warning: It is not possible to use this command to reformat the header field of a given sector without affecting the data field as well.

If a sector header is to be reformatted and the sector contains meaningful data, the data should be saved before the sector header is reformatted.

Use of this command destroys the data field content of any sector.

4.5.3 Data Field Write Option

A complete sector data field or portion thereof may be written at the same time that the sector is being formatted by appropriately increasing the Word Count Register value and following the header words in memory with the data to be written.

Note: The controller automatically generates a valid CRC value for any data written into the sector field during execution of the Format Command.

4.5.4 Multiple Sector Format Option

Multiple sectors may be formatted by the Format Command by simply increasing the Word Count value and providing the proper data buffer format.

If a multiple sector formatting technique is to be used, the user should remember that a total of two header words plus the appropriate number of data field words is required per sector.

4.5.5 Format Done Interrupt

The disk controller will become Not Busy when the Word Count Register has overflowed and the checksum value has been written at the end of the data field of the last sector accessed. When the Controller becomes ready an interrupt will be generated if bit 06 of the Control & Status Register is set.

4.6 Read Header, Header CRC, Data Field, and Data CRC Command (05_8)

4.6.1 Function

The Read Header, Header CRC, Data Field, and Data CRC Command is provided to enable the user to:

1. Read the header field (including the header CRC) of any sector for diagnostic or maintenance verification pruposes.
2. Read the data and Data CRC values for any sector for diagnostic or maintenance verification pruposes.
3. Recover data from any sector by overriding soft-seek errors encountered during execution of the normal read command.

4.6.2 Operation

4.6.2.1 This command causes the controller to go Not Ready and starts an anutomatic seek to the specified cylinder.if not inhibited by bit 14 of the Control & Status Register.

4.6.2.2 Upon completion of the seek operation, data is transferred from the addressed sector (starting with header data) to the specified memory locations until the Word Count Register Overflows.

4.6.2.3 The command will transfer in sequential order, the two header words, header CRC value, data field contents, and data CRC value if the word count value permits.

4.6.2.4 This command may be utilized to read header, data, and CRC fields for more than one sector at a time by increasing the word count value accordingly.

If the user chooses to read more than one sector at a time he should remember that the total word count per sector is equal to the data field word count +4.

4.6.2.5 During the execution of this command all softseek and CRC error detection logic is disabled to facilitate data recovery.

4.6.2.6 An interrupt is generated when the operation completes if bit 06 of the Control & Status Register is set.

4.7 Write Header, Data Field and CRC Command (06_8)

4.7.1 Function

The Write Header, Data Field and CRC Command is provided to enable the user to readily rewrite the header, data, and CRC fields of any sector for diagnostic and/or maintenance verification purposes.

4.7.2 Operation

4.7.2.1 This command causes the controller to go busy and starts an automatic seek to the specified cylinder if not inhibited by bit 14 of the Control & Status Register.

4.7.2.2 Upon completion of the seek operation, data is transferred from specified memory locations, sequentially to the header, header CRC, data field, and data CRC fields of the addressed sector until the Word Count Register Overflows.

4.7.2.3 No position verification takes place prior to initiation of the data transfer since the header position verification data itself is being written.

4.7.2.4 The first two words transferred are the header position verification words and should be formatted by the user per section 4.5.

4.7.2.5 The third word transferred is the header CRC value. This standard CRC value should be derived from the actual two header words if a valid value is to be generated.

4.7.2.6 If a full sector plus data CRC value is to be written, the CRC value, to be valid, must be derived from the actual data field contents.

4.7.2.7 More than one sector can be written by increasing the word count value as desired.

Caution:

This command should be used with great care and discretion since it affects the formatting of the disk medium itself.

Warning:

This command cannot be used with a word count of three to reformat only the header portion of a sector without affecting the data field. If the header field of a sector is to be rewritten the contents of the data field must be first recovered and saved. Reformatting the header without first saving the data will result in the loss of all data in the sector data field.

4.7.2.8 The disk controller becomes ready when the Word Count Register overflows, and will generate an interrupt at that time if bit 06 of the Control & Status Register is set.

4.8 Read Data with No Position Verification Command (07_8)

4.8.1 Function

The Read Data and CRC Command facilitates data recovery by overriding soft seek position verification prior to performing a read operation. It is identical to the Normal Read Command except that the header is not checked prior to reading the data field. (A CRC check is made on the data field contents only.)

4.8.2 Operation

4.8.2.1 This command causes the controller to go busy and starts an automatic seek to the specified cylinder if not inhibited by bit 14 of the Control & Status Register.

4.8.2.2 Upon completion of the seek operation data is transferred from the addressed sector to the specified memory locations until the Word Count Register overflows.

4.8.2.3 A CRC Check is made of all data read.

4.8.2.4 All controller position verification and position CRC error detection logic is inhibited during the execution of this command.

4.8.2.5 More than one sector may be read using this command by increasing the word count value as desired. If a CRC error is detected, the controller will abort the operation and set the CRC Error bit in the Control & Status Register.

4.8.2.6 When the operation completes an interrupt will be generated if bit 06 of the Control & Status Register is set.

4.9 Write Over Write Protected Sector (10_8)

4.9.1 Functions

This command enables the user to write on a sector which has active Write Protect Field bits in Header Word 2 if the manual Write Protect Override Closure is also enabled.

4.9.2 Operation

4.9.2.1 This command is identical in operation to the normal Write Command, except that it will not abort when a sector with active write protect bits in the header is detected, if the Write Protect Override closure is enabled.

4.9.2.2 If this command is executed and the Write Protect Override Closure is not enabled, the command will abort and generate a Write Protect Error when a sector with active Write Protect header bits is detected.

4.10 Drive Fault Clear Command (11_8)

4.10.1 Function

This command is used to clear the Disk Fault Signal and Condition in the selected disk drive.

Note: This command is the only means by which a Disk Fault can be cleared under program control.

4.10.2 Operation

4.10.2.1 This command executes immediately upon activation and therefore does not cause the controller to go busy.

4.10.2.2 An interrupt is not generated upon completion of this command, since it executes immediately.

4.11 Recalibrate Disk Drive (RTZ) Command (12_8)

4.11.1 Function

This command is utilized to clear the selected disk drive electronics of all erroneous conditions except "fault", and to cause heads of the selected disk drive to return to the cylinder Ø position.

Note: This command must be issued to any disk drive which incurs a "hard seek" error before any other operational command can be issued to it.

4.11.2 Operation

4.11.2.1 When this command is executed the controller does not become busy.

4.11.2.2 The selected drive "seeking" bit in the Disk Status Register is set at the beginning of the operation.

4.11.2.3 When the operation is complete and the selected disk head is at cylinder Ø, the selected drive "seek done" status bit will be set, the "seeking" status bit will be reset, and the seek done summary bit in the Control & Status Register will be set. A "Seek Done" Interrupt will be generated if the interrupt enable bit (06) of the Control & Status Register is set.

4.12 Write Check Disk Data Command (13_8)

4.12.1 Function

This command enables the user to automatically compare data in memory with data contained on the disk, and thus perform an additional check on data written on the disk.

4.12.2 Operation

4.12.2.1 This command causes the controller to become busy for the duration of the operation.

4.12.2.2 A seek is automatically started to the specified cylinder unless inhibited by bit 05 of the Control & Status Register.

4.12.2.3 At seek completion data is transferred from specified memory locations and the addressed disk sectors to the controller.

4.12.2.4 The controller automatically performs a bit by bit comparison of the memory and disk data until the Word Count Register overflows.

4.12.2.5 If the memory and disk data do not agree a Write Check Error will be generated and the operation will be aborted.

4.12.2.6 When the operation terminates an interrupt will be generated if bit 06 of the Control & Status Register is set.

4.13 Release Command (14_8) (Dual Port Option Disk Drives Only)

4.13.1 Function

4.13.1.2 This command releases the selected disk drive for use by another controller connected to the other port. Execution of this command specifically causes the other disk drive port to be granted a not busy port status.

4.13.2 Operation

4.13.2.1 This command is executed immediately upon activation and does not cause the controller to become Not ready.

4.13.2.2 No interrupt is generated at the end of the operation since execution and termination are immediate.

Differentiate between CDC and Calcomp.

4.14 Port Request Command

This command causes the Port Request Signal of the selected disk drive to be set.

5.0 MARGINAL DATA RECOVERY

Bits 8-11 of the Command and Status Register allow control of the servo positioner and the read recovery circuits in the disk drive. These controls are used to recover marginal data from the disk drive. Allowable combinations during Read command are:

BIT 11	BIT 10	BIT 9	BIT 8	FUNCTION	
0	0	0	0	Settings are optimum--this is the value normally used.	
0	0	0	1	Data strobe normal	Servo minus
0	0	1	0	Data strobe normal	Servo plus
0	1	0	0	Data strobe late	Servo normal
1	0	0	0	Data strobe early	Servo normal
0	1	0	1	Data strobe late	Servo minus
0	1	1	0	Data strobe late	Servo plus
1	0	0	1	Data strobe early	Servo minus
1	0	1	0	Data strobe early	Servo plus

Certain data errors are considered recoverable, and a sequence of error recovery procedures is outlined below for these. These data errors are:

Header Compare Error

CRC Error

Data Late Error

For these errors the recommended procedure is:

1. Three in-place retrys of the operation in error; then if no recovery occurs,
2. Three in-place retrys of the operation in error for each allowable marginal recovery state;
3. Perform a Recalibrate;
4. Repeat step (1);
5. Repeat step (2).

6.0 PROGRAMMING CONSIDERATIONS

6.1 Interrupts are generated upon completion of a command with the exception of Fault Clear and System Clear.

6.2 The program must set up the register parameters before 'GO' is issued. Normally the last register to be loaded will be the Command and Status Register which contains the 'GO' bit.

6.3 To perform overlapping seeks, the program must select the disk drive (as in all cases) and perform a seek only command on that drive. The Formatter goes Not Ready while the disk drive accepts the cylinder address. Formatter ready is set to signify that the drive has accepted the cylinder address. At this time, the appropriate seeking bit is set, identifying the Seeking Drive. After this, the program is free to issue any other Drive,a Seek, Recalibrate, Read, or Write command provided a seek error did not occur.

6.4 If a seek error occurs, the program must issue a Recalibrate to that disk drive which caused the seek error.

6.5 Upon Interrupt, the interrupt service routine should always check for possible error conditions.

6.6 Seek Done or Seek Error interrupts are inhibited while the Formatter is currently busy performing a data transfer.

6.7 The Formatter will abort, i.e. Formatter Ready will set, whenever an error condition is detected.

6.8 An automatic seek occurs for a read or write instruction. Therfore, the program need only give one command to cause the drive to seek and do a data transfer. The program should expect only one interrupt for a read or write command.

6.9 The Fault Clear command is used in lieu of operator intervention on the disk drive control panel.

6.10 Crossing of head and cylinder boundaries occur the following way:

When the maximum sector of a track is reached, the Formatter will increment to the next logical head and clear the Sector Address. If the maximum sector is reached on the maximum head, the Formatter will increment the cylinder address to the next logical cylinder and Reset the Head and Sector Address.

The Formatter will continue to process data in this manner until the Word Count Register overflows. Overrun Error is flagged when the formatter detects that a read or write operation has finished on the maximum sector and head of the maximum cylinder and the Word Count Register had not yet overflowed.

6.11 Seek Done is defined as on specified cylinder and two sectors before the desired sector.

6.12 The disk drive must be reselected prior to the issue of any command by loading the Unit, Head, Sector Register.

6.13 Calcomp Trident Only. To clear any condition which caused a device check (fault) both a fault clear and a RTZ command must be issued.

6.14 Servo Offset

a. Must set seek inhibit.
b. Error recovery must be limited to single Sector Transfers, or less.

6.15 Calcomp Dual Port

6.16 CDC Dual Port Operation

6.17 Power Fail

On ACLO, Failsafe sequence drive down.

6.18 Calcomp Trident Only

After the completion or issuing of any command to the 211 controller, software must not access the Controller for a minimum period of 12 μs (micro-seconds). This delay allows the Controller to clear any associated interrupts (attention) pending on the selected disk drive.

APPENDIX A

PHOENIX 211 DISK CONTROLLER

One Sector Read/Write Hand Loop

	LOCATION	CONTENTS	INSTRUCTION	COMMENT
	1000	12737	MOV #UNIT, DUSH	; Select Disk Drive
	1002	Ø		
	1004	164002		
LOOP:	1006	12700	MOV # DSTAT, RØ	; Get Disk Status Register Address
	1010	164012		
	1012	5710	TST @ RØ	; Is disk drive ready?
	1014	100401	BMI .+2	; Continue if yes
	1016	Ø	HALT	; Halt, Disk Drive Not Ready
	1020	12740	MOV # CYL, -(RØ)	; Specify cylinder address
	1022	CYL		; Cylinder Value Variable
	1024	12740	MOV #-256.,-(RØ)	; Specify one sector word count
	1026	177400		
	1030	12740	MOV #ADR,-(RØ)	; Specify origin memory address
	1032	2000		; 2000 is arbitrary
	1034	12740	MOV #TUSH,-(RØ)	; Specify disk, sector,head addresses
	1036	Ø		; Ø is arbitrary
	1040	105740	TSTB -(RØ)	; Disk controller ready?
	1042	100401	BMI .+2	; Continue if yes
	1044	Ø	HALT	; Stop if not
	1046	12740	MOV #CMP,-(RØ)	; Specify and initiate operation
	1057	7-Write 5-Read		;
	1060	105710	TSTB (RØ)	; Controller done yet?
	1062	100376	BPL.-2	; Wait till controller done
	1064	5710	TST (RØ)	; Any errors?
	1066	100001	BPL .+4	; Continue if not
	1070	Ø	HALT	; Stop on error
	1070	Ø	HALT	; Normal Stop
	1070	137	JMP LOOP	
	1072	1006.		

XYLOGIC OEM COMPONENTS GROUP, INC.

42 Third Avenue

Burlington, Massachusetts 01803

617-272-8140

Phoenix 211 Disk Controller

Diagnostic Manual

Dwg. No. 1043-05/1043-06

Revision C

Date January 3, 1978

TABLE OF CONTENTS

PHOENIX 211 DISK CONTROLLER DIAGNOSTIC

1.0 ABSTRACT

The Phoenix 211 Disk Controller Diagnostic is a series of tests that verify the operation of the Phoenix 211 Disk Controller and attached disk drives.

These tests check all Controller operations with associated disk drive (s) and host PDP-11 computer.

The Phoenix 211 Diagnostic Program and associated manual are the cheif tools provided by Xylogics to facilitate maintenance fault isolation to the board level.

The disk diagnostic is organized to exercise in each test a small, finite functional block of logic. The functional block of logic being exercised by each test or subtest is identified to the user in the diagnostic manual along with the board (s) where the logic is physically located.

In the event that the logic being exercised is contained on more than one board, the boards are listed in order of most probable failure location.

It is the intent that the disk diagnostic program and associated manual be utilized by both the user and/or third party maintenance groups to both verify the integrity of any Phoenix 211 Disk Subsystem as well as to readily isolate any failure to the board level.

2.0 REQUIREMENTS

2.1 PDP11 Minicomputer with 8K core

2.2 PHOENIX 211 Disk Controller

2.3 CDC 9760, 9762, 9764, 9766 SMD Disk Drive or CALCOMP TRIDENT T-25, T-50, T-80, T-200 Disk Drive

3.0 LOADING PROCEDURE

3.1 Verify that the Boot Loader is in memory

3.2 Set the Switch Register = #750

MEMORY SIZE		#
4K	=	17
8K	=	37
12K	=	57
16K	=	77
20K	=	117
24K	=	137
28K	=	157

3.3 Depress Load Address

3.4 Depress START

4.0 OPERATING INSTRUCTIONS

NOTE 1: This diagnostic must be run on a formatted disk pack and will destroy all data on the pack.

If it is desired to format a pack, use the formatting program included in the diagnostic (Tests 45,47,50)

NOTE 2: For CPU's without a switch register see Section 4.4.

4.1.1 Set Program Counter (PC) to program "START" address (See Diagnostic Listing)

4.1.2 Enter starting test number in the most significant byte of the console switch register.

4.1.3 Enter desired end test number in the least significant byte of the console switch register. The first test number is zero.

4.1.4 Set the most significant bit (Bit 15) of switch register if the tests are to be repeated. Leave reset if the tests are to be executed only once.

4.1.5 Depress "START" switch on computer console. The computer will start and then halt.

4.1.6 Enter starting disk drive number into the most significant byte of console switch register.

4.1.7 Enter number of the last disk drive to be tested in the least significant byte of console switch register. Note that the maximum drive number = 3 (0 through 3).

4.1.8 Depress the CONTINUE switch on the computer console.

4.1.9 The computer will then execute all tests specified on all disks specified. It will also stop if any error is encountered unless the following switch settings are used.

4.2 CONSOLE SWITCH SETTINGS (ERROR MODES)

BIT 15	BIT 14	BIT 7	SETTING REFERENCE
0	0	0	SSA
0	0	1	SSB
0	1	0	SSC
0	1	1	SSD
1	0	0	SSE
1	0	1	SSF
1	1	0	SSG
1	1	1	SSH

4.2.1 SSA

All three console bits "OFF" will allow the computer to halt upon encountering any type of error (most commonly used mode).

4.2.2 SSB

Switch Register Bit 7 set alone applies only to Data Compare errors and allows for data reliability checking. Each time a Data Compare error is flagged, location "ERRCNT" is incremented and no retry occurs. The Data Compare routine "DATCMP" then continues on with the next consecutive data word.

4.2.3 SSC

This mode applies only to soft controller errors outlined below:

SOFT CONTROLLER ERRORS

DATA LATE

HEADER COMPARE (SOFT SEEK)

DATA CRC

HEADER CRC

Console Bit 14 enables the controller to retry the command previously issued that caused one of the above error conditions. When used alone Bit 14 will allow the PHOENIX 211 up to four retries before the error is logged as a "hard" error and the general error flag "ERRFLG" is set.

4.2.4 SSD

Designed primarily for use as data error logging exercise in conjunction with TEST #32 (Complete Disk Read), this mode combines both Soft Controller Error Retry and Data Compare Error Retry. Explanation of each error counter and flag is given below.

LOCATION	LABEL	EXPLANATION
	SOFERR -	This is the total number of soft controller errors encountered (Bit 14).
	ERRFLG -	This flag is set each time a "hard" error is detected, and causes a program halt unless Bit 15 is set.
	ERRCNT -	This is the total number of data compare errors (Bit 7).
	HERCNT -	A counter for hard data errors; this location will be incremented each time a data word fails after eight attempts to reread that particular sector (Bits 7, 14, TEST 32).
	ERROR -	If Bit 15 is set and an error is flagged (ERRFLG $\neq$ 0), this location is incremented (Bits 7, 14, 15).
	SKIPER -	Hard controller error counter that is incremented in the Retry mode when four attempts to reread a sector have resulted in an error condition (Bit 14, TEST 32 only). Test 32 will then skip over that sector and continue on.

4.2.5 SSE

This mode inhibits all errors from causing a Program Halt.

When an error is detected (ERRFLG $\neq$ 0) in an exercise sequence of more than one test, control is transferred back to the program executive and testing continues with the next consecutive test. Location "ERROR" is incremented each time a failure is detected.

4.2.6 SSF

Data Compare Logging is incorporated along with Program Halt upon detection of any Controller error (errors flagged by the DERR register).

4.2.7 SSG

Data Compare Errors are considered "hard" errors and would be captured in location "ERROR". "Soft" controller errors would be retried without a Program Halt. (Bits 14, 15).

4.2.8 SSH

Data Errors would be logged and retried, "soft" Controller errors would be logged and retried, "hard" Controller errors would be logged and the sequence (TEST 32) would be restarted (all error counters would be reset at RESTART).

4.3 RECOMMENDED TEST SEQUENCES

4.3.1 Logic

For a complete test of the PHOENIX 211 and attached Disk Drive(s), it is recommended that a complete "pass" of the diagnostic is used. This sequence is defined as Tests 0 - 32 with all Error Inhibit and Retry switches OFF.

4.3.2 Reliability

Reliability testing is accomplished using Test 32 (Entire Disk Read) in the Repeat mode. Since Test 32 uses only the Read command, Test 31 (Entire Disk Write and Read) must be run prior to Test 32 .

4.4 ALTERNATE OPERATING MODES

4.4.1 NON SWITCH REGISTER MODE

This mode is used when a switch register is not present on the user's PDP11. To enter this mode deposit a non zero number into location "NONCON" (LOCATION 1266). At this point test parameters should be entered by depositing the selected sequence into memory starting at label "STPTN" (ending test number). (See diagnostic listing). The default setting is test sequence Ø thru 3Ø on drive Ø.

At the end of a test sequence or when an error is detected the program counter (PC) will be changed to 173000_8, activating the M9301 rom. Before the console is made live, the CPU registers, RØ R6, are saved in memory starting at location "REGØ." The general error flag "ERRFLG" is loaded into register RØ and the current test number is loaded into CPU register R4 just prior to activating the Emulator giving a quick detection of passage or failure.

4.4.2 ERROR CORRECTION CONTROLLERS (ECC)

Phoenix 211 Disk Controllers equipped with error correction are tested by depositing a non-zero number into location "ECC" (1074_8). Modifying this location enables the test executive to use an alternate test origin table, containing ECC logic tests.

4.4.3 AUTO-SWITCH OPTION

Controllers equipped with the auto-switch option can be tested the normal test procedure if the alternate DATA PATH, (AUTO SWITCH BIT # 7 OFF), is not enabled.

To test the alternate data path the operator must do two things:

1. Deposit a non-zero number in location "AUTOSW" ($231Ø_8$)
2. Mov the Interface Data Cable from J1 conn. to J2 conn. on the auto switch board (See Auto Switch Manual).

With the autoswitch mode enabled the program will set bit #7 on the autoswitch board before a transfer (DMA).

4.4.4 MULTIPLE COMPUTER PORT OPTION

4.4.4.1 SINGLE CPU

To run the controller with a single CPU, deposit a non-zero number into location "MULCPU" and use normal diagnostic procedures. Setting the "MULCPU" flag forces the program to request the controller each function to be performed. Because the diagnostic, in some tests, issues I/O resets and system clear commands, multiple CPU's cannot be run simultaneously.

4.4.4.2 SIMULTANEOUS CPU EXERCISE

To check possible conflict among two or more CPU's requesting a single Phoenix 211 formatter, it becomes necessary to run a special exercise. Test 33_8 has been modified to release the Phoenix controller after completing a write/read sequence. Normally during the Diagnostic Testing, all tests except 33, the formatter is not released. Operation is as follows:

1. Deposit a non-zero number in location "MULCPU" of each CPU under test.
2. Use test 33 as starting and ending test number.
3. Continue normal operating instructions.

4.4.5 NO TEST INPUT MODE

This mode is the same as non-console mode (section 4.4.1) except that it halts at normal halt locations instead of activating the console emulator. This mode is useful when:

1. CPU type is PDP11/04 or PDP11/34 and has programmers console (byte loading of test parameters) or,
2. The same test sequence is repeated often.

OPERATION: Deposit a non-zero number in location "NINPUT " Start at normal start location

5.0 GENERAL DIAGNOSTIC STRUCTURE

5.1 The PHOENIX 211 Disk Controller Diagnostic uses small subtests that are linked together through the test executive routine. The subtests use common subroutines to execute all data transfers. Ending Controller parameters are checked following each controller operation and Interrupt mode is employed in all exercises.

5.2 COMPUTER ERROR HALT

If the computer halts for an error these registers will define the condition:

RO = 2 x the disk drive number

R1 = test number

R2 = reference data

R3 = actual data received

R4 = subtest number

R5 = address where error was detected

NOTE: In some cases Registers R2 and R3 do not necessarily point to the cause of the Error Halt. The user should always use the subtest number (R4) to locate the area of failure in the test.

5.3 DISK AND CONTROLLER PARAMETER CHANGES

5.3.1 THE STANDARD DISK PARAMETERS ARE AS FOLLOWS:

Number of Cylinders = 823_{10} (1467_8)

Number of Heads = 5

Number of Sectors = 32_{10} (40_8)

Number of Words Per Sector = 256_{10} (400_8)

5.3.2 THE STANDARD CONTROLLER PARAMETERS ARE AS FOLLOWS:

Base Address (DCSR) = 164000

Interrupt Sector = 270

5.3.3 INSTRUCTIONS FOR CHANGING PARAMETERS

Located in Sections 3 and 4 of the Diagnostic Assembly listing are all the parameters listed above. To change any or all of these values, deposit into memory the desired value.

5.3.3.1 DISK CHANGES There are only 4 changes that apply, the labels are the following:

MAXSEC: Word 37 ; Last sector

MAXHD: Word 1000 ; Last head - justified for DUSH

MAXCYL: Word 1466 ; Last cylinder

WPSEC: Word 177400 ; # of words per sector in 2'scomplement

All other references to these values will be set up by subroutine "ADRSTP" during program start up.

5.3.3.2 CONTROLLER CHANGES The base address and interrupt vector labels are the following:

DCSR: Word 164000 ; Base Address

INTVEC: Word 270 ; Interrupt Vector

When a base address change is necessary, only change the location "DCSR". The remaining register assignments will be built by subroutine "ADRSTP" during program start up.

6.0 TEST ABSTRACTS

6.1 INDIVIDUAL TESTS

6.1.1 Test Ø

This test issues an I/O command and verifies that "INIT" clears all registers in the PHOENIX 211 Controller.

6.1.1.1 Initialization Test Subtest Listing

Board Being Tested	Subtest Number (R4)	Subtract Name/Error Condition
211+1	Ø	Incorrect Control & Status Register Initialization
211	1	Incorrect Bus Address Register Initialization
211	2	Incorrect Word Count Register Initialization
4	3	Incorrect Cylinder Address Register Initialization
2	4	Incorrect Disk Status Register Initialization
1	5	Incorrect Disk Error Register Initialization
3	6	Incorrect Sector, Head Register Initialization

6.1.2 Test 1: Register Load and Read Tests

This routine tests the capability of the selected formatter registers to be loaded and interrogated with any data. Each register is checked with four constants 100 times.

6.1.2.1 Patterns used are as follows:

Pattern 1 - Ø
Pattern 2 - 052525
Pattern 3 - 125252
Pattern 4 - 177777

6.1.2.2 Load and Read Sub Test Listing (R4 Value at Error Halt)

Board Being Tested	Subtest Number (R4)	Subtest Name/Error Condition
211+1	Ø	Incorrect Control & Status Register Load OR Read
211	1	Incorrect Bus Address Register Load OR Read
211	2	Incorrect Word Count Register Load OR Read
4+1,211	3	Incorrect Cylinder Address Register Load OR Read
3,1,211	4	Incorrect Unit, Sector, Head Register Load OR Read

6.1.2.3 Read only, and constant, bits in the controller registers are masked out of the test process.

6.1.3 <u>Test 2: Sliding One Register Load & Read Tests</u>

In this test all of the controller registers are loaded with a sliding 1 data pattern and interrogated for proper content. The logic one bit constant is "slid" from the bit zero position to the bit 15 position in 16 separate tests.

6.1.3.1 Sliding One Register Load and Read Subtest Listing

Board Being Tested	Subtest Number (R4)	Subtest Name/Error Condition
1	Ø	Incorrect Control & Status Register Load OR Read
3	1	Incorrect Unit, Sector, Head Register Load OR Read
211	2	Incorrect Bus Address Register Load OR Read
211	3	Incorrect Word Count Register Load OR Read
4	4	Incorrect Cylinder Address Register Load OR Read

6.1.4 <u>Test 3: Sliding Zero Register Load & Read Test</u>

In this test all of the controller registers are loaded with a "sliding zero" data pattern and interrogated for proper content.

The logic zero bit constant is slid from the bit zero position to the bit 15 position in 17 separate tests.

6.1.4.1 Sliding Zero Register Load and Read Subtest Listing

Board Being Tested	Subtest Number (R4)	Subtest Name/Error Condition
1	Ø	Incorrect Control & Status Register Load OR Read
3	1	Incorrect Unit, Sector, Head Register Load OR Read
211	2	Incorrect Bus Address Register Load OR Read
211	3	Incorrect Word Count Register Load OR Read
4	4	Incorrect Cylinder Address Register Load OR Read

6.1.5 Test 4: Disk Error and Status Register Tests

These tests verify that the nonexistent Head, Sector and Cylinder (Track) errors can be set and cleared properly.

Also tested in this test is the byte loading capability of the Command and Status register (DCSR).

6.1.5.1 Disk Error and Status Register Subtest Listing

Board Being Tested	Subtest Number (R4)	Subtest Name/Error Condition
3,1	Ø	No Illegal Sector Error Summary Generation
3,1	1	No Illegal Sector Error Bit Generation
3,1	2	Illegal Sector Error Bit did not clear
3,1	3	No Illegal Head Error Summary Generation
3,1	4	No Illegal Head Error Bit Generation
3,1	5	Illegal Head Error Bit did not clear
4,1	6	No Illegal Cylinder Error Summary Generation
4,1	7	No Illegal Cylinder Error Bit Generation
4,1	10	Illegal Cylinder Error Bit did not clear
1	11	No Illegal Sector,Head,Cylinder Error Summary Generation
3,4,1	12	Illegal Sector,Head,Cylinder Error Bits did not
3,4,1	13	Illegal sector, head, cylinder error summary set
1,211	14	Control & Status Upper Byte Did Not Load
1,211	15	Control & Status Lower Byte Did Not Load

6.1.6 TEST 5: Unit Select Tests

These tests verify that the Unit Select lines access the proper Disk Drives. This testing is accomplished by selecting each Disk Drive Port and issuing a SEEK. The DSTAT register-seeking flops are checked for proper seeking status and a Formatter Clear is then given to clear them. This test does not wait for Seek Done.

6.1.6.1 Unit Select Seeking Tests Subtest Listing

Board Being Tested	Subtest Number (R4)	Subtest Name/Error Condition
1,2,3,4	Ø	Disk Drive or Formatter, not ready
2,3,1	1	Did Not Set Unit Ø Seeking Flop Via Seek CMD
3	2	No Formatter Ready Generation on Formatter Clear CMD
2	3	Did Not Set Unit 1 Seeking Flop Via Seek Card
3	4	No Formatter Ready Generation on Formatter Clear CMD
2	5	Did Not Set Unit 2 Seeking Flop Via Seek CMD
3	6	No Formatter Ready Generation on Formatter Clear CMD
2	7	Did Not Set Unit 3 Seeking Flop via Seek CMD
3	10	No Formatter Ready Generation on Formatter Clear CMD

6.1.7 TEST 6: Seeking Test

In this exercise two seeking commands are issued, Seek and Recalibrate, and proper Seek-Done flops are checked. Also checked in this test is that an illegal track address used on a Seek command causes a Nonexistent Track Error and not a Seek Error.

6.1.7.1 Seeking Test Subtest Listing

Board Being Tested	Subtest Number (R4)	Subtest Name
1	Ø	Formatter not ready
4,2,3	1	Selected Drive Not Ready
4,2	2	Error Generation on RTZ CMD. No Seek Done Summary Status Bit Generation on RTZ
2,4	3	No Selected Drive Seek Done Status Bit Generation on RTZ
3,1	4	No Formatter Ready at RTZ Completion
4	5	No Illegal Cylinder Error Bit Generation on Seek CMD
4	6	Other Error Bit Generation on Illegal Seek CMD
3	7	No Seek Done Summary Bit Generation On Seek CMD
3	1Ø	No Selected Drive Seek Done Status Bit Generation on Seek CMD
4	11	Other Drive Seek Done Bit Generation on Seek CMD

6.1.8 TEST 7: Interrupt Test

The Interrupt tests use two formatter commands, the Recalibrate command and the Seek Only command. At this stage of testing, the Seek logic should be operable. The Interrupt logic uses these commands to cause the Seek Done identification bits to generate an Interrupt.

NOTE: If the CPU halts during this test at some memory location between 14 and 776, the Interrupt branched to a wrong location. This indicates a fault in the Interrupt Vector Address.

6.1.8.1 Interrupt Test Subtest Listing

Board Being Tested	Subtest Number (R4)	Subtest Name/Error Condition
1	Ø	Drive Not Ready
	1	None
2,1,211	2	No Interrupt Generation with RTZ CMD
2	3	Improper Disk Ident Bits set in Disk Status Register at RTZ Interrupt
3	4	No Interrupt Generation With Seek CMD
2	5	Imporper Disk Ident Bits set in Disk Status Register at Seek Interrupt

6.1.9 TEST 10: One Word Write Test

This test is the first Write attempted by the Formatter. It is only a one word transfer and only the ending formatter register parameters are checked.

6.1.9.1 This test will write data onto the first usable sector on the disk medium.

6.1.9.2 The write operation is conducted by the "DXFER" Subroutine which tests all controller parameters before, during, and after the specified operation.

6.1.9.3 If an error occurs it will be detected by DXFER Subroutine and the subtest count will be contained in location "DTCNT". For a complete description of the DXFER Subroutine subtest error conditions see Appendix A. Test 10 DTCNT subtest error conditions are defined below.

TEST 10:DXFER DTCNT Subtest Error Conditions

Note: This is the first test to use the DMA logic. If the computer bus hangs up, the 211 Interface Board or CPU backplane is suspect.

DTCNT Value	Subtext Name/Error Conditon	Board Being Tested
Ø	Controller Not Ready	1
1	Selected Disk Drive Not Ready	4,2
2	Controller Went Busy with No "GO" CMD	3
3	Controller did not go busy with "GO" CMD	3
4	Interrupt did not occur before timeout	3,4,1
5	Controller Hang in "BUSY" State	3,211,2
6	Word Count Register did not overflow	211,2,3
9	End Bus Address Register Incorrect	211
10	End Sector Value Incorrect	3
11	End Head Value Incorrect	3
12	End Cylinder Address Value Incorrect	4
13	Controller Error Bit Set	3,2,4

Note: 1. This test exercises so much incrementally new logic in the disk controller that the user should examine the disk controller register images saved by the diagnostic at the time of error detection.

Correct Ending Values for Disk Controller Registers are as follows:

	Register	Correct Contents
1	Control & Status	000306
2	Unit, Sector, Head	0x0001
3	Bus Address	xxxxxx
4	Word Count	000000
5	Cylinder	000000
6	Disk Status	100000
7	Error	00000Ø

Note: 2. Any soft seek header or header CRC error will cause the controller to abort the write operation before actually writing any data onto the disk.

6.1.10 TEST 11: One Sector Write Test:

6.1.10.1 This test writes data on one sector of the disk. The Cylinder address is normally determined by RWCYL, while the Sector, Head and Unit value is determined by RDUSH.

NOTE: 1.) This is the first time that a full sector transfer has been attempted.

2.) No data is checked during this test.

3.) The object of this test is to verify that the disk controller properly goes through the motions of performing a disk write operation:

4.) Specific functions verified during this test are:

1 - Multiple Write DMA Data Path to Controller FIFO memory
2 - Multiple Write FIFO data path to disk
3 - Multiple Word Count Register Write Incrementation
4 - Multiple Bus Address Register Write Incrementation

6.1.10.2 Optionally, the Sector and Head are dynamically selectable by the console switch register if CSWTCH is loaded with any non-zero value.

6.1.10.3 RWCYL and RDUSH locations are normally set up to write the data on the last available sector on the medium.

6.1.10.4 The actual write operation is performed under control of "DXFER" Subroutine and all appropriate errors are detected by "DXFER" and the error subtest count is contained in location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and subtest error conditons.

Note: If this test fails while test10 passes, the most probable suspect is board 2 followed by the 211 Interface Board.

6.1.10.5 DXFER DTCNT Subtest Error Conditions

DTCNT Value	Subtest Name/Error Condition	Board Being Tested
Ø	Controller Not Ready	1
1	Selected Disk Drive Not Ready	4
2	Controller went Busy with no "GO" CMD	3
3	Controller did not go Busy with "GO" CMD	3
4	Interrupt did not occur before timeout	2
5	Controller Hang in "BUSY" State	2
6	Word Count Register did not overflow	2,211
9	End Bus Address Register Incorrect	211
10	End Sector Value Incorrect	3
11	End Head Value Incorrect	3
12	End Cylinder Address Value Incorrect	4
13	Controller Error Bit Set	3,2,4

6.1.11 <u>TEST 12: One Sector Read Test</u>

6.1.11.1 This test reads data from one sector of the disk. The Cylinder address is normally determined by RWCYL, while the Sector and Head address are normally determined by RDUSH. Optionally, the Sector and Head address may be dynamically selectable by the console switch register if CSWTCH is loaded with any non-zero value.

<u>NOTE</u>: 1.) This is the first time that any disk Read data transfer operation has been attempted.

2.) No data is checked during this test.

3.) The object of this test is to verify that the disk controller properly goes through the motions of performing a disk read operation.

4.) Specific functions verified during this test are:

1 - Multiple Read DMA data path from controller FIFO memories to computer memory.
2 - Multiple Read data path from disk to FIFO memories.
3 - Word Count Register Read Incrementation
4 - Bus Address Register Read Incrementation

6.1.11.2 RWCYL and RDUSH locations are normally set up to read the <u>last</u> usable sector on the disk medium.

6.1.11.3 Data Read Back is stored in Input Buffer 1 (IDBUF1)

6.1.11.4 The actual read operation is performed by the "DXFER" Subroutine and all appropriate errors are detected by DXFER and the error subtest count is stored in location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine. Subtest error conditions for Test 12 are defined below.

6.1.11.5 DXFER DTCNT Subtest Error Conditions

DTCNT Value	Subtest Name/Error Condition	
Ø	Controller Not Ready	1
1	Selected Disk Drive Not Ready	4
2	Controller Went Busy with no "GO" CMD	3
3	Controller did not go Busy with "G" CMD	3
4	Interrupt did not occur before timeout	3
5	Controller Hang in "BUSY" state	3,2
6	Word Count Register did not overflow	2,3
9	End Bus Address Register Incorrect	211
10	End Sector Value Incorrect	3
11	End Head Value Incorrect	3
12	End Cylinder Address Value Incorrect	4
13	Controller Error Bit Set	3,2*

* Data Late Error

Note: 1. This test exercises so much incrementally new logic in the disk controller that the user should examine the disk controller register images saved by the diagnostic at the time of error detection.

Correct Ending Values for Disk Controller Registers are as follows:

	Register	Correct Status
1	Control & Status	000304
2	Unit, Sector, Head	XXXXXX
3	Bus Address	XXXXXX
4	Word Count	XXXXXX'S
5	Cylinder	100000
6	Disk Status	000000
7	Error	

Note: 2. Any soft seek header or header CRC error will cause the controller to abort the read operation before actually receiving any data from the disk.

6.1.12 TEST 13: Write Count Test

6.1.12.1 This test performs Write operations with an ever-increasing word count starting with 1 and increasing in power of 2 to a maximum of 4096 words. Each Write operation starts at absolute 0 Disk address.

NOTE:

1.) No data is checked during this test.

2.) The object of this test is to verify that the disk controller sequences logic performs correctly under data transfers less than and greater than one sector in word length.

6.1.12.2 All thirteen data transfer operations performed in this test are done by the "DXFER" Subroutine. Any errors detected by the DXFER Subroutine during any of the 13 data transfers performed will be indicated by the DXFER subtest count value contained in location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and DTCNT subtest error conditions.

6.1.12.3 Write Count Subtest Listing

R4 in this test is used to define what one of 13 write data transfer operations is to be performed by DXFER per the following table:

Board Being Tested	Subtest Number (R4)	Subtest Name/Error Condition
	Ø	Error During 1 Word Write
	1	Error During 2 Word Write
	2	Error During 4 Word Write
	3	Error During 8 Word Write
	4	Error During 16 Word Write
	5	Error During 32 Word Write
	6	Error During 64 Word Write
	7	Error During 128 Word Write
2	10	Error During 256 Word Write
3,211	11	Error During 512 Word Write
3,211	12	Error During 1024 Word Write
3,211	13	Error During 2048 Word Write
3,211	14	Error During 4096 Word Write

6.1.12.4 Logic specifically tested includes sector sequencer control, DMA word count Incrementation and Disk Word Count Incrementation

6.1.12.5 No new logic is exercised by subtests Ø - 7.

6.1.13 TEST 14 : One-Sector Read, Write, Compare Test

6.1.13.1 This Test writes data on one sector of the disk, reads it back, and compares the data obtained with the original data written. The cylinder address is specified by RWCYL while the Sector and Head values are normally supplied by RWUSH. Optionally, the Sector and Head data may be dynamically supplied from the console switch register if CSWTCH contains any non-zero value.

6.1.13.2 Data is normally written onto the last physical sector on the disk medium from the Output Buffer (ODBUF) and written back into Input Buffer 1 (IDBUF1).

6.1.13.3 All disk data transfers physically performed by the "DXFER" Subroutine. Any data transfer errors detected by DXFER will be indicated by the DXFER subtest count value contained in location DTCNT.

Please refer to Appendix A for a complete description of the DXFER Subroutine and subtest error conditions.

6.1.13.4 The Read, Write, Compare Subroutine (WRC) is used in this test to manage the Write, Read and Data Comparisons. (Reference Appendix B)

1.) If an error is encountered during the write data transfer (by DXFER) the "WRTERR" location is incremented.

2.) If an error is encountered during the write data transfer (by DXFER) the "RDERR" location is incremented.

3.) All Data comparisons between original data written onto disk from ODBUF and data received from the disk in IDBUF1 are performed by the Data Comparison Subroutine "DATCMP". (Reference Appendix C)

If an error is detected during the data comparison subroutine "CMPCNT" contains the number of words compared before the error occurred while R2 and R3 contain the correct and actual data values respectively.

6.1.13.5 R4 has <u>no</u> significance in this test with respect to subtest identification.

6.1.13.6 At this point the controller can at least go correctly through all the motions of doing both read and write operations, and is correctly reading the sector header data.

6.1.13.7 The capability to write or read data correctly has not been previously verified.

6.1.13.8 The most probable error to occur during this test is a data comparison error. If a data comparison error occurs, the most probable board at fault is Board 2 which contains all of the data path logic,followed by the 211 coupler.

6.1.14 <u>TEST 15: Read, Write, Compare Count Test</u>

6.1.14.1 This test performs Write, Read, and Data Compare operations with an ever increasing word count starting with one and increasing by powers of two to a maximum of 512 words.

Each disk operation starts at absolute disk address Ø.

6.1.14.2 All disk parameters are checked during the test by the DXFER Subroutine. Any data transfer errors detected by the "DXFER" Subroutine will be indicated by the subtest count value contained in location DTCNT.

Please refer to Appendix A for a complete description of the DXFER Subroutine and subtest error conditions.

6.1.14.3 The Write, Read, Data Comparisons are managed by the Read, Write, Data Comparison Subroutine (WRC).

Reference Appendix B.

1.) If an error is encountered during a write operation (by DXFER) the "WRTTERR" location is set to a non zero value.

2.) If an error is detected during a read operation (by DXFER) the "RDERR" location is set to a non zero value.

3.) If an error is detected during the data comparison (by DATCMP) location "CMPCNT" contains the word count value when the error was detected.

6.1.14.4 Write, Read, Compare Count Subtest Listing
R4 is utilized in the test to determine which one of 9 Write, Read, Compare Operations is being performed per the following:

Subtest Number (R4)	Subtest Name/Error Condition
Ø	Error During 1 Word Write, Read, Compare
1	Error During 2 Word Write, Read, Compare
2	Error During 4 Word Write, Read, Compare
3	Error During 8 Word Write, Read, Compare
4	Error During 16 Word Write, Read, Compare
5	Error During 32 Word Write, Read, Compare
6	Error During 64 Word Write, Read, Compare
7	Error During 128 Word Write, Read, Compare
10	Error During 256 Word Write, Read, Compare
11	Error During 512 Word Write, Read, Compare

6.1.14.5 This test does not specifically test any new controller logic but is executed to gain additional confidence in the capability of the disk controller to read and write data correctly. If an error occurs during data comparison, Board 2 is probably at fault.

6.1.15 TEST 16: Write-Protect Test*

This test verifies that the Controller will not Write over Write-Protected sectors.

6.1.15.1 First the test sets the Write-Protect bits of the absolute sector Ø and then attempts to write on that sector. The test then verifies that the Controller aborted the Write operation and that the Write-Protect error bit is activated.

6.1.15.2 All required disk data transfers are performed by the "DXFER" Subroutine and any errors detected by DXFER are indicated by the DXFER subtest count saved at location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and subtest error conditions.

* Does not apply on ECC equipped controllers.

6.1.15.3 Write Protect Subtest Listing

Subtest Number (R4)	Subtest Name/Error Condition
Ø	Write Protect Sector Format Error
1	No Error Detection on Write over Write Protected Sector
2	Write Over Write Protected Sector Error Bit Not Set
3	Error During Read of Write Protected Sector
4	Data Error-Attempted Write on Protected Sector Not Aborted
5	Error During Reformat of Write Protected Sector

6.1.15.4 This verifies that the attempted write on a Write Protected Sector was aborted by reading the data back and comparing it with original known data by the "DATCMP" Subroutine.

Reference Appendix C.

6.1.15.5 The Sector Write Protect logic exercised by this test is contained on Board 3.

6.1.15.6 Error Register logic is on Board 1.

6.1.16 TEST 17: Header Override - HEADER READ TEST *

6.1.16.1 This test verifies that the Read-Without-Header-Check logic is functioning properly. The Header is first erroneously written and then a normal Read operation is attempted to insure that we cannot read the sector.

6.1.16.2 The Read-Without-Header-Check command is then used to Read the sector. Checks are made to insure that no error occurs and that the data is correct.

6.1.16.3 All required data transfers are performed by the "DXFER" Subroutine, and any detected errors are indicated by the DXFER subtest count saved at location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and subtest error conditions.

6.1.16.4 Header Override Subtest Listing

Subtest Number (R4)	Subtest Name/Error Condition
Ø	Error During Erroneous Header Format
1	No Error During Read of Sector with Bad Header OR No Header Error Bit Set
2	Error During Header Override Read of Sector with Bad Header
3	Header Override Read Error - Data Incorrect
4	Error During Reformat of Sector
*	Does not apply to ECC equipped controllers

6.1.16.5 The header logic exercised by this test is contained on Board 3

6.1.16.6 The Error Register logic is contained on Board 1.

6.1.16.7 This section of Test 17 checks the ability of the controller to read the header field, header CRC, Data Field, and the Data Field CRC.

6.1.16.8 Header read verify subtest listing subtest number (R4)

Subtest Number (R4)	Subtest Namer/Error Condition
5	Error during reformat using write header, data, CRC command
6	Error during read of header, data, CRC, command
7	Data error on sector field
1Ø	Error during normal format

6.1.16.9 Early Phoenix 211's were shipped with the read hdr, CRC, Data, CRC command operating as a read header and CRC only command. If the Operator's controller constantly fails Subtest 7 of this Test (17), deposit a non-zero number in location "OLDHDR". This modification will cause the program to skip a certain section of the test.

Note: This difference will not cause any compatibility problems in mixing new or old controllers and, or disk packs. The Read Header, Header CRC, Data and Data CRC command is not used by any DEC* Operating Systems, and would only be used in special application programs, (pack verification), or in diagnostics.

*Digital Equipment Corporation

6.1.17 TEST 20: Write-Protect Override - Write Check Test

This test verifies that the Write-Protect-Override command logic functions properly.

6.1.17.1 Known data is written onto the disk. The sector to be Write-Protected is then so protected and new data is written using the Override command. The new data is then read back and checked to verify that the Write operation actually took place.

6.1.17.2 All required data transfers are performed by the "DXFER" Sub-routine, and any detected errors are indicated by the DXFER subtest count saved in location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and subtest error conditions.

6.1.17.3 Write Protect Override Subtest Listing

Subtest Number (R4)	Subtest Name/Error Conditions
Ø	Error During Write Protect Sector Format
1	Error During Write Over Write Protected Sector Override
2	Error During Read of Protected Sector
3	Data Error - Write Operation did Not Override Protected Sector
4	Error During Sector Reformat

6.1.17.4 The Write Protect Override Logic exercised by this test is located on Board 3.

6.1.17.5 The second section of Test 20 verifies the write check logic of the 211 controller. Known data is first written using normal write command. A write check command is issued and checks are made to insure no error was generated. The last step is to destroy a part of the memory buffer and to issue write-check commands (2) to insure that an error is generated.

6.1.17.6 Write Check Subtest Listing

SUBTEST NUMBER (R4)	SUBTEST NAME/ERROR CONDITION
5	Error during initial normal write
6	Error during write check command
7	Error not detected during write check command with destroyed buffer
1Ø	Write check error bit not set
11	Other errors bits set
12	Error not detected during second erroneous write check command
13	Write check error not set or some other error set as well
14	Corrected buffer with write check command reissued caused error

6.1.18 TEST 17: Header-Write Command Test

6.1.18.1 This test verifies that the Header can be written by the Write-Header-and Verify command (CRC). The Header is written to an erroneous value using the command and then a normal Read is attempted and the presence of the appropriate Header Error bit (s) is (are) checked.

6.1.18.2 All required data transfers are performed by the "DXFER" Subroutine, and any detected errors are indicated by the DXFER subtest count saved in location "DTCNT".

6.1.18.3 Header-Write Command Subtest Listing

Subtest Number (R4)	Subtest Name/Error Condition
Ø	Error During Erroneous Sector Header Format
1	No Detected Error During Read of Sector with Bad Sector Header OR Sector Header Error Bit Not Set
2	Bus Address Register Erroneously Incremented
3	Word Count Register Erroneously
4	Error During Sector Reformat to Illegal Head Header
5	No Detected Error During Read of Sector with Bad Head Header
6	Header Error Bit Not Set
7	Other errors generated other than sector/head error
1Ø	Error during reformat to illegal cylinder address
11	Error not detected when read attempted on sector with bad cylinder header
12	Error during final sector reformat

6.1.18.4 The Header logic exercised by this test is contained on Board 3.

6.1.18.5 Error logic is located on Board 1.

6.1.19 TEST 22: CRC and Header Compare Tests

6.1.19.1 These tests check the logic governing the detection of Header and CRC errors. Using the Write Header Data and CRC commands, erroneous Headers and CRC Checks are written in various combinations. Checks are then made for proper error bit settings.

6.1.19.2 All data transfers in this test are performed by the "DXFER" Subroutine, and any errors detected are indicated by the contents of the DXFER subtest location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and associated subtest error conditions.

6.1.19.3 CRC and Header Compare Subtest Listing

Subtest Number	Subtest Name/Error Conditions
Ø	Error During Write Format of Erroneous Header CRC
1	No Error Detected During Read of Bad Header CRC Sector
2	Header CRC Error Bit Did Not Set During 1
3	Some Other Error Bits Erroneously Set During 1
4	Error During Write Format of Erroneous Header
5	No Error Detected During Read of Bad Header Sector
6	Sector or Cylinder or CRC Header Error Bit Not Set
7	Error During Write Format of Erroneous Header
10	No Error Detected During Read of Bad Header Sector
11	CRC or Cylinder Error Bit Not Set During Read
12	Other Error Bit Erroneously Set During Read
13	Error During Write Format of Erroneous Header
14	No Error Detected During Read of Bad Header Sector
15	No CRC Header Word 1 Error
16	Other Error Bit Erroneously Set
17	Error During Write Format of Bad Sector Header
20	No Error Detected During Read of Sector with bad Header
21	No CRC Header Error Detected
22	Other Error Bit Erroneously Set
23	Error During Final Correct Format Write

6.1.19.4 The Header and CRC logic exercised in this test is contained on Board 3.

6.1.19.5 The error register logic is contained on Board 1.

6.1.20 TEST 23: Bad Sector Flag Test

This test checks the Bad Sector logic

6.1.20.1 Bad Sector bits are added to Header Word #1 of Sector Ø, Head Ø, Track Ø, and a Read command is issued.

6.1.20.2 The disk controller Error Register (DERR) is then checked to see if the Bad Sector Flag Bit is set.

6.1.20.3 A check is then made to insure that the controller "skipped" the flagged "defective" sector Ø and conducted a read operation on Sector 1.

This is done by comparing the data actually obtained during the read with data known to have been on Sector 1 of the disk.

6.1.20.4 All data transfers in this test are conducted by the "DXFER" Subroutine, and any errors detected are indicated by the contents of the DXFER subtest count location "DTCNT".

6.1.20.5 Bad Sector Flag Subtest Listing

Subtest Number (R4)	Subtest Name/Error Condition
Ø	Error During Bad Sector Flag Reformat of Sector Ø
1	Error During Write of Known Data to Sector 1
2	No Error Detected During Read of Sector Ø (Bad Sector Flag Set)
3	Bad Sector Flag Bit Not Set in Error Register
4	Other Error Bit Erroneously Set
5	Erroneous End Sector Field Value
6	Error During Reformat of Sector Ø.
7	No Error During Timeout Test
10	Time Out Error Bit Did Not Set
11	No Error Clear on Formatter Clear Command

6.1.20.6 The Bad Sector Flag logic exercised in this test is contained on Board 3 and associated error status logic is contained on Board 1.

6.1.21 TEST 24: Recalibrate Test

These tests check the RTZ command for two areas.

6.1.21.1 The first section verifies that the drive is indeed positioned on Track Ø after an RTZ.

6.1.21.2 Secondly, the conditions of the seeking flip-flops are checked (DSTAT) during and after the RTZ command.

6.1.21.3 Recalibrate Subtest Listing

Board Being Tested	Subtest Number(R4)	Subtest Number/Error Condition
1	Ø	Controller Not Ready
4	1	Selected Disk Drive Not Ready
4	2	Error During Seek Operation
2	3	Correct Seeking Flop Not Set
2	4	Other Seeking Flop Erroneously Set
2,4	5	Proper Seek Done Flop Not Set
2	6	Seeking Flop Still Set After Seek Done
3	7	Seeking Flop Erroneously Set During Seek Inhibit Read
4,3	10	Drive Did Not Go to Cylinder Ø - Error During Seek Inhibit Read of Cylinder Ø
3	11	Seek Status Flops Erroneously Set During Seek Inhibit Read
3	12	No Error Detected During Erroneous Seek Inhibit Read
3	13	Cylinder Header Error Bit Did Not Set

6.1.22 TEST 25: Implied Seek Test

This test verifies that the seek done flag will not be set during an "implied" seek operation and that no seek operation takes place when the Seek Inhibit Bit ( ) of the Control and Status Register is set.

6.1.22.1 First the implied Seek is utilized in a Read command and a check is made to see that the Seek Done flag is not set.

6.1.22.2 Secondly, two Read commands are given, one with a proper cylinder location and Seek Inhibit set to insure the original seek was to the proper cylinder, and the other with an erroneous cylinder address and Seek Inhibited to force a Cylinder Header Compare error.

6.1.22.3 All data transfers in this test are performed by the "DXFER" Subroutine, and any errors detected are indicated by the contents of the DXFER Subtest location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and associated subtest error conditions.

6.1.22.4 Implied Seek Subtest Listing

Subtest Number (R4)	Subtest Name/Error Condition
Ø	Selected Disk Drive Not Ready
1	Error During RTZ Operation
2	Error During Implied Seek Read
3	Seek Done Flag Erroneously Set
4	Error During Same Track Seek Inhibit Read
5	No Error During Seek Inhibit Read from Different Track
6	No Header Error Bit Set During Read of Subtest 5
7	No Controller Ready on Formatter Clear

6.1.22.5 The Seek Inhibit logic exercised in this test is contained on Boards 1 and 4. If Seek Done Flag in subtest 3 is set, the most probable board at fault is Board 3.

6.1.23 TEST 26: Converge/Diverge Worst Case Seek Loop Test - Address Test

6.1.23.1 This test sends the disk drive into a Worst-Case-Seeking loop, and is a test of the disk drive servo electronics rather than of the disk controller.

6.1.23.2 The seeking starts at the extreme ends of the disk cartridge (Track Ø and Track 1466_8) and alternates between an increasing track address and a decreasing track address. This process continues until each track has been searched twice.

6.1.23.3 No data is transferred during this test.

6.1.23.4 If an error occurs:

R1 = Current decreasing Cylinder Address
R2 = Current increasing Cylinder Address
R3 = Loop Count that error occurred on
R4 = subtest count
R5 = Loop Control Flag

R5 = 1 For Increasing Cylinder Seek Cycle
R5 = Ø For Decreasing Cylinder Seek Cycle

6.1.23.5 Converge/Diverge Worst Case Seek Loop Subtest Listing

Subtest Number (R4)	Subtest Name/Error Condition
Ø	Controller Not Ready
1	Selected Disk Drive Not Ready

6.1.23.6 The logic in the disk controller principally exercised by this test is the seek logic contained on Board 2 and the disk interface logic contained on Board 4.

6.1.23.7 Disk Address Test

This section of Test 26 checks the logic of the three disk address functions:

A. Sector
B. Head (Surface)
C. Cylinder

Starting at disk address ABS. Ø, one sector write/read commands are issued incrementing the sector and head values until the first cylinder boundary is crossed. (CYO Ø → CYL 1) See Fig. 6.1.

Sector X = Variable   Cylinder M = 41Ø822

fig 6.1 Disk Structure

After cylinder one is reached, transfers are issued from maximum sector and maximum head addresses. These one sector commands increment the cylinder counter each time and continue until the overrun error is detected.

The result of the above sequence is that all disk address lines, sector, head and cylinder have been exercised.

6.1.23.8 The transfers in this section of Test 26 are performed by subroutine "DXFER", and any errors detected are indicated by the subtest location "DTCNT".

6.1.24 Test 27 Functions Tested:

1. Memory Extension
2. Buss Timeout
3. Overrun Error
4. Memory Address Increment

6.1.24.1 Overrun Error

Disk address parameters are loaded as maximum values. i.e. last addressable sector, head and cylinder. A two sector write command is issued and checks are made to ensure that the overrun error was generated and that the controller aborted operations.

6.1.24.2 Memory Extension

One word write commands are issued with memory address loaded to a value that will increment an extension bit. Checks are made to ensure that the boundary was properly crossed. See Table 6.1

MEM EXT	MEM ADDR	MEM EXT	MEM ADDR
ØØ	177776	Ø1	Ø
Ø1	177776	1Ø	Ø
1Ø	177776	11	Ø
11	177776	ØØ	Ø
Before Command		After Command	

TABLE 6.1 MEMORY EXTENSION TEST

6.1.24.3 Buss Timeout Error

A one word write command is issued with memory address set to a nonexistent value. A check that the proper error bit is set and that no other errors are present.

6.1.24.4 Memory Increment Test

This test was designed to detect the failure or faulty wiring of the memory address register.

A one word write is issued from every possible starting memory address, including the memory extension bits. An illegal sector address is used to speed up the process. (Using an illegal sector aborts the command early and therefore stops the sequencer from looking for sector coincidence.)

No errors are checked or cared about in this exercise, only the incrementing of the memory address register by 2 on each transfer.

6.1.24.5 Test 27 Subtest Listing

Boards Being Tested	Subtest Number	Subtest Name/Error Condition
1,4	Ø	No error detected during overrun
1,4	1	Overrun error bit not set
1,3,4	2	Some other errors set
1,3	3	Error not cleared by system clear
211	4	First memory extension didn't set
211,1,3	5	If an error was present wasn't cleared by system clear or second memory extension didn't set
211,1,3	6	Memory ext. didn't overflow or error condition wasn't cleared by system clear
211,1	7	Didn't force error with buss timeout
211	1Ø	Buss timeout not set
1,3	11	Other error with timeout
211,1	12	Error not cleared by system clear
211 (w.w.)	13	Special wirewrap test. If this subtest fails FCO #75 is missing from your 211 (wirewrap only)

6.1.24.5 Test 27 Subtest Listing Continued

Boards Being Tested	Subtest Number	Subtest Name/Error Condition
211	14	At this subtest we are checking the memory address incrementation if an error should occur here it would be because either the increment didn't take place at all or the mem. reg incremented to an erroneous value.

6.1.25 TEST 30: Seek/Write, Seek/Read Test - Random Transfer Test

6.1.25.1 This test is a disk subsystem reliability test. The drive is made to seek a Worst Case pattern and write data generated by a subroutine (PATGEN), that creates a unique array of data each time it is called.

6.1.25.2 After every track has been written the same seeking pattern in reverse is commanded and Read and Compare is done from each track.

6.1.25.3 The same converge/diverge seeking loop utilized in Test 18 is used in this test.

6.1.25.4 All data transfers in this test are performed by the "DXFER" Subroutine, and any errors detected are indicated by the contents of the DXFER Subtest location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and associated subtest error conditions.

6.1.25.5 Seek Write, Seek Read Subtest Listing

This test is divided into two parts for write and read.

During the write operation R4 is incremented each time a data transfer is made, four transfers being made per cycle until the complete disk is written.

During the read operation R4 is reinitialized to 5 and then incremented for each read operation performed.

	Subtest Number (R4)	Subtest Name/Error Conditions
Write Cycle	Ø, 5, etc.	Error During First Decreasing Cylinder Disk Write
	1, 6, etc.	Error During Second Decreasing Cylinder Disk Write
	2, 7, etc.	None
	3, 10, etc.	Error During First Increasing Cylinder Disk Write
	4, 11, etc.	Error During Second Increasing Cylinder Disk Write
Read Cycle	5, 15, etc.	Error During #1 Decreasing Cylinder Read
	6, 16, etc.	Error During #1 Read Data Comparison
	7, 17, etc.	Error During #2 Decreasing Cylinder Read
	10, 20, etc.	Error During #2 Read Data Comparison
	11, 21, etc.	Error During #3 Increasing Cylinder Read
	12, 22, etc.	Error During Read Data Comparison Cycle
	13, 23, etc.	Error During #4 Increasing Cylinder Read
	14, 24, etc.	Error During #4 Read Data Comparison

6.1.25.6 This test is more of a disk drive exerciser than a test of disk controller logic.

6.1.25.7 Most probable error to occur during this test is a soft header seek error indicating a disk drive positioning problem.

6.1.25.8 The second portion of this test is also a disk subsystem reliability test. The only difference in the second part is that all disk addresses and data patterns as well as transfer sizes are random.

If more than one disk has been selected during Diagnostic startup, the unit selection will also be random.

6.1.25.9 Subtest Listing

The subtest number remains 5 throughout the test. If an error occurs refer to Section 6.1.25.4

6.1.26 TEST 31 : Disk Data Test

This test writes over the entire disk surface and then verifies the data by reading it back and comparing it with the original data written.

Data is written two sectors at a time and read into two sectors at a time through all of the surfaces before the cylinder register is incremented.

The data pattern used is a rotating test pattern with the exception that the first three words of every sector contain the absolute disk address.

6.1.26.1 Write Program

This program writes over all of the disk surfaces two sectors at a time. If an error is detected CURSEC, CURHD, and CURCYL indicate the disk address at which the failure occurred.

6.1.26.2 Read Program

This program reads the entire disk surface two sectors at a time. If an error is detected, CURSEC, CURHD and CURCYL indicate the disk address at which the failure occurred.

6.1.26.3 Boundary Crossing

Both the Write and Read routines command their operations to include Boundary Crossing, i.e. two-sector writes are issued at MAXSECTOR and continue on to increment the Head bits and Cylinder bits.

6.1.26.4 All data transfers in this test are conducted by the "DXFER" Subroutine, and any errors detected are indicated by the contents of the DXFER subtest count location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and associated subtest error conditions.

6.1.26.5 The Data Comparison Subroutine "DATCMP" is used to perform all of the data comparisons made in the read portion of this test. Refer to Appendix C for a complete description of this subroutine and associated error conditions.

Note: 1. The disk data test is more of a disk drive and media reliability test than a disk controller test.

2. The test will normally stop at the detection of any error condition.

3. If the test stops with an error the user should note the contents of the DUSH and DCYL Registers and rerun the test.

4. Repeated test failures at the same absolute disk address is indicative of a defective sector in the media.

6.1.27 TEST 32: Read and Compare Test

6.1.27.1 This program reads the complete disk surface and compares data received with reference data. Before entering this test, Test 31 must be run, since only Read commands are used during this exercise. Test 32 is essentially the Read portion of Test 25 modified to inclu a data error counter for use as a reliability test.

6.1.27.2 Setting Switch 7 of the console during step 4.1.7 of the starting procedure inhibits Program Halt when a Data Compare error is detected. Location "ERRCNT" is incremented each time an error is found. For explanation of Retry mode, see Section 4.2.

6.1.27.3 All data transfers in this test are conducted by the "DXFER" Subroutine, and any errors detected are indicated by the contents of the DXFER subtest count location "DTCNT".

Please refer to Appendix A for a complete description of the DXFER Subroutine and associated subtest error conditions.

6.1.27.4 The Data Comparison Subroutine "DATCMP" is used to perform all of the data comparisons made in this test.

Refer to Appendix C for a complete description of this subroutine and associated error conditions.

6.1.28 TEST 36: Simultaneous Overlapped Seek Test

6.1.28.1 Objective: This test verifies proper operation of the disk controller seek logic by initiating simultaneous seeks on multiple disk drives and verifying that:

1.) A seek complete interrupt is received from each disk drive commanded to seek.

2.) The seeking, and seek done status bits for each disk drive and seek summary status bit operate correctly.

3.) The seek drive identification logic correctly identifies the disk drive generating the Seek Done Interrupt.

6.1.28.2 Operation:

6.1.28.2.1 This test will operate on the number of disk drives from 1-4 defined to the diagnostic program by the operation via the computer switch register at the time of program initiation.

6.1.28.2.2 Initially all disk drives to be used in test are commanded to cylinder zero via repeated use of the Select and Seek Subroutine (SELASK) defined in Appendix E.

6.1.28.2.3 All disk drives utilized are then commanded to seek to the last cylinder on the disk (MAXCYL) via the Select and Seek Subroutine (SELASK) defined in Appendix E.

6.1.28.2.4 The disk controller interrupt is then enabled and the test waits for the seeks to complete via entering the WAITI Subroutine.

6.1.28.2.5 A special seek interrupt service routine (SKINT) is used for this test which checks for proper:

.1 Seek Done Summary Bit Set
.2 Seeking Status Bit Reset
.3 Drive Seek Done Status Bit Set

This interrupt service routine also increments an interrupt count location (ICNT) once for each interrupt received, and enters the Seek Ident field value received in the corresponding Interrupt Flag Table (INTTBL) location.

6.1.28.2.6 The Seek End Check Subroutine (SENCHK) is then used to verify that the proper number of seek done interrupts was received and that the Seek Ident Field values previously saved in INTTBL are correct. See Appendix F for detailed description of SENCHK.

6.1.28.3 Subtest Listing

6.1.28.3.1 A separate subtest count is maintained for each disk drive utilized in this test.

6.1.28.3.2 The subtest count for each disk drive is contained in the corresponding drive location in the Seek Subtest Count Table (SKCNT).

6.1.28.3.3 Seek Subtest Disk Drive Values are as follows:

	SKCNT Table Value	Functional Test/Error Condition
SELASK Errors	Ø	Disk Drive Not Ready When Selected For RTZ
	1	Drive Seeking Status Bit Not Set at Beginning of RTZ
	2	Disk Drive Not Ready When Selected For Seek
	3	Drive Seeking Status Bit Not Set at Beginning of Seek Command
SKINT Errors	4	Drive Seek Done Status Bit Not Set at Interrupt
	5	Drive Seeking Status Bit Not Reset at Interrupt
	6	Seek Done Summary Bit Not Set at Interrupt
	7	Disk Controller Error at Interrupt
	10	Disk Drive Not Ready at Interrupt
SENCHK Error	11	Drive Seek Identification Field Incorrect at Interrupt

6.1.28.3.4 R4 Subtest Values

R4 is incremented for subtests not unique to a disk drive. R4 subtest listings are as follows:

R4 Value	Function/Error Condition
Ø	Disk Controller Not Ready
1	Incorrect Number of Interrupts or other seek error. See SKCNT Table Subtest values.

6.1.29 TEST 37: Overlapped Seek and Data Transfer Test

6.1.29.1 Objectives

This test verifies that the Disk Controller functions properly when seek operations are overlapped with data transfer operations.

6.1.29.2 Operation

This test initiates a 8K word data transfer on the MAXCYL of unit Ø after having initiated seeks to cylinder 5 on all other units connected to the disk controller, as defined by the operator at time of program initiation.

6.1.29.2.1 The Seek operations on all units but Ø should complete before the 8K data transfer being conducted on unit Ø.

6.1.29.2.2 When unit Ø completes the data transfer:

1.) Seek status data should be present for one of the other drives when the transfer done interrupt is generated.

2.) Seek interrupts for all other drives commanded to seek will be pending.

6.1.29.2.3 The Program:

1.) Verifies that seek status is present and proper for one disk when transfer complete interrupt is received.

2.) Redirects pending seek interrupts to seek interrupt service routine.

3.) Verifies that seek interrupts are received for all other drives commanded to seek, and that status is proper.

NOTE:

This test requires a minimum of two disk drives for proper operation.

6.1.29.2.4 Initially all disk drives to be exercised are returned to cylinder Ø via executing a RTZ command via the Select and Seek Subroutine "SELASK". (See Appendix E)

6.1.29.2.5 The SELASK Subroutine is used to initiate commanded Seek Operations on all other disk drives but unit Ø.

6.1.29.2.6 The DXFER Subroutine is used to conduct the 8K write data transfer on MAXCYL of unit Ø and to check all standard controller functions for normal operation: (Please see Appendix A for description of DXFER Subroutine)

6.1.29.2.7 The Seek Interrupt Service Routine (SKINT) is used to field seek done interrupts for disk drives 8 and 3 (if used) and to check for proper drive seek status conditions. (See Appendix G for a description of the SKINT Routine)

6.1.29.2.8 The Seek End Check Subroutine (SENCHK) is used to verify that the correct number of interrupts was received and the seek identification field values received at time of interrupt are correct. (See Appendix F for description of SENCHK.)

6.1.29.3 Subtest Listing

R4 is used to indicate primary subtest values while the Seek Count Table (SKCNT) is used to catalogue subtest values associated with each disk unit.

6.1.29.3.1 R4 Subtest Listing

R4 Value	Functional Test/Error Condition
Ø	Controller Not Ready
1	RTZ Initiate Error
2	Seek Initiate Error on Unit 1,2, or 3
3	DXFER Transfer Error on Unit Ø
4	Seek Done Summary Flag not Set at Unit Ø transfer Complete
5	Drive Seek Done Flag Not Set at Unit Ø Transfer complete
6	Drive Seeking Flag Set at Unit Ø transfer complete
7	Other Disk Drive Seek Done Interrupt Errors
10	Wrong Number of Interrupts or Seek Ident. Error.

6.1.29.3.2 Seek Count Table SKCNT Drive Subtest Listing (Drives 1, 2 or 3)

	SKCNT Table Value	Functional Test/Error Condition
SELASK Errors	Ø	Disk Drive Not Ready When Selected for RTZ
	1	Drive Seeking Status Bit Not Set at Beginning of RTZ
	2	Disk Drive Not Ready When selected for Seek
	3	Drive Seeking Status Bit Not Set at Beginning of Seek
SKINT Errors	4	Drive Seek Done Status Bit not Set at Interrupt
	5	Drive Seeking Status Bit Not Reset at Interrupt
	6	Seek Done Summary Bit Not Set at Interrupt
	7	Disk Controller Error at Interrupt
	10	Disk Drive Not Ready at Interrupt
SENCHK Error	11	Drive Seek Identification Field Incorrect Interrupt

NOTE: The remainder of the tests excluding the format programs 45, 47, 50 apply only to special option controllers.

6.2 Special Option Tests

6.2.1 Error Correction Code 211 Tests

Phoenix 211's equipped with ECC have 2 more registers than standard controllers. ECC controllers have a much more complex header verification process and require several special tests. The following 6 tests are substituted into the test origin table when the operator selects the ECC option.

6.2.2 Test 16A ECC Register Checks

This test checks the ECC pattern register for loading and reading as well as the initialization of both the ECC pattern reg. and the ECC bit count register.

6.2.2.1 Subtest Listing

Subtest Number	Error Condition
Ø	ECC bit position not clear after I/O reset
1	ECC bit pattern not clear
2	Bit 15 didn't set or too many bits in ECCPW Register
3	Bit 15 didn't clear in ECCPW
4	NOP command cleared bit 15 in ECCPW
5	System clear didn't clear bit 15 in ECCPW
6	I/O reset didn't clear bit 15 of ECCPW

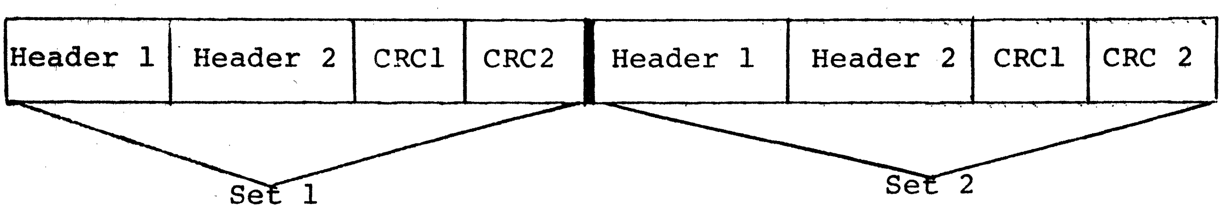
6.2.2.2 All subtests of ECC Test 16A are testing Bd. #1

6.2.3 Test 17A ECC/CRC Detection Test

This test checks the capability of the ECC detection logic to sense bad header CRC words.

In an ECC controller there are two pairs of header words and two pairs of header CRC words. The only time a header compare error will happen is if an error occurs in both pairs. The only time a header CRC error occurs is if the error is in both CRC pairs.

The header configuration is shown below.



Several different combinations of errors are tried and ending disk parameters are checked to ensure operation aborts.

Compatibility with other controllers is checked by reading a newly created CRC/ECC word and comparing it with a word previously generated at Xylogics' manufacturing facilities.

6.2.3.1 Subtest Listing Test 17A

In all subtests the order of board test is 1. Board #1
2. Board #3

Subtest Number	Error Condition
Ø	Formatter not ready
1	Error during reformat
2	No error during bad hdr read
3	CRC not detected
4	Other error with CRC
5	Word count incremented
6	Bus address incremented
7	Sector (DUSH) didn't increment properly
1Ø	Error during reformat - Sector Ø
11	Error during reformat - Sector 1
12	Error during read of Sector
13	Error during read of Sector 1 using read hdr, data, CRC command
14	First CRC word not compatible
15	Second CRC word not compatible
16	First CRC word, second pair, not compatible
17	Second CRC word, second pair, not compatible
2Ø	Reformat Sector Ø caused error
21	No error detected during read of sector Ø
22	CRC error not detected
23	Reformat of Sector Ø caused error
24	Error not detected during read of sector Ø
25	CRC error not detected
26	Error during reformat of sector Ø
27	Error not detected during read of sector Ø
3Ø	CRC error not detected
31	Error during sector Ø reformat
32	Error during sector Ø read
33	First CRC word ≠Ø
34	Second CRC Word ≠Ø
35	Third CRC Word ≠Ø
36	Fourth CRC Word ≠Ø

6.2.4 Test 21A ECC Detection Test

This test sequence checks the ECC detection logic of the Phoenix 211. All subtests use the ECC inhibit option that stops all correction actions that would normally occur if one encounters an ECC error. The purpose of inhibiting ECC is that at this time in testing the correction logic has not been checked.

6.2.4.1 ECC Sector Format and Read Routine

This routine issues 4 disk commands during the ECC test sequence. The general function of the exercise is to force ECC errors by destroying a small section of a sector by using the write header, CRC, data and ECC command. The operator can select the area in the data field where he desires to cause the error. The steps to accomplish this are as follows:

1. Using the normal write command, write a single sector executing this command will cause ECC words to be generated.

2. Read the sector using the read format command this command reads the entire sector, including headers, header CRC's, data and ECC words.

3. Rewrite the sector after changing desired data word (from 1 to 16 bits).

4. Issue normal read on that sector to cause ECC error

5. Exit routine.

Calling sequence:

```
(X)        JSR      R5,WRTECC       ;  Program call
(X+2)      .Word Ø                  ;  Data pattern desired
(X+4)      .Word Ø                  ;  Word position (X2)
(X+6)      .Word Ø                  ;  Bit (S) cleared to
                                          cause error
```

Note: This routine does not modify the ECC inhibit bit (bit 15 of the ECC bit pattern reg). It is therefore possible to use this call to check ECC correction as well as detection logic.

6.2.4.2 Subtest Listing

In all subtests the order of board under test is:

1. Board #2
2. Board #3

Subtest Number	Error Condition
Ø	Error correct sector Ø
1	Error not detected bit position 1
2	ECC error bit not set
3	Error not detected bit position 4096
4	ECC error bit not set
5	Error not detected bit position 2048
6	ECC error bit not set
7	Error all one's - good sector
1Ø	Error not detected bit position 1
11	ECC error not detected
12	Error not detected bit position 4096
13	ECC error not detected
14	Error not detected bit position 2048

Subtest Number	Error Condition
15	ECC error not detected
16	Error not detected bit pos. 2048 with checkboard pattern
17	ECC error not detected
2Ø	Error not detected bit position 2048 - reverse checkboard
21	ECC error bit not set

6.2.5 Test 22A ECC Detection/Correction Test

This test is the first to check the capabilities of the ECC logic in correcting bad data as a result of non-writable spots on a disk pack.

The Phoenix 211 has commands that enable the programmer to write data on any part of the sector field including the header CRC words and the data ECC words. This option makes it very easy to check other controller functions like the ECC generation.

The following tests will try to force ECC errors in the below manner:

1. Issue a write with preselected data field
   This command is a normal write function.

2. Issue a read command to read ECC words
   This command reads all fields of the sector.

3. Issue a second write command after having changed the data field to force the ECC error.
   This command write all fields of the sector.

4. Issue a second normal read to force the ECC error

5. Later in the test procedure the ECC position and pattern registers will be checked for proper values.

6.2.5.1 Subtest Listing

All subtests are exercising Bd. #1 and Bd. #3

Subtest Number	Error Condition
Ø	Error not detected bit position 1 bad (ECC inhibit set)
1	ECC error bit not set
2	Word count overflowed - shouldn't have
3	ECC PB ≠Ø
4	Error not detected bit position 1 bad (ECC inhibit reset)
5	ECC error bit not set
6	ECCPB ≠1
7	ECCPW ≠1 (Does not equal)
1Ø	Error not detected (11 bit error)
11	ECC error not detected
12	ECCPB ≠1
13	ECCPW ≠3777 (11 bit XOR)

Subtest Number	Error Condition
14	Error not detected (12 bit error)
15	ECC error bit not set
16	ECCPB ≠110041 (bit 15)
17	Error not detected during sliding 1 bit error
2Ø	ECC error bit not set
21	ECCPB ≠1 (count always = 1 until error bit passes 11th bit position into the sector)

6.2.6 Test 23A ECC Compatibility/Exercise

6.2.6.1 These tests check that the controller under test is generating the same CRC/ECC patterns that previous 211 controllers have generated.

6.2.6.2 A second part of this exercise is to force CRC errors on various areas of the disk and check that the ECC information generated by the ECC logic is correct. Tests TNA21 and TNA22 do similar checks however, those tests center on the first word of the data field.

6.2.6.3

Subtest Number	Error Condition
Ø	Error during sector Ø write
1	Error during sector Ø read (This reads all fields)
2	First ECC word is bad
3	Second ECC word is bad

Boards under test are Boards #1 and #3

6.3 Test 33 Multiple CPU Exercise

6.3.1 This test is an exercise for Phoenix 211 systems that are operating with the multiple CPU option. This is the only 211 Diagnostic Test that will allow more than one CPU to be requesting the 211. The format of the test is as follows:

1. The formatter is requested. If this CPU has access before the request is made than an error is logged.

2. A write/read and data compare routine is started and normal subroutines are used to drive this sequence.

3. If step #2 is successful the program releases formatter and if another CPU has a request set then it should receive access.

4. A counter of 1ØØØ.transfers is decremented and if not done, steps 1 thru 3 are repeated.

; ***This test can be run if a controller does not have a multiple CPU setup, steps 1 and 3 are omitted.

The disk address and the data pattern are random.

The units used are any and all that are on line and ready. (chosen randomly also)

6.3.2 In the Multiple CPU Test the subtest is not kept in the test body itself. The subroutine "REQUST" which does all transfer and request/release functions holds the subtest count. The subtest listing for this test will refer to the code in "REQUST".

6.3.3 <u>Subtest Listing (REQUST SUBROUTINE)</u>

Subtest Number	Error Condition
Ø	Formatter ready - should be dead
1	No interrupt from request
2	No formatter ready/interrupt enable
3	Error during write/read/data compare -Look at "DTCNT" if equal to 14 then look look for data erro
4	Formatter stayed ready after release command issued.

7.0 Simultaneous Multiple Controller Exerciser Test

7.1 Objective

This test exercises "N" Phoenix 211 Disk Controllers simultaneously in order to verify that the disk controllers will operate properly in a system with other DMA devices.

7.2 Program Loading

This program is a stand-alone program and is supplied as a loadable tape in absolute format. Program can be loaded in any computer with 8K of memory via the standard DEC supplied absolute loader.

7.3 Program Operating Instructions

7.3.1 Select the number of Phoenix 211 Disk Controllers to be simultaneously exercised by entering the appropriate binary number in location "CNUM". (i.e. for two controllers, 2 should be deposited in location CNUM.)

7.3.2 Start program execution at location "START".

NOTE!!!!

7.3.2.1 This test assumes that the disk controller base register addresses are assigned sequentially and Mod 20 starting at location 164000.

7.3.2.2 This test, once started will conduct 50 data transfers run on each disk drive and stop at location "ENDTST".

7.3.2.3 The test will stop upon detection of any error condition.

7.3.2.4 If an error is detected:

R0 = Base address of associated controller
R1 = Subtest No.
R2 = Reference data value
R3 = Actual Erroneous data value
R4 = Program address where error was detected

7.3.2.5 Actual Disk Controller register images at time of any error are saved in the error register image table "ERRIMG".

7.4 Program Operation

7.4.1 The test physically conducts simultaneous 2048 word write transfers on the number of disk controllers selected by the value inserted into location "CNUM" by the operator.

7.4.2 Data from absolute memory locations Ø-2048 is written on cylinder Ø of the disk drive connected to each disk controller.

7.4.3 All disk controller end parameters are verified for proper value.

7.4.4 Disk Controller Addressing

Since multiple disk controllers are used, all references to disk controller registers are made indirectly via the Controller Register Address Table. The actual register addresses for a given controller are computed and entered into the address table by the Controller Base Address Setup Subroutine (ADRSTP). See Appendix I for a detailed description of this subroutine.

7.4.5 Data Transfer Subroutine and Initiation

A common subroutine (XFRST) is used to initiate all data transfers in this test. This subroutine is described in Appendix J.

7.4.5.1 Initially transfers are initiated on both controllers by the program and the controller interrupts are enabled.

7.4.5.2 Subsequent data transfers on each disk controller are iniated by the common interrupt service routine at the completion of any data transfer after all controller parameters are checked for proper ending values.

7.4.6 Program End Address

When the test has been successfully run, the program will stop at location "DONE".

7.5 Program Subtest Listing

Since the test involves the use of two or more controllers simultaneously, more parameters are required to define the controller and the associated error condition per the following:

7.5.1 Disk Controller Identity (CERRNO)

The identity of the disk controller awsociated with any error condition is saved in location "CERRNO".

7.5.2 Disk Transfer Subtest Count (XFRIER)

Any Errors discovered during the initiation of any disk data transfer are identified by the subtest value save in location "XFRIER". XFRIER Subtest Values are as follows:

XFRIER Value	Functional Test/Error Condition
Ø	Selected Disk Controller Not Ready
1	Selected Disk Drive Not Ready

7.5.3 End Transfer Parameter Subtest Count (ENDCT)

Any error discovered at the completion of any data transfer during the check of all end parameter values is identified by the subtest value saved in location "ENDCT".

ENDCT Value	Functional Test/Error Condition
Ø	Controller Error at Interrupt
1	Word Count Register Did not Overflow
2	End Bus Address Register Value Incorrect
3	End Cylinder Address Register Value Incorrect
4	End Unit, Sector, Head Register Value Incorrect

NOTE:

Any error detected on one controller will cause "data late" and other errors on other disk controllers being exercised by this test since any error will cause the computer to halt.

8.0 Format Program Abstracts

There are three format programs available:

Test 45
Test 47
Test 50

For most purposes, test 45 is most convenient. It is slightly faster than test 47 and can be used for any # of sectors. Test 47 will run in 8K of memory, however the # of sectors on a surface must be evenly divisible by 4. Test 50 is useful only to add sector integrity bits to sector formats.

8.1.1 Test 45 Format Program

This program formats up to 32 sectors (256 words per sector) at each transfer. No operator intervention is required. It is the only format program that uses the Data Transfer routine "DXFER" therefore, if an error should occur, more data is saved than in the other format programs. 16K of memory is required to run this test.

8.1.2 TEST 47: Format Program #1

This program formats the disk without the operator entering any parameters. The number of sectors and the number of data words in each sector are specified in the first portion of the diagnostic. If the operator wishes to format a section or one sector of the disk surface, TEST 50 allows him to do so (Bad Sector bits, Write-Protect, etc.).

8.1.3 TEST 50: Format Program #2 (NON ECC ONLY)

This program is set up so that the user is able to format the entire pack or only one sector if he wishes.

After entering the program through Test Control the test will halt.

If it is desired to format the entire diskpack in the standard form shown below, raise Switch #15 on the front panel and depress CONTINUE. The program will proceed to format the disk.

TABLE OF STANDARD DISK PARAMETERS

(1)	#Sectors	40 (octal)	0-37
(2)	#Heads	5	0-4
(3)	#Tracks	1467 (octal)	0-1466

After the Program Halt as in the above steps, the user enters the formatting parameters in the following manner:

(1) Enter the starting sector number.
(All of these entries are to be set in lower byte of the switch register.)

(2) Depress CONTINUE.

(3) Enter ending sector number.

(4) Depress CONTINUE.

(5) Enter starting surface number.

(6) Depress CONTINUE.

(7) Enter ending surface number.

(8) Depress CONTINUE.

(9) Enter starting track (cylinder) number.

(10) Depress CONTINUE.

(11) Enter ending track number.

(12) Depress CONTINUE.

(13) Enter sector condition information:

(A) Bit 15 = 1 - Bad Sector

(B) Bit 15 = Ø - Good Sector

(14) Depress CONTINUE.

(15) Enter Write-Protect information:

(A) Bit 15 = 1 - Write Protect

(B) Bit 15 = Ø - Non-Write-Protect

(16) Depress CONTINUE.

When the program halts and the location of the halt is the normal location of Test Control, the pack will have been formatted.

Appendix A

Disk Data Transfer Subroutine

1.0 Overview

The Disk Data Transfer Subroutine is the "driver" of the disk diagnostic. This subroutine is used to perform all disk data transfers whether they be read or write in nature.

2.0 Linkage

All parameters defining the data transfer operation to be performed are contained in a control block. The origin address of this control block is contained in RØ.

3.0 Control Block Format

Entry 1 = Unit, Sector, Head Register (DUSH) Image
2 = Bus Address Register (DCAR) Image
3 = Word Count Register (WDCNT) Image
4 = Cylinder Address Register (DCYL) Image
5 = Control & Status Register (DCSR) Image

4.0 Parameter Checks Made

The data transfer routine checks to insure that the disk controller goes busy, an interrupt occurs, no error occurs, and that all end disk parameters are correct.

5.0 Interrupt Mode

All data transfers made by this subroutine utilize interrupts.

6.0 Data Transfer Subtest Count (DTCNT)

A separate data transfer subtest count is maintained in location DTCNT to facilitate error isolation.

7.0 Data Transfer Error Value Image (ERRWRD)

If an error (controller) occurs or is detected at the end of the disk operation, the image of the first error detected is saved in location -ERRWRD-.

8.0 Disk Controller Register Image (ERIMGE & ERRBLK)

The image of the disk controller error register at the time of any controller error is saved in location -ERIMGE-. All Disk Controller Register images are saved in "ERRBLK" in sequential order.

9.0 Subroutine Return Parameters (ERRFLG) Error Flag State

9.1 Control is transferred to the calling routine with or without the general error flag set (ERRFLG), as appropriate.

9.2 If an error is detected, the general error flag ERRFLG is set and control is returned to the calling program.

9.3 Registers RØ, R1, R4, and R5 are used and saved.

9.4 Registers R2 and R3 are not saved and will be returned to the calling program in the event of an Error with the reference and actual data values respectively.

10.0 Data Transfer Subroutine Subtest Listing

The following table shows the possible values of subtest numbers contained in location DTCNT and the corresponding failure conditions.

Subtest/DTCNT Value	Subtest Name/Error Condition
Ø	Controller Not Ready
1	Selected Disk Drive Not Ready
2	Controller Went Busy With No "Go" Bit Cmd.
3	Controller Did Not Go Busy With "Go" Bit Cmd.
4	Interrupt Did Not Occur Before Timeout
5	Controller Hung in "Busy" State
6	Word Count Register Did Not Overflow
7	End Bus Address Register Value Incorrect
10	End Sector Value Incorrect
11	End Head Value Incorrect
12	End Cylinder Address Value Incorrect
13	Controller Error Bit Set

Phoenix 211 Disk Controller Diagnostic Manual

Appendix B

Read, Write, Comparison Subroutines

1.0 This subroutine writes the specified data block onto the specified device at the specified address. The routine then reads the written data back from the disk and compares the data read with the original data written.

2.0 Calling Sequence:

```
JSR    R5,WRC
```

3.0 At the time of the call, RØ must contain the origin address of the control block defining the parameters of the operations to be performed.

4.0 Required Control Block Format is as follows:

```
Entry 1 = Desired DUSH Image
      2 = Blank (Used by WRC for DCAR Image)
      3 = Desired DWDCNT Image
      4 = Desired DCYL Image
      5 = Blank (Used by WRC for DCSR CMD Image)
```

5.0 Return Parameters

5.1 1. Normal return is with general error flag reset.

5.2 2. If an error is detected during any operation, control is transferred to user with the error flag set. In this case, error information is contained in the DXFER or DATCMP error flags as appropriate.

5.3 Write Transfer Error Flag (WRTERR)

If an error is detected during the write data transfer by DXFER, the "WRTERR" location is incremented to a non zero state.

5.4 Read Transfer Error Flag (RDERR)

If an error flag is detected during the read data transfer by DXFER, the "RDERR" location is incremented to a non zero state.

5.5 Data Comparison Error Flag (CMPCNT)

If a data comparison error is detected by the Data Comparison Subroutine "DATCMP":

1.) The "CMPCNT" location contains the comparison word count value at the time the error was detected.

2.) R2 contains the correct or reference dates.

3.) R3 contains the erroneous actual data.

Appendix C

Data Comparison Subroutine

1.0 This subroutine compares data obtained from the disk during any disk read operation in the input data buffer (IDBUFI) with the reference data actually written onto the disk and contained in the output data buffer (ODBUF).

2.0 At entry RØ contains the number of words that are to be compared.

3.0 If an error occurs:

3.1 The error flag is set and control is transferred back to the calling program.

3.2 The "CMPCHT" count location contains the comparison word count value when the error was detected.

3.3 R2 contains the correct or reference data value.

3.4 R3 contains the actual data value.

Appendix D

Individual Test Select Switch Register Values

Test No.	Test Name	Test Select Value
Ø	Register Initialization	Ø
1	Register Load and Read	401
2	Register Sliding Ones	1002
3	Register Sliding Zero	1403
4	Register Error and Status Register	2004
5	Unit Select	2405
6	Seeking	3006
7	Interrupt Logic	3407
10	One Word Write	4010
11	One Sector Write	4411
12	One Sector Read	5012
13	Incrementing Word Count Write	5413
14	One Sector Write, Read, and Compare	6014
15	Write, Read, Compare	6415
16	Write Protect	7016
17	Header Override	7417
20	Write Protect Override	10020
21	Header Write Command	10421
22	CRC/Header Test	11022
23	Bad Sector Flag Test	11423
24	RT & Test	12Ø24
25	Implied Seek Test	12425

Test No.	Test Name	Test Select Value
26	Converge/diverge seeking test	13026
27	Mem. Ext. - Buss Timeout Test	13427
30	Converge/diverge write/read test	14030
31	Disk Data Test (Entere Disk Write/Read)	14431
32	Disk Reliability Read	15032
33	Multiple CPU Exercise Test	15433

Phoenix 211 Controller Diagnostic Manual

Appendix E

Select and Seek Subroutine

1.0 Objective

This subroutine selects a disk drive based on the unit number contained in RØ at the time of the program call, and then issues a seek or RTZ command to the selected disk.

2.0 Usage

This subroutine is used in the Overlapped Seek Tests (TN36 & TN37).

3.0 Calling Sequence

```
(X)         JSR       R5, SELASK          ;Program Call
(X+2)        Word CYL                     ;Cylinder Address
(X+4)        Word CMD                     ;Seek or RTZ CMD
```

RØ - Disk Drive Number

4.0 Error Conditions

4.1 The general Error flag is set if the selected disk drive is not ready or if the associated seeking status flag is not set when the disk is commanded to seek or RTZ.

4.2 This subroutine increments the subtest count location associated with the specified disk drive in the SKCNT Table.

Appendix F

Seek End Check Subroutine

1.0 Objective

This subroutine is used to verify:

1.1 That all of the drives commanded to seek did generate a seek done interrupt.

1.2 That the right number of interrupts were received.

1.3 That the seek ident value generated by each drive was correct.

2.0 Usage

This Subroutine is used in tests 36 & 37, the overlapped seek tests.

3.0 Calling Sequence

```
JSR           R5, SENCHK

        RØ - Number of Disk Drives
```

4.0 Error Conditions

4.1 If an error is detected the general error flag is set.

4.2 The disk drive subtest count located in the "SKCNT" Table is incremented if the Seek Identification Field value for that drive is correct.

4.3 R4 is incremented if the correct number of seek complete interrupts was received.

Phoenix 211 Controller Diagnostic Manual

Appendix G

Seek Interrupt Service Routine

1.0 Objective

This interrupt service routine is used to verify that all of the seek parameters are correct when seek done interrupts are received at the end of commanded seek operations.

This interrupt service routine specifically

1. Extracts the unit No of the interrupting disk drive.
2. Verifies that the drive seeking status flag is reset.
3. Verifies that the dirve seek done status flag is set.
4. Verifies that the seek done summary flag is set.
5. Selects the interrupting disk drive.
6. Verifies that there is no controller errors.
7. Verifies that the interrupting drive is ready.
8. Issues a NOP CMD to drive to clear out seek done flag.
9. Sets interrupt flag in table location determined by the seek ident value at time of interrupt.

2.0 Usage

This interrupt service routine is used for tests 36 & 37, the overlapped seek logic tests.

3.0 Error Conditions

3.1 The subtest count value in the SKCNT Table for the interrupting disk drive is incremented for each subtest conducted by this routine.

3.2 If any error is detected, the general error flag is set.

Appendix H

Format Subroutine

1.0 Objective

This subroutine establishes a known one sector (256 words) data pattern in the Output Data Buffer used for most Write disk data transfer operations.

2.0 Resultant Sector Buffer Format

A 256 word buffer will be formatted per the following by this subroutine.

Buffer Word Number	Content
1 - 64	Decreasing binary count value starting at 100_8 and going to 1.
65 - 128	(177400) Low order byte = Ø, High Order Byte = all 1's
129 - 192	(125125) Low order byte = 125, High Order Byte = 252
193 - 208	Rotating Sliding zero Value Sliding from Bit 5 to Bit 15
209 - 224	Rotating Sliding One Value from Bit Ø to Bit 15
225 - 240	Rotating Sliding zero Value sliding from Bit Ø to Bit 15
241 - 256	Rotating Sliding One Value sliding from Bit Ø to Bit 15

3.0 Calling Sequence

```
JSR        R5, FORMAT
```

R1 = Origin Address of Buffer to be Formatted

4.0 Error Conditions

None

Phoenix 211 Controller Diagnostic Manual

Appendix I

Multiple Controller Register Address Setup Subroutine

1.0 Objective

This subroutine computes the proper base register addresses for the disk controller defined by the value on top of the stack after the calling sequence. The computer addresses are then loaded into the controller register address table.

2.0 Usage

This subroutine is used by the Simultaneous Multiple Controller Exerciser Test.

3.0 Calling Sequence

```
            JSR  R4, ADRSTP

   R4 = disk controller number     (1,2,3, etc.)
```

4.0 Error Conditions

None

Phoenix 211 Controller Diagnostic Manual

Appendix J

Disk Transfer Start Subroutine

1.0 Objective

Initiate all data transfers performed during the Simultaneous Multiple Disk Controller Test.

2.0 Calling Sequence

JSR R5, XFRST

1.) R5 = specified disk controller

2.) RØ must contain origin address of control block at entry.

3.) Parameters defining data transfers are contained in a control block of origin address X per the following:

(X) = Desired DUSH Image
(X+2) = Desired DBAR Image
(X+4) = Desired DWCNT Image
(X+6) = Desired DCYL Image
(X+8) = Desired DCSR Image

3.0 Error Conditions

3.1 Controller Not Ready

3.2 Selected Disk Drive Not Ready

3.3 If an error occurs, the general error flag is set and R2 contains the reference value while R3 contains the actual value.

Appendix K

Multiple Controller Test Interrupt Service Routine

1.0 Objective

This routine services all of the transfer complete interrupts generated by all of the disk controllers exercised in the Multiple Disk Controller Exerciser Test. This service routine specifically:

1. properly sets up the controller register addresses for the interrupting disk controller;
2. verifies proper end controller parameter values;
3. verifies no error occurrence;
4. initiates a new data transfer on the selected controller;
5. exits via an RTl instruction.

2.0 Error Conditions

If an error occurs:

2.1 The associated disk controller indent is saved in location "CERRNO".

2.2 The selected controller register images are saved in "ERRBLK".

2.3 The error condition is identified by the subtest number saved in location "ENDCT".
ENDCT subtest values and associated error conditions are as follows:

ENDCT VALUE	FUNCTIONAL TEST/ERROR CONDITION
Ø	Controller Error Condition at Interrupt
1	Word Count Register did not overflow
2	End Bus Address Register Value incorrect
3	End Cylinder Address Register value incorrect
4	End Unit, Sector, Head Register value incorrect

Appendix L

Typical Register Address Assignments * - Phoenix 211

CONTROL AND STATUS REGISTER 164000

Error Sum.	Seek Inhibit	Memory Extension		DSE	DSL	SV+	SV	Done (Ready)	Interrupt Enable	Command					Go
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

UNIT-SECTOR-HEAD REGISTER 164002

0	0	Unit No.		Head					Sector						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

BUSS ADDRESS REGISTER 164004

Buss Address															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

WORD COUNT REGISTER 164006

Word Count															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CYLINDER ADDRESS REGISTER 164010

0	0	0	0	0	0	Cylinder Address									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DISK STATUS REGISTER 164012 (READ ONLY)

Drive Ready	Disk Port Busy	Disk Write Protected		Drive 3 Seeking	Drive 2 Seeking	Drive 1 Seeking	Drive 0 Seeking	Seek Done	*Request Flag Status	Seek 3 Done	Seek 2 Done	Seek 1 Done	Seek 0 Done	Drive Done	Seek ID No.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ERROR STATUS REGISTER 164014 (READ ONLY)

Buss Timeout Error	Write Check Error	Bad Sector Found	Disk Drive Not Ready	Fault Error	Seek Error	Time Out Error	CRC Error	Data Late	Non Existent Sector	Non Existent Cylinder	Non Existent Head	Overrun Error	Sector Write Protected	Sector/ Head Header Compare Error	Track Header Compare Error
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

* See Configuration Chart for actual Register Addresses

Appendix M

DECIMAL TO OCTAL CONVERSION TABLE

DECIMAL	OCTAL	DECIMAL	OCTAL
Ø	Ø	26	32
1	1	27	33
2	2	28	34
3	3	29	35
4	4	30	36
5	5	31	37
6	6	32	40
7	7	33	41
8	10	34	42
9	11	35	43
10	12	36	44
11	13	37	45
12	14	38	46
13	15	39	47
14	16	40	50
15	17	41	51
16	20	42	52
17	21	43	53
18	22	44	54
19	23	45	55
20	24	46	56
21	25	47	57
22	26	48	60
23	27	49	61
24	30	50	62
25	31		

PHOENIX 211 DISK

DIAGNOSTIC MONITOR MANUAL

(Supplement To Diagnostic Manual 1043-05/06)

2/21/78

TABLE OF CONTENTS

SECTION

1.0 General Description

The Phoenix 211 Diagnostic Monitor is used to facilitate operation of the Phoenix Disk diagnostics. The monitor allows the operator to change all disk and controller parameter selections as well as test sequences via a terminal. Phoenix 211's delivered prior to March 1978 do not have diagnostics with this feature. The diagnostic tests have not been modified and the monitor version remains operable via the computer switch register.

2. Operation

2.1 Active Keyboard Operation

2.1.1 The Phoenix 211 diagnostic should be loaded via the normal procedure described in the diagnostic manual. Startup is by loading the starting address (3000) and depressing start.

2.1.2 The program will initialize the keyboard and printer (keyboard printer), print the test program heading and request the operator to enter the date.

2.1.3 After the date has been properly entered, a prompt will appear and the operator can now issue one of 13 different commands. The commands are described in detail in section 4.

2.1.4 During a test sequence the operator can produce a test halt by typing a control C. The monitor will only accept an interrupt generated by a control C character.

When the monitor detects a control C during testing it saves the CPU registers and enters The Command Line Interpreter. At this point, the operator can examine the test progress and 211 controller registers if he wishes. If the operator desires to continue testing, without re-initializing the pass count, the Continue "CO", command can be used.

3. Command Table

Table 3.1 is the valid command table for the Phoenix 211 Monitor. Typing an "H" (CR) after startup will produce this table on the keyboard printer.

Table 3.1 Monitor Commands

	Command Nmemonics	Description
A.	G	Go command - starts testing
B.	DK	Disk Parameters
C.	TS	Test Sequence
D.	FT	Controller parameters
E.	PR	Test progress report
F.	DA	Date
G.	SC	Scope Loop Parameters
H.	PR	Controller Register Read
I.	H	Help
J.	TE	Test Error Conditions
K.	FO	Format
L.	CO	Continue Testing

4.0 Individual Command Descriptions

4.1 Command Format

4.1.1 All command requests are terminated by a carriage return. (CR).

4.1.2 All numerical responses must be in octal

4.1.3 A response can be defaulted by typing only (CR) to a query.

4.1.4 A parameter select command can be terminated by typing an E (CR).

4.1.5 The default answer for numerical entries is outputed during selection.

4.1.6 The default answer for yes - no questions is the last selected or NO initially.

4.2 Command Descriptions

4.2.1 GO command

Mnemonic: G

Description: The "go" command starts testing at the specified test and drive sequence start point.

After executing this command the monitor is disabled

4.2.2 Disk Command

Mnemonic: DK

Description: The Disk command allows 4 disk oriented parameters to be loaded.

1. Maximum sector
2. Maximum head
3. Maximum cylinder
4. Words per sector

4.2.3 Test Sequence Command

Mnemonic: TS

Description: The test command allows the operator to define the test sequence he desires: there are 8 parameters.

1. Starting test number
2. Ending test number
3. Starting drive number
4. Ending drive number
5. Continuous testing mode
6. Repeat on error mode
7. Inhibit pass printout
8. Inhibit error printout

4.2.4 Controller Command

Mnemonics: FT

Description: The controller command allows selection of 5 controller oriented parameters.

1. DCSR or base address
2. Interrupt vector
3. ECC option
4. Multiple CPU option
5. Auto-switch option

4.2.5 Progress Report

Mnemonic: PR

Description: The progress command outputs current testing values. This command is primarily used when a control "C" has been typed during testing. If the operator halts the CPU and restarts the monitor, the values will still be valid. i.e. current test number and pass count are only reset during the "go" command. 5 messages are outputted:

1. Date
2. Pass count
3. Starting test number
4. Ending test number
5. Current test number
6. Error flag

4.2.6 Date Command

Mnemonic: DA

Description: The date command requests the operator to enter the date. It must be in the form:

DD-MMM-YY

4.2.7 Scope Mode

Mnemonic: SC

Description: The Scope command allows four of the five possible registers to be selected for a scope loop. The code for the actual scope loop is the subroutine labeled "DXFER" which is described in Appendix A of the Diagnostic manual. 1043/05/06

This subroutine is interrupt driven and the monitor is lost after executing the Scope command.

The disk address parameters must be loaded in image fom i.e., a write command = 6. For aid in selecting parameters refer to the Phoenix 211 Programming Reference Manual.

4.2.8 Register Read Command

Mnemonic: RR

Description: The Register Read command outputs the contents of the seven (or 9, if Error Correction Controller) Phoenix 211 registers.

4.2.9 Help Command

Mnemonic: H

Description: The Help command lists the command table and explains a few of the operating procedures.

4.2.10 Test Error Conditons Command

Mnemonic: TE

Description: This command prints the Error conditions that were reported upon failure of testing previously by the monitor. This command is useful when error conditions in the 211 controller have been cleared by an I/O reset pulse. (Start Key). The format of this printout is shown below.

```
Failing drive number = ##
Failing test number  = ##
Reference data       = ##
Actual data          = ##
Subtest number       = ##
```

If the error is a data error the output will be a message saying "Data Error"

If any other error occurs the controller register will be outputed.

4.2.11 Format Command

Mnemonic: FO

Description: The format command allows you to format any drive (0→3) however only one drive for each command call. The command responses* are below:

Disk unit to be formatted ### Ø (CR)
Are you sure??? Y or N Y (CR)

4.2.12 Continue Command

Mnemonic: CO

Description: The Continue command is useful in only one case: after testing has been halted using a control C. A Return to testing will be generated if this condition is satisfied, all other uses of the continue command are illegal and may result in destruction of memory contents.

* Operator responses are underlined

5.0 Disabling The Monitor

5.1 The Phoenix Monitor can be disabled by depositing a Ø into location "TTYPUT" (memory address 1104_8) after the diagnostic has been loaded into memory.

The program can now be run via the CPU's switch register or via the console emulator.

5.2 Non Terminal Operation

5.2.1 Instructions for operating the Phoenix diagnostics via the switch register are located both on the front section of the Diagnostic's assembly listing and in the Diagnostic Manual, #1043/05/06.

```
;
;
;
;
;	PHOENIX 211 DISK DIAGNOSTIC
;
;	****************************************
;
;	COPYRIGHT 1975
;	 XYLOGICS OEM COMPONENTS GROUP
;	 42 THIRD AVENUE
;	 BURLINGTON MASS 01803
;
;GERARD JOHNSON JR.
;
;617-272-8140
;
;	REV E	19-APR-78
;
;
;
;	TABLE OF CONTENTS
;	******************************
;
;
;	SECTION 1	OPERATING INSTRUCTIONS
;
;	SECTION 2	OPTIONAL MODES OF OPERATION
;
;			2.1  ECC CONTROLLERS
;			2.2  AUTO SWITCH OPTION
;			2.3  MULTPLE COMPUTER PORT OPTION
;			2.4  NON CONSOLE MODE(NO SWITCH REG.)
;			2.5  NO TEST INPUT MODE
;			2.6  FORMATTING INSTRUCTIONS
;
;	SECTION 3	REGISTER ASSIGNMENT TABLE
;
;	SECTION 4	DISK/CONTROLLER ASSIGNMENT TABLE
;
;	SECTION 5	MIXED CONSTANT TABLE
;
;	SECTION 6	TEST PARAMETER CONSTANT TABLE
;
;	SECTION 7	TEST ORIGIN ADDRESS TABLES
;
;	SECTION 8	REGISTER MASKS/PATTERN TABLES
;
;	SECTION 9	ERROR HALT REGISTER IMAGE TABLE
;
;	SECTION 10	TEST CODE
;
;
;
;
;
;
```

```
;
;       8. DEPRESS THE CONTINUE SWITCH ON THE COMPUTER CONSOLE.
;
;       9. THE COMPUTER WILL THEN EXECUTE ALL TESTS SPECIFIED
;          ON ALL DISKS SPECIFIED. IT WILL ALSO STOP IF
;          ANY ERROR IS ENCOUNTERED UNLESS THE FOLLOWING
;          SWITCH SETTINGS ARE USED:
;
;
;
;       (A). BIT 15 SWITCH RAISED DURING STEP 7 WILL
;       INHIBIT PROGRAM HALT IF ANY ERROR OCCURS.THE
;       PROGRAM CONTINUES TESTING STARTING WITH THE
;       NEXT CONSECUTIVE TEST.IF THE USER IS REPEATING
;       ONE OR MORE TESTS THE PROGRAM ATTEMPT THAT SEQUENCE
;       INDEFINITLY INCREMENTING LOCATION ERROR EACH TIME
;       AN ERROR OCCURS.
;
;
;       (B). BIT 7 SWITCH RAISED DURING STEP 7 WILL
;       INHIBIT PROGRAM HALT WHEN A DATA COMPARE ERROR
;       IS ENCOUNTERED.LOCATION ERRCNT IS INCREMENTED
;       EACH TIME A COMPARE IS BAD.IF SWITCH 15 IS LEFT OFF
;       IN THIS MODE THE PROGAM WILL STOP ON A CONTROLLER
;       ERROR(ERRORS FLAGGED BY REGISTER DERR).
;
;
;
;       (C). BIT 14 SWITCH RAISED DURING STEP
;       7 WILL ALLOW THE PHOENIX 211 TO RETRY CERTAIN SOFT
;       ERRORS.THESE SOFT  ERRORS CAN BE DATA OR CONTROLLER
;       IN NATURE IE. AHEADER COMPARE ERROR WOULD BE
;       CONSIDERED A SOFT CONTROLLER ERROR.IF AFTER 8
;       ATTEMPTS ON A DATA ERROR OR 4  ON A CONTROLLER ERROR
;       THE ERROR STILL EXISTS IT IS RECLASSIFIED TO A HARD ERROR
;       AND THE PROGRAM WILL HALT(IF BITS 7AND 15 ARE OFF).
;         EACH TIME A SOFT DATA ERROR IS ENCOUNTERED LOCATION
;       ERRCNT IS INCREMENTED.THE  COUNT FOR SOFT CONTROLLER
;       ERRORS IS KEPT IN LOCATION SOFERR.WHEN A HARD DATA ERROR
;       IS FOUND LOCATION HERCNT IS USED RATHER THAN THE
;       GENERAL ERROR FLAG ERRFLG SO THAT THE MODE USING
;       SWITCH 7 STILL OPERATES.
;
;
;       SWITCH SETTINGS
;    --------------------------
;
;
;       15      7       14                    EFFECT
;    -----------------------------------------------------------
;       0       0       0       PROGRAM WILL HALT ON ANY ERROR
;
;       0       0       1       PROGRAM WILL RETRY SOFT ERRORS AND
;                               STOP ON ANY HARD ERRORS
;       0       1       0       PROGRAM WILL  NO STOP ON ANY DATA ERRORS
;                               BUT WILL STOP ON CONTROLLER ERRORS
;       0       1       1       PROGRAM WILL STOP ONLY ON HARD ERORS
```

```
                                  ;
                                  ;     1       0       0       PROGRAM WILL NOT STOP ON ANY
                                  ;                             ERRORS NO RETRY OR DATA ERROR DISCRIMINATION
                                  ;     1       0       1       PROGRAM WILL RETRY AND NOT STOP ON
                                  ;                             ANY ERRORS.
                                  ;     1       1       0       NO RETRY NO STOP ON ANY ERRORS
                                  ;
                                  ;     1       1       1       NO STOP ON ANY ERRORS AND INCREMENTING
                                  ;                             DATA ERROR COUNTER
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;     10. IF THE COMPUTER HALTS FOR AN ERROR:
                                  ;             R0 CONTAINS THE DISK DRIVE NUMBER
                                  ;             R1 CONTAINS THE TEST NUMBER
                                  ;             R2 CONTAINS THE CORRECT OR REFERENCE DATA
                                  ;             R3 CONTAINS THE  ACTUAL DATA RECEIVED
                                  ;             R4 CONTAINS THE  SUBTEST NUMBER
                                  ;             R5 CONTAINS THE ADDRESS WHERE THE ERROR WAS
                                  ;                DETECTED.
                                  ;
                                  ;     11.     TEST STARTING ADDRESS IS "START"
                                  ;
000002  003000                          .WORD START
                                  ;
                                  ;     12.     ERROR HALT LOCATION IS "STOP8"
                                  ;
000004  003474                          .WORD STOP8
                                  ;
                                  ;
                                  ;     13.     NORMAL TEST HALT IS"HALT8"
                                  ;
000006  003776                          .WORD HALT8
                                  ;
                                  ;     14.     AFTER HAVING SET UP PARAMETERS,THE OPERATOR
                                  ;             CAN RESTART AT ADDRESS TNINTA AND THE PROGAM
                                  ;             WILL START RUNNING THE TEST SEQUENCE
                                  ;             PREVIOUSLY SET UP EITHER THROUGH THE
                                  ;             SWITCH REGISTER OR BY DEPOSITING INTO
                                  ;             THE START AND END TEST LOCATIONS.
                                  ;
000010  004000                                  .WORD TNINTA
                                  ;
                                  ;
                                  ;
                        ;***********************************************************************
                        ;***********************************************************************
                                  ;
                                  ;
                                  ;
                                  ;
                        ;  SECTION 2
                                  ;
                                  ; 2.1   ERROR CORRECTION CONTROLLERS
```

```
                              ;    ************************************
                              ;
                              ;       211 CONTROLLERS WITH  THE ECC OPTION REQUIRE
                              ;    A SEPARATE TEST SEQUENCE DUE TO
                              ;    DIFFERENT HEADER AND CRC WORD CONFIGURATIONS
                              ;       TO OPERATE THIS PROGRAM ON SUCH A SYSTEM THE
                              ;    OPERATOR MUST SET A FLAG  CALLED "ECC"
                              ;    THE WORD BELOW CONTAINS THE MEMORY ADDRESS OF "ECC"
                              ;    IF ECC IS INCORPOATED DEPOSIT A "1" INTO "ECC".
                              ;
                              ;       THERE ARE TWO TEST TABLES ON THIS LISTING
                              ;    AND ONE OF THEM IS DEDICATED TO ECC.
000012  001074                     .WORD ECC
                              ;
                              ;
                              ;
                              ;
                              ; 2.2 AUTO-SWITCH OPTION
                              ;    14.SYSTEMS EQUIPTED WITH PHOENIX AUTO SWITCH
                              ;    OPTION BOARD CAN BE TESTED USING LOCATION AUTOSW:
                              ;            (A). BEFORE SYSTEM POWER UP,INSTALL DATA
                              ;                 CABLE INTO CONNECTOR ON PHOENIX 211
                              ;                  BOARD #2(CONSULT AUTO SWITCH MANUAL)
                              ;            (B).  POWER UP SYSTEM AND AFTER LOADING
                              ;            (B).  POWER UP SYSTEM AND LOAD 211 DIAGNOSTIC
                              ;
                              ;            (C). START NORMAL OPERATING PROCEDURES
                              ;                 AND DURING STEP 7 RAISE CONSOLE SWITCH 13
                              ;
                              ;            (D). IF RUNNING IN A NO-INPUT MODE OR NONCONSOLE
                              ;                    MODE DEPOSIT A "1" IN LOCATION "AUTOSW"
                              ;
000014  001100                             .WORD AUTOSW
                              ;
                              ;
                              ;
                              ;**************************************************
                              ;
                              ;
                              ;
                              ; 2.3 MULTIPLE COMPUTER PORT OPTION
                              ;  ===================================
                              ;
                              ;
                              ;      TO USE THIS PROGRAM TO TEST PHOENIX 211'S
                              ;  WITH THE MULTIPLE COMPUTER PORT OPTION,DEPOSIT A "1" INTO
                              ;  CORE LOCATION "MULCPU" AND USE NORMAL OPERATING PROCEDURES.
                              ;
                              ;
                              ;
                              ; 2.3.1          SINGLE CPU TESTING
                              ;
                              ;  IN MULTIPLE CPU MODE,WITH THE EXCEPTION OF TEST 33,ONLY ONE CPU CAN
                              ;       BE EXERCISED AT A TIME.THIS IS DUE TO THE DURATION OF
                              ;       TIME THAT SOME TESTS REMAIN IN CONTROL OF THE FORMATTER.
                              ;       IN NORMAL OPERATING SYSTEM ENVIRONMENTS THE CPU WOULD BE
                              ;       RELINQUISHING POSSESSION OF THE FORMATTER WHEN
```

```
                        ;       NO DISK OPERATIONS ARE QUEUED. DURING THESE DIAGNOSTICS
                        ;       THE FORMATTER IS NEVER FORCED IDLE THEREFORE A COMPETING
                        ;       CPU RUNNING 211 DIAGNOSTICS WOULD TIME OUT AS A RESULT
                        ;       OF DIAG. REQUEST LOOPS.
                        ;
                        ;
000016  001076                          .WORD MULCPU
                        ;
                        ;
                        ; 2.3.2         SIMULTANEOUSLY RUNNING CPU'S
                        ;
                        ;
                        ;       TO EXCERCISE TWO OR MORE CPU'S RUNNING ON
                        ;       ONE 200 FORMATTER TEST 33 HAS BEEN ADDED TO
                        ;       THE DIAGNOSTIC PROCEDURE.THIS EXERCISE
                        ;       ENABLES THE OPERATOR TO DETECT ANY CONFLICT
                        ;       THAT MIGHT OCCUR BETWEEN REQUESTING CPU'S.
                        ;
                        ;         A DISCRIPTION OF THE TEST FORMAT IS LOCATED
                        ;       PRECEEDING THE ACTUAL TEST CODE.
                        ;
                        ;
                        ; =========================================
                        ;
                        ; 2.4 NON CONSOLE MODE
                        ;  =======================
                        ;
                        ;
                        ;       1.TO OPERATE DIAGNOSTIC WITHOUT  A SWITCH REGISTER
                        ;       DEPOSIT A  NON ZERO NUMBER INTO LOCATION "NONCON"
                        ;       AND ENTER TEST PARAMETERS INTO APPROPRIATE LOCS.
                        ;       ;STARTING WITH LOCATION "STPTN".
                        ;
                        ;
                        ;
000020  001244                          .WORD NONCON
                        ;
                        ;
                        ;
                        ; 2.5 NO TEST INPUT MODE
                        ;  ==================================
                        ;
                        ;
                        ;
                        ;       TO OPERATE TESTS WITHOUT ENTERING NEW TEST DATA
                        ;       EACH PASS OR STARTUP,DEPOSIT A "1" IN LOCATION
                        ;       "NINPUT" AND START OPERATION AT LOCATION "START"
                        ;
                        ;       THE TEST(S) AND DISK(S) RUN WILL BE THOSE
                        ;       ENTERED IN LOCATIONS "STPTN" THROUGH "STRDSK".
                        ;
                        ;
000022  001242                          .WORD NINPUT
                        ;
                        ;
                        ;  2.6  FORMATTING INSTRUCTIONS
                        ; ====================================
```

```
                        ;
                        ;       1.  ALL CONTROLLERS CAN USE FORMATTING PROGRAM - TEST 45
                        ;         THIS PROGRAM RUNS IN 16K OF MEMORY.
                        ;
                        ;       2.   ALL CONTROLLERS THAT HAVE A TOTAL SECTOR COUNT THAT IS
                        ;            DIVISIBLE BY 4 CAN USE FORMATTING PROGRAM - TEST 47
                        ;            THIS PROGRAM RUNS IN 8K OF MEMORY.
                        ;
                        ;       3.   ALL NON ECC CONTROLLERS THAT WISH TO ADD SECTOR INTEGRITY
                        ;            BITS TO CERTAIN OR ALL SECTORS CAN USE FORMATTING PROGRAM
                        ;            TEST 50 . THIS PROGRAM RUNS IN 8K OF MEMORY.
                        ;
                        ;
                        ;
                        .TITLE  SMDIA.MAC
                        ;
                        ;
         000000         R0=%0           ;
         000001         R1=%1           ;
         000002         R2=%2           ;
         000003         R3=%3           ;
         000004         R4=%4           ;
         000005         R5=%5           ;
         000006         SP=%6           ;
         000007         PC=%7           ;
         177570         CSWR=177570     ;CONSOLE SWITCH REGISTER
         177776         PS=177776       ;PROGRAM STATUS REGISTER
         000012         LF=12           ;LINE FEED
         000015         CR=15           ;CARRIAGE RETURN
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;       ;INTERRUPT VECTOR ASSIGNMENTS
                        ;
                        ;
         000000         .ASECT          ;DEFINE ASSEMBLY AS ABSOLUTE
                        .ENABL AMA      ;DEFINE ABSOLUTE ADDRESSING
 000000  000002         .+2             ;ERRONEOUS INTERRUPT VECTOR TRAP
 000002  000000         .WORD 0         ;H
         000004         .=4             ;BUS ERROR TRAP LOCATION
 000004  000006         .WORD 6         ;
 000006  000000         .WORD 0         ; BUS ERROR WILL HALT AT 6
 000010  000012         .WORD 12        ;ILLEGAL INSTRUCTION TRAP
 000012  000000         .WORD 0         ; ILLEGAL INSTRUCTION HALTS AT 12
 000014  000016         .+2             ;ERRONEOUS INTERRUPT VECTOR TRAP
 000016  000000         .WORD 0         ;
 000020  000022         .+2             ;ERRONEOUS INTERRUPT VECTOR TRAP

                        ;
                        ;
```

```
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
        001000                          .=1000          ;REGISTER ASSIGNMENTS WILL START HERE
                                        ;
                                        ;
                                 ; SECTION 3  REGISTER ASSIGNMENT TABLE
                                        ;    ******************************
                                        ;
                                        ;
 001000 164000                  DCSR:   .WORD 164000    ;CONTROL AND STATUS REGISTER
 001002 164002                  DUSH:   .WORD 164002    ;UNIT,SECTOR HEAD REGISTER
 001004 164004                  DCAR:   .WORD 164004    ;CORE ADDRESS REGISTER
 001006 164006                  DWCNT:  .WORD 164006    ;WORD COUNT REGISTER
 001010 164010                  DCYL:   .WORD 164010    ;CYLINDER ADDRESS REGISTER
 001012 164012                  DSTAT:  .WORD 164012    ;DISK STAUS REGISTER
 001014 164014                  DERR:   .WORD 164014    ;CONTROLLER ERROR REGISTER
 001016 164016                  ECCPB:  .WORD 164016    ;ECC BIT POSITION REGISTER
 001020 164020                  ECCPW:  .WORD 164020    ;ECC WORD PATTERN REGISTER
 001022 000000                          .WORD 0         ;BLANK
 001024 000000                          .WORD 0         ;BLANK
 001026 000000                          .WORD 0         ;BLANK
 001030 000000                          .WORD 0         ;BLANK
 001032 000000                          .WORD 0         ;BLANK
 001034 164036                  DSCAR:  .WORD 164036    ;STRIP CORE ADDRESS REGISTER
 001036 164600                  AUTO:   .WORD 164600    ;AUTO  SWITCH REGISTER
                                        ;
                                        ;
                                        ;
                                ; SECTION 4 DISK/CONTROLLER DEFINITIONS
                                        ;
 001040 000037                  MAXSEC: .WORD 37        ;MAXIMUM SECTOR ADDRESS
 001042 001000                  MAXHD:  .WORD 1000      ;MAXIMUM HEAD ADDRESS JUSTIFIED
 001044 001466                  MAXCYL: .WORD 1466      ;MAXIMUM CYLINDER ADDRESS
 001046 177400                  WPSEC:  .WORD 177400    ;NUMBER OF WORDS PER SECTOR(2'S COMPLEMENT
 001050 000400                  POSWP:  .WORD 400       ;NUMBER OF WORDS PER SECTOR
 001052 017304                  INTSAV: .WORD DSKDN     ;INTERRUPT SERVICE ROUTINE ADDRESSS
 001054 001037                  RDUSH:  .WORD 1037      ;MAXIMUM DUSH ADDRESS
 001056 001466                  RWCYL:  .WORD 1466      ;MAXIMUM CYLINDER ADDRESSE
 001060 000004                  MAXHED: .WORD 4         ;MAXIMUM HEAD VALUE
 001062 000000                  STRCNT: .WORD 0         ;STRIP COUNT
 001064 000000                  OLDHDR: .WORD 0         ;FLAG OLD HEADER ROMS
 001066 000000                  MICRO:  .WORD 0         ;NONCOMPATIBLE DISK DRIVE
 001070 164000                  DCSRA:  .WORD 164000    ; DCSR ADDRESS
 001072 000270                  INTVEC: .WORD 270       ;INTERRUPT VECTOR ADDRESS
 001074 000000                  ECC:    .WORD 0         ;FLAG ECC CONTROLLER
 001076 000000                  MULCPU: .WORD 0         ;MULTIPLE CPU FLAG
 001100 000000                  AUTOSW: .WORD 0         ;AUTOSWITCH FLAG
 001102 000000                  MCPU:   .WORD 0         ;MULTIPLE CPU SELECT BIT
 001104 000001                  TTYPUT: .WORD 1         ;KEYBOARD ACTIVE FLAG
 001106 000000                  MDRIVE: .WORD 0         ;SIMULTANEOUS DRIVE SWITCH
 001110 000000                  TDATE:  .WORD 0         ;DATE CODE FOR DIAGNOSTIC REV
                                                        ;UPPER BYTE IS WEEK (0-53)    ?
                                                        ;LOWER BYTE IS YEAR
                                        ;
```

```
                                        ;
                                        ;  SEEK INTERRUPT SUBTEST TABLE
                                        ;
                                        ;       A SEPERATE SEEK INTERRUPT SERVICE SUBTEST
                                        ;   COUNT IS MAINTAINED FOR EACH OF THE DISK DRIVES TO
                                        ;   BE EXERCISED.
                                        ;
                                        ;
001112  000000              SKCNT:      .WORD 0         ;DRIVE 0 INTERRUPT SUBTEST COUNT
001114  000000                          .WORD 0         ;DRIVE 1 INTERRUPT SUBTEST COUNT
001116  000000                          .WORD 0         ;DRIVE 2 INTERRUPT SUBTEST COUNT
001120  000000                          .WORD 0         ;DRIVE 3 INTERRUPT SUBTEST COUNT
                                        ;
                                        ;       DISK DRIVE OPERATION COMPLETE INTERRUPT FLAG TABLE
                                        ;
                                        ;
                                        ; THE CONTROLLER WILL GENERATE AN INTERRUPT AT THE END OF
                                        ; EVERY OPERATION IF THE INTERRUPT ENABLE
                                        ; BIT IS SET. THE INTERRUPT SERVICE ROUTINE WILL STORE THE
                                        ; INTERRUPTING DISK DRIVE NUMBER IN THE CORESPONDING TABLE
                                        ; LOCATION ASSIGNED TO EACH DISK DRIVE. THE TABLE CONTENTS ARE
                                        ; THEN CHECKED TO VERIFY THAT THE INTERRUPT OCCURRED AND
                                        ; THAT THE CORRECT DISK DRIVE NUMBER WAS READ.
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
001122  000000              INTTBL:     .WORD 0         ;DRIVE 1 INTERRUPT LOCATION
001124  000000                          .WORD 0         ;       2
001126  000000                          .WORD 0         ;       3
001130  000000                          .WORD 0         ;       4
                                        ;
                                        ;
                                        ;
                            ; SECTION 5 MIXED CONSTANT TABLE
                                        ;
                                        ; THIS TABLE CONTAINS VARIOUS CONSTANTS USED BY THE DIAGNOSTIC
                                        ; ROUTINES.
                                        ;
                                        ;
001132  000000              SOFTRY:     .WORD 0         ;RETRY FLAG IN READ DATA TEST
001134  000000              CMPCNT:     .WORD 0         ;DATA COMPARISON WORD COUNT USED BY DATCMP
001136  000000              CURSEC:     .WORD 0         ;CURRENT SECTOR IMAGE
001140  000000              CURHD:      .WORD 0         ;CURRENT HEAD IMAGE
001142  000000              CURCYL:     .WORD 0         ;CURRENT CYLINDER IMAGE
001144  000000              DKEND:      .WORD 0         ;END OF DISK FLAG
001146  000000              ENDFLG:     .WORD 0         ;END PARAMETER ROUTINE EXIT FLAG
001150  000000              INTFLG:     .WORD 0         ;GENERAL INTERRUPT FLAG
001152  000000              DTCNT:      .WORD 0         ;DATA XFER ROUTINE SUBTEST COUNT
001154  000000              IPFLG:      .WORD 0         ;INTERRUPT PRIORIRY FLAG(TEST 37)
001156  000000              ICNT:       .WORD 0         ;SEEK INTERRUPT COUNT(TEST 36,37)
001160  000000              ERRWRD:     .WORD 0         ;IDENT OF FIRST ERROR BIT IN ERROR REGISTER
001162  000000              ENDSEC:     .WORD 0         ;ENDING SECTOR VALUE FOR DISK OPERATION
001164  000000              ENDHD:      .WORD 0         ;ENDING HEAD VALUE
001166  000000              ENDCYL:     .WORD 0         ;ENDING CYLINDER VALUE
```

```
001170  000000          ERIMGE: .WORD 0         ;ERROR IMAGE WORD
001172  000000          RDERR:  .WORD 0         ;READ ERROR FLAG USED BY WRC
001174  000000          WRTERR: .WORD 0         ;WRITE ERROR FLAG USED BY WRC
001176  000000          SPECMD: .WORD 0         ;COMMAND FLAG TO ALLOWMORE THAN 256 WORDS
                                                ;PER SECTOR WRITE OR READ.(HEADER WRITE)
                                                ;INSTALLED
                                ;
                                ;
                                ;
        000077                  NOPCMD=77       ;NO OPERATION COMMAND
        000200                  HDINC=200       ;HEAD FIELD INCREMENT BIT
        000006                  WRTCMD=6        ;WRITE COMMAND CONSTANT
        177600                  SECMSK=177600   ;SECTOR FIELD EXTRACTION MASK
        170177                  HDMSK=170177    ;HEAD FIELD EXTRACTION MASK
        147777                  UNIMSK=147777   ;UNIT SELECT EXTRACTION MASK
        176000                  CYLMSK=176000   ;CYLINDER EXTRACTION MASK
        000004                  RDCMD=4         ;READ COMMAND CONSTANT
        000010                  FMTCMD=10       ;FORMAT COMMAND
        000012                  RDFMT=12        ;READ FORMAT COMMAND
        000016                  RWHCMD=16       ;READ WITHOUT HEADER COMMAND
        000020                  WORCMD=20       ;WRITE OVER WRITE PROTECT COMMAND
        000014                  WHSCRC=14       ;WRITE HEADER,DATA,CRC COMMAND
        000125                  INTCAL=125      ;RECALIBRATE COMMAND WITH INTERRUPT
        000001                  SYSCLR=1        ;SYSTEM CLEAR COMMAND
        000025                  RECAL=25        ;RECALIBRATE COMMAND WITH GO BIT
        000024                  RTZCMD=24       ;RECALIBRATE COMMAND
        000002                  SEKCMD=2        ;SEEK COMMAND
        000032                  STRRD=32        ;READ IGNORING STRIP WORDS
        000022                  FLTCLR=22       ;FAULT CLEAR COMMAND
        000100                  QON=100         ;QUEUE ENABLE BIT
        000336                  QDUMY=336       ;QUEUE MODE DUMMY COM MAND
        000400                  QPRDY=400       ;QUEUE PROGRAM READY BIT
        000101                  INTGO=101       ;INTERRUPT AND GO BIT
        000100                  INTON=100       ;INTERRUPT ENABLE BIT
        000010                  SEKFLP=10       ;QUEUE SEEKING BIT
        000200                  FMTRDY=200      ;FORMATTER READY BIT
        100000                  REPBIT=100000   ;REPEAT BIT
        030000                  MEMEX=30000     ;MEMORY EXTENSION BITS
        000001                  GO=1            ;GO OR ACTIVE BIT
        000001                  BIT0=1          ;BIT 0 SYMBOL
        000002                  BIT1=2          ;
        000004                  BIT2=4          ;
        000010                  BIT3=10         ;
        000020                  BIT4=20         ;
        000040                  BIT5=40         ;
        000100                  BIT6=100        ;
        000200                  BIT7=200        ;
        000400                  BIT8=400        ;
        001000                  BIT9=1000       ;
        002000                  BIT10=2000      ;
        004000                  BIT11=4000      ;
        010000                  BIT12=10000     ;
        020000                  BIT13=20000     ;
        040000                  BIT14=40000     ;
        100000                  BIT15=100000    ;BIT 15 SYMBOL
        000026                  WRCHK=26        ;WRITE CHECK COMMAND
        000074                  SKDONE=74       ;DSTAT DONE COMPARE FLOPS
```

```
            007400                      SKFLOP=7400     ;DSTAT SEEKING FLOP
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       DISK DRIVE INPUT BUFFER ADDRESS TABLE
                                        ;
                                        ;
                                        ;
                                        ;THIS TABLE CONTAINS THE ORIGIN ADDRESSES OF THE DATA
                                        ;INPUT BUFFERS FOR EACH OF THE FOUR POSSIBLE DISK DRIVES.
                                        ;
                                        ;
                                        ;
                                        ;
     001200 047314              IBFTBL: .WORD IDBUF1    ;DRIVE 0 INPUT BUFFER ORIGIN ADDRESS
     001202 047314                      .WORD IDBUF1    ;       1
     001204 047314                      .WORD IDBUF1    ;       2
     001206 047314                      .WORD IDBUF1    ;       3
                                        ;
            000033                      INTCNT=<IBFTBL-SKCNT>/2         ;INITIALIZATION LOOP COUNT
                                        ;
                                        ;
                                        .EOT            ;END OF TAPE
                                        ;
                                        ;
            000000                      R0=%0           ;
                                ; SECTION 6 TEST PARAMETER CONSTANT TABLE
                                        ;
                                        ;
                                        ;       THIS TABLE CONTAINS CONSTANTS USED TO CONTROL THE
                                        ;        TEST SEQUENCE.
                                        ;
     001210                     SKTTN:
     001210 000032              STPTN:  .WORD 32        ;ENDING TEST NUMBER(DEFAULT IS TEST 25)
     001212 000000              STRTN:  .WORD 0         ;BEGINNING TEST NUMBER
     001214 000000              STPDSK: .WORD 0         ;END DISK DRIVE NUMBER
     001216 000000              STRTDSK:.WORD 0         ;START DISK NUMBER
     001220 000000              REPFLG: .WORD 0         ;REPEAT FLAG
     001222 000000              NOSTOP: .WORD 0         ;REPEAT ON ERROR FLAG
     001224 000000              PASINH: .WORD 0         ;INHIBIT PASS COUNT PRINTOUT
     001226 000000              ERPRIN: .WORD 0         ;INHIBIT ERROR PRINTOUT
     001230 000000              CONTST: .WORD 0         ;SKIP FLAG(TEST31,32 ONLY)
     001232 000000              CURTN:  .WORD 0         ;CURRENT TEST NUMBER
     001234 000000              CURDSK: .WORD 0         ;CURRENT DISK NUMBER
     001236 000000              PASCNT: .WORD 0         ;PASS COUNTER
     001240 000000              UNIT:   .WORD 0         ;CURRENT UNIT NUMBER IN UNIT FIELD
     001242 000000              NINPUT: .WORD 0         ;FLAG NO TEST INPUT
     001244 000000              NONCON: .WORD 0         ;FLAG NO CONSOLE INPUT
     001246 000000              SKPBAD: .WORD 0         ;FLAG ABORT OR SKIP BAD SECTOR DETECT(0=SKIP)
     001250 000000              CSWTCH: .WORD 0         ;CONSOLE SWITCH FOR DYNAMIC DUSH VALUES
     001252 176345              RAMBLER:.WORD 176345    ;RANDOM NUMBER
     001254 000000              DSKSAV: .WORD 0         ;TEMPORAY LOCATION FOR SAVING CURRENT DISK NUMBER
     001256 027767              POLY:   .WORD 27767     ;XOR NUMBER FOR RANDOM POLY
```

```
001260  001223          HDRC1:  .WORD 1223      ;HEADER WORD #1 OF SECTOR 1
001262  151140          HDRC2:  .WORD 151140    ;HEADER WORD #2 OF SECTOR 1
001264  000553          ECCW1:  .WORD 553       ;ECC WORD 1 FOR DATA=1 BIT
001266  026560          ECCW2:  .WORD 26560     ;ECC WORD 2 FOR DATA=1 BIT
001270  177000          MAXTFR: .WORD 177000    ;MASK FOR LIMITING TRANSFER SIZE
001272  000000          LPCNT:  .WORD 0         ;TEMP LOCATION FOR LOOP COUNTS
001274  173000          EMULAT: .WORD 173000    ;REBOOT ADDRESS FOR PDP11 CONSOLE EMULATOR
        030000                  UNIFLD=30000    ;BIT SET FOR EXTRACTING UNIT SELECT
        000136                  FUNC=136        ;BITS ANDED TOCLEAR FUNCTION+INTERRPT BITS
        000136                  FUNC=136        ;BITS ANDED TO CLEAR FUNCTION AND INTERRUPT BITS
001276  140000          BADBIT: .WORD 140000    ;BAD SECTOR BITS
001300  002000          BUFCLR: .WORD 2000      ; DATA BUFFER CLEAR
001302  040000          SEKINB: .WORD 40000     ;SEEK INHIBIT BIT
001304  030000          WPRBIT: .WORD 30000     ;MAXIMUM DUSH VALUE
001306  003400          SEKCMP: .WORD 3400      ;COMPARE SEEKING FLOPS WORD
001310  000004          TIMLP:  .WORD 4                 ;TIME OUT LOOP COUNT
        177777                  TIME=177777             ;NUMBER USED IN WAIT LOOPS
001312  000000          SKIPER: .WORD 0         ;COUNTER FOR HADR CONTROLLER ERRORS IN READ MODE(26)
001314  000000          ERRCNT: .WORD 0         ;DATA COUNTER(ERRORS)
001316  000000          HERCNT: .WORD 0         ;HARD ERROR INDEX
001320  000000          DATERR: .WORD 0         ;DATA ERROR FLAG
001322  000000          ERRFLG: .WORD 0         ;ERROR FLAG (MAIN)
001324  000000          ERROR:  .WORD 0         ;ERROR COUNT USED WITH BIT 15 REPEAT ON ERROR
001326  000000          SOFER1: .WORD 0         ;SOFT CONTROLLER ERROR COUNTER RETRY
001330  000000          SOFERR: .WORD 0         ;SOFT  CONTROLLER ERROR COUNTER
001332  000000          RTRYIN: .WORD 0         ;DATA RETRY INDEX
                                ;
```

```
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;       DATA ERROR TABLE
                                          ;     ----------------------
                                          
                                     ;
                                          ;       THIS TABLE CONTAINS DATA ENTRY COUNTS
                                          ;  CONTAINS THE NUMBER OF DATA ERRORS AND CORRESPONDING NUMBER
                                          ;  OF RETRYS MADE BEFORE THE DATA WAS RECOVERED CORRECTLY.
                                          ;    ANY DATA STILL IN ERROR AFTER 8 RETRYS IS CONSIDERED
                                          ;  A "HARD ERROR" AND IS COUNTED AS SUCH.
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
001334  000000                            .WORD 0         ;UNUSED
001336  000000                   ERRTAB:  .WORD 0         ;        1  RETRY DATA ERROR COUNT
001340  000000                            .WORD 0         ;        2  RETRY DATA ERROR COUNT
001342  000000                            .WORD 0         ;        3
001344  000000                            .WORD 0         ;        4
001346  000000                            .WORD 0         ;        5
001350  000000                            .WORD 0         ;        6
001352  000000                            .WORD 0         ;        7
001354  000000                            .WORD 0         ;        8
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;       TTY OUTPUT MESSAGE ADDRESS TABLE
                                          ;    ************************************
001356  001400                   TSTADR:  .WORD ENTMSG            ;ENDING TEST MESSAGE ADDRESS
001360  001426                            .WORD SRTMSG            ;STARTING TEST MESSAGE ADDRESS
001362  001456                            .WORD ENDMSG            ;ENDING DISK MESSAGE ADDRESS
001364  001505                            .WORD STDMSG            ;STARTING DISK MESSAGE ADDRESS
001366                           TSTBDR:
001366  001536                            .WORD REPMSG            ;REPEAT SEQUENCE MESSAGE ADDRESS
001370  001574                            .WORD RERMSG            ;REPEAT ON ERROR MESSAGE ADDRESS
001372  007320                            .WORD PINMSG            ;INHIBIT PASS COUNT PRINTOUT
001374  007365                            .WORD EINMSG            ;INHIBIT ERROR PRINT OUT
001376  007425                            .WORD COEMSG            ;INHIBIT ERROR PRINT OUT
                                          ;
                                          ;
001400     105     116     104   ENTMSG:  .ASCII /ENDING TEST NUMBER = /
001403     111     116     107
001406     040     124     105
001411     123     124     040
001414     116     125     115
001417     102     105     122
001422     040     075     040
```

```
     52 001425     000                          .BYTE 0
     53 001426     123     124     101  SRTMSG: .ASCII /STARTING TEST NUMBER = /
        001431     122     124     111
        001434     116     107     040
        001437     124     105     123
        001442     124     040     116
        001445     125     115     102
        001450     105     122     040
        001453     075     040
     54 001455     000                          .BYTE 0
     55 001456     105     116     104  ENDMSG: .ASCII /ENDING DRIVE NUMBER = /
        001461     111     116     107
        001464     040     104     122
        001467     111     126     105
        001472     040     116     125
        001475     115     102     105
        001500     122     040     075
        001503     040
     56 001504     000                          .BYTE 0
     57 001505     123     124     101  STDMSG: .ASCII /STARTING DRIVE NUMBER = /
        001510     122     124     111
        001513     116     107     040
        001516     104     122     111
        001521     126     105     040
        001524     116     125     115
        001527     102     105     122
        001532     040     075     040
     58 001535     000                          .BYTE 0
     59 001536     122     105     120  REPMSG: .ASCII /REPEAT THIS SEQUENCE? Y OR N /
        001541     105     101     124
        001544     040     124     110
        001547     111     123     040
        001552     123     105     121
        001555     125     105     116
        001560     103     105     077
        001563     040     131     040
        001566     117     122     040
        001571     116     040
     60 001573     000                          .BYTE 0
     61 001574     122     105     120  RERMSG: .ASCII /REPEAT THIS SEQUENCE ON ERROR Y OR N /
        001577     105     101     124
        001602     040     124     110
        001605     111     123     040
        001610     123     105     121
        001613     125     105     116
        001616     103     105     040
        001621     117     116     040
        001624     105     122     122
        001627     117     122     040
        001632     131     040     117
        001635     122     040     116
        001640     040
     62 001641     000                          .BYTE 0
     63                                         ;
     64                                         ;
     65                                         ;       DISK PARAMETER MESSAGE ADDRESSES
     66                                         ;
```

```
     67 001642  001652                           DISKP:  .WORD SECMSG              ;SECTOR VALUE MESSAGE ADDRESS
     68 001644  001703                                   .WORD HDMSG               ;HEAD VALUE MESSAGE ADDRESS
     69 001646  001732                                   .WORD CYLMSG              ;CYLINDER VALUE MESSAGE ADDRESS
     70 001650  001765                                   .WORD WPMSG               ;WORDS PER SECTOR MESSAGE ADDRESSS
     71                                                  ;
     72                                                  ;
     73 001652     115     101     130   SECMSG: .ASCII /MAXIMUM SECTOR ADDRESS =/
        001655     111     115     125
        001660     115     040     123
        001663     105     103     124
        001666     117     122     040
        001671     101     104     104
        001674     122     105     123
        001677     123     040     075
     74 001702     000                                   .BYTE 0
     75 001703     115     101     130   HDMSG:  .ASCII /MAXIMUM HEAD ADDRESS =/
        001706     111     115     125
        001711     115     040     110
        001714     105     101     104
        001717     040     101     104
        001722     104     122     105
        001725     123     123     040
        001730     075
     76 001731     000                                   .BYTE 0
     77 001732     115     101     130   CYLMSG: .ASCII /MAXIMUM CYLINDER ADDRESS =/
        001735     111     115     125
        001740     115     040     103
        001743     131     114     111
        001746     116     104     105
        001751     122     040     101
        001754     104     104     122
        001757     105     123     123
        001762     040     075
     78 001764     000                                   .BYTE 0
     79 001765     127     117     122   WPMSG:  .ASCII /WORDS PER SECTOR(2'S COMP.) =/
        001770     104     123     040
        001773     120     105     122
        001776     040     123     105
        002001     103     124     117
        002004     122     050     062
        002007     047     123     040
        002012     103     117     115
        002015     120     056     051
        002020     040     075
     80 002022     000                                   .BYTE 0
     81                                                  .EVEN
     82                                                  ;
     83                                                  ;
     84                                                  ;
     85                                                  ;
     86                                                  ;
     87                                                  ;
     88                                                  ;        DATE ENTRY MESSAGE ADDRESS TABLE
     89                                                  ;
     90                                                  ;
     91 002024     104     101     124   DTADR:  .ASCII /DATE: DD-MMM-YY /
        002027     105     072     040
```

```
            002032    104     104     055
            002035    115     115     115
            002040    055     131     131
            002043    040
       92   002044    000                            .BYTE 0
       93   002045    104     101     124   DTE:     .ASCII /DATE: /
            002050    105     072     040
       94   002053    000                            .BYTE 0
       95                                            ;
       96                                            ;
       97                                            ;
       98                                            ;
       99                                            ;       CONTROLLER PARAMETER MESSAGES
      100                                            ;
      101                                            .EVEN
      102                                            ;
      103   002054    002066                DCSMSG:  .WORD DCMSG             ;DCSR MESSAGE ADDRESS
      104   002056    002114                         .WORD INTMSG            ;INTERRUPT VECTOR MESSAGE ADDRESS
      105   002060                          DECGSM:
      106   002060    002137                         .WORD ECCMSG            ;ECC OPTION MESSAGE ADDRESS
      107   002062    002204                         .WORD MULMSG            ;MULTIPLE CPU MESSAGE ADDRESS
      108   002064    002241                         .WORD AUTMSG            ;AUTO SWITCH OPTION MESSAGE ADDRESS
      109                                            ;
      110                                            ;
      111   002066    104     103     123   DCMSG:   .ASCII /DCSR (BASE) ADDRESS =/
            002071    122     040     050
            002074    102     101     123
            002077    105     051     040
            002102    101     104     104
            002105    122     105     123
            002110    123     040     075
      112   002113    000                            .BYTE 0
      113   002114    111     116     124   INTMSG:  .ASCII /INTERRUPT VECTOR =/
            002117    105     122     122
            002122    125     120     124
            002125    040     126     105
            002130    103     124     117
            002133    122     040     075
      114   002136    000                            .BYTE 0
      115   002137    105     122     122   ECCMSG:  .ASCII /ERROR CORRECTION CONTROLLER? Y OR N /
            002142    117     122     040
            002145    103     117     122
            002150    122     105     103
            002153    124     111     117
            002156    116     040     103
            002161    117     116     124
            002164    122     117     114
            002167    114     105     122
            002172    077     040     131
            002175    040     117     122
            002200    040     116     040
      116   002203    000                            .BYTE 0
      117   002204    115     125     114   MULMSG:  .ASCII /MULTIPLE CPU OPTION? Y OR N /
            002207    124     111     120
            002212    114     105     040
            002215    103     120     125
            002220    040     117     120
```

```
         002223    124     111     117
         002226    116     077     040
         002231    131     040     117
         002234    122     040     116
         002237    040
     118 002240    000                            .BYTE 0
     119 002241    101     125     124    AUTMSG: .ASCII /AUTO SWITCH OPTION? Y OR N /
         002244    117     040     123
         002247    127     111     124
         002252    103     110     040
         002255    117     120     124
         002260    111     117     116
         002263    077     040     131
         002266    040     117     122
         002271    040     116     040
     120 002274    000                            .BYTE 0
     121                                          ;
     122                                          ;
     123                                          ;
     124                                          ;
     125                                          ;
     126                                          ;          PRINT COMMAND MESSAGES
     127                                          ;
     128                                          ;
     129                                          ;
     130                                          ;
     131                                   ; SECTION 7 TEST ORIGIN ADDRESS TABLES
     132                                          ;
     133                                          ;          (NON ECC CONTROLLERS)
     134                                          ;
     135                                          ;
     136                                          ;TABLE CONTAINS ORIGIN ADDRESS OF EACH TEST IN ASCENDING
     137                                          ;ORDER.
     138                                          ;
     139                                          ;
     140                                          ;
     141                                          ;
     142                                          ;
     143                                          .EVEN
     144 002276    016530                 TNORG:  .WORD TN0         ;REGISTER INITIALIZATION TEST
     145 002300    017412                         .WORD TN1         ;REGISTER LOAD AND READ TEST
     146 002302    017654                         .WORD TN2         ;REGISTER SLIDING ONES TEST
     147 002304    020006                         .WORD TN3         ;REGISTER SLIDING ZERO TEST
     148 002306    020244                         .WORD TN4         ;REGISTER ERROR AND STATUS REGISTER TESTS
     149 002310    021110                         .WORD TN5         ;UNIT SELECT TESTS
     150 002312    021544                         .WORD TN6         ;SEEKING TESTS
     151 002314    022216                         .WORD TN7         ;INTERRUPT LOGIC TEST
     152 002316    023626                         .WORD TN10        ;ONE WORD WRITE TEST
     153 002320    023670                         .WORD TN11        ;ONE SECTOR WRITE TEST
     154 002322    023760                         .WORD TN12        ;ONE SECTOR READ TEST
     155 002324    024042                         .WORD TN13        ;INCREMENTING WORD COUNT TEST
     156 002326    024130                         .WORD TN14        ;ONE SECTOR WRITE READ COMPARE TEST
     157 002330    024220                         .WORD TN15        ;WRT/RD/CMP INCREASING WORD COUNT TEST
     158 002332    024314                         .WORD TN16        ;WRITE PROTECT TEST
     159 002334    024664                         .WORD TN17        ;HEADER OVERRIDE-HEADER READ TEST
     160 002336    025474                         .WORD TN20        ;WRITE PROTECT-WRITE CHECK TEST
     161 002340    026350                         .WORD TN21        ;HEADER WRITE COMMAND TEST
```

```
002342 027170                    .WORD TN22      ;CRC/HEADER TEST
002344 030160                    .WORD TN23      ;BADSECTOR FLAG TEST
002346 030542                    .WORD TN24      ;RTZ TEST
002350 031360                    .WORD TN25      ;IMPLIED SEEK TESTS
002352 031724                    .WORD TN26      ;CONVERGE/DIVERGE SEEKING TEST - DISK ADDRESS TEST
002354 032346                    .WORD TN27      ;MEM. EXT.-BUSS TIME OUT-OVRRUN-MEM INC TEST
002356 033712                    .WORD TN30      ;CONVERGE/DIVERGE-RANDOM TEST(DATA COMPARE)
002360 034204                    .WORD TN31      ;DISK DATA TEST(ENTIRE TEST OF DISK PACK)
002362 035532                    .WORD TN32      ;READ  ENTIRE DISK TEST
002364 036144                    .WORD TN33      ;MULTIPLE CPU EXERCISE
002366 000000G                   .WORD TN34      ;
002370 000000G                   .WORD TN35      ;
002372 000000G                   .WORD TN36      ;OVERLAPPED SEEK TEST
002374 000000G                   .WORD TN37      ;OVERLAPPED SEEK AND DATA TRANSFERED TEST
002376 000000G                   .WORD TN40      ;LINKAGE REGISTER LOAD AND READ
002400 000000G                   .WORD TN41      ;LINKAGE WRITE TEST
002402 000000G                   .WORD TN42      ;LINKAGE READ TEST
002404 000000G                   .WORD TN43      ;LINKAGE WRITE/READ/COMPARE TEST
002406 000000G                   .WORD TN44      ;LINKAGE BLOCKED READ TEST
002410 042762                    .WORD TN45      ;FORMAT PROGRAM(> 8K)
002412 043314                    .WORD TN46      ;HEADER READ TEST/PACK VERIFY TEST
002414 043506                    .WORD TN47      ;FAST FORMAT PROGRAM
002416 000000G                   .WORD TN50      ;FORMAT PROGRAM
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;     TEST ORIGIN ADDRESS TABLE       (ECC CONTROLLERS)
                                 ;
                                 ;
                                 ;
                                 ;
002420 016530            TNORGA: .WORD TN0               ;REGISTER INITIALIZATION TEST
002422 017412                    .WORD TN1               ;REGISTER LOAD AND READ TEST
002424 017654                    .WORD TN2               ;REGISTER SLIDIN ONES TEST
002426 020006                    .WORD TN3               ;REGISTER SLIDING ZEROES TEST
002430 020244                    .WORD TN4               ;REGISTER ERROR AND REGISTER STATUS TESTS
002432 021110                    .WORD TN5               ;UNIT SELECT TEST
002434 021544                    .WORD TN6               ;SEEKING TESTS
002436 022216                    .WORD TN7               ;INTERRUPT LOGIC TEST
002440 023626                    .WORD TN10              ;ONE WORD WRITE TEST
002442 023670                    .WORD TN11              ;ONE    SECTOR WRITE TEST
002444 023760                    .WORD TN12              ;ONE    SECTOR READ TEST
002446 024042                    .WORD TN13              ;INCREMENTING WORD COUNT TEST
002450 024130                    .WORD TN14              ;ONE SECTOR WRITE/READ DATA COMPARE
002452 024220                    .WORD TN15              ;WRT/RD/CMP INCREASING WORD COUNT TEST
002454 036464                    .WORD TNA16             ;ECC REGISTER TESTS
002456 036774                    .WORD TNA17             ;ECC CRC DETECTION TESTS(HEADERS ONLY)
002460 025740                    .WORD TNA20             ;WRITE CHECK TESTS
002462 041076                    .WORD TNA21             ;ECC DETECTION TESTS
002464 041674                    .WORD TNA22             ;ECC DETECTION/CORRECTION TESTS
002466 042516                    .WORD TNA23             ;ECC COMPATIBILITY/EXERCIRE TESTS
002470 030542                    .WORD TN24              ;RTZ TEST
002472 031360                    .WORD TN25              ;IMPLIED SEEKING TEST
002474 031724                    .WORD TN26              ;CONVERGE/DIVERGE SEEKING AND DISK ADDRESS TESTS
002476 032346                    .WORD TN27              ;MEM. EXT,BUSS TIME OUT,OVRRUN,MEM INC. TESTS
002500 033712                    .WORD TN30              ;CONVERGE/DIVERGE,RANDOM WR/RD CMP TEST
```

```
 002502  034204                .WORD TN31          ;DISK DATA TEST(ENTIRE DISK PACK)
 002504  035532                .WORD TN32          ;DISK RELIABILITY (READ ) TEST
 002506  036144                .WORD TN33          ;MULTIPLE CPU EXERCISE
 002510  000000G               .WORD TN34          ;
 002512  000000G               .WORD TN35          ;
 002514  000000G               .WORD TN36          ;OVERLAPPED SEEK TEST
 002516  000000G               .WORD TN37          ;OVERLAPPED SEEK/WRITE TEST
 002520  000000G               .WORD TN40          ;
 002522  000000G               .WORD TN41          ;
 002524  000000G               .WORD TN42          ;
 002526  000000G               .WORD TN43          ;
 002530  000000G               .WORD TN44          ;
 002532  042762                .WORD TN45          ;FORMAT PROGRAM ( > 8K)
 002534  043314                .WORD TN46          ;HEADER VERIFY TEST
 002536  043506                .WORD TN47          ;FORMAT TEST
 002540  000000G               .WORD TN50          ;
                               ;
                               ;
                               ;

                               ;
                               ;
                               ;
                               ;
                               ;
                               ;
                               ;       DISK DRIVE SEEK TABLE
                               ;
                               ;
                               ;THE DISK DRIVE SEEK TABLE CONTAINS THE CURRENT COMMANDED
                               ;CYLINDER ADDRESS OF EACH OF THE DISK DRIVES. THE
                               ;CYLINDER ADDRESS WORD FOR EACH DRIVE IS UPDATED BY THE
                               ; TEST PROGRAMS AND TRANSMITTED TO THE DISK DRIVE VIA
                               ; THE CONTROLLER.
                               ;
                               ;
 002542  000000        CYLADR: .WORD 0        ; DRIVE 0 COMMANDED CYLINDER ADDRESS
 002544  000000                .WORD 0        ;       1
 002546  000000                .WORD 0        ;       2
 002550  000000                .WORD 0        ;       3
                               ;
                               ;
                               ;
```

```
                                        ;
                                        ;
                                        ;
                                        ; THIS TABLE CONTAINS THE MASK FOR EACH OF THE DISK
                                        ;CONTROLLER REGISTERS THAT WILL BE USED TO ELIMINATE
                                        ;UNUSED BITS IN THE REGISTERS FOR TESTING PURPOSES.
                                        ;
                                        ;
                                        ;
002552  107577                 REGMSK:  .WORD 107577     ;DISK CONTROL AND STATUS REGISTER MASK
002554  140000                          .WORD 140000     ;UNIT-HEAD-SECTOR MASK
002556  000001                          .WORD 1          ;CORE ADDRESS MASK
002560  000000                          .WORD 0          ;WORD COUNT MASK
002562  176000                          .WORD 176000     ;CYLINDER ADDRESS MASK
002564  000000                          .WORD 0          ;DISKSTATUS DUMMY MASK
002566  000000                          .WORD 0          ;DISK ERROR REGISTER DUMMY MASK
002570  000000                          .WORD 0          ;GCSR REGISTER DUMMY MASK
002572  000000                          .WORD 0          ;OMSC REGISTER MASK
002574  000000                          .WORD 0          ;QUSH MASK
002576  000000                          .WORD 0          ;QBAR MASK
002600  000000                          .WORD 0          ;QWCNT MASK
002602  000000                          .WORD 0          ;QCYL MASK
002604  000000                          .WORD 0          ;
002606  177774                          .WORD 177774     ;STRIP CORE EXTENTION MASK
002610  000001                          .WORD 1          ;OSCAR REGISTER MASK
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       SLIDING ONE CONSTANT TABLE
                                        ;
                                        ;
                                        ;CONSTANTS ARE USED IN REGISTER TESTS
                                        ;
                                        ;
                                        .GLOBL S1ORG,S0ORG,CNSTNT
                                        ;
                                        ;
                                        ;
002612  000001                 S1ORG:   .WORD 1          ;
002614  000002                          .WORD 2          ;
002616  000004                          .WORD 4          ;
002620  000010                          .WORD 10         ;
002622  000020                          .WORD 20         ;
002624  000040                          .WORD 40         ;
002626  000100                          .WORD 100        ;
002630  000200                          .WORD 200        ;
002632  000400                          .WORD 400        ;
002634  001000                          .WORD 1000       ;
002636  002000                          .WORD 2000       ;
002640  004000                          .WORD 4000       ;
002642  010000                          .WORD 10000      ;
```

```
002644  020000                  .WORD 20000     ;
002646  040000                  .WORD 40000     ;
002650  100000                  .WORD 100000    ;
                                ;
                                ;
                                ;       SLIDING ZERO CONSTANT TABLE
                                ;
                                ;
002652  177776          SUORG:  .WORD 177776    ;
002654  177775                  .WORD 177775    ;
002656  177773                  .WORD 177773    ;
002660  177767                  .WORD 177767    ;
002662  177757                  .WORD 177757    ;
002664  177737                  .WORD 177737    ;
002666  177677                  .WORD 177677    ;
002670  177577                  .WORD 177577    ;
002672  177377                  .WORD 177377    ;
002674  176777                  .WORD 176777    ;
002676  175777                  .WORD 175777    ;
002700  173777                  .WORD 173777    ;
002702  167777                  .WORD 167777    ;
002704  157777                  .WORD 157777    ;
002706  137777                  .WORD 137777    ;
002710  077777                  .WORD 077777    ;
                                ;
                                ;
                                ;
                                ;       MIXED CONSTANT TABLE
                                ;
                                ;
002712  000000          CNSTNT: .WORD 0         ;ALL ZEROS
002714  052525                  .WORD 052525    ; CHECKERBOARD
002716  125252                  .WORD 125252    ;REVERSE CHECKERBOARD
002720  177777                  .WORD 177777    ;ALL ONES
                                ;
                                ;
                                ;
                                ;     ODT FILLER CONSTANT
                        ; SECTION 9 ERROR HALT REGISTER IMAGE (CPU REGISTERS)
                                ;
002722  000000          REGO:   .WORD 0         ;DRIVE CURRENTLY UNDER TEST
002724  000000                  .WORD 0         ;FAILED TEST NUMBER
002726  000000                  .WORD 0         ;REFERENCE DATA
002730  000000                  .WORD 0         ;ACTUAL DATA
002732  000000                  .WORD 0         ;SUBTEST NUMBER
002734  000000                  .WORD 0         ;LAST SUBROUTINE JUMP ADDR.
002736  000000                  .WORD 0         ;CPU STACK ADDR.
                                ;
                                ;
                                ;  THIS CONSTANT IS USED TO ENABLE ODT TO BE LINKED WITH
                                ; THE DIAGNOSTIC AND USED IN A STAND ALONE SYSTEM
                                ;
                                ;
                                ;
                                .GLOBL $$CON    ;
                                ;
```

```
                                                  ;
                                                  ;       START ROUTINE
                                                  ;
                                                  ;
                                                  ;
                                          ; SECTION 10 TEST CODE
                 003000                           .=3000                    ;START LOCATION =3000
                                                  ;
                                                  ;
        003000   012706   000776          START:  MOV     #DCSR-2,SP        ;POINT STACK TO SAFE VALUE
        003004   010600                           MOV     SP,R0             ;USE R0 TO CLEAR CORE
        003006   005720                           TST     (R0)+             ;LOOK  AT PROPER VECTOR
        003010   005040                   CORE:   CLR     -(R0)             ;ENTER HALT
Z       003012   010040                           MOV     R0,-(R0)          ;ENTER VECTOR TRAP
        003014   005700                           TST     R0                ;CHECK IF LOADED LOCATION 0
        003016   001401                           BEQ     CORE1             ;EXIT IF YES
        003020   000773                           BR      CORE              ;DOAGAIN
                                                  ;
                                                  ;
                                                  ;
                                                  ;
        003022   005046                   CORE1:  CLR     -(SP)             ;SET PROCESSOR STATUS WORD
        003024   012746   003032                  MOV     #PSWORD,-(SP)     ;LOAD RETURN ADDRESS
        003030   000002                           RTI                       ;FAKE INTERRUPT
        003032   005737   001104          PSWORD: TST     TTYPUT            ;KEYBOARD SETUP?
        003036   001402                           BEQ     TSTART            ;NO GO ON
        003040   000137   006170                  JMP     TTSTART           ;SET UP READER/PRINTER
                                                  ;
                                                  ;
        003044   004537   015644          TSTART: JSR     R5,ADRSTP         ;SET UP PARAMETERS NOW
                                          ;****************************************************************
        003050   005737   001244                  TST     NONCON            ;SEE IF SWITCH REGISTER PRESENT
        003054   001014                           BNE     TSTCON            ;CONTINUE IF NO PARAMETER ENTRY
        003056   005737   001242                  TST     NINPUT            ;TEST NO TEST INPUT
        003062   001011                           BNE     TSTCON            ;CONTINUE IF NO
        003064   005737   001104                  TST     TTYPUT            ;KEYBOARD?
        003070   001006                           BNE     TSTCON            ;GO ON IF YES
        003072   013700   177570          STARTA: MOV     CSWR,R0           ;ENTER FIRST PARAMETER
        003076   010003                           MOV     R0,R3             ;SAVE IN R3 ALSO
        003100   100513                           BMI     SREPFL            ;SET REPEAT FLAG IF NEGITIVE
        003102   005037   001220                  CLR     REPFLG            ;RESET REPEAT FLAG
        003106   012704   001312          TSTCON: MOV     #SKIPER,R4        ;SET UP TO CLEAR  ERROR LOCS.
        003112   010401                           MOV     R4,R1             ;
        003114   012702   001332                  MOV     #RTRYIN,R2        ;
        003120   160102                           SUB     R1,R2             ;
        003122   006202                           ASR     R2                ;
        003124   005024                   STST1:  CLR     (R4)+             ;CLEAR LOCATION
        003126   005302                           DEC     R2                ;
        003130   001375                           BNE     STST1             ;
        003132   005037   001236                  CLR     PASCNT            ;RESET PASS COUNTER
        003136   005737   001244                  TST     NONCON            ;TEST CONSOLE MODE AGAIN
        003142   001046                           BNE     STCLR             ;START TESTING IF NON ZERO
        003144   005737   001242                  TST     NINPUT            ;TEST NO DISK INPUT
        003150   001043                           BNE     STCLR             ;CONTINUE IF NO
        003152   005737   001104                  TST     TTYPUT            ;KEYBOARD?
        003156   001040                           BNE     STCLR             ;CONTINUE IF NO
        003160   042700   177400                  BIC     #177400,R0        ;EXTRACT STOP TEST NUMBER
```

```
003164  042703  100377                    BIC    #100377,R3        ;EXTRACT START TEST NUMBER
003170  000303                            SWAB   R3                ;
003172  012701  001210                    MOV    #STPTN,R1         ;POINT R1 TO STOP TEST NUMBER LOCATION
003176  010021                            MOV    R0,(R1)+          ;SAVE ENDING TEST NUMBER
003200  010321                            MOV    R3,(R1)+          ;SAVE STRTING TESTNUMBER
003202  000240                            NOP
003204  000000                            HALT                     ;WAIT FOR OPERATOR
003206  005737  001250                    TST    CSWTCH            ;SEE IF IN SCOPE MODE
003212  001403                            BEQ    TSTCA             ;CONTINUE IF NO
003214  005000                            CLR    R0                ;MAKE DRIVE SELECTION DRIVE ZERO
003216  005002                            CLR    R2                ;
003220  000415                            BR     TSTCB             ;CONTINUE INIT
003222  013700  177570           TSTCA:   MOV    CSWR,R0           ;PUT TEST DRIVE PARAMETERS IN R0
003226  010002                            MOV    R0,R2             ;ENABLE AUTO SWITCH FLAG IF APPL.
003230  042702  157777                    BIC    #157777,R2        ;EXTRACT AUTO SWITCH
003234  010237  001100                    MOV    R2,AUTOSW         ;
003240  010002                            MOV    R0,R2             ;SAVE IN R2 ALSO
003242  042700  177600                    BIC    #177600,R0        ;EXTRACT END DISK NUMBER
003246  042702  160377                    BIC    #160377,R2        ;EXTRACT STARTING DISK NUMBER
003252  000302                            SWAB   R2       ;
003254  010021                   TSTCB:   MOV    R0,(R1)+          ;SAVE LAST DRIVE NUMBER
003256  010221                            MOV    R2,(R1)+          ;SAVE BEGINNING DISK DRIVE NUMBER
003260  013737  001212  001232   STCLR:   MOV    STRTN,CURTN       ;ESTABLISH CURRENT TEST NUMBER
003266  013737  001216  001234            MOV    STRTDSK,CURDSK    ;ESTABLISH CURRENT DRIVE NUMBER
003274  012700  001336                    MOV    #ERRTAB,R0        ;GET ERROR TABLE ORIGIN ADDRESS
003300  012701  000010                    MOV    #8.,R1            ;SET UP LOOP COUNT
003304  005020                            CLR    (R0)+             ;CLR  RETRY COUNT LOCATION
003306  005301                            DEC    R1                ;DECREMENT LOOP COUNT
003310  001375                            BNE    .-4               ;REPEAT IF NOT ZERO
003312  000005                            RESET                    ;RESET I/O
003314  004737  015414                    JSR    PC,TTYSET         ;ENABLE INTERRUPTS FOR KEYBOARD
003320  004537  016352                    JSR    R5,DKINIT         ;INITIALIZE DISKS
003324  000137  003532                    JMP    PARINT            ;GO TO PARAMETER INTIALIZATION
003330  005237  001220           SREPFL:  INC    REPFLG            ;SET TEST REPEAT FLAG
003334  042700  010000                    BIC    #10000,R0         ;MASK OUT TEST PARAMETERS
003340  000662                            BR     TSTCON            ;CONTINUE
                                          ;
                                          ;
                                          ;
                                          ;      TEST CONTROL ROUTINE
                                          ;
                                          ;
                                          ; AT ENTRY:
                                          ;      1. R5 CONTAINS CALLING PROGRAM ADDRESS IF ROUTINE
                                          ;      WAS CALLED AS A RESULT OF DETECTION
                                          ;         OF ANY ERROR.
                                          ;
                                          ;      2. CURDSK= CURRENT DISK NUMBER
                                          ;
                                          ;      3. CURTN = CURRENT TEST NUMBER
                                          ;
                                          ;PROGRAM EXITS TO APPROPRIATE TEST ROUTINE OR
                                          ; HALTS DEPENDING ON SPECIFED ERROR PARAMETERS
                                          ;
                                          ;
                                          .GLOBL INTCNT          ;INITIALIZATION LOOP COUNT
                                          ;
```

```
    229                                             ;
    230                                             ;
    231 003342  004737  015414          TSTCTL: JSR     PC,TTYSET       ;ENABLE INTERRUPTS FOR KEYBOARD
    232 003346  005737  001322                  TST     ERRFLG          ;ERRORS?
    233 003352  001461                          BEQ     TNTST           ;BRANCH IF NO
    234 003354  004737  015440                  JSR     PC,REGSAV       ;SAVE CONTROLLER
    235 003360  005737  001104                  TST     TTYPUT          ;KEYBOARD?
    236 003364  001012                          BNE     STOP            ;SAVE CPU IF YES
    237 003366  005737  001244          TST2:   TST     NONCON          ;CONSOLE MODE?
    238 003372  001404                          BEQ     TST1            ;CONTINUE IF YES
    239 003374  005737  001222                  TST     NOSTOP          ;TEST NO CONSOLE CONTINUE ON ERROR FLG
    240 003400  001046                          BNE     TNTST           ;GO ON IF DESIRED
    241 003402  000403                          BR      STOP            ;HALT IF NOT
    242 003404  005737  177570          TST1:   TST     CSWR            ;IS STOP ON ERROR INHIBITED?
    243 003410  100442                          BMI     TNTST           ;CONTINUE WITH NEXT TEST IF YES
    244 003412  013700  001234          STOP:   MOV     CURDSK,R0       ;ESTABLISH FAILED DISK DRIVE NO
    245 003416  013701  001232                  MOV     CURTN,R1        ;ESTABLISH FAILED TEST NO.
    246 003422  010037  002722                  MOV     R0,REG0         ;SAVE CPU REGISTER
    247 003426  012700  002722                  MOV     #REG0,R0        ;
    248 003432  005720                          TST     (R0)+           ;POINT TO NEXT ENTRY
    249 003434  010120                          MOV     R1,(R0)+        ;
    250 003436  010220                          MOV     R2,(R0)+        ;
    251 003440  010320                          MOV     R3,(R0)+        ;
    252 003442  010420                          MOV     R4,(R0)+        ;
    253 003444  010520                          MOV     R5,(R0)+        ;
    254 003446  010620                          MOV     SP,(R0)+        ;INCLUDE STACK POINTER
    255 003450  013700  002722                  MOV     REG0,R0         ;RESTORE R0
    256 003454  005737  001104                  TST     TTYPUT          ;KEYBOARD ACTIVE?
    257 003460  001402                          BEQ     TST3            ;GO ON IF NO
    258 003462  000137  012030                  JMP     TTERR           ;PRINT FAILURE IF YES
    259 003466  005737  001244          TST3:   TST     NONCON          ;CONSOLE(SWITCHES)?
    260 003472  001002                          BNE     BOOT            ;BOOT EMULATOR IF NO
    261 003474  000000                  STOP8:  HALT                    ;QUIT++++++++++
    262 003476  000415                          BR      PARINT
    263                                             ;
    264                                             ;
    265                                             ;
    266 003500  004537  004020          BOOT:   JSR     R5,REBOOT       ;LOAD SOMETHING INTERESTING
    267                                             ;
    268                                             ;
    269                                             ;
    270                                             ;
    271 003504  005737  001220          STPTST: TST     REPFLG          ;IS TEST SEQUENCE TO BE REPEATED?
    272 003510  001740                          BEQ     STOP            ;STOP HERE IF NO
    273 003512  000137  003720                  JMP     PASINC          ;JUMP TO PASS INCREMENT IF YES
    274 003516  023737  001232  001210  TNTST:  CMP     CURTN,STPTN     ;ENDING TEST NUMBER?
    275 003524  001466                          BEQ     LSTTST          ;BRANCH IF YES
    276 003526  005237  001232                  INC     CURTN           ;INCREMENT CURRENT TEST NUMBER
    277 003532  012700  001112          PARINT: MOV     #SKCNT,R0       ;POINT TO FLAG CLEAR LOCATION
    278 003536  013777  001052  175326          MOV     INTSAV,@INTVEC  ;LOAD INTERRUPT VECTOR
    279 003544  013702  001072                  MOV     INTVEC,R2       ;ENTER PRIORITY LEVEL 5
    280 003550  005722                          TST     (R2)+           ;
    281 003552  012712  000240                  MOV     #240,(R2)       ;
    282 003556  012701  000033                  MOV     #INTCNT,R1      ;ESTABLISH LOOP COUNT IN R1
    283 003562  005002                  INITLP: CLR     R2              ;
    284 003564  010220                          MOV     R2,(R0)+        ;CLEAR LOCATION
    285 003566  005301                          DEC     R1              ;DECREMENT LOOP COUNT
```

```
003570  001374                          BNE     INITLP          ;REPEAT IF NOT DONE
003572  004537  004034                  JSR     R5,CLRBUF       ;CLEAR DISK BUFFER AREA
003576  004537  004276                  JSR     R5,RQUEST       ;GO REQUEST FORMATTER IF MULTI CPU
003602  005777  175172                  TST     @DCSR           ;ERROR STILL?
003606  100002                          BPL     FMTCLR          ;GO ON IF NO
003610  004537  016352                  JSR     R5,DKINIT       ;INITAILIZE DISKS
003614  022606                  FMTCLR: CMP     (SP)+,SP        ;POP R5 OFF THE STACK
003616  005737  001322                  TST     ERRFLG          ;ERROR DURING PAST TEST?
003622  001404                          BEQ     .+10.           ;CONTINUE IF NO
003624  005037  001322                  CLR     ERRFLG          ;RESET ERROR FLAG
003630  005237  001324                  INC     ERROR           ;INCREMENT ERROR COUNTER
003634  013700  001232                  MOV     CURTN,R0        ;ESTABLISH TEST NO. INDEX
003640  005737  001100                  TST     AUTOSW          ;AUTO SWITCH DESIRED?
003644  001403                          BEQ     .+8.            ;CONTINUE IF NO
003646  012777  000200  175162          MOV     #200,@AUTO      ;TURN AUTOSWITCH ON IF YES
003654  006300                          ASL     R0              ;MAKE VALID WORDD INDEX OF TEST  NO.
003656  012777  000041  175114          MOV     #SYSCLR!BIT5,@DCSR;CLEAR  FORMATTER
003664  005737  001074                  TST     ECC             ;IS THIS AN ECC CONTROLLER?
003670  001402                          BEQ     GOTEST          ;RUN NORMAL TEST SEQUENCE IF NO
003672  000170  002420                  JMP     @TNORGA(R0)     ;RUN EDD TESTS IF YES
003676  000170  002276          GOTEST: JMP     @TNORG(R0)      ;RUN NORMAL TESTS IF NO
003702  023737  001234  001214  LSTTST: CMP     CURDSK,STPDSK   ;HAVE WE TESTED ALL OF THE DISKS?
003710  001403                          BEQ     PASINC          ;GO ON IF YES
003712  005237  001234                  INC     CURDSK          ;REPEAT TESTS ON NEXT DISK IF NO
003716  000433                          BR      TNINT           ;
003720  005237  001236          PASINC: INC     PASCNT          ;INCREMENT PASS COUNT LOCATION
003724  005737  001104                  TST     TTYPUT          ;KEYBOARD?
003730  001402                          BEQ     PINC1           ;GO ON IF NO
003732  000137  011640                  JMP     PASS            ;GO TO PASS PRINT OUT IF YES
003736  005737  001220          PINC1:  TST     REPFLG          ;REPEAT?
003742  001016                          BNE     TNINTA          ;FINISH UP IF NO
003744  013700  001236                  MOV     PASCNT,R0       ;SET R0 TO PASS COUNT
003750  005737  001244                  TST     NONCON          ;DO WE HAVE A CONSOLE?
003754  001410                          BEQ     HALT8           ;END IT IF NO
003756  000137  003500                  JMP     BOOT            ;SPIT OUT RESULTS IF YES
003762  000405                          BR      HALT8           ;END AT 4000
                                        ;
                                        ;
                                        ;
                                        ;
        003776                          .=3776                  ;
003776  000000                  HALT8:  HALT                    ;END IT HERE
004000  013737  001216  001234  TNINTA: MOV     STRTDSK,CURDSK  ;SET UP FIRST DISK AS CURRENT DISK
004006  013737  001212  001232  TNINT:  MOV     STRTN,CURTN     ;SET UP FIRST TEST AS CURRENT TEST
004014  000137  003532                  JMP     PARINT  ;CONTINUE REINITIALIZATION
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;     EMULATOR BOOT ROUTINE
                                        ; ******************************
                                        ;
                                        ;
                                        ;
004020  013700  001322          REBOOT: MOV     ERRFLG,R0       ;LOAD ERRFLG INTO R0
004024  013704  001232                  MOV     CURTN,R4        ;LOAD ENDING TEST NUMBER
```

```
004030  013707  001274                  MOV     EMULAT,PC           ;BOOT UP THE EMULATOR
                                        ;    CAN'T DO ANYTHING NOW !
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       BUFFER CLEAR ROUTINE
                                        ;  ***************************
                                        ;
                                        ;
                                        ;
        007012                          CLRCNT=IDBUF5-<ODBUF-10>        ;CLEAR COUNT
                                        ;
                                        ;
004034  004537  015474         CLRBUF:  JSR     R5,SAVE             ;SAVE CPU
004040  012700  044302                  MOV     #ODBUF-10,R0        ;USE R0 TO POINT
004044  012701  007012                  MOV     #CLRCNT,R1          ;USE R1 TO COUNT
004050  006201                          ASR     R1                  ;MAKE USEFUL
004052  005020                 CLRB1:   CLR     (R0)+               ;CLEAR MEMORY
004054  005301                          DEC     R1                  ;DECREMENT COUNTER
004056  001375                          BNE     CLRB1               ;CONTINUE IF NOT ZERO
004060  004537  015510                  JSR     R5,REST             ;RESTORE CPU
004064  000205                          RTS     R5                  ;AND EXIT
                                        ;
                                        ;
                                        ;
        007012                          CLRCNT=IDBUF5-<ODBUF-10>
                                        ;
                                        ;       SEEK TIME OUT LOOP
                                        ;  ***************************
                                        ;
                                        ;       THIS ROUTINE ISSUES A SEEK TO MAXCYL IN CURRENT DISK
                                        ; . IT THEN ISSUES A RECALIBRATE AN USES A COUNTER TO
                                        ;   ESTABLISH A WAIT LOOP SIZE FOR TIMING A SEEK .
                                        ;
                                        ;
                                        ;
                                        ;
004066  004537  015474         SEKTIM:  JSR     R5,SAVE             ;SAVE CPU REGISTERS
004072  012701  000003                  MOV     #3,R1               ;SET UP MAXIMUM TIMEOUT LOOP
004076  004537  016332                  JSR     R5,DISKID           ;SET  UP DISK SELECT BITS
004102  004537  004276                  JSR     R5,RQUEST           ;REQUEST FORMATTER
004106  010277  174670                  MOV     R2,@DUSH            ;SELECT DISK
004112  000240                          NOP                         ;WAIT FOR SELECTION
004114  000240                          NOP
004116  005777  174670                  TST     @DSTAT              ;DISK OUT THERE?
004122  100044                          BPL     SEKER               ;EXIT NO
004124  013777  001044  174656          MOV     MAXCYL,@DCYL        ;LOAD MAXCYL
004132  052777  000002  174640          BIS     #SEKCMD,@DCSR       ;LOAD SEEK COMMAND
004140  005277  174634                  INC     @DCSR               ;SET GO BIT
004144  012700  177777         SKLP1:   MOV     #-1,R0              ;LOAD LOOP COUNT
004150  000240                          NOP                         ;
004152  105777  174634         SKLP:    TSTB    @DSTAT              ;SEEK DONE?
004156  100405                          BMI     SEKOUT              ;EXIT IF YES
004160  005300                          DEC     R0                  ;DECREMENT SEEK LOOP
```

```
 004162  001373                          BNE     SKLP            ;LOOK AGAIN
 004164  005301                          DEC     R1              ;DECREMENT OUTER LOOP
 004166  001366                          BNE     SKLP1           ;GET OUT OF HERE?
 004170  000421                          BR      SEKER           ;EXIT IF HAVN'T FINSHED
 004172  012701  000003         SEKOUT:  MOV     #3,R1           ;RESTORE OUTER LOOP
 004176  004537  004276                  JSR     R5,RQUEST       ;REQUEST FORMATTER
 004202  052777  000025  174570          BIS     #RECAL,@DCSR    ;ISSUE RECAL
 004210  000240                          NOP                     ;
 004212  012700  177777         SEKOU2:  MOV     #-1,R0          ;RESTORE LOOP COUNT
 004216  105777  174570         SEKOU1:  TSTB    @DSTAT          ;SEEK DONE?
 004222  100416                          BMI     SKDON           ;EXIT IF YES
 004224  005300                          DEC     R0              ;DECREMENT IF NO
 004226  001373                          BNE     SEKOU1          ;LOOK AGAIN
 004230  005301                          DEC     R1              ;DEC OUTER LOOP
 004232  001367                          BNE     SEKOU2          ;LOAK AGAIN IF NOT DONE
 004234  005237  001322         SEKER:   INC     ERRFLG          ;SOMETHING WRONG GET OUT OF HERE
 004240  012777  000041  174532 SEKBK:   MOV     #SYSCLR!BIT5,@DCSR      ;ISSUE SYSCLR
 004246  004537  004276                  JSR     R5,RQUEST       ;REGAIN FORMATTER
 004252  004537  015510                  JSR     R5,REST         ;RESTORE CPU REGS
 004256  000205                          RTS     R5              ;AND RETURN TO CALLER
 004260  012703  000003         SKDON:   MOV     #3,R3           ;COMPUTE LOOPSIZE
 004264  160103                          SUB     R1,R3           ;
 004266  005203                          INC     R3              ;INSURE LOOP
 004270  010337  001310                  MOV     R3,TIMLP        ;AND MAKE NUMBER PERM.
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
 004274  000761                          BR      SEKBK           ;AND LEAVE
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;        CONTROLLER REQUEST ROUTINE
                                         ;  ***********************************
                                         ;
                                         ;
                                         ;
                                         ;        THIS ROUTINE REQUESTS COMMAND OF THE PHOENIX
                                         ;     211 CONTROLLER IF  THE MULTIPLE CPU OPTION IS
                                         ;     INOPERATION.
                                         ;
                                         ;
                                         ;
                                         ;
 004276  005737  001076         RQUEST:  TST     MULCPU          ;DO WE WANT THIS CONTROLLER?
```

```
    457 004302  001434                          BEQ     RQEX            ;EXIT IF NO
    458 004304  005037  001150                  CLR     INTFLG          ;RESET INTERRUPT FLAG
    459 004310  105777  174464                  TSTB    @DCSR           ;DO WE HAVE CONTROLLER ALREADY?
    460 004314  100005                          BPL     RQ2             ;REQUEST IT IT IF NO
    461 004316  032777  000040  174454          BIT     #BIT5,@DCSR     ;IS REQUEST SET?
    462 004324  001421                          BEQ     RQ3             ;EXIT IF NO
    463 004326  000426                          BR      RQ1             ;CHECK FOR ERROR
    464 004330  012777  000100  174442  RQ2:    MOV     #INTON,@DCSR    ;SET INTERRUPT BIT
    465 004336  053777  001102  174434          BIS     MCPU,@DCSR      ;REQUEST 211
    466 004344  004537  017342                  JSR     R5,WAIT         ;WAIT FOR INTERRUPT
    467 004350  005737  001150                  TST     INTFLG          ;INTERRUPT?
    468 004354  001407                          BEQ     RQEX            ;EXIT IF NO
    469 004356  005037  001150                  CLR     INTFLG          ;CLEAR INTERRUPT FLAG(NOT A FUNCTION TESTED)
    470 004362  105777  174412                  TSTB    @DCSR           ;DO WE HAVE THE 211?
    471 004366  100406                          BMI     RQ1             ;CONTINUE IF YES
    472 004370  005237  001322          RQ3:    INC     ERRFLG          ;SET ERROR FLAG IF NO
    473 004374  042777  000136  174376  RQEX:   BIC     #FUNC,@DCSR     ;CLEAR FUNCTION BITS
    474 004402  000205                          RTS     R5              ;AND RETURN
    475 004404  005777  174370          RQ1:    TST     @DCSR           ;ERROR IN FORMATTER?
    476 004410  100371                          BPL     RQEX            ;EXIT IF NO
    477 004412  042777  000136  174360          BIC     #FUNC,@DCSR     ;LOOSE FUNCTION BITS IF YES
    478 004420  052777  000041  174352          BIS     #SYSCLR!BIT5,@DCSR      ;ISSUE FORMATTER CLEAR HERE
    479 004426  000240                          NOP
    480 004430  013777  001102  174342          MOV     MCPU,@DCSR      ;RESELECT FORMATTER(WE HAD IT ONCE)
    481 004436  000240                          NOP
    482 004440  000240                          NOP
    483 004442  000240                          NOP
    484 004444  000753                          BR      RQEX            ;EXIT EITHER WAY
    485                                         ;
    486                                         ;
    487                                         ;
    488                                         ;
    489                                         ;
    490                                         ;
    491                                         ;
    492                                         ;
    493                                         ;
    494                                         ;
    495                                         ;
    496                                         ;       PMGR.P11                6-JAN-77
    497                                         ;
    498                                         ;
    499                                         ;
    500                                         ;       PROGRAM MANAGER
    501                                         ;
    502                                         ;       THE PROGRAM MANGER IS USED TO DIRECT INPUT FROM
    503                                         ;     THE KEYBOARD READER.ONLY TWO DECISIONS ARE MADE IN THIS CODE:
    504                                         ;               1. TO WAIT FOR MORE INTERRUPTS(READER)
    505                                         ;               2.TO ENTER COMMAND LINE INTERPRETER.
    506                                         ;
    507                                         ;       ENTRY IS BY FORCING PC THRU RTI OR
    508                                         ;    BY A SUBROUTINE CALL.
    509                                         ;
    510                                         ;
    511                                         ;
    512                                         ;
    513 004446                          INWAIT:                         ;
```

```
     004446  005737  001150          PMGR:  TST     INTFLG                  ;INTERRUPT?
     004452  001034                         BNE     PM3                     ;GO SERVICE IF YES
                                            ;
                                            ;
     004454  000001                         WAIT                            ;WAIT FOR SOMETHING
     004456  000773                         BR      PMGR                    ;
                                            ;
                                            ;
     004460  004737  005346          PM1:   JSR     PC,CLI                  ;SEE IF REAL COMMAND
     004464  005037  005530                 CLR     NCR                     ;REST CR INHIBIT
     004470  005037  005534                 CLR     ECHOEM                  ;RESET ECHO MODE IF SET
     004474  012737  007060  005526         MOV     #QUEST,MSGADR           ;IF BAD RETURN HERE
     004502  004737  005134                 JSR     PC,TTYOUT               ;OUT PUT A ?
                                            ;
                                            ;
     004506  012737  007062  005526  PMG1:  MOV     #PROMPT,MSGADR          ;NOW A PROMPT
     004514  005237  005530                 INC     NCR                     ;SET CR INHIBIT
     004520  005037  005534                 CLR     ECHOEM                  ;RESET ECHOEM IF SET
     004524  004737  005134                 JSR     PC,TTYOUT               ;
     004530  005037  005530                 CLR     NCR                     ;RESET CR INHIBIT
     004534  005037  005522                 CLR     CIMODE                  ;INSURE COMMAND MODE OFF
                                            ;
                                            ;
     004540  000001                  PM2:   WAIT                            ;WAIT FOR INPUT
     004542  000741                         BR      PMGR                    ;SEE WHAT IT IS
                                            ;
                                            ;
     004544  005737  005720          PM3:   TST     TTYINT                  ;READER INTERRUPT?
     004550  001004                         BNE     PM4                     ;SERVICE IF YES
     004552  005037  001150                 CLR     INTFLG                  ;RESET INTERRUPT FLAG
                                            ;                               ;AND CONSIDER THIS,INTERRUPT SPURIOUS
     004556  000770                         BR      PM2                     ;WAIT FOR MORE
                                            ;
                                            ;
     004560  000000                         HALT                            ;HALT IF NOT READER-BAD
                                            ;
                                            ;
     004562  004737  004650          PM4:   JSR     PC,TTYIN                ;GO SERRICE AND ECHO
     004566  013737  005546  005746         MOV     PPARMB,TTWORK           ;CHECK FOR CR
     004574  005337  005746                 DEC     TTWORK                  ;LOOK AT LAST DATA WORD
     004600  127727  001142  000015         CMPB    @TTWORK,#CR             ;WAS IT A CARRIAGE RETURN?
     004606  001317                         BNE     PMGR                    ;GO ON IF YES
     004610  005737  005522                 TST     CIMODE                  ;ARE WE IN COMMAND MODE?
     004614  001003                         BNE     PM5                     ;LEAVE IF YES
                                            ;
                                            ;
     004616  005237  005532                 INC     REQFLG                  ;SET REQUEST FLAG IF YES
     004622  000716                         BR      PM1                     ;SEE IF COMMAND
                                            ;
     004624  012737  005740  005526  PM5:   MOV     #LFCODE,MSGADR          ;ISSUE LINE FEED HERE
     004632  005037  005530                 CLR     NCR                     ;INSURE CR
     004636  005037  005534                 CLR     ECHOEM                  ;RESET ECHO MODE
     004642  004737  005134                 JSR     PC,TTYOUT               ;PRINT IT
                                            ;
                                            ;
     004646  000207                         RTS     PC                      ;AND RETURN
                                            ;
```

```
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;                 TTY INPUT ROUTINE
                                          ;
                                          ;
                                          ;           THIS ROUTINE ACCEPTS INPUT FROM THE
                                          ;         MONITOR AND STORES IT IN AN AREA FOR USE
                                          ;         LATER AS A COMMAND FUNCION OR AS A DISK/TEST PARAMETER.
                                          ;         THE ONLY REQUIREMENT FOR DATA ENTRY IS THAT REGISTER
                                          ;         R4 POINTS TO THE COMMAND BLOCK USED TO DEFINE ENTRY BUFFERS
                                          ;         AND SPECIAL FUNCTIONS.THIS ROUTINE IS CALLED ONLY BY THE
                                          ;         MONITOR INTERRUPT SERVICE CODE.
                                          ;
                                          ;
                                          ;
004650  005237  005534          TTYIN:    INC       ECHOEM            ;SET ECHO MODE FOR OUTPUT
                                          ;
004654  004537  015474                    JSR       R5,SAVE           ;SAVE CPU
                                          ;
004660  005737  001150                    TST       INTFLG            ;INTERRUPT FLAG SET?
004664  001002                            BNE       TTZ               ;GO ON IF YES
004666  000137  005336                    JMP       TTBAD             ;STOP IF NO
                                          ;
                                          ;
004672  005037  001150          TTZ:      CLR       INTFLG            ;RESET FLAG IF SET
                                          ;
                                          ;
004676  005737  005720                    TST       TTYINT            ;INPUT FLAG SET?
004702  001002                            BNE       TTZZ              ;GO ON IF YES
004704  000137  005336                    JMP       TTBAD             ;STOP IF NO
                                          ;
                                          ;
004710  005037  005720          TTZZ:     CLR       TTYINT            ;TESET IF YES
                                          ;
                                          ;
004714  005737  005536                    TST       TTYACT            ;ARE WE ON A LINE?
004720  001005                            BNE       TT1               ;YES SKIP BUFFER INIT
004722  013737  005544  005546            MOV       PARMB,PPARMB      ;START INPUT AT BEGINNING OF BUFFER
004730  005237  005536                    INC       TTYACT            ;AND SET LINE ACTIVE FLAG
                                          ;
                                          ;
004734  117700  000764          TT1:      MOVB      @TKB,R0                   ;FETCH READER WORD
004740  005277  000756                    INC       @TKS                      ;SET READER ENABLE
004744  142700  000200                    BICB      #200,R0                   ;LOSE PARITY BIT
004750  001003                            BNE       TT2                       ;CONTINUE PROCESSING IF NOT NULL
004752  005037  005536                    CLR       TTYACT                    ;RESET ACTIVE STATUS
004756  000207                            RTS       PC                        ;AND LEAVE
                                          ;
                                          ;
                                          ;
004760  123700  005732          TT2:      CMPB      CONC,R0                   ;IS IT A ^C ?
004764  001447                            BEQ       TT8                       ;IF YES OUTPUT IT
                                          ;
                                          ;
```

```
                                  ;
004766  123700  005734      TT3:  CMPB    CONU,R0              ;IS IT A CONTROL U ?
004772  001450                    BEQ     TT9                  ;IF YES PRINT IT
                                  ;
                                  ;
                                  ;
004774  123700  005736            CMPB    RUB,R0               ;IS IT A RUB OUT?
005000  001412                    BEQ     TT7                  ;RUB IT OUT IF YES
005002  110077  000540      TT4:  MOVB    R0,@PPARMB           ;LOAD INPUT WORD
005006  005237  005546            INC     PPARMB               ;ADJUST POINT ER FOR NEXT ENTRY
                                  ;
                                  ;
                                  ;
005012  000240              TT5:  NOP                          ;TEMP PATCH
                                  ;
                                  ;
                                  ;
005014  004737  005134      TT6:  JSR     PC,TTYOUT            ;IF NOT ECHO CHARACTER
                                  ;
                                  ;
005020  004537  015510      TT6A: JSR     R5,REST              ;RESTORE CPU
005024  000207                    RTS     PC                   ;AND RETURN
                                  ;
                                  ;
005026  023737  005544  005546  TT7:  CMP PARMB,PPARMB         ;IS THIS THE FIRST CHARACTER?
005034  001771                    BEQ     TT6A                 ;EXIT IF YES
005036  012737  007066  005526    MOV     #SLHMSG,MSGADR       ;POINT TO MESSAGE ADDRESS
005044  005337  005546            DEC     PPARMB               ;POINT TO ENTRY OF CONCERN
005050  117737  000472  007067    MOVB    @PPARMB,SLHMSG+1     ;LOAD CHARACTER FOR DELETION
005056  005037  007070            CLR     SLHMSG+2             ;LOAD TERMINATOR
005062  005037  005534            CLR     ECHOEM               ;RESET ECHO MODE
005066  005237  005530            INC     NCR                  ;SET CR INHIBIT
005072  004737  005134            JSR     PC,TTYOUT            ;PRINT MESSAGE
005076  005037  005530            CLR     NCR                  ;RESET NCR
005102  000746                    BR      TT6A                 ;AND EXIT
                                  ;
                                  ;
005104  012737  005732  005526  TT8:  MOV  #CONC,MSGADR   ;LOAD CONTROL C MESSAGE ADDRESS
005112  000403                    BR      TT10                 ;AND GO ON
                                  ;
                                  ;
005114  012737  005734  005526  TT9:  MOV  #CONU,MSGADR   ;LOAD CONTROL U MESSAGE ADDRESS
                                  ;
                                  ;
005122  005037  005536      TT10: CLR     TTYACT          ;RESET ACTIVE FLAG
005126  005037  005534            CLR     ECHOEM          ;RESET ECHO MODE
005132  000730                    BR      TT6                  ;AND GO ECHO
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;               TTYOUT.P11   2-DEC-77
                                  ;
                                  ;       TTY OUTPUT ROUTINE
```

```
                                        ;  =================================
                                        ;
                                        ;
                                        ;
                                        ;     THIS ROUTINE CONTROLS THE OUTPUT TO THE
                                        ;   CONSOLE DEVICE.EITHER AN ECHO CHARACTER
                                        ;   FROM THE KEYBOARD OR A COMPLETE MESSAGE FROM
                                        ;   A COMMAND.
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;   CALLING SEQUENCE*   JSR PC,TTYOUT
                                        ;
                                        ;     OR                          JSR     PC,TT08
                                        ;
                                        ;  R4   =  ADDRESS OF COMMAND BLOCK
                                        ;
                                        ;
                                        ;
005134  004537  015474          TTYOUT: JSR     R5,SAVE                 ;SAVE CPU REGISTERS
005140  005037  001150                  CLR     INTFLG                  ;RESET INTERRUPT FLAGS
005144  005037  005716                  CLR     TTOINT                  ;INCLUDING PRINTER
                                        ;
                                        ;
                                        ;
005150  005737  005534          TT02:   TST     ECHOEM          ;ECHO MODE?
005154  001407                          BEQ     TT05                    ;PRINT STRAIGHT LINE IF NO
                                        ;
                                        ;
                                        ;
005156  005737  005536          TT03:   TST     TTYACT          ;END OF INPUT?
005162  001452                          BEQ     TT013                   ;PRINT CR,LF IF YES
                                        ;
                                        ;
                                        ;
005164  013702  005546          TT04:   MOV     PPARMB,R2               ;LOAD CHARACTER ADDRESS
005170  005302                          DEC     R2                      ;MAKE ACCURATE ADDRESS
005172  000402                          BR      TT06                    ;START PRINT PROCESS
                                        ;
                                        ;
                                        ;
005174  013702  005526          TT05:   MOV     MSGADR,R2               ;POINT R2 TO MESSAGE ADDRESS
                                        ;
                                        ;
                                        ;
005200  005046                  TT06:   CLR     -(SP)                   ;DRAG CPU STATUS DOWN
005202  012746  005210                  MOV     #TT07,-(SP)             ;AND FAKE INTERRUPT
005206  000002                          RTI                             ;RTI TO LOAD PROCESSOR STATUS WORD
                                        ;
                                        ;
                                        ;
005210  012777  000100  000510  TT07:   MOV     #INTON,@TPS             ;ENABLE INTERRUPTS
005216  111277  000506                  MOVB    (R2),@TPB               ;PRINT CHARACTER
005222  004537  017342                  JSR     R5,WAIT                 ;WAIT FOR INTERRUPT
                                        ;
```

```
                                                ;
                                                ;
005226  005737  001150          TT08:   TST     INTFLG                  ;DID WE INTERUPT?
005232  001441                          BEQ     TTBAD                   ;SOMETHING'S WRONG SCRAM
005234  005037  001150                  CLR     INTFLG                  ;RESET INTERRUPT FLAG
                                        ;
                                        ;
005240  005737  005716          TT09:   TST     TTOINT                  ;WAS IT A TTOUT INTERRUPT?
005244  001434                          BEQ     TTBAD                   ;GET OUT IF NO
005246  005037  005716                  CLR     TTOINT                  ;REST OUTPUT INTERRUPT FLAG
                                        ;
                                        ;
005252  121227  000015                  CMPB    (R2),#CR                ;DID WE DO A CR ?
005256  001422                          BEQ     TT015                   ;PRINT LINE FEED IF YES
                                        ;
                                        ;
                                        ;
005260  121227  000012                  CMPB    (R2),#LF                ;HOW ABUOT A LINE FEED?
005264  001403                          BEQ     TT011                   ;EXIT IF YES
                                        ;
                                        ;
                                        ;
005266  005737  005534          TT010:  TST     ECHOEM          ;ARE WE IN ECHO MODE?
005272  001403                          BEQ     TT012                   ;KEEP PRINTING IF NO
                                        ;
                                        ;
                                        ;
005274  004537  015510          TT011:  JSR     R5,REST                 ;RESTORE CPU REGISTERS
005300  000207                          RTS     PC                      ;EXIT ROUTINE
                                        ;
                                        ;
                                        ;
005302  005202                  TT012:  INC     R2                      ;POINT TO NEXT DATA ENTRY
005304  105712                          TSTB    (R2)                    ;DID WE HIT A NULL YET?
005306  001334                          BNE     TT06                    ;PRINT NEXT IN LINE IF NO
                                        ;
                                        ;
                                        ;
005310  012702  005742          TT013:  MOV     #CRCODE,R2              ;POINT R2 TO CR ASCII
                                        ;
                                        ;
                                        ;
005314  005737  005530          TT014:  TST     NCR                     ;INHIBIT CR?
005320  001365                          BNE     TT011                   ;EXIT IF YES
005322  000726                          BR      TT06                    ;PRINT CR IF NO
                                        ;
                                        ;
                                        ;
005324  012702  005740          TT015:  MOV     #LFCODE,R2              ;POINT TO LINE FEED CODE
005330  005037  005536                  CLR     TTYACT          ;RESET INPUT LINE MODE(SET OR NOT)
005334  000767                          BR      TT014                   ;AND GO LINE FEED
                                        ;
                                        ;
                                        ;
005336  062706  000002          TTBAD:  ADD     #2,SP                   ;FIX STACK
005342  004537  003342                  JSR     R5,TSTCTL               ;EXIT
                                        ;
```

```
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       COMMAND LINE INTERPRETER
                                        ;
                                        ;
                                        ;                                       6-JAN-77
                                        ;
                                        ;        THE CLI IS CALLED BY THE PROGRAM MANAGER
                                        ;       TO SEARCH THE COMMAND TABLE FOR A POSSIBLE MATCH
                                        ;       OF A LINE JUST INPUTED AND A TABLE ENTRY. IF A FUNCTION IS
                                        ;       IS FOUND ,IT IS EXECUTED VIA A SUBROUTINE JUMP TO
                                        ;       THE ASSIGNED ADDRESS IMMEDIATLY FOLLOWING THE ASCII
                                        ;       DEFINITION IN THE COMMAND TABLE
                                        ;
                                        ;
 005346  005237  005522         CLI:    INC     CIMODE          ;SET COMMAND MODE
                                        ;
                                        ;
 005352  012702  005434         CI1:    MOV     #CMDTB,R2               ;POINT R 2 TO ADDRESS OF COMMAND TABLE
 005356  013703  005544                 MOV     PARMB,R3                ;POINT R3 TO FIRST CHARACTER
 005362  000240                         NOP                             ;TEMP LOC.
 005364  001403                         BEQ     CI4                     ;EXIT IF YES
                                        ;
                                        ;
 005366  005712                 CI2:    TST     (R2)                    ;END OF COMMAND TABLE
 005370  001011                         BNE     CI6                     ;GO ON IF NO
                                        ;
                                        ;
 005372  000240                 CI3:    NOP                             ;TEMP LOCATION
                                        ;
                                        ;
 005374  005037  005522         CI4:    CLR     CIMODE                  ;RESET COMMAND MODE
 005400  005037  005534                 CLR     ECHOEM                  ;RESET ECHO MODE
 005404  005037  005532                 CLR     REQFLG                  ;RESET REQUEST FLAG
 005410  000207                         RTS     PC                      ;RETURN TO PROGRAM MANAGER TO
                                                                        ;PRINT INVALID COMMAND
                                        ;
                                        ;
 005412  000240                 CI5:    NOP                             ;TEMP LOCATION
                                        ;
                                        ;
 005414  021322                 CI6:    CMP     (R3),(R2)+              ;HAVE WE FOUND A COMMAND

 005416  001402                         BEQ     CI8                     ;GO DO IT IF YES
                                        ;
                                        ;
 005420  005722                 CI7:    TST     (R2)+                   ;INCREMENT TABLE POINTER
 005422  000761                         BR      CI2                     ;AND TRY AGAIN
                                        ;
                                        ;
 005424  062706  000002         CI8:    ADD     #2,SP                   ;FIX STACK
 005430  011203                         MOV     (R2),R3                 ;POINT TO COMMAND ADDRES
 005432  000113                         JMP     (R3)                    ;AND EXECUTE
                                        ;
```

```
                                        ;
                                        ;
                                        ;         COMMAND TABLE
                                        ;
                                        ;  THIS TABLE IS THE COMPLETE COMMAND TABLE FOR
                                        ; FUNCTIONS ACCEPTED BY THE CONSOLE MONITOR.
                                        ; THE ORIGIN OF EACH FUNCTION FOLLOWS THE ASCI
                                        ; REPRESENTATION OF THE COMMAND.
                                        ;
                                        ;
                                        ;
005434      107              CMDTB:     .ASCII /G/                      ;START TESTING
005435      015                         .BYTE 15                        ;CR CODE
005436   010064                         .WORD RUN                       ; G
                                        ;
                                        ;
005440      104      113                .ASCII /DK/                     ;DISK PARAMETERS
005442   007740                         .WORD DISKIN                    ; DK
                                        ;
005444      124      123                .ASCII /TS/                     ;TEST PARAMETERS
005446   007176                         .WORD TESTP                     ;TS
                                        ;
                                        ;
005450      106      124                .ASCII /FT/                     ;FORMATTER PARAMETERS
005452   007072                         .WORD FMTTR                     ; FT
                                        ;
                                        ;
005454      120      122                .ASCII /PR/                     ;PRINT PROGRESS
005456   010126                         .WORD PRINT                     ;PR
                                        ;
                                        ;
005460      104      101                .ASCII /DA/                     ;ENTER DATE
005462   010676                         .WORD DATE                      ; DA
                                        ;
                                        ;
005464      123      103                .ASCII /SC/                     ;SCOPE MODE
005466   011236                         .WORD SCOPE                     ; SC
                                        ;
                                        ;
005470      106      117                .ASCII /FO/                     ;FORMAT COMMAND
005472   013240                         .WORD FMTDK                     ; FO
                                        ;
                                        ;
                                        ;
005474      110                         .ASCII /H/                      ;HELP COMMAND
005475      015                         .BYTE 15                        ; H
005476   013652                         .WORD HELP                      ;
                                        ;
                                        ;
005500      124      105                .ASCII /TE/                     ;TEST ERROR CONDITIONS
005502   012030                         .WORD TTERR                     ;TE
                                        ;
                                        ;
005504      103      117                .ASCII /CO/                     ;CONTINUE COMMAND
005506   015342                         .WORD CONTU                     ;
                                        ;
                                        ;
```

```
005510     122      122            .ASCII /RR/            ;REGISTER READ COMMAND
005512  013564                     .WORD RGREAD           ;   RR
                                   ;
005514     120      103            .ASCII /PC/            ;PASS COUNT PRINTOUT
005516  011640                     .WORD PASS             ; PC
                                   ;
                                   ;
                                   ;
005520  000000                     .WORD 0
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;            COMMAND BLOCK
                                   ;
                                   ;       THIS CONTROL BLOCK CONTAINS ALL
                                   ;    INFORMATION NECESSARY TO PERFORM ALL INPUT AND
                                   ;    OUTPUT FUNCTIONS ALL WELL AS FLAG CERTAIN
                                   ;    PARAMETERS DURING A TESTING COMMAND.
                                   ;
                                   ;
                                   ;
                                   ;
005522                   CONBL:
005522  000000           CIMODE:   .WORD 0                ;COMMAND MODE FLAG
005524  000000           TSTACT:   .WORD 0                ;TEST ACTIVE FLAG
005526  000000           MSGADR:   .WORD 0                ;MESSAGE ADDRESS FOR TTY OUT
005530  000000           NCR:      .WORD 0                ;INHIBIT CR FLAG
005532  000000           REOFLG:   .WORD 0                ;CLI REQUEST FLAG
005534  000000           ECHOEM:   .WORD 0                ;ECHO MODE TTY
005536  000000           TTYACT:   .WORD 0                ;TTY ACTIVE MODE
005540  000000           NUMBER:   .WORD 0                ;NUMBER MODE(INPUT)
005542  000000           TRASH:    .WORD 0                ; SKIP HEADING FLAG
005544  005550           PARMB:    .WORD TTBUF            ;INPUT BUFFER FOR TTY
005546  005550           PPARMB:   .WORD TTBUF            ;POINTER FOR TTY BUFFER
                                   ;
005550  000000           TTBUF:    .WORD 0                ;START OF BUFFER
        005716                     .=.+100.               ;BUFFER SIZE=100.
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;      OFFSETS IN CONTROL BLOCK
                                   ;   --------------------------------
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
005716  000000           TTOINT:   .WORD 0        ;TTY INTERRUPT FLAG
005720  000000           TTYINT:   .WORD 0        ;TTY INTERRUPT FLAG (INPUT )
                                   ;
005722  177560           TKS:      .WORD 177560   ;KEYBOARD READER STATUS REGISTER
```

```
005724  177562                  TKB:    .WORD 177562    ;KEYBOARD READER BUFFER REGISTER
005726  177564                  TPS:    .WORD 177564    ;PRINTER STATUS REGISTER
005730  177566                  TPB:    .WORD 177566    ;PRINTER BUFFER REGISTER
                                        ;
005732  000003                  CONC:   .WORD 3                 ;CONTROL C
005734  000025                  CONU:   .WORD 25                ;CONTROL U
005736  000177                  RUB:    .WORD 177               ;RUB OUT
005740  000012                  LFCODE: .WORD 12                ;LINE FEED
005742  000015                  CRCODE: .WORD 15                ;CARRIAGE RETURN
005744  000000                  IOERR:  HALT                    ;CRASH IF I/O ERROR
005746  000000                  TTWORK: .WORD 0                 ;TEMP LOCATION
005750  000000                  TSTINT: .WORD 0                 ;SAVE LOCATIONS FOR ACTIVE TEST INTERRUPT
005752  000000                          .WORD 0
005754  000000                          .WORD 0
005756  000000                          .WORD 0
005760  000000                          .WORD 0
005762  000000                          .WORD 0
005764  000000                          .WORD 0
005766  000000                          .WORD 0
                                        ;
                                        ;       TTY INPUT SERVICE ROUTINES
                                        ;       ===================================
                                        ;
                                        ;
005770  005237  001150          TTISEV: INC     INTFLG                  ;SET GENERAL INTERRUPT SERVICE
005774  005237  005720                  INC     TTYINT                  ;SET READER INTERRUPT FLAG
                                        ;
                                        ;
006000  005777  177716          TS1:    TST     @TKS                    ;ERRORS?
006004  000411                          BR      TS3                     ;IGNORE THEM NOW
                                        ;
                                        ;
006006  005237  001322                  INC     ERRFLG                  ;SET GENERAL ERROR FLAG
006012  005046                          CLR     -(SP)                   ;FORCE CPU LEVEL DOWN
006014  005066  177776                  CLR     -2(SP)                  ;FORCE CPU LEVEL DOWN
006020  012766  005744  177774          MOV     #IOERR,-4(SP)           ;AND POINT PC TOERROR ROUTINE
                                        ;
                                        ;
006026  000002                  TS2:    RTI                             ;EXIT HERE
                                        ;
                                        ;
006030  005737  005524          TS3:    TST     TSTACT                  ;CURRENTLY TESTING?
006034  001774                          BEQ     TS2                     ;EXIT IF NOT
                                        ;
                                        ;
006036  127737  177662  005732          CMPB    @TKB,CONC               ;IS IT A CONTROL C
006044  001405                          BEQ     TS3A                    ;SET UP TO CANCEL TESTING
                                        ;
                                        ;
006046  005037  001150                  CLR     INTFLG                  ;RESET INTERRUPT FLAG
006052  005037  005720                  CLR     TTYINT                  ;
006056  000763                          BR      TS2                     ;AND GO BACK TO TESTING
                                        ;
                                        ;
006060  005037  005524          TS3A:   CLR     TSTACT                  ;RESET ACTIVE FLAG
006064  010037  005750                  MOV     R0,TSTINT               ;START SAVING
006070  012700  005750                  MOV     #TSTINT,R0              ;LOAD SAVE ADDRESS FOR CPU
```

```
     006074  005720                    TST     (R0)+             ;POIN TO R1 ENTRY
     006076  010120                    MOV     R1,(R0)+          ;SAVE CPU
                                       ;
     006100  010220                    MOV     R2,(R0)+          ;
     006102  010320                    MOV     R3,(R0)+          ;
     006104  010420                    MOV     R4,(R0)+          ;
     006106  010520                    MOV     R5,(R0)+          ;
     006110  011610                    MOV     (SP),(R0)         ;SAVE RETURN ADDRESS
                                       ;
                                       ;
     006112  005737  005522    TS4:    TST     CIMODE1           ;IN COMMAND MODE
     006116  001343                    BNE     TS2               ;EXIT IF YES
                                       ;
                                       ;
                                       ;
     006120  005066  177776    TS5:    CLR     -2(SP)            ;FORCE ENTRY INTO PROGRAM MANAGER
     006124  012716  004446            MOV     #PMGR,(SP)        ;
     006130  005037  005534            CLR     ECHOEM            ;AND RETURN
     006134  000734                    BR      TS2               ;AND RTI
                                       ;
                                       ;
                                       ;       TTY OUTPUT SERVICE ROUTINE
                                       ;
                                       ;
     006136  005237  005716    TTOSEV: INC     TTOINT            ;SET PRINTER INTERRUPT FLAG
     006142  005237  001150            INC     INTFLG            ;SET GENERAL INTERRUPT FLAG
                                       ;
                                       ;
     006146  005777  177554    TOS1:   TST     @TPS              ;ERRORS?
     006152  100005                    BPL     TOS2              ;EXIT IF NO ERROR
                                       ;
                                       ;
     006154  005237  001322            INC     ERRFLG            ;SET GENERAL ERROR FLAG
     006160  005046                    CLR     -(SP)             ;FORCE CPU LEVEL DOWN
     006162  012746  005744            MOV     #IOERR,-(SP)      ;POINT PC TO I/O ERROR SECTION
                                       ;
                                       ;
     006166  000002            TOS2:   RTI                       ;EXIT HERE
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
     006170  012700  000060    TTSTAR: MOV     #60,R0            ;POINT TO VECTOR AREA FOR
                                                                 ;KEYBOARD
     006174  012720  005770            MOV     #TTISEV,(R0)+     ;LOAD READER ROUTINE ADDR
     006200  012720  000340            MOV     #340,(R0)+        ;LOAD LEVEL (7)
     006204  012720  006136            MOV     #TTOSEV,(R0)+     ;LOAD PRINTER ROUTINE
     006210  012710  000340            MOV     #340,(R0)         ;LOAD LEVEL (7)
                                       ;
                                       ;
```

```
006214  005037  005522                    CLR    CIMODE                  ;RESET COMMAND MODE
006220  005037  005534                    CLR    ECHOEM                  ;REST ECHO MODE
006224  005037  005530                    CLR    NCR                     ;RESET CARRIAGE RETURN INHIBIT
006230  005037  005524                    CLR    TSTACT                  ;RESET TEST ACTIVE MODE
006234  005037  005532                    CLR    REQFLG                  ;RESET REQUEST FLAG
006240  005037  005720                    CLR    TTYINT                  ;RESET INTERRUPT FLAG
006244  005037  005716                    CLR    TTOINT                  ;BOTH IN AND OUPUT
                                          ;
006250  012777  000100  177444            MOV    #100,@TKS               ;SET INTERRUPT ENABLE
006256  012777  000100  177442            MOV    #100,@TPS               ;FOR READER AND PRINTER
006264  005277  177432                    INC    @TKS                    ;ENABLE READER
                                          ;
006270  005737  005542            TRSH:   TST    TRASH                   ;  SKIP ALL THIS BALONEY?
006274  001402                            BEQ    TRSH1                   ; NO PRINT IT
006276  000137  004506                    JMP    PMG1                    ; YES WAIT FOR INPUT
                                          ;
006302                            TRSH1:
                                          ;
                                          ;
006302  012737  005742  005526            MOV    #CRCODE,MSGADR          ;SET UP FOR CLEAN SCREEN
006310  004737  005134                    JSR    PC,TTYOUT               ;ISSUE CR
006314  012737  005740  005526            MOV    #LFCODE,MSGADR          ;NOW ISSUE LINE FEED
006322  004737  005134                    JSR    PC,TTYOUT               ;
                                          ;
                                          ;
006326  012737  006424  005526            MOV    #P2MSG,MSGADR           ;ENTER OUTPUT MESSAGE ADDRESS
006334  004737  005134                    JSR    PC,TTYOUT               ;PRINT HEADING
006340  012737  006507  005526            MOV    #STRMSG,MSGADR          ;UNDER LINE HEADING
006346  004737  005134                    JSR    PC,TTYOUT               ;HERE
                                          ;
                                          ;
006352  012737  005740  005526            MOV    #LFCODE,MSGADR          ;PRINT 2 LINE FEEDS HERE
006360  004737  005134                    JSR    PC,TTYOUT               ;
006364  004737  005134                    JSR    PC,TTYOUT               ;
                                          ;
006370  012737  006603  005526            MOV    #P3MSG,MSGADR           ;NOW TELL PEOPLE ABOUT ANSWERS
006376  004737  005134                    JSR    PC,TTYOUT               ;
                                          ;
006402  012737  005740  005526            MOV    #LFCODE,MSGADR          ;ONE LINE FEED HERE
006410  004737  005134                    JSR    PC,TTYOUT               ;
                                          ;
                                          ;
006414  004737  005134                    JSR    PC,TTYOUT               ;
                                          ;
                                          ;
006420  000137  010676                    JMP    DATE                    ;ASK FOR DATA NOW
                                                                         ;PROGRAM MANAGER
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;      DIAGNOSTIC HEADING MESSAGE
                                          ;
                                          ;
                                          ;
006424     040     040     040    P2MSG:  .ASCII /        PHOENIX 211 DISK CONTROLLER DIAGNOSTIC /
```

```
          006427   040    040    040
          006432   040    040    040
          006435   040    040    120
          006440   110    117    105
          006443   116    111    130
          006446   040    062    061
          006451   061    040    104
          006454   111    123    113
          006457   040    103    117
          006462   116    124    122
          006465   117    114    114
          006470   105    122    040
          006473   104    111    101
          006476   107    116    117
          006501   123    124    111
          006504   103    040
     1141 006506   000                          .BYTE 0
     1142 006507   052    052    052   STRMSG:  .ASCII /************************************************************/
          006512   052    052    052
          006515   052    052    052
          006520   052    052    052
          006523   052    052    052
          006526   052    052    052
          006531   052    052    052
          006534   052    052    052
          006537   052    052    052
          006542   052    052    052
          006545   052    052    052
          006550   052    052    052
          006553   052    052    052
          006556   052    052    052
          006561   052    052    052
          006564   052    052    052
          006567   052    052    052
          006572   052    052    052
          006575   052    052    052
          006600   052    052
     1143 006602   000                          .BYTE 0
     1144 006603   040    040    040   P3MSG:   .ASCII /      TYPE  H <CR>  FOR HELP/
          006606   040    040    124
          006611   131    120    105
          006614   040    040    110
          006617   040    074    103
          006622   122    076    040
          006625   040    106    117
          006630   122    040    110
          006633   105    114    120
     1145 006636   000                          .BYTE 0
     1146                                       ;
     1147                                       .EVEN
     1148                                       ;
     1149                                       ;
     1150                                       ;
     1151                                       ;
     1152                                       ;
     1153                                       ;
     1154                                       ;
```

```
                                        ;       ASCII NUMBER CONVERSION
                                        ;
                                        ;        THS ROUTINE CONVERTS AN ASCII NUMBER
                                        ;       INTO OCTAL FORMAT.
                                        ;        THE NUMBER TO BE CONVERTED IS LOCATED STARTING
                                        ;       AT INPUT BUFFER "PARMB" AND IS ENDED
                                        ;       BY DETECTING A CR WORD
                                        ;
                                        ;       NO CHARACTERS LARGER THAN A 7 ARE ACCEPTED
                                        ;       AND IF A BAD CHARACTER IS RECEIVED,
                                        ;       THE FLAG "BADNUM" IS SET.
                                        ;
                                        ;        THE OCTAL NUMBER IS LOCATED IN "DATA" AT
                                        ;       EXIT
                                        ;

006640  010046                  OCTAL:  MOV     R0,-(SP)                ;SAVE R0,R1,R2
006642  010146                          MOV     R1,-(SP)                ;
006644  010246                          MOV     R2,-(SP)                ;
006646  005037  006750                  CLR     BADNUM                  ;START CLEAN
006652  013701  005544                  MOV     PARMB,R1                ;POINT TO INPUT BUFFER
006656  005000                          CLR     R0                      ;CLR R0
006660  111102                  COCTI:  MOVB    (R1),R2                 ;FETCH CHARACTER
006662  162702  000060                  SUB     #60,R2                  ;LOSE ASCII PART
006666  100417                          BMI     COCTER                  ;LEAVE IF ERROR
006670  022702  000010                  CMP     #10,R2                  ;SEE IF GREATER THAN 7
006674  002414                          BLT     COCTER                  ;LEAVE IF YES WITH ERROR
006676  006100                          ROL     R0                      ;SHIFT DOWN THREE BITS FOR
006700  103412                          BCS     COCTER                  ;OVERFLOW = ERROR
006702  006100                          ROL     R0                      ;
006704  103410                          BCS     COCTER                  ;
006706  006100                          ROL     R0                      ;
006710  103406                          BCS     COCTER
006712  060200                          ADD     R2,R0                   ;LOAD FINISHED CHARACTER
006714  105721                          TSTB    (R1)+                   ;LOOK AT NEXT CHARACTER
006716  121127  000015          COCTDN: CMPB    (R1),#CR                ;CARRIAGE RETURN?
006722  001404                          BEQ     COCEX                   ;EXIT IF YES
006724  000755                          BR      COCTI                   ;
006726  005237  006750          COCTER: INC     BADNUM                  ;SET BAD NUMBER FLAG
006732  000402                          BR      CEXIT                   ;AND LEAVE
                                        ;
                                        ;
006734  010037  006752          COCEX:  MOV     R0,DATA                 ;LOAD GOOD NUMBER
                                        ;
                                        ;
006740  012602                  CEXIT:  MOV     (SP)+,R2                ;RELOAD REGS
006742  012601                          MOV     (SP)+,R1                ;
006744  012600                          MOV     (SP)+,R0                ;
006746  000207                          RTS     PC                      ;AND LEAVE
                                        ;
006750  000000                  BADNUM: .WORD 0                 ;BAD DATA WORD FLAG
                                        ;
006752  000000                  DATA:   .WORD 0                 ;CONVERTED ASCI TO OCTAL WORD
                                        ;
                                        ;
                                        ;
                                        ;
```

```
                                          ;
                                          ;
                                          ;
                                          ;       OCTAL TO ASCII CONVERSION ROUTINE
                                          ;
                                          ;
                                          ;         THIS ROUTINE CONVERTS AN OCTAL NUMBER TO IT'S
                                          ;       ASCII EQUIVILENT.
                                          ;
                                          ;
                                          ;
                                          ;        THE NUMBER TO BE WORKED ON SHOLD
                                          ;        THE NUMBER TO BE CONVERTED IS IN REG R3
                                          ;        THE OUTPUT GENERATED WILL BE IN OUTBUF AREA
                                          ;       AND A NULL WILL TERMINATE STRING.
                                          ;
                                          ;
006754  010146                   ASCII:   MOV     R1,-(SP)                ;SAVE R1,R2,R3
006756  010246                            MOV     R2,-(SP)                ;
006760  012746  000005                    MOV     #5,-(SP)                ;SET COUNTER
                                          ;
006764  012701  010574                    MOV     #OUTBUF,R1              ;POINT TO BUFFER AREA
006770  000241                            CLC                             ;CLEAR CARRY BIT
006772  006103                            RCL     R3                      ;SHIFT CARRY IN
006774  006103                            ROL     R3                      ;INTO BIT 0
006776  010302                            MOV     R3,R2                   ;LOAD INTO R2
007000  006003                            ROR     R3                      ;RETURN CARRY INTO R3
007002  042702  177776                    BIC     #177776,R2              ;LOOSE GARBARGE
007006  062702  000060                    ADD     #60,R2                  ;MAKE ASCII
007012  110221                            MOVB    R2,(R1)+                ;LOAD INTO BUFFER AREA
007014  006103                   CASCI:   ROL     R3                      ;NOW SHIFT HALF BYTES IN
007016  006103                            ROL     R3                      ;
007020  006103                            ROL     R3                      ;
007022  006103                            ROL     R3                      ;
007024  010302                            MOV     R3,R2                   ;LOAD INTO R2 FOR CONVERSION
007026  006003                            ROR     R3                      ;RETURN CARRY BIT INTO R3
007030  042702  177770                    BIC     #177770,R2              ;LOOSE GARBARGE AGAIN
007034  062702  000060                    ADD     #60,R2                  ;MAKE HALF BYTE ASCII
007040  110221                            MOVB    R2,(R1)+                ;LOAD IN OUPUT BUFFER
007042  005316                            DEC     (SP)                    ;DECREMENT SHIFT COUNTER
007044  001363                            BNE     CASCI                   ;GO ON IF MORE
007046  005011                            CLR     (R1)                    ;SET NULL FOR OUPUT FLAG FINISH
007050  005726                            TST     (SP)+                   ;FIX STACK
007052  012602                            MOV     (SP)+,R2                ;RESTORE R2 REG.
007054  012601                            MOV     (SP)+,R1                ;AND R1
007056  000207                            RTS     PC                      ;LEAVE THIS PLACE
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;       MISC. OUTPUT MESSAGES
                                          ;
                                          ;
007060     077                   QUEST:   .ASCII /?/
007061     000                            .BYTE 0
007062     052                   PROMPT:  .ASCII /*/
007063     007                            .BYTE 7
```

```
007064     000                          .BYTE 0
                                        .EVEN
007066     134                SLHMSG:   .ASCII /\/
                                        ;
007067     000                          .BYTE 0
007070     000000                       .WORD 0
                                        .EVEN
                                        ;
                                        ;
                                        ;
                                        ;       CONTROLLER PARAMETER COMMAND
                                        ;
                                        ;        THE CONTROLLER COMMAND ALLOWS THE OPERATOR
                                        ;       TO CHANGE SEVERAL CONTROLLER ORIENTED COMMANDS
                                        ;       THERE ARE 5 CHANGES :
                                        ;
                                        ;
                                        ;       1.DCSR OR BASE ADDRESS
                                        ;       2.INTERRUPT VECTOR
                                        ;       3.ECC OPTION
                                        ;       4.MULTIPLE CPU OPTION
                                        ;       5.AUTO-SWITCH OPTION
                                        ;
                                        ;
                                        ;
007072     005037  005532     FMTTR:    CLR     REQFLG                  ;RESET REQUEST FLAG
007076     005037  005524               CLR     TSTACT                  ;RESET TEST ACTIVE
007102     005237  005522               INC     CIMODE                  ;SET COMMAND MODE
007106     004537  015474               JSR     R5,SAVE                 ;SAVE CPU
007112     012700  002054               MOV     #DCSMSG,R0              ;R0=MESSGAE ADDRESS OF ADDRESS
007116     012701  000002               MOV     #2,R1                   ;FIRST ENTRY COUNR = 2
007122     012702  001070               MOV     #DCSRA,R2               ;R2 = ENTRY AREA
007126     005237  005540               INC     NUMBER                  ;SET NUMBER FLAG
007132     004737  007502               JSR     PC,LOADIN               ;ENTER P'S
007136     005037  005540               CLR     NUMBER                  ;RESET NUMBER FLAG
007142     012700  002060               MOV     #DECGSM,R0              ;LOAD NEW MESSAGE
007146     012702  001074               MOV     #ECC,R2                 ;AND NEW DATA AREA
007152     012701  000003               MOV     #3,R1                   ;LETTER COUNT=3
007156     004737  007502               JSR     PC,LOADIN               ;ENTER YES/NO DATA
007162     005037  005522               CLR     CIMODE                  ;RESET COMMAND MODE
007166     004537  015510               JSR     R5,REST                 ;RESTORE CPU
007172     000137  004506               JMP     PMG1                    ;WAIT FOR SOMETHING ELSE
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       TEST PARAMETERS
                                        ;
                                        ;         THE TEST PARAMETER COMMAND ALLOWS
                                        ;       THE OPERATOR TO ENTER 8 TEST PARAMETERS
                                        ;
                                        ;
                                        ;       1.STARTING TEST NUMBER
                                        ;       2.ENDING TEST NUMBER
                                        ;       3.STARTING DRIVE NUMBER
                                        ;       4.ENDING DRIVE NUMBER
```

```
1326                                              ;       5.CONTINUOUS TESTING MODE
1327                                              ;       6.REPEAT ON ERROR MODE
1328                                              ;       7.INHIBIT PASS PRINTOUT
1329                                              ;       8.INHIBIT ERROR PRINTOUT
1330                                              ;
1331                                              ;
1332                                              ;
1333 007176 004537 015474            TESTP:  JSR     R5,SAVE                         ;SAVE CPU
1334 007202 005237 005522                    INC     CIMODE                  ;SET COMMAND MODE
1335 007206 023727 001232 000045             CMP     CURTN,#45                       ;DID WE FORMAT LAST?
1336 007214 001005                           BNE     TESTP1                          ;GO NO IF NO
1337 007216 005037 001212                    CLR     STRTN                           ;IF YES RESET
1338 007222 012737 000030 001210             MOV     #30,SRTTN                       ;SET OLD DEFAULTS IF YES
1339 007230                          TESTP1:
1340 007230 005037 005524                    CLR     TSTACT                          ;RESET TEST ACTIVE
1341 007234 005037 005532                    CLR     REQFLG                          ;RESET REQUEST FLAG
1342 007240 012700 001356                    MOV     #TSTADR,R0                      ;R0=MESSAGE ADDRESS
1343 007244 012701 000004                    MOV     #4,R1                           ;R1=# OF ENTRIES
1344 007250 012702 001210                    MOV     #SRTTN,R2                       ;R2=ENTRY AREA
1345 007254 005237 005540                    INC     NUMBER                          ;SET NUMBER FLAG
1346 007260 004737 007502                    JSR     PC,LOADIN                       ;DO IT
1347 007264 005037 005540                    CLR     NUMBER                          ;RESET NUMBER FLAG
1348 007270 012701 000005                    MOV     #5,R1                           ;RELOAD FOR 5 ENTRIES
1349 007274 012700 001366                    MOV     #TSTBDR,R0                      ;LOAD NEW MESSAGE
1350 007300 012702 001220                    MOV     #REPFLG,R2                      ;AND ANSWE ADDRESS
1351 007304 004737 007502                    JSR     PC,LOADIN                       ;FETCH THOSE FLAGS
1352 007310 005037 005522                    CLR     CIMODE                  ;RESET COMMAND MODE
1353 007314 000137 004506                    JMP     PMG1                            ;WAIT FOR WORK
1354                                              ;
1355                                              ;
1356 007320    040    111    116  PINMSG: .ASCII / INHIBIT PASS COUNT PRINTOUT? Y OR N/
     007323    110    111    102
     007326    111    124    040
     007331    120    101    123
     007334    123    040    103
     007337    117    125    116
     007342    124    040    120
     007345    122    111    116
     007350    124    117    125
     007353    124    077    040
     007356    131    040    117
     007361    122    040    116
1357 007364    000                            .BYTE 0
1358 007365    040    111    116  EINMSG: .ASCII / INHIBIT ERROR PRINTOUT? Y OR N/
     007370    110    111    102
     007373    111    124    040
     007376    105    122    122
     007401    117    122    040
     007404    120    122    111
     007407    116    124    117
     007412    125    124    077
     007415    040    131    040
     007420    117    122    040
     007423    116
1359 007424    000                            .BYTE 0
1360 007425    040    123    113  COEMSG: .ASCII / SKIP ON ERROR (TESTS 31,32 ONLY) ? Y OR N /
     007430    111    120    040
```

```
     007433    117     116     040
     007436    105     122     122
     007441    117     122     040
     007444    050     124     105
     007447    123     124     123
     007452    040     063     061
     007455    054     063     062
     007460    040     117     116
     007463    114     131     051
     007466    040     077     040
     007471    131     040     117
     007474    122     040     116
     007477    040
     007500    000                              .BYTE 0
                                                .EVEN
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;                 LOADIN .P11  11-JAN-78
                                                ;
                                                ;
                                                ;
                                                ;            THIS SUBROUTINE IS USED BY SEVERAL MONITOR
                                                ;        COMMANDS TO HANDLE INPUT FROM THE KEYBOARD.
                                                ;        THE EXPECTED DATA IS IN TWO FORMS;ONE IS
                                                ;        AN OCTAL NUMBER AND THE SECOND IS EITHER
                                                ;        A "Y" OR AN "N".ANY OTHER INPUT
                                                ;        WILL BE IGNORED.UPON ENTRY THE FLAG "NUMBER" MUST
                                                ;        BE SET IF THE INPUT DESIRED IS TO BE IN OCTAL FORM.
                                                ;
                                                ;
                                                ;
                                                ;        CALLING SEQUENCE:  JSR  PC,LOADIN
                                                ;
                                                ;        R0=MESSAGE ADDRESS
                                                ;        R1=# OF EXPECTED DATA LINES
                                                ;        R2=ADDRESS OF DATA LOCATIONS
                                                ;
                                                ;
                  000040                        SPACE=40                          ;SPACE DEFINITION
     007502       012037  005526        LOADIN: MOV     (R0)+,MSGADR              ;POINT TO MESSAGE ADDRESS
     007506       005037  005534                CLR     ECHOEM                    ;RESET ECHO MODE
     007512       005237  005530                INC     NCR                       ;SET NO CR FLAG
     007516       004737  005134                JSR     PC,TTYOUT                 ;TYPE QUESTION
     007522       005737  005540                TST     NUMBER                    ;PRINT DEFAULT?
     007526       001415                        BEQ     LD1A                      ;GO NO IF NO
     007530       011203                        MOV     (R2),R3                   ;SET UP TO PRINT IF YES
     007532       004737  006754                JSR     PC,ASCII                  ;MAKE ASCII
     007536       012737  010574  005526        MOV     #OUTBUF,MSGADR            ;LOAD OUTPUT BUFFER
     007544       004737  005134                JSR     PC,TTYOUT                 ;OUTPUT IT
                                                ;
     007550       012737  000040  010574        MOV     #SPACE,OUTBUF             ;ENCLUDE SPACE
     007556       004737  005134                JSR     PC,TTYOUT                 ;PRINT IT
                                                ;
```

```
007562  004737  004446           LD1A:   JSR     PC,INWAIT               ;COLLECT ANSWER
                                         ;
                                         ;
007566  117703  175752           LD1:    MOVB    @PARMB,R3               ;FETCH FIRST ENTRY
007572  042703  000200                   BIC     #BIT7,R3                ;LOSE PARITY BIT
007576  020327  000015                   CMP     R3,#CR                  ;CARRIAGE RETURN?
007602  001013                           BNE     LD3                     ;GO ON IF NO
                                         ;
                                         ;
007604  005722                   LD2:    TST     (R2)+                   ;DEFAULT ANSWER
007606  012737  005740  005526           MOV     #LFCODE,MSGADR          ;ISSUE LINE FEED HERE
007614  005037  005534                   CLR     ECHOEM                  ;RESET ECHO MODE
007620  004737  005134                   JSR     PC,TTYOUT               ;PRINT IT
                                         ;
                                         ;
007624  005301                           DEC     R1                      ;DECREMENT ENTRY NUMBER
007626  001325                           BNE     LOADIN                  ;DO ANOTHER
007630  000207                   LDEX:   RTS     PC                      ;RETURN HERE
                                         ;
                                         ;
007632  020327  000105           LD3:    CMP     R3,#105                 ;IS IT AN "E"
007636  001774                           BEQ     LDEX                    ;EXIT IF YES
                                         ;
                                         ;
007640  005737  005540           LD4:    TST     NUMBER                  ;EXPECTING NUMBER?
007644  001423                           BEQ     LD7                     ;CHRCK IT IF YES
007646  004737  006640                   JSR     PC,OCTAL                ;
007652  005737  006750                   TST     BADNUM                  ;GOOD?
007656  001413                           BEQ     LD6                     ;GO ON IF YES
                                         ;
                                         ;
007660  012737  007060  005526   LD5:    MOV     #QUEST,MSGADR           ;LOAD ADDR OF "?"
007666  005037  005534                   CLR     ECHOEM                  ;RESET ECHO MODE
007672  005037  005530                   CLR     NCR                     ;RESET CR INHIBIT
007676  004737  005134                   JSR     PC,TTYOUT               ;
007702  024042                           CMP     -(R0),-(R2)             ;RESET POINTERS
007704  000676                           BR      LOADIN                  ;TRY IT AGAIN
                                         ;
                                         ;
007706  013712  006752           LD6:    MOV     DATA,(R2)               ;ENTER DATA WORD
007712  000734                           BR      LD2                     ;AND GO ON
                                         ;
                                         ;
007714  020327  000131           LD7:    CMP     R3,#131                 ;SEE IF
007720  001002                           BNE     LD10                    ;IT IS A "Y"
007722  005212                           INC     (R2)                    ;SET FLAG IF YES
007724  000727                           BR      LD2                     ;AND GO ON
                                         ;
                                         ;
007726  020327  000116           LD10:   CMP     R3,#116                 ;IS IT AN "N"
007732  001352                           BNE     LD5                     ;IGNORE IT IF NO
007734  005012                           CLR     (R2)                    ;RESER FLAG IF YES
007736  000722                           BR      LD2                     ;AND GO ON
                                         ;
                                         ;
                                         ;
                                         ;
```

```
                                        ;
                                        ;       DISK INPUT COMMAND
                                        ;
                                        ;        THIS COMMAND INPUTS THE DISK PARAMETERS
                                        ;       USED IN TESTING THE PHOENIX 211.
                                        ;        THE FOLLOWING ENTRIES ARE ALLOWED:
                                        ;
                                        ;               1.MAXIMUM SECTOR
                                        ;               2.MAXIMUM HEAD
                                        ;               3.MAXIMUM CYLINDER
                                        ;               4.WORDS PER SECTOR
                                        ;
                                        ;
007740  005037  005532          DISKIN: CLR     REQFLG                  ;RESET REQUEST FLAG
007744  005037  005524                  CLR     TSTACT                  ;RESET TEST ACTIVE
007750  005037  005534                  CLR     ECHOEM                  ;RESET ECHO MODE
007754  005237  005522                  INC     CIMODE          ;SET COMMAND MODE
007760  004537  015474                  JSR     R5,SAVE                 ;SAVE CPU
007764  012700  001642                  MOV     #DISKP,R0               ;R0 IS ADDRESS OF MESSAGES
007770  012701  000004                  MOV     #4,R1                   ;R1= NUMBER OF ENTRIES
007774  012702  001040                  MOV     #MAXSEC,R2              ;R2=PARAMETER ADDRESS
010000  005237  005540                  INC     NUMBER
010004  004737  007502                  JSR     PC,LOADIN               ;ENTER DATA
010010  005037  005540                  CLR     NUMBER                  ;CLR NUMBER FLAG
010014  005037  005522                  CLR     CIMODE                  ;RESET COMMAND MODE
010020  013737  001042  001060          MOV     MAXHD,MAXHED            ;FIX HEAD SELECTION
010026  012702  000007                  MOV     #7,R2                   ;SHIFT
010032  006237  001060          DISKHD: ASR     MAXHED                  ;SHIFT
010036  005302                          DEC     R2                      ;
010040  001374                          BNE     DISKHD                  ;REPEAT UNTIL DONE
010042  013737  001046  001050          MOV     WPSEC,POSWP             ;
010050  005437  001050                  NEG     POSWP                   ;MAKE POSITIVE WORDS PER SECTOR
010054  004537  015510                  JSR     R5,REST                 ;RESTORE CPU
010060  000137  004506                  JMP     PMG1                    ;WAIT FOR NEXT STUFF
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       "GO" COMMAND
                                        ;
                                        ;         THE GO COMMAND IS A VERY SIMPLE FUNCTION.
                                        ;       IT IS ASSUMED THAT ALL TEST PARAMETERS ARE SETUP
                                        ;       AND DISK/CONTROLLER VALUES ARE CORRECT.
                                        ;        A JUMP TO TEST EXECUTIVE IS THEN PERFORMED
                                        ;
                                        ;
010064  005037  005532          RUN:    CLR     REQFLG                  ;RESET REQUEST FLAG
010070  005037  005522                  CLR     CIMODE          ;
010074  005237  005524                  INC     TSTACT          ;SET TEST ACTIVE FLAG
010100  023727  001212  000045          CMP     STRTN,#45       ; DID WE JUST DO A FORMAT?
010106  001005                          BNE     RUN1            ; GO ON IF NO
                                        ;
010110  005037  001212                  CLR     STRTN           ; DEFAULT TO TEST 0
010114  012737  000030  001210          MOV     #30,STPTN       ; AND END WITH 30
                                        ;
010122                          RUN1:
```

```
010122  000137  003044                  JMP     TSTART                  ;START TESTING
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       PRINT PROGRESS COMMAND
                                        ;
                                        ;         THIS COMMAND PRINTS THE CURRENT TEST PROGRESS
                                        ;       OF THE PHOENIX 211 CONTROLLER.THE VALUES OUTPUTED
                                        ;       ARE THE FOLLOWING:
                                        ;
                                        ;               1.PASS COUNT
                                        ;               2.PASS DEFINITION (STARTING AND ENDING TEST)
                                        ;               3.CURRENT TEST
                                        ;               4.ERROR COUNT
                                        ;               5.DATE
                                        ;
                                        ;
                                        ;
                                        ;
010126  004537  015474          PRINT:  JSR     R5,SAVE                 ;SAVE CPU
010132  005037  005524                  CLR     TSTACT                  ;RSET TEST ACTIVE
010136  005237  005522                  INC     CIMODE                  ;SET COMMAND MODE
010142  005037  005534                  CLR     ECHOEM                  ;CLEAR ECHO MODE
010146  005037  005532                  CLR     REQFLG                  ;RESET REQUEST FLAG
010152  012737  010334  005526          MOV     #PNTMSG,MSGADR          ;LOAD PRINT HEADING
010160  004737  005134                  JSR     PC,TTYOUT               ;PRINT IT
010164  012737  005740  005526          MOV     #LFCODE,MSGADR          ;LINE FEED
010172  004737  005134                  JSR     PC,TTYOUT               ;
010176  004737  005134                  JSR     PC,TTYOUT               ;
                                        ;
                                        ;
010202  005237  005530                  INC     NCR                     ;INHIBIT CR
010206  012737  002045  005526          MOV     #DTE,MSGADR             ;POINT TO DATE MESSAGE
010214  004737  005134                  JSR     PC,TTYOUT               ;PRINT DATE:
010220  005037  005530                  CLR     NCR                     ;RESER CR INHIBIT
010224  005037  005524                  CLR     TSTACT                  ;RESET TEST ACTIVE
010230  012737  011222  005526          MOV     #DATEW,MSGADR           ;POINT TO ACTUAL DATE
010236  004737  005134                  JSR     PC,TTYOUT               ;PRINT DATE
                                        ;
                                        ;
010242  012737  005740  005526          MOV     #LFCODE,MSGADR          ;PRINT LINE FEED (2)
010250  004737  005134                  JSR     PC,TTYOUT               ;
010254  004737  005134                  JSR     PC,TTYOUT               ;
                                        ;
                                        ;
010260  004537  013104                  JSR     R5,POUT                 ;PRINT TEST PARAMETERS
010264  010322                          .WORD PRMSG                     ;TEST MESSAGES
010266  010306                          .WORD PRDATA                    ;TEST P'S
010270  000005                          .WORD 5                         ;NUMBER OF MESSAGES
                                        ;
                                        ;
```

```
010272  005037  005522                 CLR     CIMODE                     ;RESET COMMAND MODE
010276  004537  015510                 JSR     R5,REST                    ;RESTORE CPU
010302  000137  004506                 JMP     PMG1                       ;DO SOMETHING ELSE
                                       ;
                                       ;       DATA TABLE OF LOCS FOR PRINT COMMAND
                                       ;
                                       ;
010306  001236                 PRDATA: .WORD PASCNT               ;PASS COUNTER
010310  001212                         .WORD STRTN                ;STARTING TEST NUMBER
010312  001210                         .WORD STPTN                ;ENDING TEST NUMBER
010314  001232                         .WORD CURTN                ;CURRENT TEST NUMBER
010316  001322                         .WORD ERRFLG               ;ERROR FLAG
010320  000000                         .WORD 0                    ;WHO KNOWS?
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;       PRINT COMMAND MESSAGES
                                       ;
                                       ;
010322  010403                 PRMSG:  .WORD PASMSG               ;PASS COUNT MESSAGE
010324  010434                         .WORD T1MSG                ;STARTING TEST MESSAGE
010326  010464                         .WORD T2MSG                ;ENDING TEST MESSAGE
010330  010514                         .WORD CMSSG                ;CURRENT TEST MESSAGE
010332  010544                         .WORD ERCNTM               ;ERROR FLAG MESAGE
                                       ;
                                       ;
                                       ;
010334     011     120     110 PNTMSG: .ASCII /        PHOENIX 211 DIAGNOSTIC TEST PROGRESS /
010337     117     105     116
010342     111     130     040
010345     062     061     061
010350     040     104     111
010353     101     107     116
010356     117     123     124
010361     111     103     040
010364     124     105     123
010367     124     040     120
010372     122     117     107
010375     122     105     123
010400     123     040
010402     000                         .BYTE 0
010403     120     101     123 PASMSG: .ASCII /PASS COUNTER         = /
010406     123     040     103
010411     117     125     116
010414     124     105     122
010417     040     040     040
010422     040     040     040
010425     040     040     040
010430     075     040
010432     000                         .BYTE 0
                                       .EVEN
010434     123     124     101 T1MSG:  .ASCII /STARTING TEST NUMBER = /
010437     122     124     111
010442     116     107     040
```

```
          010445    124     105     123
          010450    124     040     116
          010453    125     115     102
          010456    105     122     040
          010461    075     040
     1612 010463    000                               .BYTE 0
     1613 010464    105     116     104     T2MSG:  .ASCII /ENDING   TEST NUMBER = /
          010467    111     116     107
          010472    040     040     040
          010475    124     105     123
          010500    124     040     116
          010503    125     115     102
          010506    105     122     040
          010511    075     040
     1614 010513    000                               .BYTE 0
     1615 010514    103     125     122     CMSSG:  .ASCII /CURRENT  TEST NUMBER = /
          010517    122     105     116
          010522    124     040     040
          010525    124     105     123
          010530    124     040     116
          010533    125     115     102
          010536    105     122     040
          010541    075     040
     1616 010543    000                               .BYTE 0
     1617 010544    105     122     122     ERCNTM: .ASCII /ERROR    FLAG        = /
          010547    117     122     040
          010552    040     040     040
          010555    106     114     101
          010560    107     040     040
          010563    040     040     040
          010566    040     040     040
          010571    075     040
     1618 010573    000                               .BYTE 0
     1619                                             .EVEN
     1620                                             ;
     1621                                             ;
     1622                                             ;
     1623 010574  000000                    OUTBUF:   .WORD 0                       ;OUTPUT BUFFER ARE (TTY)
     1624           010676                            .=.+100                       ;
     1625                                             ;
     1626                                             ;
     1627                                             ;
     1628                                             ;
     1629                                             ;
     1630                                             ;
     1631                                             ;
     1632                                             ;       DATE COMMAND
     1633                                             ;
     1634                                             ;
     1635                                             ;        THE DATE COMMAND ENTERS THE DATE
     1636                                             ;       INTO A MEMORY BUFFER FOR LATER USE
     1637                                             ;       AS OUTPUT MESSAGE HEADING DATA.
     1638                                             ;
     1639                                             ;
     1640                                             ;
     1641 010676  004537  015474            DATE:     JSR     R5,SAVE                 ;SAVE CPU
     1642 010702  005037  005532                      CLR     REQFLG                  ;RESET REQUEST FLAG
```

```
1643 010706 005237 005522                  INC   CIMODE             ;SET COMMAND MODE
1644 010712 005037 005534                  CLR   ECHOEM             ;RESET ECHO MODE
1645 010716 012737 002024 005526   DT1:    MOV   #DTADR,MSGADR      ;POINT TO MESSAGE ADDRESS
1646 010724 005237 005530                  INC   NCR                ;SET CR INHIBIT
1647 010730 004737 005134                  JSR   PC,TTYOUT          ;PRINT "DATE"
1648 010734 005037 005530                  CLR   NCR                ;RESET CR INHIBIT
1649 010740 004737 004446                  JSR   PC,INWAIT          ;WAIT FOR RESPONSE
1650 010744 013702 005546                  MOV   PPARMB,R2          ;LOAD ENDING DATA ADDRESS
1651 010750 013703 005544                  MOV   PARMB,R3           ;LOAD STARTING DATA ADDRESS
1652 010754 160302                         SUB   R3,R2              ;COMPUTE LENGTH OF DATE
1653 010756 020227 000012                  CMP   R2,#12             ;11 CHARACTERS?
1654 010762 001027                         BNE   DT5                ;PRINT "?" IF NO
1655 010764 012700 011154                  MOV   #MONTH,R0          ;POINT TO MONTH TABLE
1656 010770 013702 005544          DT2:   MOV   PARMB,R2           ;LOAD FIRST CHARACTER
1657 010774 010201                         MOV   R2,R1              ;SET UP R1 TO POINT TO DATA
1658 010776 062701 000003                  ADD   #3,R1              ;FIRST DATE LETTER IS THE
1659                                       ;                        ;FOURTH CHARACTER RECEIVED
1660 011002 012702 000003                  MOV   #3,R2              ;LOAD MONTH COUNT
1661 011006 111003                 DT3:    MOVB  (R0),R3            ;LOAD TABLE ENTRY
1662 011010 120311                         CMPB  R3,(R1)            ;IS IT A LEGAL ENTRY?
1663 011012 001404                         BEQ   DT4                ;LOOK AT NEXT IF YES
1664 011014 060200                         ADD   R2,R0              ;UPDATE MONTH TABLE
1665 011016 105710                         TSTB  (R0)               ;END OF MONTHS?
1666 011020 001410                         BEQ   DT5                ;GO PRINT "?"
1667 011022 000762                         BR    DT2                ;TRY AGAIN IF NOT
1668                                       ;
1669                                       ;
1670 011024 020227 000001          DT4:    CMP   R2,#1              ;LAST CHARACTER?
1671 011030 001416                         BEQ   DT6                ;STORE DATE IF YES
1672 011032 005200                         INC   R0                 ;CHECK NEXT IF MORE
1673 011034 005201                         INC   R1                 ;UPDATE DATA POINTER
1674 011036 005302                         DEC   R2                 ;DECREMENT CHARACTER COUNTER
1675 011040 000762                         BR    DT3                ;CHECK THIS ONE
1676                                       ;
1677                                       ;
1678 011042 012737 007060 005526   DT5:    MOV   #QUEST,MSGADR      ;LOAD "?" ADDRESS
1679 011050 005037 005534                  CLR   ECHOEM             ;RESET ECHOEM MODE
1680 011054 005037 005530                  CLR   NCR                ;RESET CR INHIBIT
1681 011060 004737 005134                  JSR   PC,TTYOUT          ;PRINT IT
1682 011064 000714                         BR    DT1                ;START OVER
1683                                       ;
1684                                       ;
1685 011066 013700 005544          DT6:    MOV   PARMB,R0           ;R0 DATA BUFFER
1686 011072 012701 000011                  MOV   #11,R1             ;R1=DATA COUNTER
1687 011076 012702 011222                  MOV   #DATBUF,R2         ;R2=DATE RESIDENCE
1688 011102 112022                 DT7:    MOVB  (R0)+,(R2)+        ;LOAD DATE
1689 011104 005301                         DEC   R1                 ;DECREMENT COUNTER
1690 011106 001375                         BNE   DT7                ;CONTINUE IF NOT DONE
1691 011110 105012                         CLRB  (R2)               ;INSURE PROPER OUPUT
1692                                       ;
1693                                       ;
1694                                       ;
1695 011112 012737 007062 005526           MOV   #PROMPT,MSGADR     ;SET CR INHIBIT
1696 011120 005037 005534                  CLR   ECHOEM             ;REST ECHOEMODE
1697 011124 005237 005530                  INC   NCR                ;SET CR INHIBIT
1698 011130 004737 005134                  JSR   PC,TTYOUT          ;OUT PUT PROMPT
1699 011134 005037 005530                  CLR   NCR                ;RESET CR INHIBIT
```

```
                                        ;
                                        ;
                                        ;
                                        ;
011140  005037  005522          DT10:   CLR     CIMODE                          ;RESET COMMAND MODE
011144  004537  015510                  JSR     R5,REST                         ;BRING BACK CPU
011150  000137  004446                  JMP     PMGR                            ;DO SOMETHING ELSE
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       MONTH TABLE FOR FINDING CORRECT DATE
                                        ;
                                        ;
011154     112     101     116  MONTH:  .ASCII /JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC/
011157     106     105     102
011162     115     101     122
011165     101     120     122
011170     115     101     131
011173     112     125     116
011176     112     125     114
011201     101     125     107
011204     123     105     120
011207     117     103     124
011212     116     117     126
011215     104     105     103
011220     000                          .BYTE 0
                                        .EVEN
                                        ;
                                        ;
                                        ;       DATE BUFFER (HOLDS CURRENT DATE IN ASCII)
                                        ;
                                        ;
011222                          DATEM:
011222  000000                  DATBUF: .WORD 0
        011236                          .=.+10.
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       SCOPE   MODE COMMAND
                                        ;
                                        ;
                                        ;         THE SCOPE COMMAND IS USED ONLY FOR
                                        ;       DEBUGGING PURPOSES.IT ALLOWS THE OPERATOR TO
                                        ;       CHOSE THE FOLLOWING VALUES IN HIS SCOPR LOOP:
                                        ;
                                        ;       1. TYPE OF COMMAND
                                        ;       2.STARTING DUSH ADDRESS
                                        ;       3.STARTING CYLINDER(DCYL) ADDRESS
                                        ;       4.WORD COUNT (2'S COMPLEMENT)
                                        ;
                                        ;        THE COMMAND WILL BE ISSUED  VIA THE DATA TRANSFER
                                        ;       SUBROUTINE (DXFER).IF THE CONTROLLER DOES NOT
                                        ;       COMPLETE THE COMMAND,THE NORMAL WAIT ROUTINE WILL
```

```
                                              ;       TERMINATE THE OPERATION.
                                              ;
                                              ;
                                              ;        THE PROGRAM WILL NOT HALT EVEN IF THE
                                              ;       THE OPERATION IS ERROR FREE.THE OPERATOR MUST
                                              ;       STOP THE PROCESS WITH A CONTROL "C" OR BY THE
                                              ;       CONSOLE HALT KEY.
                                              ;
                                              ;
011236  005237  005522           SCOPE:  INC     CIMODE                          ;SET COMMAND MODE
011242  005037  005532                   CLR     REQFLG                          ;RESET REQUEST FLAG
011246  012700  011446                   MOV     #SCOADR,R0                      ;POINT TO QUESTIONS
011252  005237  005530                   INC     NCR                             ;
011256  012702  011434                   MOV     #SCBLTP,R2                      ;LOAD TEMPORARY DATABUFFER
011262  012701  000004                   MOV     #4,R1                           ;SET UP TO CLEAR ENTRY BUFFER
011266  010203                           MOV     R2,R3
011270  005023                   SC1:    CLR     (R3)+                           ;CLEAR SPOT
011272  005301                           DEC     R1                              ;UNTIL DONE
011274  001375                           BNE     SC1                             ;
011276  005237  005540                   INC     NUMBER                          ;SET NUMBER INPUT FLAG
011302  012701  000004                   MOV     #4,R1                           ;SET ENTRY COUNT
011306  004737  007502                   JSR     PC,LOADIN                       ;FETCH DATA
                                         ;
                                         ;
011312  005037  005522           SC2:    CLR     CIMODE                          ;RESET COMMAND MODE
011316  005237  005524                   INC     TSTACT                          ;SET TEST ACTIVE
011322  012702  011434                   MOV     #SCBLTP,R2                      ;POINT TO TEMP BLOCK
011326  012703  011420                   MOV     #SCBLK,R3                       ;POINT TO REAL BLOCK
011332  005712                           TST     (R2)                            ;CHANGE COMMAND?
011334  001002                           BNE     SC2A                            ;JUMP IF CHANGE OR DEFAULT
011336  012712  000006                   MOV     #WRTCMD,(R2)                    ;LOAD DEFAULT COMMAND WRITE
011342  012263  000010           SC2A:   MOV     (R2)+,10(R3)                    ;LOAD DESIRED COMMAND
011346  005712                           TST     (R2)                            ;DUSH DIFFERENT?
011350  001001                           BNE     SC2B                            ;GO ON IF YES
011352  005012                           CLR     (R2)                            ;RESET TO ZERO IF NO
011354  012223                   SC2B:   MOV     (R2)+,(R3)+                     ;LOAD NEW COMMAND
011356  005723                           TST     (R3)+                           ;SKIP CORE ADDRESS
011360  005712                           TST     (R2)                            ;CHANGEING WRD COUNT?
                                                                                 ;WE DON'T ALLOW A 64 K
                                         ;                                       TRANSFER,BUT WE WILL ALLOW
                                         ;                                       64K -1 WORD.
011362  001002                           BNE     SC2C                            ;SO LOAD IT
011364  013712  001046                   MOV     WPSEC,(R2)                      ;
011370  012223                   SC2C:   MOV     (R2)+,(R3)+                     ;ENTER WORD COUNT
011372  005712                           TST     (R2)                            ;CHANGE CYLINDER ADDRESS?
011374  001001                           BNE     SC2D                            ;DO IT IF YES
011376  005012                           CLR     (R2)                            ;RESET IT IF NOT
011400  011213                   SC2D:   MOV     (R2),(R3)                       ;LOAD HERE
011402  012700  011420           SC2H:   MOV     #SCBLK,R0                       ;LOAD COMMAND BLOCK ADDRESS
011406  004537  022700           SC5:    JSR     R5,DXFER                        ;GO DO TRANSFER
011412  005037  001322                   CLR     ERRFLG                          ;IGNORE ERROR FLAG
011416  000773                           BR      SC5                             ;HANG UP IN THIS LOOP
                                         ;
                                         ;
                                         ;
                                         ;
011420  000000                   SCBLK:  .WORD 0                         ;DUSH IMAGE
```

```
1803 011422  044312                          .WORD ODBUF                  ;CORE ADDRESS
1804 011424  177400                          .WORD 177400                 ;WORDS PER SECTOR
1805 011426  000000                          .WORD 0                      ;CYLINDER ADDRESS
1806 011430  000006                          .WORD WRTCMD                 ;COMMAND
1807 011432  000000                          .WORD 0                      ;
1808                                         ;
1809                                         ;
1810                                         ;
1811 011434  000000                  SCBLTP: .WORD 0                      ;TEMP LOCS
1812 011436  000000                          .WORD 0
1813 011440  000000                          .WORD 0
1814 011442  000000                          .WORD 0
1815 011444  000000                          .WORD 0
1816                                         ;
1817                                         ;
1818                                         ;        MESSAGE ADDRESSES FOR SCOPE COMMAND
1819                                         ;
1820                                         ;
1821                                         ;
1822 011446  011456                  SCOADR: .WORD COMMSG         ;COMMAND TYPE MESSAGE
1823 011450  011510                          .WORD DUMSG          ;DUSH MESSAGE
1824 011452  011550                          .WORD WSMSG          ;WORDS PER SECTOR MESSAGE
1825 011454  011612                          .WORD CYMSG          ;CYLINDER ADDRESS MESSAGE
1826                                         ;
1827                                         ;
1828 011456     103     117     115  COMMSG: .ASCII /COMMAND TYPE (OCTAL) = 6 /
     011461     115     101     116
     011464     104     040     124
     011467     131     120     105
     011472     040     050     117
     011475     103     124     101
     011500     114     051     040
     011503     075     040     066
     011506     040
1829 011507     000                          .BYTE 0
1830 011510     104     125     123  DUMSG:  .ASCII /DUSH ADDRESS (SECTOR,HEAD) = 0 /
     011513     110     040     101
     011516     104     104     122
     011521     105     123     123
     011524     040     050     123
     011527     105     103     124
     011532     117     122     054
     011535     110     105     101
     011540     104     051     040
     011543     075     040     060
     011546     040
1831 011547     000                          .BYTE 0
1832 011550     127     117     122  WSMSG:  .ASCII /WORD COUNT (2'S COMPL.) = 177400 /
     011553     104     040     103
     011556     117     125     116
     011561     124     040     050
     011564     062     047     123
     011567     040     103     117
     011572     115     120     114
     011575     056     051     040
     011600     075     040     061
     011603     067     067     064
```

```
011606  060    060    040
011611  000                             .BYTE 0
011612  103    131    114     CYMSG:    .ASCII /CYLINDER ADDRESS = 0 /
011615  111    116    104
011620  105    122    040
011623  101    104    104
011626  122    105    123
011631  123    040    075
011634  040    060    040
011637  000                             .BYTE 0
                                        .EVEN
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       PASS COMMAND
                                        ;
                                        ;        THE PASS COMMAND IS CALLED ONLY BY THE TEST
                                        ;       EXECUTIVE.IT OUTPUTS THE PASS COUNTER AND RETURNS
                                        ;       VIA A RTS PC.THE OUTPUT IS IN THE FORM:
                                        ;
                                        ;       PASS COUNT = ###
                                        ;
                                        ;
011640  004537 015474          PASS:    JSR     R5,SAVE                 ;SAVE CPU
011644  005037 005532                   CLR     REQFLG                  ;RESET REQUEST FLAG
011650  005037 005534                   CLR     ECHOEM                  ;RESET ECHOEM
011654  005737 001224                   TST     PASINH                  ;NO PRINT OUT?
011660  001031                          BNE     PSS4                    ;EXIT IF NO
011662  005237 005522                   INC     CIMODE                  ;SET COMMAND MODE
011666  005237 005530                   INC     NCR                     ;SET CR INHIBIT
011672  012737 012006 005526            MOV     #PSMSG,MSGADR           ;POINT TO PASS MESSAGE
011700  004737 005134                   JSR     PC,TTYOUT               ;OUTPUT PASS =
011704  005037 005530                   CLR     NCR                     ;REST CR INHIBIT
011710  013703 001236                   MOV     PASCNT,R3               ;LOAD PASS COUNT
011714  004737 006754                   JSR     PC,ASCII                ;MAKE ASCII FOR OUTPUT
011720  012737 010574 005526            MOV     #OUTBUF,MSGADR          ;POINT TO MESSAGE BUFFER
011726  004737 005134                   JSR     PC,TTYOUT               ;OUTPUT PASS COUNT
011732  012737 005740 005526            MOV     #LFCODE,MSGADR          ;LETS DO ANOTHER LINE FEED
011740  004737 005134                   JSR     PC,TTYOUT               ;NOW
                                        ;
011744  005037 005522          PSS4:    CLR     CIMODE                  ;RESET COMMAND MODE
011750  004537 015510                   JSR     R5,REST                 ;BRING BACK REGISTERS
011754  005737 005524                   TST     TSTACT                  ;TEST ACTIVE?
011760  001002                          BNE     PSS1                    ;SEE IF REPEAT IF YES
                                        ;
011762  000137 004506          PSS2:    JMP     PMG1                    ;
                                        ;
                                        ;
011766  005737 001220          PSS1:    TST     REPFLG                  ;REPEAT THIS SEQUENCE?
011772  001402                          BEQ     PSS3                    ;NO KILL TEST ACTIVE
011774  000137 004000                   JMP     TNINTA                  ;YES DO IT
                                        ;
012000  005037 005524          PSS3:    CLR     TSTACT                  ;RESET TEST ACTIVE
012004  000766                          BR      PSS2                    ;AND LEAVE
```

```
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;       PASS MESSAGE
                                                ;
                                                ;
     012006                             PSMSG:
     012006     040     040     040             .ASCII /    PASS COUNT = /
     012011     040     120     101
     012014     123     123     040
     012017     103     117     125
     012022     116     124     040
     012025     075     040
     012027     000                             .BYTE 0
                                                .EVEN
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;       ERROR REPORTING ROUTINE
                                                ;
                                                ;
                                                ;         THIS ROUTINE PRINTS OUT THE TEST AND DRIVE
                                                ;       DATA NECESSARY TO DIAGNOSTIC PROBLEMS OCCURRING
                                                ;       WITH THE PHOENIX 211 OR THE ATTATCHED DRIVE(S).
                                                ;
                                                ;
                                                ;         THE OUTPUT IS IN THE FOLLOWING FORM
                                                ;
                                                ;       FAILING DRIVE NUMBER = ##
                                                ;       FAILING TEST NUMBER  = ##
                                                ;       REFERENCE DATA       = ##
                                                ;       ACTUAL DATA          = ##
                                                ;       SUBTEST NUMBER       = ##
                                                ;
                                                ;
                                                ;  $$$  IF THE ERROR IS A DATA ERROR THE OUTPUT WILL
                                                ;       BE A MESSAGE SAYING "DATA ERROR"
                                                ;
                                                ;       IF ANY OTHER ERROR OCCURS THE CONTROLLER
                                                ;       REGISTER WILL BE OUTPUTED.
                                                ;
                                                ;
                                                ;
     012030     005737  001222          TTERR:  TST     NOSTOP                  ;SKIP PRINTOUT?
     012034     001402                          BEQ     TER1                    ;GO ON IF NO
     012036     000137  003516                  JMP     INTST                   ;REPEAT TEST SEQUENCE IF YES
                                                ;
                                                ;
     012042     005037  005524          TER1:   CLR     TSTACT                  ;RESET TEST ACTIVE
     012046     005737  001226                  TST     ERPRIN                  ;NO PRINTOUT?
     012052     001056                          BNE     TER5                    ;THEN EXIT
     012054     005237  005522                  INC     CIMODE                  ;SET COMMAND MODE HERE
     012060     004537  013104                  JSR     R5,POUT                 ;OUTPUT WHAT HAPPENED
     012064     012352                          .WORD   ERMSG                   ;ERROR MESSAGE ADDRESSES
```

```
012066  013220                            .WORD   ERDATA              ;ERROR DATA WORDS
012070  000005                            .WORD   5                   ;MESSAGE COUNT
                                          ;
                                          ;
012072  005737  001320            TER2:   TST     DATERR              ;DATA ERROR?
012076  001427                            BEQ     TER4                ;PRINT CONTROLLER REGISTERS IF NO
                                          ;
                                          ;
012100  012737  005740  005526    TER3:   MOV     #LFCODE,MSGADR      ;POINT TO LINE FEED
012106  005037  005534                    CLR     ECHOEM              ;REST ECHO
012112  005037  005530                    CLR     NCR                 ;
012116  004737  005134                    JSR     PC,TTYOUT           ;PRINT IT
012122  004737  005134                    JSR     PC,TTYOUT           ;TWICE
                                          ;
012126  012737  013034  005526            MOV     #DATMSG,MSGADR      ;NOW PRINT ERROR MESSAGE
                                          ;
012134  004737  005134                    JSR     PC,TTYOUT           ;PRINT MESASAGE
                                          ;
                                          ;
012140  012737  005740  005526            MOV     #LFCODE,MSGADR      ;TWO MORE LF'S
012146  004737  005134                    JSR     PC,TTYOUT           ;PRINT
012152  004737  005134                    JSR     PC,TTYOUT           ;NOW
                                          ;
                                          ;
012156  004537  013104            TER4:   JSR     R5,POUT             ;PRINT CONTROLLER REGISTERS
012162  012560                            .WORD   DCCMSG              ;CONTROLLER MESSAGES(REGISTER)
012164  013176                            .WORD   EBMSG               ;ERROR REGISTER SAVE BLOCK
012166  000007                            .WORD   7                   ;OUTPUT COUNT
                                          ;
012170  005737  001074                    TST     ECC                 ;IS THIS AN ECC CONTROLLER?
012174  001405                            BEQ     TER5                ;EXIT IF NO
                                          ;
012176  004537  013104                    JSR     R5,POUT             ;PRINT REGISTERS IF YES
012202  012576                            .WORD   ECRMSG              ;ECC REGISTER MESSAGES
012204  013214                            .WORD   ECCBB               ;SAVE LOCATIONS FOR REGISTERS
012206  000002                            .WORD   2                   ;PRINT COUNT
                                          ;
                                          ;
012210  005737  001230            TER5:   TST     CONTST              ;CONTINUE TEST MODE?(TEST31,32)
012214  001454                            BEQ     TER6                ;EXIT IF NOT
                                          ;
012216  023727  001232  000031            CMP     CURTN,#31           ;TEST 31?
012224  001404                            BEQ     TER7                ;SET UP FOR GO AHEAD IF YES
012226  023727  001232  000032            CMP     CURTN,#32           ;TEST 32?
012234  001044                            BNE     TER6                ;EXIT IF NEITHER
                                          ;
012236  017700  166536            TER7:   MOV     @DCSR,R0            ;FIND OUT WHAT COMMAND
012242  042700  177770                    BIC     #177770,R0          ;LOOSE GARBARGE
012246  020027  000006                    CMP     R0,#WRTCMD          ;WAS IT A WRITE?
012252  001030                            BNE     TER8                ;GO ON IF NO
                                          ;
012254  012700  023602                    MOV     #WRTBLK,R0          ;POINT TO WRITE BLOCK IF YES
                                          ;
012260                            TER9:
012260  062700  000004                    ADD     #4,R0               ;ADJUST FOR BLOCK
012264  004537  022436                    JSR     R5,ENDPAR           ;COMPUTE AS IF NO ERROR
012270  005737  001162                    TST     ENDSEC              ;SECTOR =0
```

```
012274  001402                          BEQ     TER9A                   ;SKIP THIS PART IF 0
012276  005337  001162                  DEC     ENDSEC                  ;DECREMENT IF OK
012302                          TER9A:
012302  013737  001162  001136          MOV     ENDSEC,CURSEC   ;SECTOR
012310  013737  001164  001140          MOV     ENDHD,CURHD             ;HEAD
012316  013737  001166  001142          MOV     ENDCYL,CURCYL           ;CYLINDER
012324  005037  001322                  CLR     ERRFLG                  ;
                                        ;
012330  000137  003614                  JMP     FMTCLR                  ;JUMP TO FORMATTER CLEAR AREA
                                        ;
012334  012700  023614          TER8:   MOV     #RDBLK,R0               ;POINT TO READ BLOCK
012340  005037  001320                  CLR     DATERR                  ;RESET DATA ERROR
012344  000745                          BR      TER9                    ;AND GO ON
                                        ;
012346  000137  004506          TER6:   JMP     PMG1                    ;RETURN TO MONITOR
                                        ;
                                        ;
                                        ;
                                        ;       ERROR PRINTOUT MESSAGES
                                        ;
012352  012367                  ERMSG:  .WORD FDRMSG                    ;FAILED DRIVE MESSAGE
012354  012417                          .WORD FTTMSG                    ;FAILED TEST MESSAGE
012356  012447                          .WORD RFRMSG                    ;REFERENCE DATA MESSAGE
012360  012477                          .WORD ACTMSG                    ;ACTUAL DATA MESSAGE
012362  012527                          .WORD SUBMSG                    ;SUBTEST NUMBER MESSAGE
012364  000000                          .WORD 0                         ;THATS ALL THE MESSAGES
                                        ;
                                        ;
012366     000                          .BYTE   0
012367     106     101     111  FDRMSG: .ASCII /FAILING DRIVE NUMBER = /
012372     114     111     116
012375     107     040     104
012400     122     111     126
012403     105     040     116
012406     125     115     102
012411     105     122     040
012414     075     040
012416     000                          .BYTE   0
012417     106     101     111  FTTMSG: .ASCII /FAILING TEST  NUMBER = /
012422     114     111     116
012425     107     040     124
012430     105     123     124
012433     040     040     116
012436     125     115     102
012441     105     122     040
012444     075     040
012446     000                          .BYTE   0
012447     122     105     106  RFRMSG: .ASCII /REFERENCE   DATA     = /
012452     105     122     105
012455     116     103     105
012460     040     040     040
012463     104     101     124
012466     101     040     040
012471     040     040     040
012474     075     040
012476     000                          .BYTE   0
012477     101     103     124  ACTMSG: .ASCII /ACTUAL      DATA     = /
```

```
     012502   125   101   114
     012505   040   040   040
     012510   040   040   040
     012513   104   101   124
     012516   101   040   040
     012521   040   040   040
     012524   075   040
2028 012526   000                      .BYTE   0
2029 012527   123   125   102  SUBMSG: .ASCII /SUBTEST NUMBER       = /
     012532   124   105   123
     012535   124   040   116
     012540   125   115   102
     012543   105   122   040
     012546   040   040   040
     012551   040   040   040
     012554   075   040
2030 012556   000                      .BYTE 0
2031                                   .EVEN
2032                                   ;
2033                                   ;
2034                                   ;
2035                                   ;
2036                                   ;        CONTROLLER REGISTER PRINT OUT MESSAGES
2037                                   ;
2038                                   ;
2039                                   ;
2040 012560 012602             DCCMSG: .WORD DCRMSG                  ;DCSR MESSAGE ADDRESS
2041 012562 012623                     .WORD DUSMSG                  ;DUSH MESSAGE ADDRESS
2042 012564 012644                     .WORD DARMSG                  ;DCAR MESSAGE ADDRESS
2043 012566 012665                     .WORD DWCMSG                  ;DWCNT MESSAGE ADDRESS
2044 012570 012706                     .WORD DCYMSG                  ;DCYL MESSAGE ADDRESS
2045 012572 012727                     .WORD DSKMSG                  ;DSTAT MESSAGE ADDRESS
2046 012574 012750                     .WORD DERMSG                  ;DERR MESSAGE ADDRESS
2047                                   .EVEN                                 ;MAKE EVEN
2048                                   ;
2049 012576 012771             ECRMSG: .WORD ECBMSG                  ;ECCPB MESSAGE ADDRESS
2050 012600 013012                     .WORD ECPMSG                  ;ECCPW MESSAGE ADDRESS
2051                                   ;
2052                                   ;
2053 012602   040   040   040  DCRMSG: .ASCII /     DCSR    =    /
     012605   040   040   104
     012610   103   123   122
     012613   040   040   040
     012616   075   040   040
     012621   040
2054 012622   000                      .BYTE   0
2055 012623   040   040   040  DUSMSG: .ASCII /     DUSH    =    /
     012626   040   040   104
     012631   125   123   110
     012634   040   040   040
     012637   075   040   040
     012642   040
2056 012643   000                      .BYTE   0
2057 012644   040   040   040  DARMSG: .ASCII /     DCAR    =    /
     012647   040   040   104
     012652   103   101   122
     012655   040   040   040
```

```
     012660   075   040   040
     012663   040
2058 012664   000                             .BYTE   0
2059 012665   040   040   040   DWCMSG: .ASCII /      DWCNT  =   /
     012670   040   040   104
     012673   127   103   116
     012676   124   040   040
     012701   075   040   040
     012704   040
2060 012705   000                             .BYTE   0
2061 012706   040   040   040   DCYMSG: .ASCII /      DCYL   =   /
     012711   040   040   104
     012714   103   131   114
     012717   040   040   040
     012722   075   040   040
     012725   040
2062 012726   000                             .BYTE   0
2063 012727   040   040   040   DSKMSG: .ASCII /      DSTAT  =   /
     012732   040   040   104
     012735   123   124   101
     012740   124   040   040
     012743   075   040   040
     012746   040
2064 012747   000                             .BYTE   0
2065 012750   040   040   040   DERMSG: .ASCII /      DERR   =   /
     012753   040   040   104
     012756   105   122   122
     012761   040   040   040
     012764   075   040   040
     012767   040
2066 012770   000                             .BYTE 0
2067 012771   040   040   040   ECBMSG: .ASCII /      ECCPB  =   /
     012774   040   040   105
     012777   103   103   120
     013002   102   040   040
     013005   075   040   040
     013010   040
2068 013011   000                             .BYTE 0
2069 013012   040   040   040   ECPMSG: .ASCII /      ECCPW  =   /
     013015   040   040   105
     013020   103   103   120
     013023   127   040   040
     013026   075   040   040
     013031   040
2070 013032   000                             .BYTE 0
2071                                          .EVEN
2072                                          ;
2073                                          ;
2074                                          ;
2075                                          ;DATA ERROR MESSAGE
2076                                          ;
2077                                          ;
2078 013034   040   040   040   DATMSG: .ASCII /      **********  DATA ERROR  *********/
     013037   040   040   052
     013042   052   052   052
     013045   052   052   052
     013050   052   052   052
```

```
     013053   040     040     104
     013056   101     124     101
     013061   040     105     122
     013064   122     117     122
     013067   040     040     052
     013072   052     052     052
     013075   052     052     052
     013100   052     052
     013102   000                              .BYTE 0
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;         PRINT OUT ROUTINE
                                               ;
                                               ;           THIS SUBROUTINE PRINTS OUT AN ASCII LINE
                                               ;         FOLLOWED BY A SINGLE OCTAL WORD.
                                               ;         THE FORMAT OF THE OUTPURT IS THE FOLLOWING
                                               ;
                                               ;         ENDING TEST NUMBER = ######
                                               ;
                                               ;
                                               ;         THE SUBROUTINE CALL IS:
                                               ;
                                               ;         (X)      JSR      R5,POUT       ;CALL HERE
                                               ;         (X)2     .WORD NNNN             ;ADDRESS OF MESSAGES
                                               ;         (X)4     .WORD NNNN             ;ADDRESS OF DATA WORDS
                                               ;         (X)6     .WORD #                ;NUMBER OF MESSAGES
                                               ;
                                               ;
                                               ;
                                               .EVEN
     013104   000240                   POUT:   NOP                                     ;
     013106   011500                           MOV     (R5),R0                         ;LOAD R0 WITH MESSAGE ADDRESS
     013110   016501  000002                   MOV     2(R5),R1                        ;LOAD R1 WITH DATA ADDRESS
     013114   016502  000004                   MOV     4(R5),R2                        ;LOAD R2 ITH MESSAGE COUNT
                                               ;
     013120   005237  005530           PT1:    INC     NCR                             ;INHIBIT CARRIAGE RETURNS
     013124   012037  005526                   MOV     (R0)+,MSGADR                    ;
     013130   005037  005534                   CLR     ECHOEM                          ;ENSURE NO ECHOE MODE
     013134   004737  005134                   JSR     PC,TTYOUT                       ;PRINT ASCII MESSAGE
     013140   005037  005530                   CLR     NCR                             ;RESET CR RETURN MODE
     013144   013103                           MOV     @(R1)+,R3                       ;GET READY FOR DATA WORD CONVERSION
     013146   004737  006754                   JSR     PC,ASCII                        ;CONVERT TO ASCII
     013152   012737  010574  005526           MOV     #OUTBUF,MSGADR                  ;LOAD OUTPUT ADDRESS
     013160   004737  005134                   JSR     PC,TTYOUT                       ;OUTPUT DATA WORD
     013164   005302                           DEC     R2                              ;DECREMENT MESSAGE COUNTER
     013166   001354                           BNE     PT1                             ;GO AGAIN IF MORE
     013170   062705  000006                   ADD     #6,R5                           ;UPDATE RETURN
     013174   000205                           RTS     R5                              ;AND RETURN NOW
                                               ;
                                               ;
                                               ;
                                               ;
```

```
                                        ;
                                        ;
                                        ;
                                        ;       MEMORY POINTERS FOR ERROR PRINTOUT
                                        ;
                                        ;       1.CONTROLLER REGISTER IMAGES
                                        ;
                                        ;       2.CPU REGISTERS AT ERROR DETECTION
                                        ;
                                        ;
     013176  035334             EBMSG:  .WORD ERRBLK            ;DCSR ADDRESS
     013200  035336                     .WORD ERRBLK+2          ;DUSH ADDRESS
     013202  035340                     .WORD ERRBLK+4          ;DCAR ADDRESS
     013204  035342                     .WORD ERRBLK+6          ;DWCNT ADDRESS
     013206  035344                     .WORD ERRBLK+10         ;DCYL ADDRESS
     013210  035346                     .WORD ERRBLK+12         ;DSTAT ADDRESS
     013212  035350                     .WORD ERRBLK+14         ;DERR ADDRESS
     013214  035352             ECCBB:  .WORD ERRBLK+16         ;ECCPB ADDRESS
     013216  035354                     .WORD ERRBLK+20         ;ECCPW ADDRESS
                                        ;
                                        ;
                                        ;
     013220  002722             ERDATA: .WORD REG0              ;CPU R0
     013222  002724                     .WORD REG0+2            ;CPU R1
     013224  002726                     .WORD REG0+4            ;CPU R2
     013226  002730                     .WORD REG0+6            ;CPU R3
     013230  002732                     .WORD REG0+10           ;CPU R4
     013232  002734                     .WORD REG0+12           ;CPU R5
     013234  002736                     .WORD REG0+14           ;CPU R6(SP)
     013236  002740                     .WORD REG0+16           ;CPU R7(PC)
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       FORMAT COMMAND
                                        ;
                                        ;         THIS COMMAND WIL FORMAT ONE DISK UNIT
                                        ;       THE COMMAND WILL ASK THE THE OPERATOR WHICH UNIT
                                        ;       HE WISHES TO FORMAT  ( 0 THRU 3).
                                        ;
                                        ;       A SECOND QUERY WILL BE MADE IF A BAD CHOICE WAS
                                        ;       WAS ENTERED.THIS QUESTION WILL BE OF THE FORM:
                                        ;
                                        ;       "ARE YOU SURE???? Y OR N
                                        ;
                                        ;
     013240  004537  015474     FMTDK:  JSR     R5,SAVE                 ;SAVE CPU
     013244  005037  013450             CLR     DISKTP+2                        ;CLEAR TEMP
                                        ;
     013250  005037  005532             CLR     REQFLG                  ;RESET REQUEST FLAG
     013254  005237  005522             INC     CIMODE                  ;SET COMMAND MODE
                                        ;
     013260  012702  013446     FMTK1:  MOV     #DISKTP,R2              ;POINT TO TEMP DISK UNIT
     013264  012700  013452             MOV     #DUNFT,R0               ;POINT TO MESSAGE OUT PUT
     013270  012701  000001             MOV     #1,R1                   ;INCUDE COUNT
```

```
013274  005237  005540                  INC     NUMBER                  ;SET NUMBER FLAG
013300  004737  007502                  JSR     PC,LOADIN               ;FETCH DESIRED UNIT  TO BE
                                                                        ;FORMATTED.
013304  023727  013446  000003          CMP     DISKTP,#3               ;UNIT OK?
013312  003410                          BLE     FMTD1                   ;GO ON IF YES
                                        ;
                                        ;
013314  005037  005534                  CLR     ECHOEM                  ;RESET ECHO MODE
013320  012737  007060  005526          MOV     #QUEST,MSGADR           ;POINT TO ? MARK
013326  004737  005134                  JSR     PC,TTYOUT               ;PRINT ?
013332  000752                          BR      FMTK1                   ;TRY THIS AGAIN
                                        ;
                                        ;
013334  013737  013446  001214  FMTD1:  MOV     DISKTP,STPDSK           ;LOAD SELECTED UNIT
013342  013737  013446  001216          MOV     DISKTP,STRTDSK          ;
                                        ;
013350  005037  005540                  CLR     NUMBER                  ;ASK IF SURE
013354  012701  000001                  MOV     #1,R1                   ;COUNT =1
013360  004737  007502                  JSR     PC,LOADIN               ;FETCH ANSWER
013364  005737  013450                  TST     DISKTP+2                ;OK
013370  001006                          BNE     FMTD2                   ;GO ON IF YES
013372  005037  005534                  CLR     ECHOEM                  ;RESET ECHO MODE
013376  005037  005522                  CLR     CIMODE                  ;RESET COMMANDMODE
013402  000137  004506                  JMP     PMG1                    ;GO BACK TO PMGR IF NO
                                        ;
013406  012737  000045  001212  FMTD2:  MOV     #45,STRTN               ;LOAD TEST P'S FROM HERE
013414  012737  000045  001210          MOV     #45,STPTN               ;
013422  005037  001220                  CLR     REPFLG                  ;
013426  005037  001222                  CLR     NOSTOP                  ;
013432  005037  005522                  CLR     CIMODE                  ;RESET COMMAND MODE
013436  005237  005524                  INC     TSTACT                  ; SET TEST ACTIVE
                                        ;
                                        ;
013442  000137  003044                  JMP     TSTART                  ; START OFF NOW
                                        ;
                                        ;
013446  000000                  DISKTP: .WORD 0                         ;TEMP
013450  000000                          .WORD 0                         ;
013452  013456                  DUNFT:  .WORD DFMSG                     ;FORMAT MESAGE UNITE
013454  013525                          .WORD ASRMSG                    ;ARE YOU SURE MESSAGE
                                        ;
                                        ;
013456     040     040     040  DFMSG:  .ASCII /      DISK UNIT TO BE FORMATTED   = 0  /
013461     040     040     104
013464     111     123     113
013467     040     125     116
013472     111     124     040
013475     124     117     040
013500     102     105     040
013503     106     117     122
013506     115     101     124
013511     124     105     104
013514     040     040     040
013517     075     040     060
013522     040     040
013524     000                          .BYTE 0
013525     040     040     040  ASRMSG: .ASCII /     ARE YOU SURE??? Y OR N   /
```

```
     013530    040     040     101
     013533    122     105     040
     013536    131     117     125
     013541    040     123     125
     013544    122     105     077
     013547    077     077     040
     013552    131     040     117
     013555    122     040     116
     013560    040     040     040
     013563    000                                .BYTE 0
                                                  ;
                                                  ;
                                                  ;
                                                  ;
                                                  ;       REGISTER READ COMMAND
                                                  ;
                                                  ;    THIS COMMAND REDS AND OUPUTS THE PHOENIX 211 REGISTERS
                                                  ;
                                                  ;
     013564    004537  015474           RGREAD:   JSR     R5,SAVE                 ;SAVE CPU
     013570    005037  005532                     CLR     REQFLG                  ;RESET REQUEST FLAG
     013574    005037  005534                     CLR     ECHOEM                  ;RESET ECHO MODE
     013600    005237  005522                     INC     CIMODE                  ;SET COMMAND MODE
                                                  ;
                                                  ;
     013604    004537  013104                     JSR     R5,POUT                 ;PRINT REGISTERS
     013610    012560                             .WORD DCCMSG                    ;REGISTER HEADINGS
     013612    001000                             .WORD   DCSR                    ;REGISTE ADDRESSES
     013614    000007                             .WORD 7                         ;COUNT FOR OUPUT
                                                  ;
     013616    005737  001074                     TST     ECC                     ;IS THIS AN ECC CONTROLLER?
     013622    001405                             BEQ     RGR                     ;EXIT NOW IF NOT
                                                  ;
     013624    004537  013104                     JSR     R5,POUT                 ;OUPUT ECC REGS IF YES
     013630    012576                             .WORD   ECRMSG                  ;ECC REGISTER HEADING
     013632    001016                             .WORD   ECCPB                   ;ADDRESS OF REAL REGISTER
     013634    000002                             .WORD 2                         ;COUNT FOR REG'S
                                                  ;
     013636    005037  005522           RGR:      CLR     CIMODE                  ;RESET COMMAND MODE
     013642    004537  015510                     JSR     R5,REST                 ;BRING BACK CPU
                                                  ;
                                                  ;
     013646    000137  004506                     JMP     PMG1                    ;PROMPT
                                                  ;
                                                  ;
                                                  ;       HELP COMMAND
                                                  ;
                                                  ;         THIS COMMAND IS A BRIEF DESCIPTION OF THE
                                                  ;       PHOENIX 211 DIAGNOSTIC MONITOR TO AID THE
                                                  ;       OPERATOR DURING TESTING
                                                  ;
     013652    004537  015474           HELP:     JSR     R5,SAVE                 ;SAVE CPU
     013656    005237  005522                     INC     CIMODE                  ;SET COMMAND MODE
     013662    005037  005532                     CLR     REQFLG                  ;RESET REQUEST FLAG
     013666    005037  005534                     CLR     ECHOEM                  ;RESET ECHO MODE
                                                  ;
     013672    012737  014046  005526             MOV     #HLPMSG,MSGADR          ;POINT TO MESSAGE ADDRESSES
```

```
 013700  004737  005134                   JSR    PC,TTYOUT          ;PRINT IT
 013704  012737  005740  005526           MOV    #LFCODE,MSGADR     ;ISSUE LINE FEED
 013712  004737  005134                   JSR    PC,TTYOUT          ;
                                          ;
 013716  012702  000004                   MOV    #4,R2              ;PRINT FIRST INSTRUCTIONS
 013722  012703  014004                   MOV    #HPMSG,R3          ;R3=MESSAGE ADDRESS POINTER
 013726  012337  005526          HLP1:    MOV    (R3)+,MSGADR       ;LOAD MESSAGE ADDRESS
 013732  004737  005134                   JSR    PC,TTYOUT          ;PRINT IT
 013736  005302                           DEC    R2                 ;SEE IF DONE
 013740  001372                           BNE    HLP1               ;GO ON IF NO
                                          ;
                                          ;
 013742  012737  005740  005526  HLP2:    MOV    #LFCODE,MSGADR     ;ANOTHER LINE FEED HERE
 013750  004737  005134                   JSR    PC,TTYOUT          ;
                                          ;
 013754  012702  000015                   MOV    #13.,R2            ;COUNT =13
 013760  012337  005526          HLP3:    MOV    (R3)+,MSGADR       ;LOAD UP AN ADDRESS
 013764  004737  005134                   JSR    PC,TTYOUT          ;PRINT IT
 013770  005302                           DEC    R2                 ;DEC COUNTER
 013772  001372                           BNE    HLP3               ;AND CONTINUE UNTIL DONE
 013774  005037  005522                   CLR    CIMODE             ;RESET COMMAND MODE
 014000  000137  004506                   JMP    PMG1               ;AND PROMPT
                                          ;
                                          ;
 014004  014131                  HPMSG:   .WORD  MSG1               ;LIST OF MESSAGE FOR HELP COMMAND
 014006  014210                           .WORD  MSG2               ;
 014010  014271                           .WORD  MSG3
 014012  014347                           .WORD  MSG4
 014014  014436                           .WORD  MSG5
 014016  014513                           .WORD  MSG6
 014020  014574                           .WORD  MSG7
 014022  014634                           .WORD  MSG8
 014024  014705                           .WORD  MSG9
 014026  014754                           .WORD  MSG10
 014030  015021                           .WORD  MSG11
 014032  015046                           .WORD  MSG12
 014034  015101                           .WORD  MSG13
 014036  015152                           .WORD  MSG14
 014040  015177                           .WORD  MSG15
 014042  015245                           .WORD  MSG16
 014044  015301                           .WORD  MSG17              ;
                                          ;
 014046     040     040     040  HLPMSG:  .ASCII /        PHOENIX 211 DIAGNOSTIC MONITOR INSTRUCTIONS/
 014051     040     040     040
 014054     040     120     110
 014057     117     105     116
 014062     111     130     040
 014065     062     061     061
 014070     040     104     111
 014073     101     107     116
 014076     117     123     124
 014101     111     103     040
 014104     115     117     116
 014107     111     124     117
 014112     122     040     111
 014115     116     123     124
 014120     122     125     103
```

```
          014123    124     111     117
          014126    116     123
     2321 014130    000                          .BYTE 0
     2322                                        ;
     2323 014131    040     040     040   MSG1:  .ASCII /     1. TYPE A CONTROL C TO ABORT THIS PRINTOUT/
          014134    040     061     056
          014137    040     124     131
          014142    120     105     040
          014145    101     040     103
          014150    117     116     124
          014153    122     117     114
          014156    040     103     040
          014161    124     117     040
          014164    101     102     117
          014167    122     124     040
          014172    124     110     111
          014175    123     040     120
          014200    122     111     116
          014203    124     117     125
          014206    124
     2324 014207    000                          .BYTE 0
     2325 014210    040     040     040   MSG2:  .ASCII /     2. DEFAULT ANSWER FOR YES-NO QUESTIONS IS NO/
          014213    040     062     056
          014216    040     104     105
          014221    106     101     125
          014224    114     124     040
          014227    101     116     123
          014232    127     105     122
          014235    040     106     117
          014240    122     040     131
          014243    105     123     055
          014246    116     117     040
          014251    121     125     105
          014254    123     124     111
          014257    117     116     123
          014262    040     111     123
          014265    040     116     117
     2326 014270    000                          .BYTE 0
     2327 014271    040     040     040   MSG3:  .ASCII /     3. ALL NUMERICAL ANSWERS MUST BE IN OCTAL/
          014274    040     063     056
          014277    040     101     114
          014302    114     040     116
          014305    125     115     105
          014310    122     111     103
          014313    101     114     040
          014316    101     116     123
          014321    127     105     122
          014324    123     040     115
          014327    125     123     124
          014332    040     102     105
          014335    040     111     116
          014340    040     117     103
          014343    124     101     114
     2328 014346    000                          .BYTE 0
     2329 014347    040     040     040   MSG4:  .ASCII /     4. TYPING A CONTROL C DURING TESTING CALLS MONITOR/
          014352    040     064     056
          014355    040     124     131
```

```
      014360   120   111   116
      014363   107   040   101
      014366   040   103   117
      014371   116   124   122
      014374   117   114   040
      014377   103   040   104
      014402   125   122   111
      014405   116   107   040
      014410   124   105   123
      014413   124   111   116
      014416   107   040   103
      014421   101   114   114
      014424   123   040   115
      014427   117   116   111
      014432   124   117   122
 2330 014435   000                       .BYTE 0
 2331 014436   040   040   040   MSG5:   .ASCII /    5. THE FOLLOWING ARE THE LEGAL COMMANDS:/
      014441   040   065   056
      014444   040   124   110
      014447   105   040   106
      014452   117   114   114
      014455   117   127   111
      014460   116   107   040
      014463   101   122   105
      014466   040   124   110
      014471   105   040   114
      014474   105   107   101
      014477   114   040   103
      014502   117   115   115
      014505   101   116   104
      014510   123   072
 2332 014512   000                       .BYTE 0
 2333 014513   040   040   040   MSG6:   .ASCII /       A. G    ;GO COMMAND -STARTS TEST SEQUENCE/
      014516   040   040   040
      014521   040   101   056
      014524   040   107   040
      014527   040   040   040
      014532   073   107   117
      014535   040   103   117
      014540   115   115   101
      014543   116   104   040
      014546   055   123   124
      014551   101   122   124
      014554   123   040   124
      014557   105   123   124
      014562   040   123   105
      014565   121   125   105
      014570   116   103   105
 2334 014573   000                       .BYTE 0
 2335 014574   040   040   040   MSG7:   .ASCII /       B. DK   ;DISK PARAMETERS/
      014577   040   040   040
      014602   040   102   056
      014605   040   104   113
      014610   040   040   040
      014613   073   104   111
      014616   123   113   040
      014621   120   101   122
```

```
     014624  101  115  105
     014627  124  105  122
     014632  123
2336 014633  000                  .BYTE 0
2337 014634  040  040  040 MSG8:  .ASCII /         C. TS    ;TEST SEQUENCE PARAMETERS/
     014637  040  040  040
     014642  040  103  056
     014645  040  124  123
     014650  040  040  040
     014653  073  124  105
     014656  123  124  040
     014661  123  105  121
     014664  125  105  116
     014667  103  105  040
     014672  120  101  122
     014675  101  115  105
     014700  124  105  122
     014703  123
2338 014704  000                  .BYTE 0
2339 014705  040  040  040 MSG9:  .ASCII /         D. FT    ;CONTROLLER PARAMETERS./
     014710  040  040  040
     014713  040  104  056
     014716  040  106  124
     014721  040  040  040
     014724  073  103  117
     014727  116  124  122
     014732  117  114  114
     014735  105  122  040
     014740  120  101  122
     014743  101  115  105
     014746  124  105  122
     014751  123  056
2340 014753  000                  .BYTE 0
2341 014754  040  040  040 MSG10: .ASCII /         E. PR    ;TEST PROGRESS REPORT/
     014757  040  040  040
     014762  040  105  056
     014765  040  120  122
     014770  040  040  040
     014773  073  124  105
     014776  123  124  040
     015001  120  122  117
     015004  107  122  105
     015007  123  123  040
     015012  122  105  120
     015015  117  122  124
2342 015020  000                  .BYTE 0
2343 015021  040  040  040 MSG11: .ASCII /         F. DA    ;DATE/
     015024  040  040  040
     015027  040  106  056
     015032  040  104  101
     015035  040  040  040
     015040  073  104  101
     015043  124  105
2344 015045  000                  .BYTE 0
2345 015046  040  040  040 MSG12: .ASCII /         G. SC    ;SCOPE LOOP/
     015051  040  040  040
     015054  040  107  056
```

```
        015057  040   123   103
        015062  040   040   040
        015065  073   123   103
        015070  117   120   105
        015073  040   114   117
        015076  117   120
   2346 015100  000                      .BYTE 0
   2347 015101  040   040   040  MSG13:  .ASCII /        H. RR   ;CONTROLLER REGISTER READ/
        015104  040   040   040
        015107  040   110   056
        015112  040   122   122
        015115  040   040   040
        015120  073   103   117
        015123  116   124   122
        015126  117   114   114
        015131  105   122   040
        015134  122   105   107
        015137  111   123   124
        015142  105   122   040
        015145  122   105   101
        015150  104
   2348 015151  000                      .BYTE 0
   2349 015152  040   040   040  MSG14:  .ASCII /        I. H    ;HELP/
        015155  040   040   040
        015160  040   111   056
        015163  040   110   040
        015166  040   040   040
        015171  073   110   105
        015174  114   120
   2350 015176  000                      .BYTE 0
   2351 015177  040   040   040  MSG15:  .ASCII /        J. TE   ;TEST ERROR CONDITIONS/
        015202  040   040   040
        015205  040   112   056
        015210  040   124   105
        015213  040   040   040
        015216  073   124   105
        015221  123   124   040
        015224  105   122   122
        015227  117   122   040
        015232  103   117   116
        015235  104   111   124
        015240  111   117   116
        015243  123
   2352 015244  000                      .BYTE 0
   2353 015245  040   040   040  MSG16:  .ASCII /        K. FO   ;FORMAT DISK/
        015250  040   040   040
        015253  040   113   056
        015256  040   106   117
        015261  040   040   040
        015264  073   106   117
        015267  122   115   101
        015272  124   040   104
        015275  111   123   113
   2354 015300  000                      .BYTE 0
   2355 015301  040   040   040  MSG17:  .ASCII /        L. CO   ;CONTINUE TESTING/
        015304  040   040   040
        015307  040   114   056
```

```
                015312   040     103     117
                015315   040     040     040
                015320   073     103     117
                015323   116     124     111
                015326   116     125     105
                015331   040     124     105
                015334   123     124     111
                015337   116     107
          2356  015341   000                                     .BYTE 0
          2357                                                   .EVEN
          2358                                                   ;
          2359                                                   ;
          2360                                                   ;
          2361                                                   ;
          2362                                                   ;        CONTINUE COMMAND
          2363                                                   ;
          2364                                                   ;   THIS COMMAND IS USED AFTER TESTING HAS BEEN HALTED
          2365                                                   ;  AND THE MONITOR HAS BEEN ACTIVATED.
          2366                                                   ;   WITH THIS COMMAND TESTING CAN BE STARTED FROM
          2367                                                   ; THE LAST TEST THAT WAS BEING EXECUTED THEREBY
          2368                                                   ;  NOT LOSING PREVIOUS PASS COUNTS AND OR PREVIOUS PASSED
          2369                                                   ;  TESTS.
          2370                                                   ;
          2371                                                   ;
          2372  015342   012700  005750          CONTU:  MOV     #TSTINT,R0              ;START FILLING CPU AGAIN
          2373  015346   005037  005522                  CLR     CIMODE                  ;RESET COMMAND MODE
          2374  015352   005037  001150                  CLR     INTFLG                  ;RESER INTERRUPT FLAGS
          2375  015356   005037  005720                  CLR     TTYINT                  ;
          2376  015362   005237  005524                  INC     TSTACT                  ;SET TEST MODE BIT
          2377  015366   005720                          TST     (R0)+                   ;LOOK AT R1 IMAGE
          2378  015370   012001                          MOV     (R0)+,R1                ;FILL CPU
          2379  015372   012002                          MOV     (R0)+,R2                ;
          2380  015374   012003                          MOV     (R0)+,R3                ;
          2381  015376   012004                          MOV     (R0)+,R4                ;
          2382  015400   012005                          MOV     (R0)+,R5
          2383  015402   005046                          CLR     -(SP)                   ;ENTER BACK NOW
          2384  015404   011046                          MOV     (R0),-(SP)              ;LOAD OLD ADDRESS
          2385  015406   013700  005750                  MOV     TSTINT,R0               ;RESTORE R0 LAST
          2386                                           ;
          2387                                           ;
          2388  015412   000002                          RTI                             ;AND EXIT
          2389                                           ;
          2390                                           ;       KEYBOARD INTERRUPT SET ROUTINE
          2391                                           ;
          2392                                           ;
          2393  015414   005737  001104          TTYSET: TST     TTYPUT          ;KEYBOARD HRER?
          2394  015420   001406                          BEQ     TTYSEX          ;EXIT IF NO
          2395  015422   012777  000100  170272          MOV     #INTON,@TKS     ;ENABLE READER
          2396  015430   012777  000100  170270          MOV     #INTON,@TPS     ;ENABLE PRINTER
          2397  015436   000207                  TTYSEX: RTS     PC              ;AND EXIT
          2398                                           ;
          2399                                           ;
          2400                                           ;
          2401                                           ;
          2402                                           ;
          2403                                           ;       CONTROLLER REGISTER SAVE ROUTINE
          2404                                           ;
```

```
                                                        ;
                                                        ;
015440  004537  015474              REGSAV: JSR      R5,SAVE              ;SAVE CPU
015444  012700  035334                      MOV      #ERRBLK,R0           ;POINT TO SAVE BLOCK
015450  012701  001000                      MOV      #DCSR,R1             ;POINT T O REGISTER ADDR.
015454  012702  000007                      MOV      #7,R2                ;COUNT
015460  013120                      REG1:   MOV      @(R1)+,(R0)+         ;SAVE
015462  005302                              DEC      R2                   ;DECREMENT COUNT
015464  001375                              BNE      REG1                 ;GO ON UNTIL DONE
015466  004537  015510                      JSR      R5,REST              ;BRING BACK CPU
                                            ;
015472  000207                              RTS      PC                   ;RETURN
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;        3M.ECC   TEST NUMBERS 0>>> 3    NOV-13-77
                                            ;        REGISTER STACK SAVE ROUTINE
                                            ;     ********************************
                                            ;
                                            ;
                                            ;
                                            ;
015474  010046                      SAVE:   MOV      R0,-(SP)             ;SAVE R0.>>>>>>>>>>R4
015476  010146                              MOV      R1,-(SP)             ;
015500  010246                              MOV      R2,-(SP)             ;
015502  010346                              MOV      R3,-(SP)             ;
015504  010446                              MOV      R4,-(SP)             ;
015506  000205                              RTS      R5                   ;GO BACK
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;        REGISTER STACK RETURN ROUTINE
                                            ;    ******************************************
                                            ;
                                            ;
                                            ;
015510  012604                      REST:   MOV      (SP)+,R4             ;RESTORE R4>>>>>>>>>>>>>>>>R0
015512  012603                              MOV      (SP)+,R3             ;
015514  012602                              MOV      (SP)+,R2             ;
015516  012601                              MOV      (SP)+,R1             ;
015520  012600                              MOV      (SP)+,R0             ;
015522  000205                              RTS      R5                   ;AND RETURN
                                            ;
                                            ;
                                            ;
                                            ;
                                    ;
                                    ;
                                            ;        RANDOM BUFFER FORMATTER
                                            ;    ******************************
```

```
                                        ;
                                        ;       THIS ROUTINE FORMATS THE OUTPUT BUFFER
                                        ;    WITH THE RANDOM NUMBER "RAMBLER"
                                        ;
                                        ;
015524  004537  015474         RANBUF:  JSR     R5,SAVE         ;SAVE REGISTERS
015530  004537  015564                  JSR     R5,RANDOM       ;GENERATE RANDOM NUMBER
015534  013700  001252                  MOV     RAMBLER,R0      ;FETCH RANDOM NUMBER
015540  012701  044312                  MOV     #ODBUF,R1       ;POINT TO OUTPUT BUFFER
015544  012702  047314                  MOV     #IDBUF1,R2      ;POINT R2 TO END OF BUFFER
015550  010021                 RANB:    MOV     R0,(R1)+        ;ENTER DATA
015552  020102                          CMP     R1,R2           ;DONE?
015554  001375                          BNE     RANB            ;CONTINUE IF NO
015556  004537  015510                  JSR     R5,REST         ;RELOAD REGISTERS
015562  000205                          RTS     R5              ;AND EXIT
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       RANDOM NUMBER GENERATOR
                                        ;   ********************************
                                        ;
                                        ;           THIS ROUTINE WILL GENERATE A 16 BIT
                                        ;       RANDOM NUMBER FORUSE AS DATA AND DISK ADDRESS
                                        ;
                                        ;
                                        ;
                                        ;
015564  004537  015474         RANDOM:  JSR     R5,SAVE         ;SAVE R0>>>R4
015570  013704  001252                  MOV     RAMBLER,R4              ;FETCH OLD RANDOM
015574  010403                          MOV     R4,R3           ;LOAD INTO R3
015576  005001                          CLR     R1              ;CLEAR R1
015600  012702  000020                  MOV     #20,R2          ;SET UP SHIFT LOOP
015604  043704  001256                  BIC     POLY,R4         ;CLEAR XOR BITS
015610  000401                          BR      $RN2            ;START RANDOMMING????????
015612  006304                 $RN1:    ASL     R4              ;SHIFT R4
015614  100001                 $RN2:    BPL     $RN3            ;MSB NOT SET
015616  005101                          COM     R1
015620  005302                 $RN3:    DEC     R2              ;LAST BITS?
015622  100373                          BPL     $RN1            ;CONTINUE IF NO
015624  006303                          ASL     R3              ;SHIFT OLD NUMBER
015626  005401                          NEG     R1              ;CLEARX XOR
015630  050103                          BIS     R1,R3           ;UPDATE LSB OF OLD NUMBER
015632  010337  001252                  MOV     R3,RAMBLER      ;UPDATE RANDOM NUMBER
015636  004537  015510                  JSR     R5,REST         ;RESTORE REGISTERS
015642  000205                          RTS     R5              ;GO BACK
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
```

```
                                           ;
                                           ;
                                           ;
                                           ;
                                      ;    ;
                                           ;
                                           ;
                                           ;
                                           ;               ADDRESS COMPUTE ROUTINE
                                           ;     ***********************************
                                           ;
                                           ;
                                           ;
                                           ;
                                           ;
 015644  004537  015474          ADRSTP:   JSR    R5,SAVE            ;PUSH GENERAL REGISTERS
 015650  012700  001000                    MOV    #DCSR,R0           ;R0=ADDRESS TABLE
 015654  013710  001070                    MOV    DCSRA,(R0)         ;LOAD DESIRED DCR ADDRESS
 015660  011002                            MOV    (R0),R2            ;R2=DCSR ADDRESS
 015662  012701  000011                    MOV    #11,R1             ;R1 = # OF CONTROLLER REGISTERS
 015666  010220                  ADR:      MOV    R2,(R0)+           ;LOAD ADDRESS
 015670  062702  000002                    ADD    #2,R2              ;COMPUTE NEXT ADDRESS
 015674  005301                            DEC    R1                 ;DECREMENT LOOP COUNT
 015676  001373                            BNE    ADR                ;LOAD ANOTHER ADDRESS IF NOT DONE
 015700  013737  001046  001050            MOV    WPSEC,POSWP        ;SET UP WORD COUNT LABLES
 015706  005437  001050                    NEG    POSWP              ;MAKE TRUE WORD COUNT VALUE
 015712  013737  001040  001054            MOV    MAXSEC,RDUSH       ;SET UP MAX DISK ADDRESSES
 015720  053737  001042  001054            BIS    MAXHD,RDUSH        ;INCLUDE HEAD ADDRESS
 015726  013737  001044  001056            MOV    MAXCYL,RWCYL       ;AND CYLINDER
 015734  013777  001052  163130            MOV    INTSAV,@INTVEC     ;LOAD INTERRUPT SERVICE ROUTINE ADDR.
 015742  013702  001072                    MOV    INTVEC,R2          ;R2= INTERRUPT VECTOR ADDR
 015746  005722                            TST    (R2)+              ;POINT TO PS WORD LOCATION
 015750  012722  000240                    MOV    #240,(R2)+         ;LOAD LEVEL 5 WORD (PS)
 015754  012737  000004  023624            MOV    #RDCMD,RDBLK+10    ;RESTORE RD COMMAND
 015762  012737  000006  023612            MOV    #WRTCMD,WRTBLK+10;AND WRITE
 015770  005737  001066                    TST    MICRO              ;NONCOMPATIBLE DISK?
 015774  001403                            BEQ    ADR1               ;EXIT IF NO
 015776  005037  030720                    CLR    TN24D+2            ;CLEAR DRIVE READY CIT-THIS BIT
                                                                     ;IS CLEAR IN NONCOMPATIBLE DRIVES
                                                                     ;DURING A SEEK.
 016002  000403                            BR     ADR2               ;AND CONTINUE
 016004  012737  100000  030720  ADR1:     MOV    #BIT15,TN24D+2     ;SET UP DRIVE READY
 016012  005737  001076          ADR2:     TST    MULCPU             ;MULTIPLE CPU?
 016016  001404                            BEQ    ADR3               ;GO TO CLEAR REQUEST BIT
 016020  012737  000040  001102            MOV    #BIT5,MCPU         ;LOAD REQUEST BIT IF YESD
 016026  000402                            BR     ADREX              ;AND EXIT
 016030  005037  001102          ADR3:     CLR    MCPU               ;RESET REQUEST BIT
 016034  004537  015510          ADREX:    JSR    R5,REST            ;RESTORE CPU REGS
 016040  000205                            RTS    R5                 ;RETURN TO CALLING ROUTINE
                                           ;
                                           ;
                                           ;
                                           ;
                                           ;
                                           ;
                                           ;
                                           ;
```

```
                                        ;
                                        ;
                                        ;
                                        ;       DISK SEARCH ROUTINE
                                        ;  ************************
                                        ;
                                        ;       THIS ROUTINE LOOKS FOR ACTIVE DRIVES
                                        ;
                                        ;
016042  004537  015474  FIUNIT: JSR     R5,SAVE         ;SAVE REGISTERS
016046  004537  004276          JSR     R5,RQUEST       ;REQUEST FORMATTER
016052  005002                  CLR     R2              ;R2= NUMBER OF DRIVES
016054  005000                  CLR     R0              ;JUSTIFIED UNTI SELECTION
016056  010077  162720  FI1:    MOV     R0,@DUSH        ;SELECT DRIVE
016062  012701  000010          MOV     #10,R1  ;INSURE RESPONSE FROM DRIVE
016066  005301          FI2:    DEC     R1              ;WAIT
016070  001376                  BNE     FI2             ;
016072  005777  162714          TST     @DSTAT          ;DRIVE OUT THERE?
016076  100006                  BPL     FIEXIT  ;EXIT IF NO
016100  062700  010000          ADD     #10000,R0       ;SELECT NEXT UNIT
016104  005202                  INC     R2              ;UPDATE COUNTER
016106  020227  000003          CMP     R2,#3           ;ALL DONE LOOKING
016112  003761                  BLE     FI1             ;LOOK AGAIN IF NO
016114  005302          FIEXIT: DEC     R2              ;ADJUST DRIVE NUMBER
016116  010237  001214          MOV     R2,STPDSK       ;LOAD INTO DRIVE FLAG
016122  004537  015510          JSR     R5,REST         ;GIVE BACK REGISTERS
016126  000205                  RTS     R5              ;AND RETURN
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       RANDOM BLOCK GENERATOR
                                        ;    *****************************
                                        ;
                                        ;
                                        ;       THIS ROUTINE USES A PREVIOUSLY GENERATED
                                        ;  RANDOM NUMBER(16 BITS) AND FORMATS A COMMAND
                                        ;   BLOCK WITH A RANDOM SECTOR,HEAD AND CYLINDER ADDRESS
                                        ;  AS WELL AS A RANDOM WORD COUNT VALUE.
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
016130  004537  015474  RANBLK: JSR     R5,SAVE         ;SAVE REGISTERS
016134  013701  001252          MOV     RAMBLER,R1      ;FETCH RANDOM NUMBER
016140  010102                  MOV     R1,R2           ;USE R2 FOR WORK
016142  042702  177600  RAN5:   BIC     #SECMSK,R2      ;EXTRACT SECTOR FIELD
016146  020237  001040          CMP     R2,MAXSEC       ;TOO BIG?
016152  003405                  BLE     RAN1            ;CONTINUE IF NO
```

```
016154 004537 015564                JSR   R5,RANDOM     ;GET NEW NUMBER
016160 013702 001252                MOV   RAMBLER,R2    ;ENTER INTO R2
016164 000766                       BR    RAN5          ;CHECK AGAIN
016166 010210               RAN1:   MOV   R2,(R0)       ;LOAD INTO COMMAND BLOCK
016170 010102                       MOV   R1,R2         ;RELOAD RANDOM NUMBER
016172 042702 170177        RAN2:   BIC   #HDMSK,R2     ;EXTRACT SECTOR FIELD
016176 020237 001042                CMP   R2,MAXHD      ;TOO BIG?
016202 003405                       BLE   RAN3          ;CONTINUE IF NO
016204 004537 015564                JSR   R5,RANDOM     ;GET NEW NUMBER
016210 013702 001252                MOV   RAMBLER,R2    ;LOAD INTO R2
016214 000766                       BR    RAN2          ;CHECK AGAIN
016216 050210               RAN3:   BIS   R2,(R0)       ;AND LOAD INTO COMMAND BLOCK
016220 010102                       MOV   R1,R2         ;RELOAD RANDOM NUMBER
016222 042702 176000        RAN8:   BIC   #CYLMSK,R2    ;EXTRACT CYLINDER FIALD
016226 020237 001044                CMP   R2,MAXCYL     ;TOO BIG?
016232 003405                       BLE   RAN4          ;CONTINUE IF NO
016234 004537 015564                JSR   R5,RANDOM     ;GET NEW NUMBER
016240 013702 001252                MOV   RAMBLER,R2    ;LOAD INTO R2
016244 000766                       BR    RAN8          ;TRY AGAIN
016246 010260 000006        RAN4:   MOV   R2,6(R0)      ;LOAD INTO COMMAND BLOCK
016252 010102                       MOV   R1,R2         ;RELOAD RANDOM NUMBER
016254 053702 001270                BIS   MAXTFR,R2     ;MAKE SMALL ENOUGH WORD COUNT
016260 010260 000004                MOV   R2,4(R0)      ;LOAD INTO COMMAND BLOCK
016264 010102                       MOV   R1,R2         ;RESTORE RANDOM NUMBER
016266 005737 001106                TST   MDRIVE        ;TESTING MULTIPLE DRIVES?
016272 001414                       BEQ   RAN10         ;EXIT IF NO
016274 042702 147777        RAN6:   BIC   #UNIMSK,R2    ;EXTRACT UNITS
016300 020237 001214                CMP   R2,STPDSK     ;SEE IF OVER
016304 003405                       BLE   RAN7          ;CONTINUE IF NO
016306 004537 015564                JSR   R5,RANDOM     ;GET NEW NUMBER
016312 013702 001252                MOV   RAMBLER,R2    ;LOAD INTO R2
016316 000766                       BR    RAN6
016320 010237 001234        RAN7:   MOV   R2,CURDSK     ;FORCE RANDOM DISK ADDRESS
016324 004537 015510        RAN10:  JSR   R5,REST       ;RESTORE REGISTERS
016330 000205                       RTS   R5            ;GO BACK
                                    ;
                                    ;
                                    ;     CURRENT DISK UNIT SUBROUTINE
                                    ;     ******************************
                                    ;     THIS ROUTINE SETS UP THE
                                    ;     CURRENT DISK UNIT ID. FOR USE IN THE
                                    ;     CONTROLLER DUSH REGISTER
                                    ;
                                    ;
                                    ;
016332 013702 001234        DISKID: MOV   CURDSK,R2     ;FETCH CURRENT UNIT NUMBER
016336 000302                       SWAB  R2            ;MAKE VALID DISK SELECTION BITS
016340 006302                       ASL   R2
016342 006302                       ASL   R2
016344 006302                       ASL   R2
016346 006302                       ASL   R2
016350 000205                       RTS   R5            ;AND RETURN TO CALLING PROGRAM
                                    ;
                                    ;
                                    ;
                                    ;     DISK INITIALIZATION ROUTINE
                                    ;  ***********************************
```

```
                                               ;
                                               ;
                                               ;
016352  013746  001234         DKINIT: MOV     CURDSK,-(SP)     ;SAVE CURRENT DISK ID
016356  004537  004066                 JSR     R5,SEKTIM        ;SET UP LOOP FOR SEEK TIME OUT
016362  004537  004276                 JSR     R5,RQUEST        ; SELCT FORMATTER
016366  005002                         CLR     R2               ;USE R2 FOR WORK
016370  004537  016332         DKI1:   JSR     R5,DISKID        ;FORMAT DISK SLECT BITS
016374  010277  162402                 MOV     R2,@DUSH         ;SELECT DISK
016400  000240                         NOP                      ;WAIT FOR DISK READY
016402  000240                         NOP                      ;
016404  000240                         NOP                      ;
016406  052777  000022  162364         BIS     #FLTCLR,@DCSR    ;LOAD FAULT CLEAR COMMAND
016414  005277  162360                 INC     @DCSR            ;SET GO BIT
016420  000240                         NOP                      ;
016422  012703  000020                 MOV     #20,R3           ;SET UP WAIT LOOP
016426  105777  162346         DKI3:   TSTB    @DCSR            ;WAIT FOR DONE
016432  100402                         BMI     DKI4             ;CONTINUE IF DONE
016434  005303                         DEC     R3               ;DECREMENT PROTECT LOOP
016436  001373                         BNE     DKI3             ;CONTINUE IF REASONABLE TIME
016440  042777  000136  162332 DKI4:   BIC     #FUNC,@DCSR      ;LOSE FUCNTION BITS
016446  052777  000024  162324         BIS     #RTZCMD,@DCSR    ;ENTER RECALIBRATE COMMAND
016454  005277  162320                 INC     @DCSR            ;SET GO
016460  004537  022164                 JSR     R5,WAIT1         ;WAIT DONE
016464  052777  000077  162306         BIS     #NOPCMD,@DCSR    ;ISSUE NOP FOR SEEK DONE
016472  020227  030000                 CMP     R2,#UNIFLD       ;ALL DONE WITH DISKS?
016476  001403                         BEQ     DKI2             ;CONTINUE IF NO
016500  005237  001234                 INC     CURDSK           ;LOOK AT ANOTHER DISK
016504  000731                         BR      DKI1             ;AND DO ANOTHER DISK
016506  042777  000136  162264 DKI2:   BIC     #FUNC,@DCSR      ;LOOSE FUNCTION BITS
016514  012777  000041  162256         MOV     #SYSCLR!BIT5,@DCSR;CLEAR ERRORS(DRIVE NOT READY)
016522  012637  001234                 MOV     (SP)+,CURDSK     ;RESTORE CURRENT DISK FLAG
016526  000205                         RTS     R5               ;AND RETURN TO CALLING PROGRAM
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;
                                       ;       INITIALIZATION TEST
                                       ;     -----------------------------
                                       ;
                                       ;
                                       ;       THIS TEST VERIFIES THAT INIT CLEARS ALL THE
                                       ;    DISK CONTROLLER REGISTERS
                                       ;
016530  004537  004276         TNO:    JSR     R5,RQUEST        ;REQUEST FORMATTER
016534  000005                         RESET                    ;INITIALIZE ALL HARDWARE
016536  012702  000200                 MOV     #FMTRDY,R2       ;ESTABLISH REFERENCE
016542  004537  004276                 JSR     R5,RQUEST        ;GET FORMATTER AGAIN
016546  005004                         CLR     R4               ;INITIALIZE SUBTEST NUMBER
016550  017703  162224                 MOV     @DCSR,R3         ;WAS CONTROL AND STATUS INITALIZED?
016554  043703  001102                 BIC     MCPU,R3          ;LOOSE REQUEST BIT IF SET
016560  042703  000100                 BIC     #INTON,R3        ;LOOSE INTERRUPT REQUEST BIT
016564  020203                         CMP     R2,R3            ;COMPARE THE DCSR IS IT RIGHT
016566  001411                         BEQ     TNOA             ;IF RIGHT CONTINUE
016570  005777  162216                 TST     @DSTAT           ;TEST FORD DRIVE READY
```

```
2747 016574  100454                        BMI     TNOERR          ;EXIT WITH ERROR IF DRIVE READY
2748 016576  012702  000200                MOV     #FMTRDY,R2      ;SET UP NEW REFERENCE WORD
2749 016602  052702  100000                BIS     #BIT15,R2       ;ADD ERROR SUMMARY
2750 016606  020203                        CMP     R2,R3           ;CPMPARE AGAIN
2751 016610  001046                        BNE     TNOERR          ;EXIT WITH ERROR IF REGISRES NOT EQUAL
2752 016612  005204                TNOA:   INC     R4              ;SUBTEST NUMBER=1
2753 016614  017703  162164                MOV     @DCAR,R3        ;WAS CORE ADDRESS INITIALIZED?
2754 016620  005002                        CLR     R2              ;SET UP REFERENCE WORD
2755 016622  020203                        CMP     R2,R3           ;INTIALIZED?
2756 016624  001040                        BNE     TNOERR          ;EXIT WITH ERROR IF NOT
2757 016626  005204                        INC     R4              ;SUBTEST NO. =2
2758 016630  017703  162152                MOV     @DWCNT,R3       ;WAS WORD COUNT INITIA;IZED?
2759 016634  020203                        CMP     R2,R3           ;COMPARE
2760 016636  001033                        BNE     TNOERR          ;EXIT WITH ERROR IF NOT
2761 016640  005204                        INC     R4              ;SUBTEST NO.=3
2762 016642  017703  162142                MOV     @DCYL,R3        ;WAS CYLINDER ADDRESS INITIALIZED?
2763 016646  020203                        CMP     R2,R3           ;COMPARE
2764 016650  001026                        BNE     TNOERR          ;EXIT WITH ERROR IF NOT
2765 016652  005204                        INC     R4              ;SUBTEST NO.=4
2766 016654  017703  162132                MOV     @DSTAT,R3       ;FETCH DISK REGISTER
2767 016660  042703  100000                BIC     #BIT15,R3       ;LOSE DRIVE READY IF PRESENT
2768 016664  042703  020000                BIC     #BIT13,R3       ;AND WRITE PROTECT IF PRESENT
2769 016670  005002                        CLR     R2              ;SET UP REFERENCE WORD
2770 016672  020203                        CMP     R2,R3           ;IS THE DATA RIGHT?
2771 016674  001014                        BNE     TNOERR          ;EXIT IF NOT THE SAME
2772 016676  005204                TNOB:   INC     R4              ;SUBTEST NUMBER=5(DISK STATUS OK)
2773 016700  017703  162110                MOV     @DERR,R3        ;WAS DISK ERROR INITIALIZED?
2774 016704  005002                        CLR     R2              ;SET UP REFERENCE
2775 016706  020203                        CMP     R2,R3           ;RESET?
2776 016710  001006                        BNE     TNOERR          ;EXIT IF NO
2777 016712  005204                TNOC:   INC     R4              ;SUBTEST NUMBER=6
2778 016714  017703  162062                MOV     @DUSH,R3        ;WAS UNTI,HEAD SECTOR INIT.?
2779 016720  005002                        CLR     R2              ;SET UP REFERENCE WORD
2780 016722  020203                        CMP     R2,R3           ;RESET?
2781 016724  001402                        BEQ     TNOOUT          ;EXIT IF YES
2782 016726  005237  001322        TNOERR: INC     ERRFLG          ;SET ERROR FLAG
2783 016732  004537  003342        TNOOUT: JSR     R5,TSTCTL       ;GO TO NEXT TESAT
2784                                       ;
2785                                       ;
2786                                       .EOT            ;END OF TAPE
```

```
                        ;
                        ; ********** SECTION 1 **********
                        ;
                        ;
                        ;         OPERATING INSTRUCTIONS
                        ;
                        ;
                        ;
                        ;
                        ;**************************************************************
                        ;**************************************************************
                        ;
                        ;
                        ;         NOTE:   THIS DIAGNOSTIC MUST BE RUN ON A
                        ;                 FORMATTED DISK PACK AND WILL DESTROY
                        ;                 ALL DATA ON THE PACK.
                        ;                         IF IT IS DESIRED TO FORMAT A PACK
                        ;                 USE THE FORMATTING PROGRAM INCLUDED IN THE
                        ;                 THE DIAGNOSTIC(TESTS 47 AND 50)
                        ;
                        ;
                        ;**************************************************************
                        ;**************************************************************
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;
                        ;         1. SET PROGRAM COUNTER (PC) TO "START"
                        ;         ADDRESS .
                        ;
000000  003000                    .WORD START
                        ;
                        ;         2. ENTER STARTING TEST NUMBER IN THE MOST SIGNIFICANT
                        ;            BYTE OF THE CONSOLE SWITCH REGISTER.
                        ;
                        ;         3. ENTER DESIRED END TEST NUMBER IN THE LEAST
                        ;            SIGNIFICANT BYTE OF THE CONSOLE SWITCH REGISTER.
                        ;            THE FIRST TEST NUMBER IS ZERO.
                        ;
                        ;         4. SET THE MOST SIGNIFICANT BIT (BIT 15) OF SWITCH
                        ;            REGISTER IF THE TESTS ARE TO BE REPEATED. LEAVE
                        ;            RESET IF THE TESTS ARE TO BE EXECUTED ONLY ONCE.
                        ;
                        ;         5. DEPRESS "START" SWITCH ON COMPUTER CONSOLE. THE
                        ;            COMPUTER WILL START AND THEN HALT.
                        ;
                        ;         6. ENTER STARTING DISK DRIVE NUMBER INTO THE MOST
                        ;            SIGNIFICANT BYTE OF CONSOLE SWITCH REGISTER.
                        ;
                        ;         7. ENTER NUMBER OF THE LAST DISK DRIVE TO BE TESTED IN
                        ;            THE LEAST SIGNIFICANT BYTE OF CONSOLE SWITCH REGISTER/
                        ;             NOTE THAT THE MAXIMUM DRIVE NUMBER=3(0 THRU 3)
```

```
                                  ;
                                  ;
                                  ;    FORMAT ROUTINE                          9/29/75
                                  ;
                                  ;  THIS SUBROUTINE FORMATS THE DATA FOR THE
                                  ; OUTPUT BUFFER USED IN ALL DISK WRITE
                                  ; OPERATIONS.THE MESSAGE FORMATTED CONSISTS OF
                                  ; 64 WORDS OF AN DECREASING COUNT,
                                  ; 64 WORDS OF ALTERNATING 1 AND 0 CHARACTERS,
                                  ; 64 WORDS OF ALTERNATING CHECKERBOARD AND
                                  ; REVERSE CHECKERBOARD CHARACTERS, AND 64 WORDS OF ALTERNATING
                                  ; SLIDING ONES AND SLIDING ZERO WORDS
                                  ;
                                  ; AT ENTRY R1 POINTS TO THE BUFFER ORIGIN THAT IS TO BE FORMATTED.
                                  ;
                                  ;
                                  ;
016736  010046           FORMAT:  MOV  R0,-(SP)     ;SAVE REGISTERS
016740  010146                    MOV  R1,-(SP)     ;
016742  010246                    MOV  R2,-(SP)     ;
016744  010346                    MOV  R3,-(SP)     ;
016746  010546                    MOV  R5,-(SP)     ;
016750  012700  000100            MOV  #64.,R0      ;GET CHARACTER COUNT
016754  010005                    MOV  R0,R5        ;SAVE IN R5 ALSO
016756  010021           F1LP:    MOV  R0,(R1)+     ;FORMAT DATA WORD
016760  005300                    DEC  R0           ;DECREMENT CHARACTER COUNT,CHANGE DATA
016762  001375                    BNE   F1LP        ;REPEAT IF NOT DONE
016764  005002                    CLR  R2           ;ESTABLISH 0 CHARACTER
016766  012703  000377            MOV  #377,R3      ;ESTABLISH ALL 1'S CHARACTER
016772  010500                    MOV  R5,R0        ;REINITIALIZE CHARACTER COUNT
016774  110221           F2LP:    MOVB R2,(R1)+     ;INITIALIZE 0 CHARACTER
016776  110321                    MOVB R3,(R1)+     ;INITIALIZE 1'S CHARACTER
017000  005300                    DEC   R0          ;DECREMENT LOOP COUNT
017002  001374                    BNE  F2LP         ;REPEAT IF NO;T DONE
017004  010500                    MOV  R5,R0        ;GET NEW CHARACTER LOOP COUNT
017006  012702  000125            MOV  #125,R2      ;ESTABLISH CHECKERBOARD  REFERENCE
017012  012703  000252            MOV  #252,R3      ;ESTABLISH REVERSE LOOP COUNT
017016  110221           F3LP:    MOVB R2,(R1)+     ;INITIALIZE CHECKERBOARD CHARACTER
017020  110321                    MOVB R3,(R1)+     ;INITIALIZE REVERSE CHECKERBOARD CHARACTER
017022  005300                    DEC  R0           ;DECREMENT LOOP COUNT
017024  001374                    BNE  F3LP         ;REPEAT IF NOT DONE
017026  012705  000002            MOV  #2,R5        ;ESTABLISH OUTER LOOP COUNT
017032  012700  000020   F0LP:    MOV     #16.,R0   ;ESTABLISH INNER LOOP COUNT
017036  012702  002652            MOV  #S0ORG,R2    ;POINT R2 TO THE ORIGIN OF SLIDING ZERO
017042  012221           F4LP:    MOV  (R2)+,(R1)+  ;INITIALIZE BUFFER LOCATION
017044  005300                    DEC  R0           ;DECREMENT LOOP COUNT(INNER)
017046  001375                    BNE  F4LP         ;REPEAT IF NOT DONE
017050  012700  000020            MOV  #16.,R0      ;REESTABLISH INNER LOOP COUNT
017054  012702  002612            MOV  #S1ORG,R2    ;POINT R2 TO ORIGIN OF SLIDING 1 TABLE
017060  012221           F5LP:    MOV  (R2)+,(R1)+  ;INITIALIZE LOCATION WITH SLIDING 1 CONSTANT
017062  005300                    DEC  R0           ;DECREMENT LOOP COUNT
017064  001375                    BNE  F5LP         ;REPEAT IF NOT DONE
017066  005305                    DEC  R5           ;DECREMENT OUTER LOOP COUNT
017070  001360                    BNE  F0LP         ;REPEAT IF NOT DONE
017072  012605                    MOV  (SP)+,R5     ;RESTORE REGISTERS
017074  012603                    MOV  (SP)+,R3     ;
017076  012602                    MOV  (SP)+,R2     ;
```

```
017100  012601                  MOV  (SP)+,R1    ;
017102  012600                  MOV  (SP)+,R0    ;
017104  000205                  RTS  R5          ;RETURN TO TEST PROGRAM.
                                ;
                                ;                                9/29/75
                                ;
                                ;
                                ;  DATA COMPARE SUBROUTINE
                                ;
                                ;
                                ; THIS SUBROUTINE COMPARES DATA OBTAINED FROM THE
                                ; DISK DURING ANY DISK READ OPERATION IN THE INPUT
                                ; DATA BUFFER WITH THE REFERENCE DATA ACTUALLY WRITEN
                                ; ONTO THE DISK AND CONTAINED IN THE OUTPUT DATA BUFFER.
                                ;
                                ;
                                ;
                                ; AT ENTRY R0 CONTAINS THE NUMBER OF WORDS THAT
                                ; ARE TO BE COMPARED.
                                ;
                                ;
                                ; IF AN ERROR OCCURS:
                                ;
                                ;      1. THE ERROR FLAG IS SET AND CONTROL
                                ;         IS TRANSFERRED BACK TO THE CALLING PROGRAM.
                                ;
                                ;      2. THE CMPCNT COUNT LOCATION CONTAIN THE
                                ;         NUMBER OF THE COMPARISON AT WHIOCH THE ERROR WAS
                                ;         DETECTED.
                                ;
                                ;
                                ;
                                ;
                                ;
017106  010046          DATCMP: MOV  R0,-(SP)    ;SAVE REGISTERS
017110  010146                  MOV  R1,-(SP)    ;
017112  010446                  MOV  R4,-(SP)    ;
017114  005037  001134          CLR   CMPCNT     ;RESET COMPARISON WEORD COUNT
017120  013701  001234          MOV  CURDSK,R1   ;GET CURRENT DISK NUMBER
017124  006301                  ASL  R1          ;MAKE VALID WORD INDEX
017126  016101  001200          MOV  IBFTBL(R1),R1;GET INPUT BUFFER ORIGIN ADDRESS
017132  012704  044312          MOV  #ODBUF,R4   ;GET OUTPUT BUFFER ADDRESS
017136  012402          DTCMLP: MOV  (R4)+,R2    ;GET REFERENCE DATA WORD
017140  012103                  MOV  (R1)+,R3    ;GET ACTUAL DATA WORD
017142  020203                  CMP  R2,R3       ;ARE THEY EQUAL?
017144  001012                  BNE  DTCMER      ;EXIT WITH ERROR IF NOT
017146  005237  001134  DCMPA:  INC     CMPCNT          ;INCREMENT COMPARISON COUNT
017152  005300                  DEC  R0          ;DECREMENT WORD COUNT
017154  001370                  BNE  DTCMLP      ;REPEAT IF NOT DONE
017156  005037  001332          CLR     RTRYIN   ;RESET RETRY INDEX
017162  012604          DTCMEX: MOV  (SP)+,R4    ;RESTORE REGISTERS
017164  012601                  MOV  (SP)+,R1    ;
017166  012600                  MOV  (SP)+,R0    ;
017170  000205                  RTS  R5          ;RETURN
017172  005737  001244  DTCMER: TST     NONCON          ;CONSOLE ACTIVE?
017176  001035                  BNE     ERSET           ;SET ERROR IF NO
017200  005737  001104          TST     TTYPUT          ;TELETYPE HERE?
```

```
017204  001032                        BNE     ERSET           ;SET ERROR IF YES
017206  005737  177570                TST     CSWR    ;INHIBIT DATA ERROR ON?
017212  100027                        BPL     ERSET           ;SET ERROR FLAG IF NO
017214  005237  001314                INC     ERRCNT          ;INCREMENT ERRCNT IF YES
017220  032737  000001  177570        BIT     #1,CSWR         ;ARE WE IN RETRY MODE?
017226  001747                        BEQ     DCMPA           ;CHECK NEXT WORD IF NOT
017230  023727  001332  000020        CMP     RTRYIN,#16.     ;IS REENTRY INDEX LESS THAN 16.
017236  002012                        BGE     HDERR           ;THIS IS A "HARD ERROR" IF NO
017240  013700  001332                MOV     RTRYIN,R0       ;GET ENTRY INDEX
017244  005360  001336                DEC     ERRTAB(R0)      ;DECREMENT CURRENT RETRY INDEX COUNT
017250  005720                        TST     (R0)+           ;INCREMENT  INDEX BY TWO
017252  010037  001332                MOV     R0,RTRYIN       ;SAVE UPDATED RETRY INDEX
017256  005260  001336                INC     ERRTAB(R0)      ;INCREMENT APPROPRIATE RETRY INDEX
017262  000737                        BR      DTCMEX          ;GO TRY READ OPERATION AGAIN
017264  005237  001316        HDERR:  INC     HERCNT          ;INCREMENT HARD ERROR COUNT
017270  000726                        BR      DCMPA           ;GO ON TO NEXT WORD
017272  005237  001322        ERSET:  INC     ERRFLG          ;SET GENERAL ERROR FLAG
017276  005237  001320                INC     DATERR          ;SET DATA ERROR FLAG
017302  000727                        BR      DTCMEX          ;EXIT
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;    DISK OPERATION COMPLETE INTERRUPT SERVICE ROUTINE
                                      ;
                                      ;
                                      ;  THIS ROUTINE IS ENTERED WHENEVER A DISK OPERATION
                                      ; COMPLETE INTERRUPT IS GENERATED.THE ROUTINE SETS
                                      ; FLAGS IN THE INTERRUPT TABLE TO REMEMBER THE OCURRENCE
                                      ; OF THE INTERRUPT AS WELL AS TO IDENTIFY THE DISK
                                      ; DRIVE ASSOCIATED WITH THE DISK OPERATION
                                      ; THAT GENERATED THE INTERRUPT.
                                      ;
                                      ;
                                      ;
                                      ;
017304  010246                DSKDN:  MOV  R2,-(SP)    ;SAVE  R2
017306  105777  161500                TSTB    @DSTAT  ;IS THIS A SEEK INTERRUPT?
017312  100007                        BPL     DKDNEX  ;SET INTERRUPT FLAG IF NO
017314  017702  161472                MOV     @DSTAT,R2       ;GET CURRENT STATUS REGISTER
017320  042702  177774                BIC  #177774,R2 ;EXTRACT DISK DRIVE IDENTIFICATION BITS
017324  006302                        ASL  R2          ;MAKE DISK DRIVE NUMBER A VALID WORD INDEX
017326  010262  001122                MOV  R2,INTTBL(R2);SAVE IDENT. IN TABLE LOCATION
017332  005237  001150        DKDNEX: INC     INTFLG  ;SET GENERAL INTERRUPT FLAG
017336  012602                        MOV  (SP)+,R2    ;RESTORE R2
017340  000002                        RTI              ;EXIT FROM INTERRUPT
                                      ;
                                      ;                                   9/29/75
                                      ;    WAIT SUBROUTINE
                                      ;
                                      ;
                                      ;  THIS ROUTINE WAITS A MAXIMUM OF APPROX.
                                      ; 100MILLISECONDS OR UNTIL AN INTERRUPT OCCURS.
                                      ; WHICHEVER COMES FIRST.CONTROL IS THEN TRANS-
                                      ; FERRED TO THE CALLING PROGRAM.
```

```
                                                ;
                                                ;
                                                ;
                                                ;
017342  010046                         WAIT:    MOV     R0,-(SP)        ;SAVE R0 AND R1
017344  010146                                  MOV     R1,-(SP)        ;
017346  013700  001310                          MOV     TIMLP,R0                ;ESTABLISH OUTER WAIT LOOP
017352  012701  177777                 WTPLA:   MOV     #TIME,R1                ;ESTABLISH WAIT
017356  005737  001150                 WTPL:    TST     INTFLG                  ;HAS INTERRUPT OCURRED YET?
017362  001004                                  BNE     WTEXIT          ;EXIT IF YES
017364  005301                                  DEC     R1              ;DECREMENT LOOP COUNT
017366  001373                                  BNE     WTPL            ;RPEAT IF NOT DONE
017370  005300                                  DEC     R0              ;DECREMENT OUTER WAIT LOOP
017372  001367                                  BNE     WTPLA           ;LOOP AGAIN IF NO INTERRUPT
017374  012601                         WTEXIT:  MOV     (SP)+,R1        ;RESTORE R1 AND R0
017376  012600                                  MOV     (SP)+,R0        ;
017400  005046                                  CLR     -(SP)           ;LOWER CPU HERE
017402  012746  017410                          MOV     #WTPL1,-(SP)    ;
017406  000002                                  RTI                             ;
017410  000205                         WTPL1:   RTS     R5                      ;RETURN NOW
                                                ;       LDREAD.PAL
                                                ;
                                                ;
                                                ;       REGISTER LOAD AND READ TESTS
                                                ;
                                                ; THIS ROUTINE TESTS THE CAPABILITY OF THE
                                                ; SELECTED FORMATTER REGISTERS TO BE LOAD-
                                                ; ED AND INTERROGATED WITH ANY DATA. EACH REGISTER
                                                ; IS CHECKED WITH FOUR CONSTANTS 100 TIMES.
                                                ;
                                                ;
                                                ;   AT ENTRY,THE REGISTER INDEX IS IN R0.
                                                ; IF AN ERROR IS DETECTED,THE TEST EXITS TO
                                                ; PROGRAM CONTROL WITH THE ERROR FLAG SET.
                                                ;
                                                ;
017412  005004                         TN1:     CLR     R4              ;INITIALIZE SUBTEST NUMBER
017414  004537  004276                          JSR     R5,RQUEST               ;REQUEST 211
017420  005000                                  CLR     R0              ;INITIALIZE REGISTER INDEX
017422  004537  017530                          JSR     R5,LDRD         ;JUMP TO LOAD AND READ ROUTINE
017426  005737  001322                          TST     ERRFLG          ;ERROR OCCUR?
017432  001034                                  BNE TN1ERR              ;EXIT IF ERROR
017434  005204                         TN1A:    INC     R4              ;SUBTEST NUMBER=1 (DCSR)
017436  022010                                  CMP     (R0)+,(R0)      ;SELECT NEXT REGISTER
017440  004537  017530                          JSR     R5,LDRD         ;JUMP TO LOAD AND READ ROUTINE
017444  005737  001322                          TST     ERRFLG          ;ANY ERRORS?
017450  001025                                  BNE     TN1ERR          ;EXIT IF ERROR
017452  005204                         TN1B:    INC     R4              ;SUBTEST NUMBER=2 (DCAR)
017454  022010                                  CMP     (R0)+,(R0)      ;SELECT NEXT REGISTER
017456  004537  017530                          JSR     R5,LDRD         ;JMP YO LOAD AND READ ROUTINE
017462  005737  001322                          TST     ERRFLG          ;ANY ERRORS?
017466  001016                                  BNE     TN1ERR          ;EXIT IF ERROR
017470  005204                         TN1C:    INC     R4              ;SUBTEST NUMBER=3 (DWCNT)
017472  022010                                  CMP     (R0)+,(R0)      ;SELECT NEXT REGISTER
017474  004537  017530                          JSR     R5,LDRD         ;JUMP TO LOAD AND READ ROUTINE
017500  005737  001322                          TST     ERRFLG          ;ANY ERRORS?
017504  001007                                  BNE     TN1ERR          ;EXIT IF ERROR
```

```
017506  005204                     TN1D:   INC   R4              ;SUBTEST NUMBER=4 (DCYL)
017510  022010                             CMP   (RO)+,(RO)      ;SELECT NEXT REGISTER
017512  004537  017530                     JSR   R5,LDRD         ;JUMP TO LOAD AND READ ROUTINE
017516  005737  001322                     TST   ERRFLG          ;ANY ERRORS?
017522  001000                             BNE   TN1ERR          ;EXIT IF ERROR
017524  004537  003342             TN1ERR: JSR   R5,TSTCTL       ;EXIT TO PROGRAM CONTROL WITH ERROR
                                                                 ;FLAG SET IF A JUMP TO THIS INSTRUCTION
                                                                 ;WAS FROM LOAD AND READ ROUTINE.
                                                                 ;OTHERWISE ,ALL THE REGISTERS ARE CKECKED
                                                                 ;AND THE NEXT TEST CAN BE RUN
                                           ;
                                           ;
                                           ;LOAD AND READ ROUTINE
                                           ;
                                           ;
017530  010146                     LDRD:   MOV   R1,-(SP)        ;SAVE REGISTERS;R1,R4,AND R5
017532  010446                             MOV   R4,-(SP)        ;
017534  010546                             MOV   R5,-(SP)        ;
017536  012704  000004                     MOV   #4,R4           ;ESTABLISH CONSTANT LOOP COUNT
017542  012701  002712                     MOV   #CNSTNT,R1      ;POINT R1TO THE CONSTANT TABLE
017546  010005                             MOV   RO,R5           ;PUT THE REGISTER INDEX INTO  R5
017550  063705  001000                     ADD      DCSR,R5      ;COMPUTE REGISTER I/O ADDRESS
017554  012102                     LRLP1:  MOV   (R1)+,R2        ;GET REFERENCE CONSTANT
017556  046002  002552                     BIC   REGMSK(RO),R2;MASK OUT UNWANTED BITS
017562  005700                             TST      RO           ;TEST RO FOR REGISTER DCSR
017564  001006                             BNE      LRLP3        ;CONTINUE IF NOT DCSR
017566  053702  001102                     BIS      MCPU,R2 ;ENTER MULTPI
017572  042702  000100                     BIC      #INTON,R2;AND INTERRUPT ENABLE
017576  052702  000200                     BIS      #FMTRDY,R2;MASK IN FORMATTER READY
017602  012746  000144             LRLP3:  MOV      #100.,-(SP);LOOP COUNT
017606  010215                     LRLP2:  MOV      R2,@R5       ;LOAD REGISTER WITH CONSTANT
017610  011503                             MOV   @R5,R3          ;READ REGISTER BACK
017612  020203                     LRLP4:  CMP      R2,R3        ;IS THE DATA RIGHT?
017614  001014                             BNE   LRERR           ;NO:EXIT TO ERROR ROUTINE
017616  005316                             DEC   @SP             ;DECREMENT REPEAT LOOP COUNT
017620  001372                             BNE LRLP2             ;REPEAT IF NOT DONE WITH LOOPS
017622  005304                             DEC   R4              ;DECREMENT CONSTANT LOOP COUNT
017624  001403                             BEQ      LREX         ;EXIT IF DONE
017626  022606                             CMP      (SP)+,SP            ;FIX UP STACK
017630  000137  017554                     JMP      LRLP1        ;REPEAT TEST WITH NEXT CONSTANT
017634  022606                     LREX:   CMP   (SP)+,SP;FIX UP THE STACK POINTER
017636  012605                             MOV   (SP)+,R5        ;RESTORE REGISTERS;R5,R4 AND R1
017640  012604                             MOV   (SP)+,R4        ;
017642  012601                             MOV   (SP)+,R1        ;
017644  000205                             RTS   R5              ;RETURN TO TEST NUMBER ONE
017646  005237  001322             LRERR:  INC   ERRFLG          ;SET ERROR FLAG
017652  000770                             BR    LREX            ;EXIT TO TN1 WITH ERROR FLAG SET
                                           ;
                                           ;        SLIDING ONES AND ZEROES TESTS
                                           ;
                                           ; IN THE FOLLOWING TESTS ALL THE CONTROLLER
                                           ; REGISTERS ARE LOADED AND READ BACK WITH 15
                                           ; PATTERNS OF BOTH SINGLE BIT ON AND SINGLE
                                           ; BIT OFF DATA WORDS
                                           ;
017654  005004                     TN2:    CLR   R4              ;INITIALIZE SUBTEST NUMBER
017656  004537  004276                     JSR      R5,RQUEST         ;REQUEST  211
```

```
017662  012700  000000              MOV     #0,R0           ;INITIALIZE REGISTER INDEX NUMBER
017666  012701  002612              MOV     #S1ORG,R1       ;LOAD ADDRESS OF SLIDING ONE TABLE
017672  004537  020144     TN2A:    JSR     R5,SLIDE        ;JUMP TO SLIDE ROUTINE
017676  005737  001322              TST     ERRFLG          ;ANY ERRORS? (CONTROL AND STATUS)
017702  001035                      BNE     TN2ERR          ;EXIT IF ERROR
017704  005204                      INC     R4              ;SUBTEST NUMBER =1
017706  012010                      MOV     (R0)+,(R0)      ;SELECT NEXT REGISTER
017710  004537  020144     TN2B:    JSR     R5,SLIDE        ;JUMP TO SLIDE
017714  005737  001322              TST     ERRFLG          ;ANY ERRORS? (UNIT HEAD AND SECTOR )
017720  001026                      BNE     TN2ERR          ;EXIT IF ERROR
017722  005204                      INC     R4              ;SUBTEST NUMBER=2 (CORE ADDR. REG)
017724  012010                      MOV     (R0)+,(R0)      ;SELECT NEXT REGISTER
017726  004537  020144     TN2C:    JSR     R5,SLIDE        ;JMP TO SLIDE
017732  005737  001322              TST     ERRFLG          ;ANY ERRORS?
017736  001017                      BNE     TN2ERR          ;EXIT IF ERRORS
017740  005204                      INC     R4              ;SUBTEST NUMBER=3 (WORD COUNT REG.)
017742  012010                      MOV     (R0)+,(R0)      ;SELECT NEXT REGISTER
017744  004537  020144     TN2D:    JSR     R5,SLIDE        ;JUMP TO SLIDE ROUTINE
017750  005737  001322              TST     ERRFLG          ;ANY ERRORS
017754  001010                      BNE     TN2ERR          ;EXIT IF ERROR
017756  005204                      INC     R4              ;SUBTEST NUMBER =4 (CYLINDER ADDRESS REG)
017760  012010                      MOV     (R0)+,(R0)      ;SELECT NEXT REGISTER
017762  004537  020144     TN2E:    JSR     R5,SLIDE        ;JUMP TOSLIDE ROUTINE
017766  005737  001322              TST     ERRFLG          ;ANY ERRORS?
017772  001001                      BNE     TN2ERR          ;EXIT IF ERROR
017774  000402                      BR          TN2OUT      ;EXIT TO NEXT TEST
017776  005237  001322     TN2ERR:  INC     ERRFLG          ;SET ERROR FLAG
020002  004537  003342     TN2OUT:  JSR         R5,TSTCTL           ;EXIT TO NEXT TEST
                                    ;
                                    ;       SLIDING ZERO TEST
                                    ;
                                    ;
020006  005004             TN3:     CLR     R4              ;INITIALIZE SUBTEST NUMBER
020010  004537  004276              JSR         R5,RQUEST       ;REQUEST 211
020014  012700  000000              MOV     #0,R0           ;INITIALIZE REGISTER INDEX
020020  012701  002652              MOV     #S0ORG,R1        ;LOAD ADDRESS OF SLIDING ZERO TABLE
020024  004537  020144     TN3A:    JSR     R5,SLIDE        ;JUMP TOSLIDE ROUTINE
020030  005737  001322              TST     ERRFLG          ;ANY ERRORS? (CONTROL AND STATUS)
020034  001402                      BEQ         .+6         ;CONTINUE IF NO
020036  000137  020136              JMP         TN3ERR      ;EXIT IF YES
020042  005204                      INC     R4              ;SUBTEST NUMBER =1 (UNIT,HEAD AND SECTOR)
020044  012010                      MOV     (R0)+,(R0)      ;SELECT NEXT REGISTER
020046  004537  020144     TN3B:    JSR     R5,SLIDE        ;JUMP TO SLIDE ROUTINE
020052  005737  001322              TST     ERRFLG          ;ANY ERRORS?
020056  001027                      BNE     TN3ERR          ;EXIT IF ERROR
020060  005204                      INC     R4              ;SUBTEST NUMBER=2 (CORE ADDR. REGISTER)
020062  012010                      MOV     (R0)+,(R0)      ;SELECT NEXT REGISTER
020064  004537  020144     TN3C:    JSR     R5,SLIDE        ;JUMP TO SLIDE ROUTINE
020070  005737  001322              TST     ERRFLG          ;ANY ERRORS?
020074  001020                      BNE     TN3ERR          ;EXIT IF ERROR
020076  005204                      INC     R4              ;SUBTEST NUMBER=3 (WORD COUNT REGISTER)
020100  012010                      MOV     (R0)+,(R0)      ;SELECT NEXT REGISTER
020102  004537  020144     TN3D:    JSR     R5,SLIDE        ;JUMP TO SLIDE ROUTINE
020106  005737  001322              TST     ERRFLG          ;ANY ERRORS?
020112  001011                      BNE     TN3ERR          ;EXIT IF ERROR
020114  005204                      INC     R4              ;SUB TEST NUMBER =4
020116  012010                      MOV     (R0)+,(R0)      ;SEL. NEXT REG. (CYLINDER ADDRESSS REG)
```

```
020120  004537  020144          TN3E:    JSR   R5,SLIDE     ;JMP TO SLIDE ROUTINE
020124  005737  001322                   TST   ERRFLG       ;ANY ERRORS?
020130  001002                           BNE   TN3ERR       ;EXIT IF ERROR
                                         ;
        001004                           DBAR=DCAR                  ;REDEFINE CORE ADDRESS REGISTER FOR THOSE
                                         ;                          PEOPLE WHO CALL IT BUSS ADDRESS REGISTER
                                         ;
020132  004537  003342          TN3OUT:  JSR      R5,TSTCTL         ;EXIT IF DONE
                                         ;
                                         ;
                                         ;
                                         ;
020136  005237  001322          TN3ERR:  INC      ERRFLG            ;SET ERROR FLAG
020142  000773                           BR       TN3OUT            ;AND EXIT
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;    SLIDE ROUTINE
                                         ;
                                         ; THIS ROUTINE CONTROLS THE  LOADING OF THE
                                         ; SLIDING ZERO AND ONE PATTERNS INTO THE
                                         ; DISK CONTROLLER REGISTERS.
                                         ;
                                         ;
020144  010146                  SLIDE:   MOV   R1,-(SP)     ;SAVE REGISTERS R1,R4,R5
020146  010446                           MOV   R4,-(SP)     ;
020150  010546                           MOV   R5,-(SP)     ;
020152  012704  000020                   MOV   #20,R4       ;ESTABLISH LOOP COUNT
020156  010005                           MOV   R0,R5        ;EST REGISTER INDEX INTO R5
020160  063705  001000                   ADD      DCSR,R5 ;COMPUTE REGISTER I/O ADDRESS
020164  012102                  SL1:     MOV   (R1)+,R2     ;GET REFERENCE CONSTANT
020166  046002  002552                   BIC   REGMSK(R0),R2;MASK OUT UNWANTED BITS
020172  005700                           TST      R0        ;IS THIS THE DCSR?
020174  001006                           BNE      SLIDE1    ;NO CONTINUE
020176  052702  000200                   BIS      #FMTRDY,R2;OR IN FORMATTER READY
020202  053702  001102                   BIS      MCPU,R2 ;ENTER REQUEST BIT
020206  042702  000100                   BIC      #INTON,R2;LOOSE INTERRUPT ENABLE
020212  010215                  SLIDE1:  MOV      R2,@R5    ;LOAD REGISTER
020214  011503                           MOV   @R5,R3       ;READ REGISTER BACK
020216  020203                  SLIDE2:  CMP      R2,R3     ;IS THE DATA CORRECT?
020220  001006                           BNE   SLERR        ;NO EXIT TO ERROR ROUTINE
020222  005304                           DEC   R4           ;DECREMENT LOOP COUNT
020224  001357                           BNE   SL1          ;REPEAT TEST LOOP WITH NEXT CONSTANT
020226  012605                  SLEX:    MOV   (SP)+,R5     ;RETRIEVE REGISTERS FROM STACK
020230  012604                           MOV   (SP)+,R4     ;
020232  012601                           MOV      (SP)+,R1;
020234  000205                           RTS   R5           ;EXIT TO FETCH ANOTHER REGISTER
020236  005237  001322          SLERR:   INC   ERRFLG       ;ERROR OCURRED SET ERROR FLAG
020242  000771                           BR    SLEX         ;JUMP TO EXIT ROUNTINE
                                         ;ERREG.PAL
                                         ;
                                         ;
                                         ;
                                         ;       4M.ECC  TEST NUMBERS 4>> 7      NOVEMBER-13-77
```

```
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       DISK ERROR AND STATUS REGISTER TESTS
                                        ;
                                        ;
                                        ;     THESE TESTS VERIFY THAT THE NONEXISTANT HEAD,
                                        ; SECTOR AND CYLINDER(TRACK) ERRORS CAN
                                        ; BE SET AND CLEARED PROPERLY.
                                        ;
                                        ;       ALSO TESTED INTHIS TEST IS THE BYTE LOADING
                                        ;       CAPABILITY OF THE COMMAND AND STATUS REG(DCSR). ;
020244  004537  004276          TN4:    JSR     R5,RQUEST       ;FETCH 211
020250  000005                          RESET           ;RESET I/O BUSS
020252  004537  004276                  JSR     R5,RQUEST       ;GET 211 AGAIN
020256  005004                          CLR  R4         ;INITIALIZE SUBTEST NUMBER=0
020260  004537  021100                  JSR  R5,TSTDSK  ;FORMATTER READY?,WAIT IF NOT
020264  013701  001040                  MOV     MAXSEC,R1       ;LOAD ILLEGAL SECTOR NUMBER
020270  005201                          INC     R1              ;
020272  010177  160504                  MOV     R1,@DUSH        ;
020276  004537  016332                  JSR     R5,DISKID       ;SET UP CURRENT DISK SELECT
020302  050277  160474                  BIS     R2,@DUSH        ;SELECT DISK
020306  052777  000077  160464          BIS     #NOPCMD,@DCSR   ;ISSUE NOP COMAND
020314  004537  021100                  JSR  R5,TSTDSK  ;WAIT FORMATTER READY
020320  005777  160454                  TST     @DCSR   ;ERROR
020324  100402                          BMI   .+6       ;CONTINUE IF YES
020326  000137  021020          TN4EX:  JMP  TN4ERR     ;NO:GOTO NEXT TEST
020332  005204                          INC     R4              ;SUBTEST NUMBER=1(ERROR DETECTED)
020334  012702  000100                  MOV     #100,R2 ;SET UP REFERENCE WORD
020340  017703  160450                  MOV     @DERR,R3        ;FETCH ERROR REGISTER
020344  042703  010000                  BIC     #10000,R3       ;CLEAR DRIVE NOT READY IF SET
020350  020203                          CMP     R2,R3   ;ERROR DETECTED?
020352  001365                          BNE     TN4EX   ;NOT DETECTED SET ERROR FLAG
020354  005204                  TN4A:   INC     R4      ;SUBTEST NUMBER=2(CLR SECTOR ERROR)
020356  004537  021030                  JSR  R5,CLRERR  ;JUMP TO CLEAR ERROR
020362  005737  001322                  TST  ERRFLG     ;ERROR FLAG SET?
020366  001402                          BEQ   .+6       ;NO,CONTINUE
020370  000137  021024                  JMP  TN4FIN     ;YES:EXIT TO NEXT TEST
020374  005204                  TN4B:   INC     R4              ;SUBTEST NUMBER=3(FORCE HEAD ERROR)
020376  013702  001042                  MOV     MAXHD,R2        ;LOAD MAXIMUM HEAD ADDRESS
020402  062702  000200                  ADD     #HDINC,R2       ;MAKE ILLEGAL ADDRESS
020406  010277  160370                  MOV     R2,@DUSH        ;LOAD INTO DUSH REGISTER
020412  004537  016332                  JSR     R5,DISKID       ;SET UP CURENT DISK SELECT
020416  050277  160360                  BIS     R2,@DUSH        ;OR IN SELECT
020422  052777  000077  160350          BIS     #NOPCMD,@DCSR   ;ISSUE NOP COMMAND
020430  004537  021100                  JSR     R5,TSTDSK       ;WAIT FORMATTER READY
020434  005777  160340                  TST     @DCSR   ;ERROR DETECTED?
020440  100332                          BPL  TN4EX      ;NOT DETECTED GO TO NEXT TEST
020442  005204                          INC     R4              ;SUBTEST NUMBER=4(ERROR DETECTED)
020444  012702  000020                  MOV     #20,R2          ;SET UP REFERENCE WORD
020450  017703  160340                  MOV     @DERR,R3        ;FETCH ERROR REGISTER
020454  042703  010000                  BIC     #10000,R3       ;CLR DRIVE NOT READY IF SET
020460  020203                          CMP     R2,R3           ;EQUAL?
020462  001321                          BNE     TN4EX           ;NOT EQUAL EXIT
020464  005204                  TN4C:   INC     R4              ;SUBTEST NUMBER=5(HEAD ERROR DETECTED)
020466  004537  021030                  JSR  R5,CLRERR  ;CLEAR ERROR FLAG
```

```
020472 005737 001322                 TST    ERRFLG          ;ERROR FLAG CLEARED?
020476 001152                        BNE    TN4FIN          ;NOT CLEARED:EXIT TO NEXT TEST
020500 005204                 TN4D:  INC    R4              ;SUBTEST NUMBER=6(ERROR CLEARED)
020502 004537 016332                 JSR    R5,DISKID       ;SET UP CURRENT DISK SELECT
020506 010277 160270                 MOV    R2,@DUSH        ;SELECT DISK
020512 013702 001044                 MOV    MAXCYL,R2       ;LOAD ILLEGAL CYLINDER ADDRESS
020516 005202                        INC    R2              ;
020520 010277 160264                 MOV    R2,@DCYL        ;LOAD INTO CYLINDER ADDRESS REGISTER
020524 052777 000077 160246          BIS    #NOPCMD,@DCSR   ;ISSUE NOP COMMAND
020532 004537 021100                 JSR    R5,TSTDSK       ;WAIT FORMATTER READY
020536 005777 160236                 TST    @DCSR           ;ERROR DETECTED?
020542 100126                        BPL    TN4ERR          ;NOT DETECTED:EXIT TO NEXT TEST
020544 005204                        INC    R4              ;SUBTEST NUMBER=7(ERROR DETECTED)
020546 012702 000040                 MOV    #40,R2          ;SET UP REFERENCE WORD
020552 017703 160236                 MOV    @DERR,R3        ;FETCH ERROR REGISTER
020556 042703 010000                 BIC    #10000,R3       ;EX TRACT DRIVE NOT READY
020562 020203                        CMP    R2,R3           ;
020564 001260                        BNE    TN4EX           ;EXIT IF NOT EQUAL
020566 005204                 TN4E:  INC    R4              ;SUBTEST NUMBER=10(NONEX CYL OK)
020570 004537 021030                 JSR    R5,CLRERR       ;CLEAR ERROR FLAG
020574 005737 001322                 TST    ERRFLG          ;ERROR FLAG CLEARED?
020600 001111                        BNE    TN4FIN          ;NOT CLEARED:EXIT TO NEXT TEST
020602 005204                 TN4F:  INC    R4              ;SUBTEST NUMBER=15(ERROR CLEARED)
020604 013702 001040                 MOV    MAXSEC,R2       ;FORCE ALL THREE NONEXISTANT ERRORS
020610 005202                        INC    R2              ;
020612 063702 001042                 ADD    MAXHD,R2        ;INCLUDE HEAD
020616 062702 000200                 ADD    #HDINC,R2       ;
020622 010277 160154                 MOV    R2,@DUSH        ;LOAD INTO DUSH
020626 004537 016332                 JSR    R5,DISKID       ;SET UP CURRENT DISK SELECT
020632 050277 160144                 BIS    R2,@DUSH        ;OR IT IN
020636 013777 001044 160144          MOV    MAXCYL,@DCYL    ;LOAD ILLEGAL CYLINDER ADDR
020644 005277 160140                 INC    @DCYL           ;
020650 052777 000077 160122          BIS    #NOPCMD,@DCSR   ;SEND NOP COMMAND
020656 004537 021100                 JSR    R5,TSTDSK       ;WAIT FORMATTER READY
020662 005777 160112                 TST    @DCSR           ;ERROR DETECTED?
020666 100054                        BPL    TN4ERR          ;NOT DETECTED:EXIT TO NEXT TEST
020670 005204                        INC    R4              ;SUBTEST NUMBER=12(ERROR DETECTED)
020672 012702 000160                 MOV    #160,R2 ;SET UP REFERENCE WORD
020676 017703 160112                 MOV    @DERR,R3;FETCH ERROR REGISTER
020702 042703 010000                 BIC    #10000,R3       ;CLEAR DRIVE NOT READY
020706 020203                        CMP    R2,R3   ;EQUAL?
020710 001043                        BNE    TN4ERR  ;EXIT IF NOT
020712 005204                 TN4G:  INC    R4              ;SUBTEST NUMBER=13(ERRORS OK)
020714 004537 021030                 JSR    R5,CLRERR       ;CLEAR ERROR FLAG
020720 005737 001322                 TST ERRFLG     ;ERROR FLAGS CLEARED?
020724 001035                        BNE    TN4ERR  ;EXIT IF NOT CLEARED
020726 005204                        INC    R4              ;SUBTEST NUMBER=14(ERRORS CLEARED)
                                                            ;CHECK DCSR BYTE LOAD
020730 013701 001000                 MOV    DCSR,R1         ;ATTEMPT BYTE LOAD OF DCSR
020734 005201                        INC    R1              ;POINT R1 TO DCSR+1
020736 012702 000177                 MOV    #177,R2         ;SET UP DATA WORD
020742 150211                        BISB   R2,(R1)         ;LOAD I BYTE OF DCSR
020744 111103                        MOVB   (R1),R3         ;REREAD DCSR+1
020746 020203                        CMP    R2,R3           ;READ OK?
020750 001023                        BNE    TN4ERR  ;EXIT IF NO
020752 005204                        INC    R4              ;SUBTEST NUMBER=15(UPPER BYTE OK)
020754 152777 000176 160016          BISB   #176,@DCSR      ;LOAD LOWER BYTE OF DCSR
```

```
020762  112702  000176                       MOVB#176,R2      ;LOAD R2 WITH REFERENCE
020766  117703  160006                       MOVB    @DCSR,R3         ;FETCH DCSR REGISTER
020772  042703  177600                       BIC     #177600,R3;EXTRACT LOWER BITS
020776  005737  001076                       TST     MULCPU           ;IS THIS A MULTIPLE CPU SET UP?
021002  001404                               BEQ     TN4H             ;CONTINUE IF NO
021004  042702  000040                       BIC     #BIT5,R2         ;LOOSE BIT 5 IF YES
021010  042703  000040                       BIC     #BIT5,R3         ;IN BOTH WORDS
021014  020203                       TN4H:   CMP     R2,R3            ;ARE THEY EQUAL?
021016  001402                               BEQ     TN4FIN  ;EXIT IF YES
021020  005237  001322               TN4ERR: INC  ERRFLG     ;SET ERROR FLAG
021024  004537  003342               TN4FIN: JSR  R5,TSTCTL  ;EXIT TO PROGRAM CONTROL
                                                              ;
                                                              ;
                                             ;    CLEAR ERROR FLAGS
                                             ;
                                             ;
                                             ; THIS ROUTINE CLEARS ERROR FLAGS AND TESTS THAT
                                             ;THEY ARE ACTUALLY CLEARED,THEN RETURNS TO THE
                                             ;ERROR  STATUS REGISTER TESTS
                                             ;
                                             ;
021030  042777  000136  157742  CLRERR: BIC     #FUNC,@DCSR      ;LOOSE FINCUION BITS
021036  052777  000001  157734          BIS     #SYSCLR,@DCSR    ;ISSUE FORMATTER CLREAR
021044  004537  004276                  JSR     R5,RQUEST        ;GET 211
021050  004537  021100                  JSR  R5,TSTDSK  ;WAIT FORMATTER READY
021054  005777  157720                  TST     @DCSR    ;ERROR(S) CLEARED?
021060  100404                          BMI     CERR     ;NOT CLEARED EXIT TO SET ERROR FLAG
021062  005777  157726                  TST     @DERR    ;ERROR CLEARED INDERR?
021066  001001                          BNE  CERR        ;NO: SET ERROR FLAG
021070  000402                          BR   .+6         ;EXIT
021072  005237  001322          CERR:   INC  ERRFLG      ;SET ERROR FLAG
021076  000205                          RTS  R5          ;RETURN TO  TN4(ERROR REG TEST)
021100  105777  157674          TSTDSK: TSTB @DCSR       ;TEST FORMATTER READY?
021104  100375                          BPL     TSTDSK   ;WAIT
021106  000205                          RTS  R5          ;JUMP BACK TO TN4
                                        ;UNSEL.PAL
                                        ;                                 9/23/75
                                        ;
                                        ;
                                        ;
                                        ;  THE UNIT SELECT TESTS VERIFY THAT THE UNIT
                                        ; SELECT LINES ACCESS THE PROPER DISK DRIVES.
                                        ;  THIS TESTING IS ACCOMPLISHED BY SELECTING
                                        ;       EACH DRIVE AND ISSING A RECALIBRATE.
                                        ;       THE DSTAT REGISTER SEEKING FLOPS ARE
                                        ;  CHECKED FOR PROPER SEEKING STATUS AND
                                        ;  A FORMATTER CLEAR IS THEN GIVEN TO CLEAR THEM.
                                        ;       THIS TEST DOES NOT WAIT FOR SEEK DONE.
                                        ;
                                        ;
                                        ;
021110  005004                  TN5:    CLR  R4          ;INITIALIZE SUBTEST NUMBER
021112  004537  004276                  JSR     R5,RQUEST        ;GET 211
021116  105777  157656                  TSTB    @DCSR    ;FORMATTER READY?
021122  100126                          BPL  TN5ERR      ;NO:EXIT
021124  004537  004066                  JSR     R5,SEKTIM        ;SET UP WAIT LOOP
021130  005737  001322                  TST     ERRFLG           ;ERRORS?
```

```
021134  001421                           BEQ     TN5A            ;CONTINUE IF OK
021136  013702  001234                   MOV     CURDSK,R2       ;SET UP REFERENCE FOR ERROR REPORTING
021142  013703  002616                   MOV     S1ORG+4,R3      ;USE R3 TO HELP SET UP REFERENCE
021146  005302                   TN5AA:  DEC     R2              ;DECREMENT COUNTER
021150  100402                           BMI     TN5AB           ;GO ON IF DONE
021152  006303                           ASL     R3              ;SLIDE SEEK INDIVIDUAL DONE BIT
021154  000774                           BR      TN5AA           ;CHECK AGAIN
021156  013702  001234           TN5AB:  MOV     CURDSK,R2       ;RELOAD CURRENT DISK
021162  006302                           ASL     R2              ;MAKE USEFUL
021164  060302                           ADD     R3,R2           ;COMBINE TWO
021166  052702  100200                   BIS     #BIT15!BIT7,R2  ;INCLUDE DRIVE READY AND SEEK DONE SUMMARY
021172  017703  157614                   MOV     @DSTAT,R3       ;FETCH CULPRIT
021176  000500                           BR      TN5ERR          ;AND THEN
021200  052777  000077  157572   TN5A:   BIS     #NOPCMD,@DCSR   ;CLEAR SEEK DONE
021206  005204                           INC     R4              ;SUBTEST NUMBER=1(GOT FORMATER AND DISK)
021210  013702  002632                   MOV     S1ORG+20,R2     ;SET UP SEEKING REFERENCE
021214  005001                           CLR     R1              ;CLEAR INUT SELECT BIT
021216  004537  021406                   JSR     R5,SELECT       ;SELECT UNIT 0
021222  005737  001322                   TST     ERRFLG          ;ERROR FLAG SET?
021226  001062                           BNE     TN5OUT          ;YES:EXIT
021230  005204                           INC     R4              ;SUBTEST NUMBER=2
021232  004537  021510                   JSR     R5,CLRFTR       ;CLEAR FORMATTER
021236  005737  001322                   TST     ERRFLG          ;ERROR FLAG SET?
021242  001054                           BNE     TN5OUT          ;YES:EXIT
021244  005204                   TN5B:   INC     R4              ;SUBTEST=3
021246  006302                           ASL     R2              ;UP DATE SEEK COMPARE BIT
021250  063701  002642                   ADD     S1ORG+30,R1     ;SELECT NEXT UNIT
021254  004537  021406                   JSR     R5,SELECT       ;SELECT UNIT 1
021260  005737  001322                   TST     ERRFLG          ;ERROR FLAG SET
021264  001043                           BNE     TN5OUT
021266  005204                           INC     R4              ;SUBTEST NUMBER=4
021270  004537  021510                   JSR     R5,CLRFTR       ;CLEAR FORMATTER
021274  005737  001322                   TST     ERRFLG          ;ERROR FLAG SET?
021300  001035                           BNE     TN5OUT          ;YES:EXIT
021302  005204                   TN5C:   INC     R4              ;SUBTEST NUMBER=5
021304  006302                           ASL     R2              ;UPDATRE SEEK COMPARE BIT
021306  006301                           ASL     R1              ;AND UNIT SELECT BITS
021310  004537  021406                   JSR     R5,SELECT       ;SELECT UNIT 2
021314  005737  001322                   TST     ERRFLG          ;ERRROR FLAG SET?
021320  001025                           BNE     TN5OUT          ;YES:EXIT
021322  005204                           INC     R4              ;SUBTEST NUMBER=6
021324  004537  021510                   JSR     R5,CLRFTR       ;CLEAR FORMATTER
021330  005737  001322                   TST     ERRFLG          ;ERROR FLAG SET?
021334  001017                           BNE     TN5OUT          ;YES:EXIT
021336  005204                   TN5D:   INC     R4              ;SUBTEST NUMBER=7
021340  006302                           ASL     R2              ;UPDATE UNIT SEEK COMPARE BIT
021342  063701  002642                   ADD     S1ORG+30,R1     ;SELECT NEXT UNIT
021346  004537  021406                   JSR     R5,SELECT       ;SELECT UNIT 3
021352  005737  001322                   TST     ERRFLG          ;ERROR FLAG SET?
021356  001006                           BNE     TN5OUT          ;YES:EXIT
021360  005204                           INC     R4              ;SUBTEST NUMBER=8
021362  004537  021510                   JSR     R5,CLRFTR       ;CLEAR FORMATTER
021366  005737  001322                   TST     ERRFLG          ;ERROR FLAG SET?
021372  001000                           BNE     TN5OUT          ;YES:EXIT
021374  004537  003342           TN5OUT: JSR     R5,TSTCTL       ;EXIT TO TEST CONTROL
021400  005237  001322           TN5ERR: INC     ERRFLG          ;SET ERROR FLAG
021404  000773                           BR      TN5OUT          ;EXIT
```

```
                                         ;
                                         ;
                                         ;
021406  010177  157370          SELECT: MOV     R1,@DUSH        ;SELECT NEXT UNIT
021412  042777  000136  157360          BIC     #FUNC,@DCSR     ;CLEAR FUNCTION BITS
021420  052777  000024  157352          BIS     #RTZCMD,@DCSR   ;ISSUE RECALIBRATE
021426  005277  157346                  INC     @DCSR   ;SET GO
021432  012703  000020                  MOV     #20,R3  ;INSURE FORMATTER BUSY
021436  005303                          DEC     R3      ;
021440  001376                          BNE     .-2     ;
021442  105777  157332          TSTB    @DCSR           ;WAIT FORMATTER READY
021446  100375                          BPL  .-4        ;WAIT
021450  017703  157336                  MOV     @DSTAT,R3       ;FETCH DISK STATUS REGISTER
021454  042703  100000                  BIC     #BIT15,R3       ;EXTRACT POSSIBLE DRIVE READY
021460  042703  040000                  BIC     #BIT14,R3       ;EXTRACT POSSIBLE PORT BUSY
021464  042703  020000                  BIC     #BIT13,R3       ;AND ALSO POSSIBLE WRITE PROTECT
021470  020203                          CMP     R2,R3           ;RIGHT FLOP SET?
021472  001003                          BNE     SELA            ;NO EXIT
021474  004537  022164                  JSR     R5,WAIT1        ;WAIT FOR SEEK DONE IF DRIVE READY
021500  000205                          RTS     R5      ;RETURN
021502  005237  001322          SELA:   INC  ERRFLG     ;SET ERROR FLAG
021506  000205                          RTS  R5
                                        ;
                                        ;
                                        ;
021510  042777  000136  157262  CLRFTR: BIC     #FUNC,@DCSR     ;LOOSE FUNCTION BITS
021516  052777  000001  157254          BIS     #SYSCLR,@DCSR   ;ISSUE SYSCLR
021524  004537  004276                  JSR     R5,RQUEST       ;GET 211
021530  105777  157244                  TSTB    @DCSR   ;FORMATTER READY?
021534  100402                          BMI     .+6     ;EXIT IF YES
021536  005237  001322                  INC     ERRFLG  ;SET ERROR FLAG IF  NO
021542  000205                          RTS  R5         ;RETURN TO PROGRAM TN5
                                        ;SEKTST.PAL
                                        ;                               9/23/75
                                        ;
                                        ;
                                        ;               SEEKING TESTS
                                        ;
                                        ;   THESE TESTS SEND SEEK COMMANDS (2) TO THE
                                        ; DRIVE UNIT CURRENTLY UNDER TEST AND CHECKS THAT
                                        ; THE PROPER DRIVE HAS DONE A SEEK.THERE IS NO ACTUAL
                                        ; DATA TRANSFER BETWEEN THE PROCESSOR AND THE DISK
                                        ; IN THESE TESTS.
                                        ;
                                        ;
                                        ;
021544  005004                  TN6:    CLR  R4         ;INITIALIZE SUBTEST NUMBER
021546  004537  004276                  JSR     R5,RQUEST       ;GET 211
021552  052777  000001  157220          BIS     #SYSCLR,@DCSR   ;ISSUE SYSTEM CLEAR
021560  004537  004276                  JSR     R5,RQUEST       ;GET IT AGAIN
021564  105777  157210                  TSTB    @DCSR   ;TEST FORMATTER READY
021570  100145                          BPL  TN6ERR     ;NOT READY:EXIT
021572  005204                          INC     R4      ;SUBTEST NUMBER=1
021574  004537  016332                  JSR     R5,DISKID       ;SET UP UNIT SELECT BIT
021600  010201                          MOV     R2,R1           ;SAVE IN REG R1
021602  013702  002616                  MOV     S1ORG+4,R2      ;SET UP SEEKING FLOP
021606  013700  001234                  MOV     CURDSK,R0       ;
```

```
021612  005300                        ST61:   DEC     R0                      ;FIND REFERENCE BIT
021614  100402                                BMI     ST62                    ;EXIT IF DONE
021616  006302                                ASL     R2                      ;SHIFT REFERENCE
021620  000774                                BR      ST61                    ;CONTINUE IF NOT DONE
021622  010177  157154                ST62:   MOV     R1,@DUSH                ;SELECT DRIVE
021626  010200                                MOV     R2,R0                   ;SAVE SEEK COMPARE DONE
021630  005777  157156                        TST     @DSTAT  ;IS SELECTED UNIT READY?
021634  100123                                BPL     TN6ERR  ;NO:EXIT
021636  005204                        TN6C:   INC     R4      ;SUBTEST NUMBER=2 UNIT READY
021640  052777  000025  157132                BIS     #RECAL,@DCSR    ;RECALIBRATE
021646  004537  022112                        JSR     R5,TN6CHK       ;WAIT SEEK DONE
021652  005737  001322                        TST     ERRFLG  ;ERROR FLAG SET?
021656  001110                                BNE     TN6OUT  ;YES:EXIT
021660  005204                        TN6D:   INC     R4      ;SUBTEST NUMBER=3
021662  004537  022140                        JSR     R5,T6DONE       ;PROPER DRIVE DONE?
021666  005737  001322                        TST     ERRFLG  ;ERROR FLAG SET?
021672  001102                                BNE     TN6OUT  ;YES:EXIT
021674  005204                        TN6E:   INC     R4      ;SUBTEST NUMBER=4
021676  105777  157076                        TSTB    @DCSR   ;FORMATTER READY?
021702  100100                                BPL     TN6ERR  ;NO:EXIT
021704  005204                        TN6F:   INC     R4      ;SUBTEST NUMBER=5
021706  013777  001044  157074                MOV     MAXCYL,@DCYL    ;LOAD ILLEGAL TRACK ADDRESS
021714  005277  157070                        INC     @DCYL           ;
021720  042777  000136  157052                BIC     #FUNC,@DCSR     ;CLEAR FUNCTION BITS
021726  052777  000002  157044                BIS     #SEKCMD,@DCSR   ;ISSUE SEEK
021734  005277  157040                        INC     @DCSR           ;SET GO
021740  004537  022164                        JSR     R5,WAIT1        ;WAIT
021744  017703  157044                        MOV     @DERR,R3        ;GET ERROR REGISTER
021750  042703  177737                        BIC     #177737,R3;MASK OUT ALL OTHER ERRORS
021754  012702  000040                        MOV     #40,R2  ;SET UP REFERNECE WORD
021760  020203                                CMP     R2,R3   ;ILLEGAL CYLINDER BIT SET?
021762  001050                                BNE     TN6ERR  ;EXIT IF NO
021764  005204                        TN6J:   INC     R4      ;SUBTEST NUMBER=6
021766  017703  157022                        MOV     @DERR,R3        ;REREAD ERROR REGISTER
021772  042703  000040                        BIC     #40,R3  ;CLEAR TRACK ERROR
021776  005002                                CLR     R2      ;SET UP REFERENCE WORD
022000  020203                                CMP     R2,R3   ;EQUAL?
022002  001040                                BNE     TN6ERR  ;EXIT IF NO
022004  005204                        TN6K:   INC     R4      ;SUBTEST NUMBER=7(NO OTHER ERRORS)
022006  004537  021510                        JSR     R5,CLRFTR       ;CLEAR FORMATTER
022012  010177  156764                        MOV     R1,@DUSH;RESELECT DRIVE
022016  052777  000002  156754                BIS     #SEKCMD,@DCSR   ;ISSUE SEEK
022024  005277  156750                        INC     @DCSR   ;AND GO BIT
022030  004537  022164                        JSR     R5,WAIT1        ;ISSUE SEEK COMMAND
022034  105777  156752                        TSTB    @DSTAT  ;SEEK DONE?
022040  100021                                BPL     TN6ERR  ;EXIT IF NO
022042  005204                                INC     R4      ;SUBTEST NUMBER=10(SEEK DONE)
022044  004537  022140                        JSR     R5,T6DONE ;SEE IF PROPER DRIVE DONE
022050  005737  001322                        TST     ERRFLG  ;ERRORS?
022054  001011                                BNE     TN6OUT  ;EXIT IF YES
022056  005204                                INC     R4      ;SUBTEST NUMBER=11(PROPER DRIVE DONE)
022060  017703  156726                        MOV     @DSTAT,R3       ;FETCH DISK STATUS REGISTER
022064  042703  100777                        BIC     #100777,R3      ;ETRACT SEEKING FLOPS
022070  005002                                CLR     R2              ;SET UP REFERENCE WORD
022072  020203                                CMP     R2,R3           ;EQUAL?
022074  001003                                BNE     TN6ERR          ;EXIT IF NO
022076  005204                                INC     R4      ;SUBTEST NUMBER=12(NO ERRONEOUS FLOPS SET)
```

```
                                              ;
                                              ;
022100  004537  003342          TN6OUT: JSR   R5,TSTCTL   ;EXIT TO PROGRAM CONTROL
022104  005237  001322          TN6ERR: INC   ERRFLG      ;SET ERROR FLAG
022110  000773                          BR    TN6OUT      ;EXIT
                                        ;
                                        ;
                                        ;
022112  005777  156662          TN6CHK: TST      @DCSR    ;ANY ERRORS?
022116  100405                          BMI   CHKERR      ;YES:EXIT
022120  004537  022164                  JSR   R5,WAIT1    ;WAIT
022124  105777  156662                  TSTB     @DSTAT   ;SEEK DONE?
022130  100402                          BMI   EXIT6       ;YES:GO BACK TO TN6
022132  005237  001322          CHKERR: INC   ERRFLG      ;SET ERROR FLAG
022136  000205                  EXIT6:  RTS   R5          ;RETURN TO TN6
                                        ;
                                        ;
                                        ;
022140  017703  156646          T6DONE: MOV      @DSTAT,R3         ;MOVE DISK REG IMAGE INTO R3
022144  042703  177703                  BIC      #177703,R3        ;EXTRACT SEEK DONE BITS
022150  010002                          MOV      R0,R2             ;SET UP REFERENCE WORD
022152  020203                          CMP      R2,R3             ;PROPER DRIVE DONE?
022154  001402                          BEQ      .+6      ;YES EXIT
022156  005237  001322                  INC   ERRFLG      ;NO:SET ERROR FLAG
022162  000205                          RTS   R5          ;RETURN TO TN6
022164  010146                  WAIT1:  MOV   R1,-(SP)    ;SAVE REGISTER IN STACK
022166  010046                          MOV      R0,-(SP);
022170  013700  001310                  MOV      TIMLP,R0          ;SET OUTER LOOP SIZE
022174  012701  177777          WAIT1A: MOV      #TIME,R1          ;LOAD INNER LOOP COUNT
022200  005301                          DEC   R1          ;DECREMENT LOOP COUNT
022202  001376                          BNE   .-2         ;DONE?,IF NOT LOOP AGAIN
022204  005300                          DEC      R0       ;DECREMENT OUTER LOOP
022206  001372                          BNE      WAIT1A   ;CONTINUE IF NOT DONE
022210  012600                          MOV      (SP)+,R0;
022212  012601                          MOV   (SP)+,R1    ;GET R1 BACK
022214  000205                          RTS   R5          ;RETURN TO PROGRAM
                                        ;
                                        ;
                                        ;
                                        ;                                       9/24/75
                                        ;
                                        ;
                                        ;   INTERRUPT TESTS
                                        ;
                                        ;
                                        ;  THE INTERRUPT TESTS USE TWO FORMATTER
                                        ; COMMANDS,THE RECALIBRATE COMMAND AND THE SEEK
                                        ; ONLY COMMAND.AT THIS STAGE OF TESTING THE SEEK
                                        ; LOGIC SHOULD BE OPERABLE.
                                        ;  THE INTERRUPT LOGIC USES THESE COMMANDS TO
                                        ; CAUSE THE SEEK DONE IDENTIFICATION BITS FLAG AN
                                        ; INTERRUPT.
                                        ;  IF THE CPU HALTS DURING THIS TEST TO SOME
                                        ;  MEMORY LOCATION BETWEEN 14 AND 776 THE
                                        ; THE INTERRUPT BRANCHED TO A WRONG LOCATION.
                                        ;
                                        ;
```

```
                                                ;
     022216  005004                      TN7:   CLR  R4              ;INITIALIZE SUBTEST NUMBER
     022220  004537  004276                     JSR      R5,RQUEST        ;GET 211
     022224  004537  016332                     JSR      R5,DISKID        ;FORMAT CURRENT DISK FOR DUSH
     022230  010200                             MOV      R2,R0            ;SAVE IN R0
     022232  010277  156544                     MOV      R2,@DUSH         ;LOAD CURRENT DISK SELECT BITS
     022236  005737  001012                     TST  DSTAT         ;IS DRIVE READY?
     022242  100072                             BPL   TN7ERR       ;NO:EXIT
     022244  005204                      TN7A:  INC  R4              ;SUBTEST NUMBER=1
     022246  052777  000001  156524             BIS      #SYSCLR,@DCSR    ;ISSUE SYSTEM CLEAR
     022254  004537  004276                     JSR      R5,RQUEST        ;RECUEST 211
     022260  105777  156514                     TSTB     @DCSR      ;FORMATTER READY?
     022264  100375                             BPL   .-4          ;WAIT
     022266  010277  156510                     MOV      R2,@DUSH         ;RESELECT UNIT
     022272  052777  000125  156500             BIS      #INTCAL,@DCSR    ;ISSUE RECAL AND INTERRUPT BIT
     022300  004537  017342                     JSR  R5,WAIT       ;WAIT
     022304  005204                      TN7B:  INC  R4              ;SUBTEST NUMBER=2
     022306  005737  001150                     TST  INTFLG        ;DID IT INTERRUPT?
     022312  001446                             BEQ  TN7ERR        ;NO:SET ERROR FLAG
     022314  005204                             INC  R4              ;SUBTEST NUMBER=3
     022316  017703  156470                     MOV      @DSTAT,R3        ;EXTRACT SEEK DONE BITS
     022322  042703  177774                     BIC  #177774,R3           ;
     022326  013702  001234                     MOV      CURDSK,R2        ;FETCH CURRENT DISK
     022332  020203                             CMP      R2,R3            ;RIGHT DISK DINE?
     022334  001035                             BNE  TN7ERR                ;EXIT IF NO
     022336  005204                      TN7C:  INC  R4              ;SUBTEST NUMBER=4
     022340  005037  001150                     CLR  INTFLG        ;CLEAR INTERRUPT FLAG
     022344  042777  000136  156426             BIC      #FUNC,@DCSR      ;CLEAR FUNCTION BITS
     022352  052777  000100  156420             BIS      #INTON,@DCSR     ;ENABLE INTERRUPTS
     022360  052777  000002  156412             BIS      #SEKCMD,@DCSR    ;ISSUE SEEK
     022366  052777  000001  156404             BIS      #GO,@DCSR        ;SET GO BIT
     022374  004537  017342                     JSR  R5,WAIT       ;WAIT
     022400  005737  001150                     TST  INTFLG        ;TEST INRERRUPT FLAG
     022404  001411                             BEQ  TN7ERR        ;NO INT.,EXIT
     022406  005204                      TN7D:  INC  R4              ;SUBTEST NUMBER=5
     022410  017703  156376                     MOV      @DSTAT,R3;EXTRACT SEEK DONE BITS
     022414  042703  177774                     BIC  #177774,R3 ;
     022420  020203                             CMP      R2,R3    ;RIGHT DISK DONE?
     022422  001002                             BNE  TN7ERR        ;EXIT IF NO
     022424  004537  003342              TN7OUT: JSR     R5,TSTCTL        ;RETURN TO TEST CONTROL
     022430  005237  001322              TN7ERR: INC ERRFLG        ;SET ERROR FLAG
     022434  000773                             BR   TN7OUT         ;EXIT
                                                ;
                                                ;

                                                ;

                                                ;  ENDPAR.PAL
                                                ;                                          9/30/75
                                                ;
                                                ;  END PARAMETER COMPUTE ROUTINE
                                                ;
                                                ;
                                                ;  THIS ROUTINE COMPUTES THE ENDING SECTOR
                                                ; HEAD AND CYLINDER ADDRESS PARAMETERS BASED
                                                ; UPON THE SPECIFIED WORD COUNT.
                                                ;
```

```
                                          ;  AT ENTRY:
                                          ;         (R0-4)=UNIT,SECTOR AND HEAD ADDRESS
                                          ;         (R0-2)=BUSS ADDRESS REGISTER
                                          ;         (R0)  =WORD COUNT REGISTER
                                          ;         (R0+2)=CYLINDER ADDRESS REGISTER
                                          ;
                                          ;  AT EXIT: ENDSEC,ENDHD,AND ENDCYL CONTAIN THE
                                          ; COMPUTED END SECTOR,HEAD AND CYLINDER PARAMETERS
                                          ; RESPECTIVELY
                                          ;
                                          ;
                                          ;
022436  010046                   ENDPAR:  MOV  R0,-(SP)    ;SAVE R0 AND R1
022440  010146                            MOV  R1,-(SP)    ;
022442  024040                            CMP  -(R0),-(R0);POINT R0 TO UNIT,SECTOR,HEAD REGISTER
022444  005037  001146                    CLR  ENDFLG      ;INTIALIZE END FLAG
022450  011001                            MOV  (R0),R1     ;GET UNIT,SECOR,HEAD REGISTER
022452  042701  177600                    BIC  #SECMSK,R1  ;EXTRACT SECTOR FIELD
022456  010137  001162                    MOV  R1,ENDSEC   ;INITIALIZE END SECTOR WORD
022462  011001                            MOV  (R0),R1     ;GET UNIT,SECTOR,HEAD REGISTER AGAIN
022464  042701  170177                    BIC  #HDMSK,R1   ;EXTRACT HEAD BITS
022470  010137  001164                    MOV  R1,ENDHD    ;INITIALIZE END HEAD WORD
022474  016037  000006  001166            MOV  6(R0),ENDCYL;INITIALIZE END CYLINDER WORD
022502  016001  000004                    MOV     4(R0),R1;GET SPECIFIED WORD COUNT
022506  005401                            NEG  R1          ;FORM POSITIVE VALUE
022510  163701  001050           CMPLP:   SUB     POSWP,R1;SUBTRACT NUMBER OF WORDS PER SECTOR
022514  003002                            BGT  .+6         ;CONTINUE IF THE RESULT IS GREATER THAN ZERO
022516  005237  001146                    INC  ENDFLG      ;SET EXIT FLAG IF NEGATIVE
022522  023737  001040  001162            CMP     MAXSEC,ENDSEC    ;LAST SECTOR?
022530  003030                            BGT  SECINC      ;INC SECTOR IF NOT
022532  005037  001162                    CLR  ENDSEC      ;RESET SECTOR COUNTER IF YES
022536  023737  001042  001164            CMP     MAXHD,ENDHD      ;LAST HEAD?
022544  003051                            BGT  INCHD       ;INCREMENT HEAD IF NO
022546  005037  001164                    CLR  ENDHD       ;RESET HEAD COUNTER IF YES
022552  005237  001166                    INC  ENDCYL      ;INCREMENT CYLINDER COUNTER
022556  013700  001044                    MOV     MAXCYL,R0        ;LOAD LAST CYLINDER
022562  005200                            INC     R0               ;MAKE ONE GREATER
022564  023700  001166                    CMP     ENDCYL,R0        ;LEND OF DISK?
022570  001002                            BNE   .+6       ;CONTINUE IF NO
022572  005237  001144                    INC   DKEND      ;SET DISK END FLAG IF YES
022576  005737  001146           ENDTST:  TST  ENDFLG      ;ARE WE DONE?
022602  001742                            BEQ  CMPLP       ;REPEAT IF NO
022604  012601                            MOV  (SP)+,R1    ;RESTORE REGISTERS
022606  012600                            MOV  (SP)+,R0    ;
022610  000205                            RTS  R5          ;EXIT
022612  005237  001162           SECINC:  INC  ENDSEC      ;INCREMENT SECTOR COUNTER
022616  005737  001176                    TST  SPECMD      ;IS THIS A > 256 WORD TRANSFER?
022622  001765                            BEQ  ENDTST      ;CONTINUE NORMAL IF NO
022624  005237  001146                    INC  ENDFLG      ;SET ENDFLG IF YES
022630  017737  156146  001162            MOV     @DUSH,ENDSEC     ;FORCE ENDING PARAMETERS
022636  042737  177600  001162            BIC     #SECMSK,ENDSEC   ;
022644  017737  156132  001164            MOV     @DUSH,ENDHD      ;SECTOR,HEAD AND CYLINDER
022652  042737  170177  001164            BIC     #HDMSK,ENDHD     ;
022660  017737  156124  001166            MOV     @DCYL,ENDCYL     ;
022666  000743                            BR   ENDTST      ;CONTINUE
022670  062737  000200  001164   INCHD:   ADD  #HDINC,ENDHD;INCREMENT HEAD COUNTR FIELD
022676  000737                            BR   ENDTST      ;CONTINUE
```

```
                                        ;
                                        ; DXFER.PAL
                                        ;
                                        ;
                                        ;  DISK DATA TRANSFER SUBROUTINE
                                        ;
                                        ;
                                        ; THIS SUBROUTINE IS USED TO PERFORM ALL DISK
                                        ; DATA TRANSFER OPERATIONS WETHER THEY BE READ
                                        ; OR WRITE IN NATURE.
                                        ;
                                        ;
                                        ; LINKAGE:
                                        ;       PARAMETERS USED BY THE DATA TRANSFER SUB-
                                        ;    ROUTINE ARE CONTAINED IN A CONTROL BLOCK.
                                        ;       THE ORIGIN ADDRESS OF THE CONTROL BLOCK
                                        ;     IS CONTAINED IN R0.
                                        ;
                                        ;
                                        ; CONTROL BLOCK FORMAT:
                                        ;              ENTRY 1 =DESIRED DUSH IMAGE
                                        ;              ENTRY 2 =DESIRED DCAR IMAGE
                                        ;              ENTRY 3 =DESIRED DWDCNT IMAGE
                                        ;              ENTRY 4 =DESIRED DCYL IMAGE
                                        ;              ENTRY 5 =DESIRED DCSR IMAGE
                                        ;
                                        ;
                                        ;  THE DATA TRANSFER ROUTINE CHECKS TO INSURE THAT
                                        ; THE DISK CONTROLLER GOES BUSY ,AN INTERRUPT OCCURS,
                                        ; NO ERROR OCCURS,AND THAT ALL END DISK PARAMETERS
                                        ; ARE CORRECT.
                                        ;
                                        ;
                                        ;  A SEPARATE DATA TRANSFER SUBTEST COUNT IS
                                        ; MAINTAINED IN LOCATION DTCNT TO FACILITATE
                                        ; ERROR ISOLATION.IF AN ERROR (CONTROLLER) OCCURS
                                        ; OR IS DETECTED AT THE END OF THE DISK OPERATION
                                        ; THE IMAGE OF THE FIRST ERROR DETECTED IS CONTAINED
                                        ; IN LOCATION -ERRWRD-.
                                        ;
                                        ;
                                        ;
                                        ;  THE IMAGE OF THE DISK CONTROLLER ERROR REGISTER
                                        ; AT THE TIME OF ANY CONTROLLER ERROR IS CONTAINED
                                        ; IN LOCATION -ERIMGE-.
                                        ;
                                        ;
                                        ;  CONTROL IS TRANSFERED TO THE CALLING ROUTINE
                                        ; WITH OR WITHOUT THE GENERAL ERROR FLAG SET
                                        ; (ERRFLG), AS APPROPRIATE.
                                        ;
                                        ;
 022700  010046                 DXFER:  MOV  R0,-(SP)            ;SAVE REGISTERS
 022702  010146                         MOV  R1,-(SP)   ;
 022704  010446                         MOV  R4,-(SP)            ;
 022706  010546                         MOV     R5,-(SP)         ;
 022710  004537  004276                 JSR     R5,RQUEST        ;GET FORMATTER
```

```
022714  005037  001150                  CLR     INTFLG          ;INITIALIZE INTERRUPT FLAG
022720  005004                          CLR R4                  ;INITIALIZE SUBTEST NUMBER =0
022722  105777  156052                  TSTB    @DCSR           ;IS THE CONTROLLER READY?
022726  100174                          BPL     DTERR           ;NO, EXIT WITH ERROR
022730  005204                          INC R4                  ;SUBTEST NUMBER=1(CONTROLLER READY)
022732  004537  016332                  JSR     R5,DISKID       ;SET UP DRIVE SELECT
022736  010201                          MOV     R2,R1           ;SAVE IN R1
022740  042710  030000                  BIC     #30000,(R0)     ;CLEAR PREVIOUS UNIT SELECT
022744  050110                          BIS R1,(R0)             ;OR IN UNIT SELECT INTO USH REGISTER
022746  013701  001002                  MOV     DUSH,R1         ;GET UNIT,SECTOR,HEAD REGISTER ADDR.
022752  012021                          MOV (R0)+,(R1)+         ;SELECT DISK DRIVE(AND INIT. SEEK)
022754  000240                          NOP
022756  023727  001232  000027          CMP     CURTN,#27       ;DO WE CARE IF DISK READY?
022764  001403                          BEQ     DXF1            ;CONTINUE IF NO
022766  005777  156020                  TST     @DSTAT          ;DRIVE READY?
022772  100152                          BPL     DTERR           ;NO EXIT DRIVE NOT READY
022774  005204                  DXF1:   INC     R4              ;SUBTEST NUMBER=2(DRIVE READY OR DON'T CARE)
022776  012021                          MOV (R0)+,(R1)+         ;LOAD BUSS ADDRESS REGISTER WITH
                                                                ;DATA BUFFER ORIGIN ADDRESS
023000  012021                          MOV (R0)+,(R1)+         ;LOAD WORD COUNT REGISTER
023002  012021                          MOV (R0)+,(R1)+         ;LOAD CYLINDER ADDRESS REGISTER
023004  053710  001102                  BIS     MCPU,(R0)       ;LOAD REQUEST BIT IF APPL.
023010  011077  155764                  MOV     (R0),@DCSR      ;LOAD COMMAND (NO GO BIT)
023014  043710  001102                  BIC     MCPU,(R0)       ;LOOSE REQUEST BIT
023020  105777  155754                  TSTB    @DCSR           ;DID THE CONTROLLER GO BUSY?
023024  001535                          BEQ DTERR               ;EXIT WITH ERROR IF YES
023026  005204                          INC R4                  ;SUBTEST NUMBER=3(NO ERRONEOUS CONT.BUSY)
023030  005737  001062                  TST     STRCNT          ;SEE IF STRIP APPLICABLE
023034  001403                          BEQ     .+6.            ;CONTINUE IF NO
023036  012777  000000G 155770          MOV     #STRBUF,@DSCAR  ;ENTER STRIP BUFFER ADDRESS
023044  052777  000101  155726          BIS     #INTGO,@DCSR    ;ENABLE INTERRUPTS AND GO!!!!!!!!!!!!!!!!
023052  000240                          NOP
023054  000240                          NOP
023056  000240                          NOP
023060  105777  155714                  TSTB    @DCSR           ;CONTROLLER BUSY?
023064  100515                          BMI DTERR               ;EXIT WITH ERROR IF NO
023066  005204                          INC R4                  ;SUBTEST NUMBER=4(CONTROLLER BUSY)
023070  004537  017342                  JSR R5,WAIT             ;WAIT FOR INTERRUPT OR TIME OUT
023074  005737  001150                  TST INTFLG              ;DID AN INTERRUPT OCCUR?
023100  001507                          BEQ DTERR               ;EXIT WITH ERROR IF NO
023102  005204                          INC R4                  ;SUBTEST NUMBER=5
023104  005037  001150                  CLR INTFLG              ;CLEAR INTERRUPT FLAG IF IT WAS SET
023110  105777  155664                  TSTB    @DCSR           ;IS THE CONTROLLER STILL BUSY?
023114  100101                          BPL DTERR               ;EXIT WITH ERROR IF YES
023116  005204                          INC     R4              ;SUBTEST NUMBER =6
023120  005002                          CLR R2                  ;EST. REFERENCE FOR WORD COUNT
023122  017703  155660                  MOV     @DWCNT,R3       ;GET ACTUAL WORD COUNT REGISTER
023126  020203                          CMP R2,R3               ;DID WORD COUNT OVERFLOW GO TO ZERO?
023130  001073                          BNE DTERR               ;EXIT WITH ERROR IF NO
023132  005204                          INC     R4              ;SUBTEST NUMBER =7
023134  024040                          CMP -(R0),-(R0)         ;POINT R5 TO WORD COUNT VALUE
023136  011002                          MOV @R0,R2              ;GET ORIGINAL WORD COUNT
023140  005402                          NEG R2                  ;FORM POSITIVE VALUE
023142  006302                          ASL R2                  ;MAKE VALID WORD COUNT
023144  064002                          ADD     -(R0),R2        ;COMPUTE REFERENCE ENDING BUS ADDRESS
023146  005720                          TST     (R0)+           ;RESTORE R0 FOR ENDPAR ROUTINE
023150  017703  155630                  MOV     @DCAR,R3        ;GET ACTUAL END BUSS ADDR. VALUE
```

```
     023154  020203                  CMP   R2,R3            ;IS END BUS ADDRESS CORRECT?
     023156  001060                  BNE   DTERR            ;EXIT WITH ERROR IF NO
     023160  005204                  INC     R4             ;SUBTEST NUMBER=10
     023162  004537  022436          JSR   R5,ENDPAR        ;COMPUTE END DISK PARAMETERS
     023166  013702  001162          MOV   ENDSEC,R2        ;GET COMPUTED END SECTOR VALUE
     023172  017703  155604          MOV     @DUSH,R3       ;GET UNIT,SECTOR HEAD REG.
     023176  010301                  MOV   R3,R1            ;SAVE IN R1 ALSO
     023200  042703  177600          BIC   #SECMSK,R3       ;EXTRACT ACTAUL SECTOR VALUE
     023204  020203                  CMP   R2,R3            ;IS THE SECTOR VALUE CORRECT?
     023206  001044                  BNE   DTERR            ;EXIT WITH ERROR IF NO
     023210  005204                  INC     R4             ;SUBTEST NUMBER=11
     023212  013702  001164          MOV   ENDHD,R2         ;GET COMPUTED END HEAD VALUE
     023216  010103                  MOV   R1,R3            ;GET UNIT,SEC,HEAD VALUE AGAIN
     023220  042703  170177          BIC   #HDMSK,R3        ;EXTRACT HEAD FIELD BITS
     023224  020203                  CMP   R2,R3            ;IS THE ACTUAL HEAD VALUE CORRECT?
     023226  001034                  BNE   DTERR            ;EXIT WITH ERROR IF NO
     023230  005204                  INC     R4             ;SUBTEST NUMBER=12
     023232  013702  001166          MOV   ENDCYL,R2        ;GET COMPUTED END CYLINDER ADDRESS
     023236  017703  155546          MOV     @DCYL,R3       ;GET ACTUAL CYLINDER ADDRESS
     023242  020203                  CMP   R2,R3            ;IS THE CYLINDER ADDRESS CORRECT?
     023244  001025                  BNE   DTERR            ;EXIT WITH ERROR IF NO
     023246  005204                  INC     R4             ;SUBTEST NUMBER=13
     023250  017703  155524          MOV     @DCSR,R3       ;TAKE A LOOK AT MEMORY EXTENSION BITS
     023254  042703  147777          BIC     #147777,R3     ;EXTRACT MEMORY EXTENSION BITS
     023260  005002                  CLR     R2             ;SET UP REFERENCE WORD
     023262  020203                  CMP     R2,R3          ;ANY BITS SET?
     023264  001015                  BNE     DTERR          ;EXIT IF YES
     023266  005204                  INC     R4             ;SUBTEST NUMBER=14(NO MEMORY EXT. BITS SET)
     023270  005777  155504          TST     @DCSR          ;ANY CONTROLLER ERRORS STILL?
     023274  100477                  BMI   CTLERR           ;FIND OUT WHICH ONE IF YES
     023276  005037  001132          CLR     SOFTRY         ;CLEAR  REREAD FLAG IF SET
     023302  005037  001326          CLR     SOFER1         ;CLEAR RETRY ATTEMPT COUNTER
     023306  012605          DTEXIT: MOV   (SP)+,R5         ;RESTORE REGISTERS
     023310  012604                  MOV   (SP)+,R4         ;
     023312  012601                  MOV   (SP)+,R1         ;
     023314  012600                  MOV   (SP)+,R0         ;
     023316  000205                  RTS     R5             ;EXIT TO CALLING PROGRAM
     023320  005777  155454  DTERR:  TST     @DCSR          ;CONTROLLER READY?
     023324  100006                  BPL     DTERRA         ;EXIT NORMALLY IF NO
     023326  017705  155462          MOV     @DERR,R5       ;GET ERROR REGISTER IMAGE
     023332  042705  175000          BIC     #175000,R5     ;EXTRACT POSSIBLE SOFT ERROR BITS
     023336  005705                  TST     R5             ;ARE THERE ANY ON
     023340  001023                  BNE     DSERR          ;RETRY TRANSFER IF YES
     023342  005237  001322  DTERRA: INC     ERRFLG         ;SET GENERAL ERROR FLAG
     023346  013700  001000          MOV     DCSR,R0        ;GET ADDR OF CONTROL AND STATUS REG
     023352  012701  035334          MOV     #ERRBLK,R1     ;GET ERROR BLOCK ADDRESS
     023356  012705  000007          MOV     #7,R5          ;ESTABLISH LOOP COUNT
     023362  012021                  MOV     (R0)+,(R1)+    ;SAVE DISK CONT REGISTER IMAGES
     023364  005305                  DEC     R5             ;DECREMENT LOOP COUNT
     023366  001375                  BNE     .-4            ;REPEAT IF NOT DONE
     023370  005737  001074          TST     ECC            ;IS THIS ECC?
     023374  001402                  BEQ     DERECC         ;GO NO IF NO
                                     ;
     023376  012021                  MOV     (R0)+,(R1)+    ;LOAD ECCPB IF YES
     023400  012021                  MOV     (R0)+,(R1)+    ;AND ECCPW ALSO
                                     ;
     023402                  DERECC:
```

```
1084 023402  010437  001152                       MOV  R4,DTCNT            ;SAVE SUBTEST COUNT
1085 023406  000737                               BR   DTEXIT              ;EXIT
1086 023410  005737  001244             DSERR:    TST      NONCON          ;CONSOLE ACTIVE?
1087 023414  001352                               BNE      DTERRA          ;EXIT IF NO
1088 023416  032737  040000  177570               BIT      #40000,CSWR     ;RETRY?
1089 023424  001746                               BEQ      DTERRA          ;EXIT WITH ERROR IF NO
1090 023426  023727  001326  000004               CMP      SOFER1,#4       ;HAVE WE TRIED 4 TIMES YET?
1091 023434  001412                               BEQ      DSERRA          ;EXIT IF YES
1092 023436  005237  001326                       INC      SOFER1   ;INCREMENT SOFT RETRY COUNTER
1093 023442  005237  001330                       INC      SOFERR   ;INCREMENT SOFT ERROR COUNTER
1094 023446  012605                               MOV      (SP)+,R5        ;
1095 023450  012604                               MOV      (SP)+,R4        ;
1096 023452  012601                               MOV      (SP)+,R1        ;
1097 023454  012600                               MOV      (SP)+,R0        ;
1098 023456  000137  022700                       JMP      DXFER           ;GO RETRY TRANSFER
1099 023462  005037  001326             DSERRA:   CLR      SOFER1          ;CLEAR RETRY COUNTER
1100 023466  005237  001132                       INC      SOFTRY          ;SET SOFT RETRY STOP FLAG
1101 023472  000723                               BR       DTERRA          ;AND EXIT
1102 023474  004537  023502             CTLERR:   JSR      R5,ERRID        ;SAVE CONTROLLER REGISTERS
1103 023500  000707                               BR       DTERR           ;AND GO INTO ERROR ROUTINE
1104                                              ;
1105                                              ;
1106                                              ;
1107                                              ;   ERROR IDENTIFICATION ROUTINE
1108                                              ;
1109                                              ;  THIS SUBROUTINE EXAMINES THE DISK CONTROLLER
1110                                              ; ERROR STATUS REGISTER STARTING WITH THE
1111                                              ; FIRST BIT(0),AND STORES THE MASK OF THE FIRST
1112                                              ; ERROR BIT ENCOUNTERED THAT IS ACTIVE IN LOCATION
1113                                              ; ERRWRD.THE COMPLETE DISK ERROR REGISTER STATE
1114                                              ; AT THE TIME OF THE TEST IS STORED IN LOCATION
1115                                              ; ERIMGE.
1116                                              ;
1117                                              ;
1118                                              ;
1119 023502  010046                     ERRID:    MOV  R0,-(SP)            ;SAVE REGISTER R0,R1,R2,R3
1120 023504  010146                               MOV  R1,-(SP)            ;
1121 023506  010246                               MOV  R2,-(SP)            ;
1122 023510  010346                               MOV  R3,-(SP)            ;
1123 023512  017700  155276                       MOV      @DERR,R0        ;GET DISK ERROR REG.
1124 023516  010046                               MOV  R0,-(SP)            ;SAVE DERR ALSO ON THE STACK
1125 023520  001416                               BEQ  EREXIT+2            ;EXIT IF TIME OUT ERROR
1126 023522  012703  000020                       MOV  #16.,R3             ;ESTABLISH LOOP COUNT OF 16
1127 023526  012701  002612                       MOV  #S1ORG,R1           ;POINT R1 TO SLIDING 1 TBLE
1128 023532  012702  002652                       MOV  #S0ORG,R2           ;POINT R2  TO SLIDING 0 TBLE
1129 023536  011600                     IDLP:     MOV      (SP),R0         ;GET ERROR REGISTER IMAGE
1130 023540  042200                               BIC  (R2)+,R0            ;EXTRACT ERROR STATUS BIT
1131 023542  005700                               TST  R0                  ;IS ERROR BIT SET?
1132 023544  001011                               BNE  IDERR               ;FLAG IT IF YES
1133 023546  022101                               CMP  (R1)+,R1            ;POINT R1 TO NEXT SLIDING 1 CONSTANT
1134 023550  005303                               DEC  R3                  ;DECREMENT LOOP CONSTANT
1135 023552  001371                               BNE  IDLP                ;REPEAT IF NOT DONE
1136 023554  022616                     EREXIT:   CMP  (SP)+,(SP)          ;POP STACK
1137 023556  012603                               MOV  (SP)+,R3            ;RESORE REGISTERS
1138 023560  012602                               MOV  (SP)+,R2            ;
1139 023562  012601                               MOV  (SP)+,R1            ;
1140 023564  012600                               MOV  (SP)+,R0            ;
```

```
023566  000205                   RTS     R5                      ;RETURN
023570  011137  001160   IDERR:  MOV     (R1),ERRWRD             ;PUT IDENTIFYING WORD IN ERRWRD
023574  011637  001170           MOV     (SP),ERIMGE             ;STORE WHOLE ERROR REGISTER
023600  000765                   BR      EREXIT                  ;EXIT TO REG EXIT ROUTINE
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;   WRITE CONTROL BLOCK
                                 ;
                                 ;
                                 ;  THIS TABLE CONTAINS PARAMETERS USED TO DEFINE ALL
                                 ; WRITE DISK OPERATIONS AND IS REFERENCED  BY THE
                                 ; DATA TRANSFER SUBROUTINE,(DXFER).
                                 ;
                                 ;
                                 ;
                                 ;
023602  000000           WRTBLK: .WORD 0                         ;UNIT 0,SECTOR 0,HEAD 0 (DUSH)
023604  044312                   .WORD ODBUF                     ;OUTPUT DATA BUFFER (CAR)
023606  177777                   .WORD -1                        ;ONE WORD TRANSFER (WDCNT)
023610  000000                   .WORD 0                         ;CYLINDER 0 (DCYL)
023612  000006                   .WORD WRTCMD                    ;WRITE COMMAND (NO GO)
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;       READ CONTROL BLOCK
                                 ;
                                 ;
                                 ;
                                 ;  THIS TABLE CONTAINS PARAMETERS USED TO DEFINE ALL
                                 ; READ DISK OPERATIONS AND IS REFERENCED BY THE
                                 ; DATA TRANSFER SUBROUTINE (DXFER)
                                 ;
                                 ;
                                 ;
023614  000000           RDBLK:  .WORD 0                         ;UNIT 0,HEAD 0,SECTOR 0,(DUSH)
023616  047314                   .WORD IDBUF1                    ;INPUT DATA BUFFER 1 ORIGIN
023620  177777                   .WORD -1                        ; 1 WORD TRANSFER
023622  000000                   .WORD 0                         ;CYLINDER 0  (DCYL)
023624  000004                   .WORD RDCMD                     ;READ COMMAND (NO GO) , (DCSR)
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
```

```
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;       5M.ECC  THEST NUMBERS 10 >> 16  13-NOV-77
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;       ONE WORD WRITE TEST
                                               ;     ---------------------
                                               ;
                                               ;
                                               ;       THIS TEST IS FIRST WRITE ATTEMTED BY THE FORMATTER
                                               ;       IT ONLY A ONE WORD TRANSFER AND ONLY THE ENDING
                                               ;       FORMATTER REGISTER PARAMETERS ARE CHECKED
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
 023626                             TN10:      ;CLR    R4              ;SUBTEST NUMBER=0
 023626  004537  004276                         JSR    R5,RQUEST       ;GET 211
 023632  012701  044312                         MOV    #ODBUF,R1       ;GET OUT PUT DATA BUFFER ADDRESS
 023636  004537  016736                         JSR    R5,FORMAT       ;FORMAT DATA
 023642  012700  023610                         MOV    #WRTBLK+6,R0    ;GET CYLINDER ADDRESS ORIGIN
 023646  005010                                 CLR    (R0)            ;MAKE CYLINDER=0
 023650  012740  177777                         MOV    #-1,-(R0)       ;MAKE WORD COUNT=1
 023654  010140                                 MOV    R1,-(R0)        ;MAKE ODBUF CORE ADDRESS START
 023656  005040                                 CLR    -(R0)           ;MAKE DUSH=0
 023660  004537  022700                         JSR    R5,DXFER        ;GO WRITE ONE WORD!!!!!!!
 023664  004537  003342                         JSR    R5,TSTCTL       ;EXIT TO TEST CONTROL
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
```

```
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;       ONE SECTOR WRITE TEST
                                                ;
                                                ;       THIS TEST WRITES DATA ON ONE SECTOR OF THE DISK.
                                                ;THE CYLINDER ADDRESS IS NORMALLY DETERMINED BY RWCYL,WHILE
                                                ;THE SECTOR,HEAD,AND UNIT VALUE IS DETERMINED BY RWUSH.
                                                ;       OPTIONALLY THE SECTOR AND HEAD IS DYNAMICALLY SELECTABLE
                                                ;BY THE CONSOLE SWITCH REGISTER IF CSWTCH IS LOADED WITH
                                                ;ANY NON ZERO VALUE.
                                                ;
                                                ;
 023670  012701  044312          TN11:  MOV     #ODBUF,R1       ;GET OUT PUT BUFFER ADDR
 023674  004537  004276                 JSR     R5,RQUEST       ;REQUEST 211
 023700  004537  016736                 JSR     R5,FORMAT       ;FORMAT DATA BUFFER
 023704  012700  023610                 MOV     #WRTBLK+6,R0    ;GET CYL ENTRY CB ADDRESS
 023710  013710  001056                 MOV     RWCYL,(R0)      ;SPECIFY CYLINDER (NORM =MAXCYL)
 023714  013740  001046                 MOV     WPSEC,-(R0)     ;SPECIFY ONE SECTOR WORD COUNT
 023720  010140                         MOV     R1,-(R0)        ;DCAR =ODBUF ORIGIN
 023722  005737  001250                 TST     CSWTCH          ;DOES DUSH COME FROM CSWR?
 023726  001406                         BEQ     .+14.           ;BRANCH IF NO
 023730  013702  177570                 MOV     CSWR,R2         ;GET DUSH VALUE
 023734  042702  100000                 BIC     #REPBIT,R2      ;REMOVE REPEAT BIT IF ON
 023740  010240                         MOV     R2,-(R0)        ;LOAD IN DUSH VALUE
 023742  000402                         BR      .+6             ;CONTINUE
 023744  013740  001054                 MOV     RDUSH,-(R0)     ;SPECIFY CONSTANT DUSH  (MAX VALUE)
 023750  004537  022700                 JSR     R5,DXFER        ;PERFORM ONE SECTOR WRITE
 023754  004537  003342                 JSR     R5,TSTCTL       ;GO TO NEXT TESTIF NO ERROR
                                                ;
                                                ;
                                                ;
                                                ;       ONE SECTOR READ TEST
                                                ;
                                                ;
                                                ;       THIS TEST READS DATA FROM ONE SECTOR OF THE DISK.
                                                ;THE CYLINDER ADDRESS IS NORMALLY DETERMINED BY RWCYL,WHILE
                                                ;THE SECTOR AND HEAD ADDRESS IS NORMALLY DETERMINED BYRWUSH.
                                                ;OPTIONALLY THE SECTOR AND HEAD ADDRESS MAY BE DYNAMICALLY
                                                ;SPECIFIED BY THE CONSOLE SWITCH REGISTER IF THE CSWTCH
                                                ;LOCATION IS NON ZERO.
                                                ;
                                                ;
 023760  012700  023622          TN12:  MOV     #RDBLK+6,R0     ;GET CYL ENTRY ADDR
 023764  004537  004276                 JSR     R5,RQUEST       ;REQUEST 211
 023770  013710  001056                 MOV     RWCYL,(R0)      ;SPECIFY CYLINDER ( NORM =MAXCYL)
 023774  013740  001046                 MOV     WPSEC,-(R0)     ;SPECIFY ONE SECTOR READ
 024000  012740  047314                 MOV     #IDBUF1,-(R0)   ;PUT DATA IN IDBUF1
 024004  005737  001250                 TST     CSWTCH          ;DOES DUSH COME FROM CSWR?
 024010  001406                         BEQ     .+14.           ;USE RDUSH VALUE IF NO
 024012  013702  177570                 MOV     CSWR,R2         ;GET SWITCH REGISTER
 024016  042702  100000                 BIC     #REPBIT,R2      ;CLEAR REPEAT BIT IF ON
 024022  010240                         MOV     R2,-(R0)        ;LOAD IN DUSH VALUE
 024024  000402                         BR      .+6             ;CONTINUE
 024026  013740  001054                 MOV     RDUSH,-(R0)     ;LOAD DUSH  (MAXIMUM VALUE)
 024032  004537  022700                 JSR     R5,DXFER        ;PERFORM READ OPERATION
```

```
024036  004537  003342            JSR     R5,TSTCTL       ;GO TO NEXT TEST IF NO ERROR
                                  ;
                                  ;
                                  .EOT
                                  ;
                                  ;
                                  ;       WRITE COUNT TEST
                                  ;
                                  ;       THIS TEST PERFORMS WRITE OPERATIONS WITH AN EVER
                                  ;INCREASING WORD COUNT STARTING WITH 1 AND INCREASING IN
                                  ;POWERS OF 2 TO A MAXIMUM OF 8192. WORDS
                                  ;       EACH WRITE OPERATION STARTS A ABSOLUTE 0 DISK
                                  ;ADDRESS.
                                  ;
                                  ;
                                  ;
024042  005004            TN13:   CLR     R4              ;SUBTEST NUMBER=0
024044  004537  004276            JSR     R5,RQUEST       ;REQUEST 211
024050  005001                    CLR     R1              ;CLEAR INDEX REGISTER
024052  012705  000016            MOV     #14.,R5         ;ESTABLISH MASTER LOOP COUNT
024056  012700  023610    TN13LP: MOV     #WRTBLK+6,R0    ;GET CYL CB ENTRY
024062  005010                    CLR     (R0)            ;CYLINDER=0
024064  016102  002612            MOV     S1ORG(R1),R2    ;GET WORD COUNT (=2 NEXP)
024070  005402                    NEG     R2              ;FORM NEGATIVE VALUE
024072  010240                    MOV     R2,-(R0)        ;SPECIFY WORD COUNT
024074  012740  000000            MOV     #0,-(R0)        ;BAR=ZERO ADDR. CORE
024100  005040                    CLR     -(R0)           ;DUSH =0
024102  004537  022700            JSR     R5,DXFER        ;PERFORM SPECIFIED WRITE
024106  005737  001322            TST     ERRFLG          ;ANY ERRORS?
024112  001004                    BNE     TN13EX          ;EXIT IF YES
024114  005204                    INC     R4              ;INCREMENT SUBTEST NO.
024116  005721                    TST     (R1)+           ;INCREMENT WORD COUNT INDEX
024120  005305                    DEC     R5              ;DECREMENT LOOP COUNT
024122  001355                    BNE     TN13LP          ;REPEAT IF NOT DONE
024124  004537  003342    TN13EX: JSR     R5,TSTCTL       ;EXIT
                                  ;
                                  ;
                                  ;
                                  ;       ONE SECTOR READ,WRITE,COMPARE TEST
                                  ;
                                  ;
                                  ;       THIS TEST WRIITES DATA ON ONE SECTOR OF THE DISK,
                                  ;READS IT BACK, AND COMPARES THE DATA OBTAINED WITH
                                  ;THE ORIGINAL DATA WRITTEN.
                                  ;       THE CYLINDER ADDRESS IS SPECIFIED BY RWCYL WHILE
                                  ;THE SECTOR AND HEAD VALUES ARE NORMALLY SUPPLIED BY
                                  ;RWUSH.
                                  ;       OPTIONALLY THE SECTOR AND HEAD DATA MAY BE
                                  ;DYNAMICALLY SUPPLIED FROM THE CONSOLE SWITCH REGISTER
                                  ;IF CSWTCH CONTAINS ANY NON ZERO VALUE.
                                  ;
                                  ;
                                  ;
024130  012700  035526    TN14:   MOV     #WRCB+6,R0      ;GET CYLINDER ENTRY ADDR
024134  004537  004276            JSR     R5,RQUEST       ;REQUEST 211
024140  013710  001056            MOV     RWCYL,(R0)      ;SPECIFY CYLINDER ADDRESS
024144  013740  001046            MOV     WPSEC,-(R0)     ;SPECIFY ONE SECTOR OPERATION
```

```
1369 024150  005740                  TST     -(R0)             ;SKIP OVER BAR CB ENTRY
1370 024152  005737  001250          TST     CSWTCH            ;DOES USH DATA COME FROM CSWR?
1371 024156  001406                  BEQ     .+14.             ;USE RDUSH DATA IF NO
1372 024160  013702  177570          MOV     CSWR,R2           ;GET DESIRED DATA
1373 024164  042702  100000          BIC     #REPBIT,R2        ;REMOVE REPEAT BIT IF ON
1374 024170  010240                  MOV     R2,-(R0)          ;ENTER DUSH VALUE
1375 024172  000402                  BR      .+6               ;CONTINUE
1376 024174  013740  001054          MOV     RDUSH,-(R0)       ;PUT MAXIMUM DUSH VALUE IN
1377 024200  012701  044312          MOV     #ODBUF,R1         ;GET OUTPUT DATA BUFFER ADDRESS
1378 024204  004537  016736          JSR     R5,FORMAT         ;FORMAT BUFFER
1379 024210  004537  035356          JSR     R5,WRC            ;PERFORM WRITE,READ AND DATA COMPARISON
1380 024214  004537  003342          JSR     R5,TSTCTL         ;GO TO NEXT TEST IF NO ERROR
1381                                 ;
1382                                 ;
1383                                 ;
1384                                 ;       READ, WRITE, COMPARE COUNT TEST
1385                                 ;
1386                                 ;
1387                                 ;       THIS TEST PERFORMS WRITE, READ AND DATA COMPARISONS
1388                                 ;WITH AN EVER INCREASING WORD COUNT STARTING WITH ONE AND
1389                                 ;INCREASING BY POWERS OF TWO TO A MAXIMUM OF 512 WORDS.
1390                                 ;       EACH DISK OPERATION STARTS AT ABSOLUTE DISK ADDRESS

1391                                 ;ZERO.
1392                                 ;ALL DISK AND DATA PARAMETERS ARE CHECKED DURING THE TEST.
1393                                 ;
1394                                 ;
1395                                 ;
1396 024220  012701  044312  TN15:   MOV     #ODBUF,R1         ;GET OUTPUT BUFFER ADDR
1397 024224  004537  004276          JSR     R5,RQUEST         ;REQUEST 211
1398 024230  004537  016736          JSR     R5,FORMAT         ;FORMAT BUFFER
1399 024234  012705  000011          MOV     #11,R5            ;ESTABLISH MASTER LOOP
1400 024240  005001                  CLR     R1                ;CLEAR INDEX REGISTER
1401 024242  005004                  CLR     R4                ;SUBTEST NUMBER=0
1402 024244  012700  035526  TN15LP: MOV     #WRCB+6,R0        ;GET WRITE/READ CYL ENTRY ADDR
1403 024250  005010                  CLR     (R0)              ;CYLINDER = 0
1404 024252  016102  002612          MOV     S1ORG(R1),R2      ;GET WORD COUNT =2N (N=EXP)
1405 024256  005402                  NEG     R2                ;MAKE NEGATIVE
1406 024260  010240                  MOV     R2,-(R0)          ;PUT IN WORD COUNT ENTRY
1407 024262  005740                  TST     -(R0)             ;SKIP OVER BAR CB ENTRY
1408 024264  005040                  CLR     -(R0)             ;DUSH = 0
1409 024266  004537  035356          JSR     R5,WRC            ;DO WRITE,READ,DATA COMPARISON
1410 024272  005737  001322          TST     ERRFLG            ;ANY ERRORS?
1411 024276  001004                  BNE     T15EX             ;EXIT IF YES
1412 024300  005204                  INC     R4                ;INCREMENT SUBTEST COUNT
1413 024302  005721                  TST     (R1)+             ;INCREMENT WORD COUNT INDEX
1414 024304  005305                  DEC     R5                ;DECREMENT LOOP COUNT
1415 024306  001356                  BNE     TN15LP            ;REPEST IF NOT DONE
1416 024310  004537  003342  T15EX:  JSR     R5,TSTCTL         ;EXIT
1417                                 ;
1418                                 ;
1419                                 ;
1420                                 .EOT
1421                                 ;
1422                                 ;       WRITE PROTECT TEST
1423                                 ;
1424                                 ;
1425                                 ;   THIS TEST VERIFIES THAT THE CONTROLLER WILL NOT
```

```
                                                        ; WRITE OVER WRITE PROTECTED SECTORS.
                                                        ;
                                                        ;  FIRST THE TEST SETS THE WRITE PROTECT BITS OF THE
                                                        ; ABSOLUTE SECTOR 0 AND THEN ATTEMPT TO WRITE
                                                        ; ON THAT SECTOR.THE TEST THEN VERIFIES THAT THE
                                                        ; CONTROLLER ABORTED THE WRITE OPERATION AND THAT
                                                        ; THE WRITE PROTECT ERROR BIT IS ACTIVATED.
                                                        ;
                                                        ;
                                                        ;
024314  005004                          TN16:   CLR     R4                      ;INIT SUBTEST NUMBER
024316  004537  004276                          JSR     R5,RQUEST               ;REQUEST 211
024322  005237  001176                          INC     SPECMD                  ;FLAG SPECIAL TRANSFER(GREATER THAN 256 WORDS)
024326  012701  044312                          MOV     #ODBUF,R1               ;GET ODBUF ADDRESS
024332  063701  001062                          ADD     STRCNT,R1               ;ADJUST DATA BUFFER FOR STRIP OPTION
024336  004537  016736                          JSR     R5,FORMAT               ;GO FORMAT BUFFER
024342  013737  001304  044310                  MOV     WPRBIT,ODBUF-2          ;ADD WRITE PROTECT BITS
024350  005037  044306                          CLR     ODBUF-4                 ;MAKE SECTOR-HEAD HDR VALID
024354  012700  025450                          MOV     #ISECFB,R0              ;ENTER FORMAT BLOCK ADDR
024360  013760  001046  000004                  MOV     WPSEC,4(R0)             ;ADD 2 WORDS FOR HEADERS
024366  162760  000002  000004                  SUB     #2,4(R0)                ;
024374  004537  022700                          JSR     R5,DXFER                ;GO REFORMAT SECTOR
024400  005037  001176                          CLR     SPECMD                  ;CLEAR SPECIAL COMMAND FLAG
024404  005737  001322                          TST     ERRFLG                  ;ERRORS?
024410  001116                                  BNE     TN16EX                  ;EXIT IF YES
024412  005204                                  INC     R4                      ;SUBTEST NUMBER=1(WRITE PROTECT OK)
024414  012700  044312                          MOV     #ODBUF,R0               ;GET BUFFER ORIGIN
024420  013701  001050                          MOV     POSWP,R1                ;ESTABLISH WORD COUNT
024424  005420                                  NEG     (R0)+                   ;COMPLEMENT REFERENCE DATA
024426  005301                                  DEC     R1                      ;DECREMENT LOOP
024430  001375                                  BNE     .-4                     ;REPEAT IF NOT DONE
024432  012700  024652                          MOV     #T16WB,R0               ;GET WRITE BLOCK ADDR
024436  013760  001046  000004                  MOV     WPSEC,4(R0)             ;ENTER WORD COUNT
024444  004537  022700                          JSR     R5,DXFER                ;
024450  005737  001322                          TST     ERRFLG                  ;ERROR ON WRITE ATTEMPT?
024454  001003                                  BNE     .+8.                    ;EXIT IF NO
024456  005237  001322                          INC     ERRFLG                  ;SET ERROR FLAG
024462  000471                                  BR      TN16EX                  ;EXIT ON ERROR
024464  005204                                  INC     R4                      ;SUBTEST NUMBER=2(ERROR DETECTED)
024466  017703  154322                          MOV     @DERR,R3                ;GET ERROR REGISTER
024472  042703  077773                          BIC     #77773,R3               ;EXTRACT ERROR
024476  012702  000004                          MOV     #4,R2                   ;SET UP REFERENCE ERROR WORD
024502  020203                                  CMP     R2,R3                   ;ARE THEY EQUAL?
024504  001060                                  BNE     TN16EX                  ;EXIT IF NO
024506  005037  001322                          CLR     ERRFLG                  ;CLEAR EOROORROR FLAG
024512  005204                                  INC     R4                      ;SUBTEST NUMBER=3(WRITE PROTECT)
024514  012701  044312                          MOV     #ODBUF,R1               ;GET BUFFER ORIGIN
024520  004537  016736                          JSR     R5,FORMAT               ;GO FORMAT DATA
024524  012700  025470                          MOV     #ISECRB+6,R0            ;GET ONE SECTOR READ BLOCK ADDR
024530  005010                                  CLR     (R0)                    ;MAKE=0
024532  013740  001046                          MOV     WPSEC,-(R0)             ;ENTER WORD COUNT
024536  013701  001234                          MOV     CURDSK,R1               ;GET CURRENT DISK NUMBER
024542  006301                                  ASL     R1                      ;MAKE VALID INDEX
024544  016140  001200                          MOV     IBFTBL(R1),-(R0);
024550  005040                                  CLR     -(R0)                   ;SECTOR,HEAD=0
024552  004537  022700                          JSR     R5,DXFER                ;READ SECTOR WITH WRITE PROTECT
024556  005737  001322                          TST     ERRFLG                  ;ANY ERRORS?
```

```
024562  001031                    BNE     TN16EX          ;EXIT IF YES
024564  005204                    INC     R4              ;SUBTEST NUMBER=4(DATA READ OK)
024566  013700  001050            MOV     POSWP,R0        ;SPECIFY 256. WORD COMPARE
024572  004537  017106            JSR     R5,DATCMP       ;SEE IF DATA WAS ERRONEOUSLY OVERWRITTEN
024576  005737  001322            TST     ERRFLG          ;ANY ERRORS?
024602  001021                    BNE     TN16EX          ;EXIT IF YES
024604  005204                    INC     R4              ;SUBTEST NUMBER=5(DATA OK)
024606  005037  044306            CLR     ODBUF-4         ;RESTORE NRMAL HEADER FORMAT
024612  005037  044310            CLR     ODBUF-2         ;FOR SECTOR 0
024616  012700  025450            MOV     #ISECFB,R0      ;GET FORMAT BLOCK ADDR
024622  005237  001176            INC     SPECMD          ;SET SPECIAL TRANSFER COMMAND
024626  004537  022700            JSR     R5,DXFER        ;RESTORE ORIGINAL FORMAT TO SECTOR
024632  005037  001176            CLR     SPECMD          ;CLEAR SPECIAL TRANSFER FLAG
024636  005737  001322            TST     ERRFLG          ;ANY ERRORS?
024642  001001                    BNE     TN16EX          ;EXIT IF YES
024644  005204                    INC     R4              ;SUBTEST NUMBER=6(REFORMAT OK)
024646  004537  003342    TN16EX: JSR     R5,TSTCTL       ;RETURN
                                  ;
                                  ;
                                  ;
                                  ;       TEST 16         TRANSFER BLOCK
                                  ;
                                  ;
                                  ;
024652  000000            T16WB:  .WORD 0         ;DUSH =0
024654  044312                    .WORD ODBUF     ;DBAR=ODBUF
024656  177700                    .WORD 177700    ;WORD COUNT LESS THAN ONE SECTOR
024660  000000                    .WORD 0         ;CYLINDER 0
024662  000006                    .WORD WRTCMD    ;WRITE OPERATION
        024652                    ISECWB=T16WB
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;       6M.ECC  TEST NUMBERS 17>> 21
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;       HEADER OVERRIDE TEST
                                  ;     **************************
                                  ;
                                  ;
                                  ;
                                  ;
```

```
                                        ;
                                        ;
                                        ;
                                        ;  THIS TEST VERIFIES THAT THE READ WITHOUT
                                        ; HEADER CHECK LOGIC IS FUNCTIONING PROPERLY.
                                        ;  THE HEADER IS FIRST ERRONEOUSLY WRITTEN AND
                                        ; THEN A  NORMAL READ OPERATION IS ATTEMPTED
                                        ; TO INSURE THAT WE CANNOT READ THE SECTOR.
                                        ;  THE READ WITHOUT HEADER CHECK COMMAND IS USED
                                        ; TO THEN READ THE SECTOR.CHECKS ARE MADE TO INSURE
                                        ; THAT NO ERROR OCCURS AND THAT THE DATA IS CORRECT.
                                        ;
                                        ;  THE SECOND SECTION OF THIS PROGRAM CHECKS
                                        ; THE ABILITY OF THE FORMATTER TO READ THE HEADER FIELD
                                        ; HEADER CRC DATA FIELD AND DATA CRC.
                                        ;
                                        ;
                                        ;
024664  005004                  TN17:   CLR     R4              ;INITIALIZE SUBTEST COUNT
024666  004537  004276                  JSR     R5,RQUEST       ;REQUEST 211
024672  012701  044306                  MOV     #ODBUF-4,R1     ;GET OUTPUT BUFFER ADDRESS
024676  012721  177774                  MOV     #177774,(R1)+   ;SET FIRST HEADER TO BE ERRONEOUS
024702  005021                          CLR     (R1)+           ;MAKE CYLINDER HEADER WORD VALID
024704  063701  001062                  ADD     STRCNT,R1       ;ADJUST BUFFER FOR MCOMPARE
024710  004537  016736                  JSR     R5,FORMAT       ;FORMAT OUTPUT BUFFER
024714  012700  025450                  MOV     #TN17FB,R0      ;GET FORMAT CONTROL BLOCK ADDRESS
024720  005237  001176                  INC     SPECMD          ;SET SPECIAL TRANSFER FLAG
024724  004537  022700                  JSR     R5,DXFER        ;WRITE ERRONEOUS HEADER AND DATA
024730  005037  001176                  CLR     SPECMD          ;CLEAR SPECIAL TRANSFER FLAG
024734  005737  001322                  TST     ERRFLG          ;ANY ERRORS?
024740  001402                          BEQ     TN17A           ;CONTINUE IF NO
024742  000137  025416                  JMP     TN17EX          ;EXIT IF YES
024746  005204                  TN17A:  INC     R4              ;SUBTEST NUMBER=1(REFORMAT OK)
024750  012700  025472                  MOV     #ISECRB+8.,R0   ;GET READ BLOCK ADDR
024754  012710  000004                  MOV     #RDCMD,(R0)     ;INSURE READ COMMAND PRESENT
024760  005040                          CLR     -(R0)           ;CLEAR CYLINDER
024762  013740  001046                  MOV     WPSEC,-(R0)     ;SET WORD COUNT TO FULL SECTOR
024766  013701  001234                  MOV     CURDSK,R1       ;GET CURRENT DISK NUMBER
024772  006301                          ASL     R1              ;MAKE WORD INDEX
024774  016140  001200                  MOV     IBFTBL(R1),-(R0);UCAR=INPUT BUFFER ORIGIN
025000  005040                          CLR     -(R0)           ;DUSH=0
025002  004537  022700                  JSR     R5,DXFER        ;READ SECTOR WITH BAD HEADER
025006  005737  001322                  TST     ERRFLG          ;ANY ERRORS?
025012  001003                          BNE     .+8.            ;CONTINUE IF YES
025014  005237  001322                  INC     ERRFLG          ;SET ERROR FLAG IF NO ERROR DETECTED
025020  000576                          BR      TN17EX          ;EXIT IF NO ERROR FOUND
025022  017703  153766                  MOV     @DERR,R3        ;LOAD ERROR REGISTER
025026  012702  000002                  MOV     #2,R2           ;LOAD REFERENCE ERROR WORD
025032  020203                          CMP     R2,R3           ;IS THE DERR RIGHT?
025034  001170                          BNE     TN17EX          ;EXIT IF NO
025036  005204                          INC     R4              ;SUBTEST NUMBER=2(ERROR READ OK )
025040  005037  001322                  CLR     ERRFLG          ;RESET ERROR FLAG
025044  012737  000016  025472          MOV     #RWHCMD,ISECRB+8.;SPECIFY READ WITHOUT HEADER CHECK
025052  012700  025462                  MOV     #ISECRB,R0      ;GET READ BLOCK ADDR.
025056  004537  022700                  JSR     R5,DXFER        ;READ IGNORING HEADER WORD
025062  005737  001322                  TST     ERRFLG          ;ANYERRORS?
025066  001153                          BNE     TN17EX          ;EXIT IF YES
```

```
     025070  005204                        INC     R4              ;SUBTEST NUMBER=3 (READ IGNORING HEADER OK
     025072  012701  044312                MOV     #ODBUF,R1       ;LOAD BUFFERR ADDR IN R1
     025076  004537  016736                JSR     R5,FORMAT       ;REFORMAT BUFFER FOR COMPARE
     025102  013700  001050                MOV     POSWP,R0        ;SPECIFY WORD COMPARE COUNT
     025106  004537  017106                JSR     R5,DATCMP       ;CHECK DATA
     025112  005737  001322                TST     ERRFLG          ;ANY ERRORS?
     025116  001137                        BNE     TN17EX          ;EXIT IF YES
     025120  005204                        INC     R4              ;SUBTEST NUMBER=4(DATA IS CORRECT)
     025122  005037  044306                CLR     ODBUF-4         ;WRITE PROPER FORMAT ON SECTOR
     025126  005037  044310                CLR     ODBUF-2         ;
     025132  012700  025450                MOV     #TN17FB,R0      ;GET FORMAT CONTROL BLOCK
     025136  005237  001176                INC     SPECMD          ;SET SPECIAL TRANSFER FLAG
     025142  004537  022700                JSR     R5,DXFER        ;REWRITE CORRECT FORMAT
     025146  005037  001176                CLR     SPECMD          ;CLEAR SPECIAL TRANSFER FLAG
                                   ;
                                   ;
                                   ;       READ HDR,HDRCRC,DATA,DATACRC COMMAND VERIFY
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
     025152  005737  001322                TST     ERRFLG          ;ERRORS?
     025156  001117                        BNE     TN17EX          ;EXIT IF YES
     025160  005204                        INC     R4              ;SUBTEST NUMBER=5(REFORMAT OK)
     025162  005737  001064                TST     OLDHDR          ;DO WE HAVE BAD READ HDR ROMS?
     025166  001113                        BNE     TN17EX          ;EXIT WITHOUT ERROR IF YES
     025170  012700  027156                MOV     #WHSCCB,R0      ;POINT TO WRITE HDR,HDRCRC,DATA,DATACRC
     025174  012760  044312  000002        MOV     #ODBUF,2(R0)    ;LOAD OUTPUT BUFFER ADDR.
     025202  013760  001046  000004        MOV     WPSEC,4(R0)     ;LOAD WORD COUNT
     025210  162760  000004  000004        SUB     #4,4(R0)        ;AD JUST FOR EXTRA DATA
     025216  012701  044312                MOV     #ODBUF,R1       ;FORMAT DATA AREA
     025222  004537  016736                JSR     R5,FORMAT       ;
     025226  062701  000010                ADD     #8.,R1          ;ADJUST DATA AREA FOR EXTRA WORDS
     025232  004537  016736                JSR     R5,FORMAT       ;REFORMAT BUFFER
     025236  005237  001176                INC     SPECMD          ;FLAG SECTOR DATA WORD CHANGE
     025242  004537  022700                JSR     R5,DXFER        ;GO REFORMAT
     025246  005037  001176                CLR     SPECMD          ;CLEAR SPECIAL COMMAND FLAG
     025252  005737  001322                TST     ERRFLG          ;ERRORS?
     025256  001057                        BNE     TN17EX          ;EXIT IF YES
     025260  005204                        INC     R4              ;SUBTEST NUMBER=6(REFORMAT OK)
     025262  012700  025436                MOV     #RDHDBK,R0      ;POINT TO READ ALL DATA WORDS BLOCK
     025266  013760  001046  000004        MOV     WPSEC,4(R0)     ;ENTER WORD COUNT
     025274  162760  000004  000004        SUB     #4,4(R0)        ;ADJUST WORD COUNT
     025302  013702  001234                MOV     CURDSK,R2       ;SET UP INPUT ADDR.
     025306  006302                        ASL     R2              ;USING CURRENT DISK ADDR
     025310  016237  001200  025440        MOV     IBFTBL(R2),RDHDBK+2     ;
     025316  005237  001176                INC     SPECMD          ;FLAG ODD TRANSFER
     025322  004537  022700                JSR     R5,DXFER        ;TRY READING
     025326  005037  001176                CLR     SPECMD          ;CLEAR SPECIAL COMMAND FLAG
     025332  005737  001322                TST     ERRFLG          ;ERRORS?
     025336  001027                        BNE     TN17EX          ;EXIT IF YES
     025340  005204                        INC     R4              ;SUBTEST NUMBER=7(NO ERROR ON READ)
```

```
025342  016001  000004                   MOV     4(R0),R1         ;SET UP FOR DATA COMPARE
025346  010100                           MOV     R1,R0            ;
025350  005400                           NEG     R0               ;
025352  004537  017106                   JSR     R5,DATCMP        ;CHEACL DATA WORDS
025356  005737  001322                   TST     ERRFLG           ;ERRORS?
025362  001015                           BNE     TN17EX           ;EXIT IF YES
025364  005204                           INC     R4               ;SUBTEST NUMBER=10(DATA OK)
025366  012700  025450                   MOV     #ISECFB,R0       ;REFORMAT SECTOR
025372  005037  044306                   CLR     ODBUF-4          ;RESTORE SECTOR
025376  005037  044310                   CLR     ODBUF-2          ;
025402  005237  001176                   INC     SPECMD           ;FLAG ODD TRANSFER
025406  004537  022700                   JSR     R5,DXFER         ;RESTORE SECTOR 0
025412  005037  001176                   CLR     SPECMD           ;CLEAR TRANSFER FLAG
025416  012737  044306  027160  TN17EX:  MOV     #ODBUF-4,WHSCCB+2         ;RELOAD TRANSFER ADDR.
025424  012737  000004  025472           MOV     #RDCMD,ISECRB+8.;RELOAD READ COMMAND
025432  004537  003342                   JSR     R5,TSTCTL        ;AND EXIT
                                         .EOT
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                ;
                                         ;       READ HEADER,HEADER CRC,DATA,DATA CRC BLOCK
                                         ;     ******************************************
                                         ;
                                         ;
025436  000000                  RDHDBK:  .WORD 0          ;TRANSFER AT SECTOR,HEAD 0
025440  044312                           .WORD ODBUF      ;OUTPUT BUFFER
025442  177400                           .WORD 177400     ;WORD COUNT VARIABLE
025444  000000                           .WORD 0          ;CYLINDER ADDRESS= 0
025446  000012                           .WORD 12         ;READ EVERYTHING COMMAND
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;       TEST 17 FORMAT CONTROL BLOCK
                                         ;
                                         ;
                                         ;
025450  000000                  TN17FB:  .WORD 0          ;DUSH=0
025452  044306                           .WORD ODBUF-4    ;DBAR=OUTPUT BUFFER
025454  177700                           .WORD 177700     ;WORD COUNT LESS ONE SECTORH
025456  000000                           .WORD 0          ;CYLINDER =0
025460  000010                           .WORD FMTCMD     ;FORMAT OPERATION
                                         ;
                                         ;
                                         ;
        025450                           ISECFB=TN17FB
                                         ;
                                         ;
                                         ;
                                         ;       TEST 17 READ CONTROL BLOCK
                                         ;
                                         ;
                                         ;
025462  000000                  TN17RB:  .WORD 0          ;DUSH =0
```

```
     025464  047314                   .WORD IDBUF1     ;INPUT BUFFER DEFAULT CASE
     025466  177400                   .WORD 177400     ;ONE SECTOR WORD COUNT
     025470  000000                   .WORD 0          ;DCYL VALUE
     025472  000004                   .WORD RDCMD      ;READ COMMAND
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
             025462                   ISECRB=TN17RB          ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;       WRITE PROTECT OVERRIDE TEST
                                      ;
                                      ;
                                      ;       WRITE CHECK TEST
                                      ;
                                      ;
                                      ;
                                      ;  THIS TEST VERIFIES THAT THE WRITE PROTECT OVERRIDE
                                      ; COMMAND LOGIC FUNCTIONS PROPERLY.
                                      ;  KNOWN DATA IS WRITTEN ON TH DISK.THE SECTOR TO BE
                                      ; WRITE PROTECTED IN THEN DONE SO AND NW DATA IS
                                      ; WRITTEN USING THE OVERRIDE COMMAND.THE NEW
                                      ; DATA IS THEN READ BACK AND CHECKED TO VERIFY THAT
                                      ; THE WRITE OPERATION ACTUALLY TOOK PLACE
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;         THE SECOND PORTION OF THIS TEST
                                      ;       (SUBTEST NUMBERS 5-14) VERIFIES THE
                                      ;       WRITE CHECK LOGIC OF THE 211.
                                      ;          FIRST A DATA BUFFER IS FORMATTED AND
                                      ;       A NORMAL WRITE COMMAND IS ISSUED. A WRITE
                                      ;       CHECK COMMAND IS NEXT USED AND CHECKS FOR
                                      ;       ERRORS ARE MADE.FINALLY THE DATA BUFFER IS
                                      ;       MODIFIED AND  TWO WRITE CHECK COMMANDS ARE GIVEN TO
                                      ;       INSURE THAT A WRITE CHECK ERROR IS GENERATED.
                                      ;
                                      ;
     025474  005004           TN20:   CLR     R4              ;INITIALIZE SUBTEST NUMBER
     025476  004537  004276           JSR     R5,RQUEST       ;REQUEST 211
     025502  012701  044312           MOV     #ODBUF,R1       ;GET DATA BUFFER ADDRESS
     025506  004537  016736           JSR     R5,FORMAT       ;FORMAT DATA
     025512  013737  001304  044310   MOV     WPRBIT,ODBUF-2  ;WRITE PROTECT SECTOR ZERO
     025520  005037  044306           CLR     ODBUF-4         ;SECTOR,HEAD HEADER =0
     025524  012700  025450           MOV     #ISECFB,R0      ;GET ADDRESS OF ONE SECTOR FORMAT BLOCK
     025530  005237  001176           INC     SPECMD          ;SET SPECIAL TRANSFER FLAG
     025534  004537  022700           JSR     R5,DXFER        ;WRITE PROTECT SECTOR 0 AND REFERENCE DATA
     025540  005037  001176           CLR     SPECMD          ;CLEAR SPECIAL TRANSFER FLAG
     025544  005737  001322           TST     ERRFLG          ;ANY ERRORS?
```

```
025550  001161                   BNE    $TN20             ;EXIT IF YES
025552  005204                   INC    R4                ;SUBTEST NUMBER=1(WRITE PROTECT WRITE OK )
025554  013700  001050           MOV    POSWP,R0          ;ONE SECTOR WORD COUNT
025560  012701  044312           MOV    #ODBUF,R1         ;GET REFERENCE BUFFER ORIGIN
025564  005421                   NEG    (R1)+             ;COMPLEMENT REFERENCE DATA
025566  005300                   DEC    R0                ;DECREMENT WORD COUNT
025570  001375                   BNE    .-4               ;CONTINUE IF NOT ZERO
025572  012700  026336           MOV    #WPORCB,R0        ;GET OVERRIDE CB ADDRESS
025576  013760  001046  000004   MOV    WPSEC,4(R0)       ;LOAD ONE SECTOR WORD COUNT
025604  004537  022700           JSR    R5,DXFER          ;WRITE OVER PROTECTED DATA
025610  005737  001322           TST    ERRFLG            ;ANY ERRORS?
025614  001137                   BNE    $TN20             ;EXIT IF ERROR
025616  005204                   INC    R4                ;SUBTEST NUMBER=2(OVERRIDE OK 0
025620  012700  023622           MOV    #RDBLK+6,R0       ;GET CYLINDER ENTRY CB ADDRESS
025624  005010                   CLR    (R0)              ;CYLINDER=0
025626  013740  001046           MOV    WPSEC,-(R0)       ;ONE SECTOR WORD COUNT
025632  013701  001234           MOV    CURDSK,R1         ;GET CURRENT DISK DRIVE
025636  006301                   ASL    R1                ;MAKE INDEX WORD
025640  016140  001200           MOV    IBFTBL(R1),-(R0)  ;SPECIFY INPUT BUFFER ADDRESS
025644  005040                   CLR    -(R0)             ;SECTOR,HEAD=0
025646  004537  022700           JSR    R5,DXFER          ;READ DATA FROM SECTOR
025652  005737  001322           TST    ERRFLG            ;ANY ERRORS?
025656  001116                   BNE    $TN20             ;EXIT IF ERROR
025660  005204                   INC    R4                ;SUBTEST NUMBER =3
025662  013700  001050           MOV    POSWP,R0          ;ONE SECTOR COMPARE COUINT
025666  004537  017106           JSR    R5,DATCMP         ;COMPARE DATA
025672  005737  001322           TST    ERRFLG            ;ANY ERRORS?
025676  001106                   BNE    $TN20             ;EXIT IF ERROR
025700  005204                   INC    R4                ;SUBTEST NUMBER=4(DATA OVERRIDE OK)
025702  005037  044306           CLR    ODBUF-4           ;REFORMAT SECTOR WITH NO WRITE PROTECT
025706  005037  044310           CLR    ODBUF-2           ;
025712  012700  025450           MOV    #ISECFB,R0        ;GET ORIGIN ADDRESS OF FORMAT BLOCK
025716  005237  001176           INC    SPECMD            ;SET SPECIAL TRANSFER FLAG
025722  004537  022700           JSR    R5,DXFER          ;REFORMAT SECTOR
025726  005037  001176           CLR    SPECMD            ;RESET SPECIAL TRANSFER FLAG
025732  005737  001322           TST    ERRFLG            ;ERROT?
025736  001066                   BNE    $TN20             ;EXIT IF ERROR
025740  012704  000005   TNA20:  MOV    #5,R4             ;SUBTEST NUMBER=5(REFORMAT OK)
                                                          ;  THE SUBTEST NUMBER IS FORCED
                                                          ; HERE BCAUSE THIS SECTION IS
                                                          ;USED BY ECC CONTROLLERS
                                                          ;
025744  012701  044312           MOV    #ODBUF,R1         ;POINT TO DATA BUFFER
025750  004537  016736           JSR    R5,FORMAT         ;FORMAT DATA BUFFER
025754  013702  001050           MOV    POSWP,R2          ;COMPUTE LAST SECTOR WORD ADDR
025760  006302                   ASL    R2                ;
025762  062702  044312           ADD    #ODBUF,R2         ;
025766  012742  177777           MOV    #-1,-(R2)         ;MAKE LAST WORD ALL ONES
025772  012737  177777  044312   MOV    #-1,ODBUF         ;MAKE FORST WORD ALL ONES
026000  012700  023610           MOV    #WRTBLK+6,R0      ;POINT TO WRITE BLOCK ADDRESS
026004  013710  001056           MOV    RWCYL,(R0)        ;SPECIFY CYLINDER
026010  013740  001046           MOV    WPSEC,-(R0)       ;ONE SECTOR WORD COUNT
026014  012740  044312           MOV    #ODBUF,-(R0)      ;BUFF ADDRESS
026020  013740  001054           MOV    RDUSH,-(R0)       ;DUSH ADDRESS
026024  004537  022700           JSR    R5,DXFER          ;ISSUE WRITE OPERATION
026030  005737  001322           TST    ERRFLG            ;ERROR SET?
026034  001133                   BNE    TN20EX            ;EXIT IF YES
```

```
026036 005204                           INC  R4               ;SUBTEST NUMBER=6(FIRST WRITE OK)
026040 012737 000026 023612             MOV  #WRCHK,WRTBLK+8. ;LOAD WRITE CHECK COMMAND IN  WRITE BLOCK
026046 012700 023602                    MOV  #WRTBLK,R0       ;SET UP TO WRITE CHECK
026052 004537 022700                    JSR  R5,DXFER         ;GO WRITE CHECK
026056 005737 001322                    TST  ERRFLG           ;ERROR SET?
026062 001120                           BNE  TN20EX           ;EXIT IF YES
026064 005204                           INC  R4               ;SUBTEST NUMBER=7(WRITE CHECK OK)
026066 005337 044312                    DEC  ODBUF            ;CHANGE DATA BUFFER (FIRST BIT ONLY)
                                                              ;TO CAUSE WRITE CHECK ERROR
026072 012700 023602                    MOV  #WRTBLK,R0       ;SET UP FOR SECOND WRITE CHECK COMMAND
026076 004537 022700                    JSR  R5,DXFER         ;GO WRITE CHECK
026102 005737 001322                    TST  ERRFLG           ;ERROR FLAG SET?
026106 001004                           BNE  $TN20A           ;CONTINUE IF YES
026110 005237 001322                    INC  ERRFLG           ;SET IT IF NOT
026114 000137 026324           $TN20:   JMP  TN20EX           ;AND EXIT
026120 005204                  $TN20A:  INC  R4               ;SUBTEST NUMBER=10(ERROR DETECTED)
026122 017703 152666                    MOV  @DERR,R3         ;GET ERROR REG.
026126 042703 137777                    BIC  #137777,R3       ;EXTRACT WRITE CHECK ERROR
026132 013702 001302                    MOV  SEKINB,R2        ;SET UP REFERENCE
026136 020203                           CMP  R2,R3            ;WRITE CHECK ERROR SET?
026140 001071                           BNE  TN20EX           ;EXIT IF NOT
026142 005204                           INC  R4               ;SUBTEST NUMBER=11(WRITE CHECK ERROR SET)
026144 017703 152644                    MOV  @DERR,R3         ;FETCH ERROR REG AGAIN
026150 043703 001302                    BIC  SEKINB,R3        ;MASK OUT WRITE CHECK ERROR
026154 005002                           CLR  R2               ;SET UP REFERENCE WORD
026156 020203                           CMP  R2,R3            ;ANY OTHER ERRORS SET?
026160 001061                           BNE  TN20EX           ;EXIT IF YES
026162 005204                           INC  R4               ;SUBTEST NUMBER=12(NO OTHER ERRORS SET)
026164 005237 044312                    INC  ODBUF            ;RESTORE FIRST DATA BIT
026170 013702 001050                    MOV  POSWP,R2         ;COMPUTE LAST CORE WORD LOCATION
026174 006302                           ASL  R2               ;
026176 062702 044312                    ADD  #ODBUF,R2        ;
026202 042742 100000                    BIC  #BIT15,-(R2)     ;REMOVE LAST BIT OF DATA
026206 005037 001322                    CLR  ERRFLG           ;RESET ERROR FLAG
026212 012700 023602                    MOV  #WRTBLK,R0       ;SET UP FOR WRITE CHECK
026216 004537 022700                    JSR  R5,DXFER         ;GO WRITE CHECK!!!!!!!!!!!!!!
026222 005737 001322                    TST  ERRFLG           ;ERROR SET?
026226 001003                           BNE  $TN20B           ;CONTINUE IF YES
026230 005237 001322                    INC  ERRFLG           ;SET ERROR FLAG
026234 000433                           BR   TN20EX           ;AND EXIT
026236 005204                  $TN20B:  INC  R4               ;SUBTEST NUMBER=13(SECOND ERROR GENERATED)
026240 017703 152550                    MOV  @DERR,R3         ;FETCH ERROR REGISTER
026244 013702 001302                    MOV  SEKINB,R2        ;SET UP REFERENCE WORD
026250 030203                           BIT  R2,R3            ;WRITE CHECK ERROR SET?
026252 001424                           BEQ  TN20EX           ;EXIT IF NO
026254 005204                           INC  R4               ;SUBTEST NUMBER=14(WRITE CHECK ERROR )
026256 005037 001322                    CLR  ERRFLG           ;RESET ERROR FLAG
026262 012701 044312                    MOV  #ODBUF,R1        ;ENTER BUFFER LOCATION FOR FORMAT
026266 013702 001050                    MOV  POSWP,R2         ;RECOMPUTE LAST DATA WORD LOCATION
026272 006302                           ASL  R2               ;
026274 062702 044312                    ADD  #ODBUF,R2        ;
026300 012742 177777                    MOV  #-1,-(R2)        ;RESTORE LAST DATA WORD
026304 012700 023602                    MOV  #WRTBLK,R0       ;POINT TO COMMAND BLOCK
026310 004537 022700                    JSR  R5,DXFER         ;CHECK COMMAND AGAIN
026314 005737 001322                    TST  ERRFLG           ;ERROR FLAG SET?
026320 001001                           BNE  TN20EX           ;EXIT IF YES
026322 005204                           INC  R4               ;SUBTEST NUMBER=17(WRITE CHECK OK)
```

```
 026324  012737  000006  023612  TN20EX: MOV     #WRTCMD,WRTBLK+8.;RESTORE WRITE COMMAND
 026332  004537  003342                  JSR     R5,TSTCTL       ;EXIT TO TEST CONTROL
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;       TEST 20 WRITE PROTECT CONTROL BLOCK
                                         ;
                                         ;
 026336  000000                  WPORCB: .WORD 0         ;SECTOR,HEAD ADDR=0,0
 026340  044312                          .WORD ODBUF     ;DBAR=ODBUF
 026342  177400                          .WORD 177400    ;WORD COUNT =ONE SECTOR
 026344  000000                          .WORD 0         ;CYLINDER ADDR=0
 026346  000020                          .WORD WORCMD    ;WRITE OVERRIDE COMMAND
                                         ;       CDC6.P11        15-FEB-77
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;       HEADER WRITE COMMAND TEST
                                         ;
                                         ;
                                         ; THIS TEST VERIFIES THAT THE HEADER CAN BE WRITTEN
                                         ; BY THE WRITE HEADER AND DATA COMMAND,(CRC).
                                         ;  THE HEADER IS WRITTEN TO AN ERRONEOUS VALUE
                                         ; USING THE COMMAND AND,THEN A NORMAL READ IS
                                         ; ATTEMPTED AND THE PRESENCE OF THE APPROPRIATE
                                         ; HEADER ERROR BIT(S) IS(ARE) CHECKED.
                                         ;
                                         ;
                                         ;
                                         ;
 026350  005004                  TN21:   CLR     R4              ;INITIALIZE SUSTEST NUMBER
 026352  004537  004276                  JSR     R5,RQUEST       ;REQUEST 211
 026356  012701  044306                  MOV     #ODBUF-4,R1     ;GET DATA BUFFER ORIGIN ADDRESS
 026362  013702  001040                  MOV     MAXSEC,R2       ;GENERATE ERROREOUS HEADER
 026366  005202                          INC     R2              ;
 026370  010221                          MOV     R2,(R1)+        ;LOAD INTO BUFFER
 026372  005021                          CLR     (R1)+           ;MAKE CYLINDER HEADER VALID
 026374  004537  016736                  JSR     R5,FORMAT       ;FORMAT DATA BUFFER
 026400  012700  027156                  MOV     #WHSCCB,R0      ;GETWRITE SECTOR,HEAD,CRC,CB
 026404  013760  001046  000004          MOV     WPSEC,4(R0)     ;INCLUDE HEADR/CRC IN FORMAT
 026412  162760  000004  000004          SUB     #4,4(R0)        ;
 026420  005237  001176                  INC     SPECMD          ;SET SPECIAL TRANSFER FLAG
 026424  004537  022700                  JSR     R5,DXFER        ;PERFORM SPECIAL WRITE OPERATION
 026430  005037  001176                  CLR     SPECMD          ;CLEAR SPECIAL TRANSFER FLAG
 026434  005737  001322                  TST     ERRFLG          ;ANY ERRORS?
 026440  001076                          BNE     TN21A           ;EXIT IF YES
 026442  005204                          INC     R4              ;SUBTEST NUMBER=1(WRITE FUNCTION OK)
 026444  012700  023622                  MOV     #RDBLK+6,R0     ;GET CYLINDER ENTRY CB ADDRESS
 026450  005010                          CLR     (R0)            ;CYLINDER=0
```

```
026452  013740  001046          MOV     WPSEC,-(R0)     ;ONE SECTOR WORDC COUNT
026456  013701  001234          MOV     CURDSK,R1       ;GET CURRENT DISK
026462  006301                  ASL     R1              ;MAKE VALID INDEX
026464  016140  001200          MOV     IBFTBL(R1),-(R0);SPECIFY INPUT BUFFER ADDRESS
026470  005040                  CLR     -(R0)           ;SECTOR,HEAD=0
026472  004537  022700          JSR     R5,DXFER        ;ATTEMPT TO READ SECTOR ZERO
026476  005737  001322          TST     ERRFLG          ;ANY ERRORS?
026502  001004                  BNE     .+10.           ;CONTINUE IF YES
026504  005237  001322          INC     ERRFLG          ;SET ERROR FLAG
026510  000137  027152          JMP     TN21EX          ;EXIT TO TEST CONTROL
026514  017703  152274          MOV     @DERR,R3        ;GET ERROR REGISTER
026520  042703  077774          BIC     #77774,R3       ;EXTRACT HEADER ERROR BITS
026524  012702  000002          MOV     #2,R2           ;SET UP REFERENCE ERROR WORD
026530  020203                  CMP     R2,R3           ;ARE THE ERROR BITS CORRECT
026532  001041                  BNE     TN21A           ;EXIT IF NO
026534  005204                  INC     R4              ;SUBTEST  NUMBER=2(ERROR READ OK )
026536  017703  152242          MOV     @DCAR,R3        ;GET CORE ADDRESS REGISTER
026542  013702  023616          MOV     RDBLK+2,R2      ;GET ORIGINAL ORIGIN ADDRESS
026546  020203                  CMP     R2,R3           ;ARE THEY EQUAL
026550  001032                  BNE     TN21A           ;EXIT IF NOT
026552  005204                  INC     R4              ;SUBTEST NUMBER=3(NO DBAR INCREMENTATION)
026554  017703  152226          MOV     @DWCNT,R3       ;GET WORD COUNT REGISTER
026560  013702  023620          MOV     RDBLK+4,R2      ;GET ORIGINAL WORD COUNT
026564  020203                  CMP     R2,R3           ;ARE THEY EQUAL?
026566  001171                  BNE     TN21EX          ;EXIT IF NO
026570  005037  001322          CLR     ERRFLG          ;RESET ERROR FLAG
026574  005204                  INC     R4              ;SUBTEST NUMBER=4(NO WORD COUNT INCREMENTATION)
026576  013702  001042          MOV     MAXHD,R2        ;NOW GENERATE HEADER USING
                                                        ;BAD HEAD ADDRESS
026602  062702  000200          ADD     #HDINC,R2       ;
026606  010237  044306          MOV     R2,ODBUF-4      ;LOAD INTO BUFFER
026612  012700  027156          MOV     #WHSCCB,R0      ;POINT TO WRITE BLOCK
026616  005237  001176          INC     SPECMD          ;SET  SPECTIAL COMMAND FLAG
026622  004537  022700          JSR     R5,DXFER        ;REFORMAT SECTOR
026626  005037  001176          CLR     SPECMD          ;RESET SPECIAL COMMAND FLAG
026632  005737  001322          TST     ERRFLG          ;ERROR FLAG SET?
026636  001145          TN21A:  BNE     TN21EX          ;EXIT IF ERROR SET
026640  005204                  INC     R4              ;SUBTEST NUMBER=5(SECOND FORMAT OK)
026642  012700  023622          MOV     #RDBLK+6,R0     ;POINT TO  READ BLOCK ADDRESS
026646  005010                  CLR     (R0)            ;CLEAR CYLINDER VALUE
026650  013740  001046          MOV     WPSEC,-(R0)     ;ENTER WORD COUNT (ONE SECTOR)
026654  013701  001234          MOV     CURDSK,R1       ;SET UP PROPER BUFFER ADDR.
026660  006301                  ASL     R1              ;
026662  016140  001200          MOV     IBFTBL(R1),-(R0);ENTER INTO READ BLOCK
026666  005040                  CLR     -(R0)           ;READ SECTOR0,HEAD 0
026670  004537  022700          JSR     R5,DXFER        ;EXECUTE READ
026674  005737  001322          TST     ERRFLG          ;ERROR FLAG SET?
026700  001003                  BNE     .+8.            ;CONTINUE IF YES
026702  005237  001322          INC     ERRFLG          ;SET IT IF NO
026706  000521                  BR      TN21EX          ;AND EXIT
026710  005204                  INC     R4              ;SUBTEST NUMBER=6(AN ERROR IS DETECTED)
026712  017703  152076          MOV     @DERR,R3        ;FETCH ERROR REG.
026716  042703  177775          BIC     #177775,R3      ;EXTRACT SECTOR HEADER ERROR
026722  012702  000002          MOV     #2,R2           ;SET UP REFERENCE ERROR WORD
026726  020203                  CMP     R2,R3           ;DETECTED?
026730  001110                  BNE     TN21EX          ;EXIT IF NOT DETECTED
026732  005204                  INC     R4              ;SUBTEST NUMBER=7(HEAD HEADER DETECTED)
```

```
 026734  017703  152054            MOV     @DERR,R3          ;ERROR REG AGAIN
 026740  005002                    CLR     R2                ;SET UP REFERENCE WORD
 026742  042703  000002            BIC     #2,R3             ;MASK OUT HEADER ERROR
 026746  020203                    CMP     R2,R3             ;ANY OTHER ERRORS DETECTED?
 026750  001100                    BNE     TN21EX            ;EXIT IF YES
 026752  005204                    INC     R4                ;SUBTEST NUMBER=10(NO OTHER  ERRORS DETECTED)
 026754  005037  001322            CLR     ERRFLG            ;RESET ERROR FLAG
 026760  005037  044306            CLR     ODBUF-4           ;MAKE FIRST HEADER VALID
 026764  013702  001044            MOV     MAXCYL,R2         ;CONFIGURE ILLEGAL CYLINDER WORD
 026770  005202                    INC     R2                ;
 026772  010237  044310            MOV     R2,ODBUF-2        ;LOAD ILLEGAL CYLINDER WORD
 026776  012700  027156            MOV     #WHSCCB,R0        ;WRITE NEW FORMAT ONTO DISK
 027002  005237  001176            INC     SPECMD            ;SET SPECIAL TRANSFER FLAG
 027006  004537  022700            JSR     R5,DXFER          ;
 027012  005037  001176            CLR     SPECMD            ;CLEAR SPECIAL TRANSFER FLAG
 027016  005737  001322            TST     ERRFLG            ;ERROR FLAG SET?
 027022  001053                    BNE     TN21EX            ;EXIT IF YES
 027024  005204                    INC     R4                ;SUBTEST NUMBER=11(REFORMAT CYL HDR OK)
 027026  012700  023622            MOV     #RDBLK+6,R0       ;GET WRITE BLOCK ADDRESS
 027032  005010                    CLR     (R0)              ;CLEAR CYLINDER VALUE
 027034  013740  001046            MOV     WPSEC,-(R0)       ;WORD COUNT=ONE SECTOR
 027040  013701  001234            MOV     CURDSK,R1         ;GET CURRENT DISK DRIVE
 027044  006301                    ASL     R1                ;MAKE VALID INDEX
 027046  016140  001200            MOV     IBFTBL(R1),-(R0)  ;GET OUTPUT BUFFER ADDRESS
 027052  005040                    CLR     -(R0)             ;SECTOR,HEAD = 0
 027054  004537  022700            JSR     R5,DXFER          ;READ SECTOR
 027060  005737  001322            TST     ERRFLG            ;ERROR FLAG SET?
 027064  001003                    BNE     .+8.              ;CONTINUE IF YES
 027066  005237  001322            INC     ERRFLG            ;SET ERROR FLAG IF NO
 027072  000427                    BR      TN21EX            ;AND EXIT TO TEST CONTROL
 027074  017703  151714            MOV     @DERR,R3          ;GET ERROR REGISTER
 027100  042703  177776            BIC     #177776,R3        ;EXTRACT CYLINDER ERROR BIT
 027104  012702  000001            MOV     #1,R2             ;SET UP REFERENCE ERROR
 027110  020203                    CMP     R2,R3             ;ARE THEY EQUAL?
 027112  001017                    BNE     TN21EX            ;EXIT IF NO
 027114  005037  001322            CLR     ERRFLG            ;CLEAR ERROR FLAG IF YES
 027120  005204                    INC     R4                ;SUBTEST NUMBER=12(CYL HDR ERROR DETECTED)
 027122  005037  044310            CLR     ODBUF-2           ;SET UP CYLINDER HEADER WORD
 027126  005237  001176            INC     SPECMD            ;SET SPECIAL TRANSFER FLAG
 027132  005037  044306            CLR     ODBUF-4           ;SET UP SECTOR HEAD HEADER WORD
 027136  012700  025450            MOV     #ISECFB,R0        ;GET FORMAT BLOCK ORIGIN
 027142  004537  022700            JSR     R5,DXFER          ;REWRITE HEADERS
 027146  005037  001176            CLR     SPECMD            ;CLEAR SPECIAL TRANSFER FLAG
 027152  004537  003342    TN21EX: JSR     R5,TSTCTL         ;EXIT TO TEST CONTROL
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;       TN21 WRITE HEADER,DATA,SECTOR,CONTROL BLOCK
                                   ;
                                   ;
                                   ;
 027156  000000            WHSCCB: .WORD 0          ;DUSH=0
 027160  044306                    .WORD ODBUF-4    ;DCAR=ODBUF-4
 027162  177374                    .WORD 177374     ;WORD COUNT = ONE SECTOR +4 WORDS
 027164  000000                    .WORD 0          ;CYLINDER 0
 027166  000014                    .WORD WHSCRC     ;WRITE HEADER,SECTOR,CRC COMMAND
```

```
2053                    ;
2054                    ;
2055                    ;
2056                    ;
2057                    ;
2058                    ;
```

```
                                             ;
                                             ;
                                             ;      7M.ECC  TEST NUMBERS 22 >> 25   13-NOV-77
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;

                                             ;
                                             ;
                                             ;
                                             ;      CRC AND HEADER COMPARE TESTS
                                             ;
                                             ;
                                             ;
                                             ;      THESE TESTS CHECK THE LOGIC GOVERNNING
                                             ;  THE DETECTION OF HEADER AND CRC ERRORS.USING
                                             ;  THEWRITE HEADER DATA AND CRC COMMAND,ERRONEOUS
                                             ;  HEADERS AND CRC CHECKS ARE WRITTEN IN VARIOUS
                                             ;  COMBINATIONS.CHECKS ARE THEN MADE FOR PROPER
                                             ;  ERROR BIT SETTINGS.
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
027170  005004                      TN22:    CLR     R4                 ;SUBTEST NUMBER=0
027172  004537  004276                       JSR     R5,RQUEST          ;REQUEST 211
027176  012701  044312                       MOV     #ODBUF,R1          ;LOAD BUFFER ADDRESS
027202  004537  016736                       JSR     R5,FORMAT          ;FORMAT DATA
027206  005037  044306                       CLR     ODBUF-4            ;MAKE HEADERS VALID
027212  005037  044310                       CLR     ODBUF-2            ;
027216  005037  044312                       CLR     ODBUF              ;CLEAR WORD FOR HEADER CRC CHECK
027222  005237  001176                       INC     SPECMD             ;SET SPECIAL COMMAND FLAG
027226  012700  027156                       MOV     #WHSCCB,R0         ;GET TRANSFER BLOCK ADDRESS
027232  013760  001046  000004               MOV     WPSEC,4(R0)        ;LOAD WORD COUNT
027240  162760  000004  000004               SUB     #4,4(R0)           ;MODIFY FOR HEADERS/CRC
027246  004537  022700                       JSR     R5,DXFER           ;GO WRITE BAD CRC CHECK
027252  005037  001176                       CLR     SPECMD             ;CLEAR SPECIAL COMMAND FLAG
027256  005737  001322                       TST     ERRFLG             ;ANY ERRORS?
027262  001067                               BNE     TN22A              ;EXIT IF YES
027264  005204                               INC     R4                 ;SUBTEST NUMBER=1(DATA CRC ERRONEOUSLY WRITTEN)
027266  012737  000004  025472               MOV     #RDCMD,ISECRB+8.        ;PUT READ COMMAND INTO TRANSFER BLK
027274  012700  025462                       MOV     #ISECRB,R0         ;POINT TO READ BLOCK
027300  013760  001046  000004               MOV     WPSEC,4(R0)        ;LOAD WORD COUNT -ONE SECTOR
027306  004537  022700                       JSR     R5,DXFER           ;GO READ
027312  005737  001322                       TST     ERRFLG             ;ERROR?
027316  001451                               BEQ     TN22A              ;EXIT IF NO
027320  005204                               INC     R4                 ;SUBTEST NUMBER=2(ERROR OCCURRED)
027322  017703  151466                       MOV     @DERR,R3           ;GET ERROR REGISTER
027326  042703  177377                       BIC     #177377,R3         ;EXTRACT CRC ERROR
027332  012702  000400                       MOV     #400,R2            ;SET UP REFERENCE WORD
027336  020203                               CMP     R2,R3              ;ARE THEY EQUAL?
```

```
027340 001040                 BNE   TN22A         ;EXIT IF NO
027342 005204                 INC   R4            ;SUBTEST NUMBER=3(CRC ERROR )
027344 017703 151444          MOV   @DERR,R3      ;GET ERROR REGISTER
027350 042703 000600          BIC   #600,R3       ;MASK OUT CRC AND DATA LATE ERRROR
027354 005002                 CLR   R2            ;SET UP REFERENCE ERROR WORD
027356 020203                 CMP   R2,R3         ;ARE THEY EQUAL?
027360 001030                 BNE   TN22A         ;EXIT IF NO
027362 005204                 INC   R4            ;SUBTEST NUMBER=4(NO ERRORS EXCEPT DATA LATE)
027364 005037 001322          CLR   ERRFLG        ;CLEAR ERROR FLAG
027370 013702 001040          MOV   MAXSEC,R2     ;FORMAT ERRONEOUS HEADER #1
027374 005202                 INC   R2            ;
027376 010237 044306          MOV   R2,ODBUF-4    ;ENTER INTO BUFFER AREA
027402 013702 001044          MOV   MAXCYL,R2     ;ALSO SECOND HEADER(CYLINDER)
027406 005202                 INC   R2            ;MAKE BAD
027410 010237 044310          MOV   R2,ODBUF-2    ;LOAD INTO WRITE BUFFER
027414 012700 027156          MOV   #WHSCCB,R0    ;LOAD WRITE BLOCK ADDR
027420 005237 001176          INC   SPECMD        ;SET SPECIAL TRANSFER FLAG
027424 004537 022700          JSR   R5,DXFER      ;GO WRITE HEADERS
027430 005037 001176          CLR   SPECMD        ;CLEAR SPECIAL TRANSFER FLAG
027434 005737 001322          TST   ERRFLG        ;ANY ERRORS?
027440 001402                 BEQ   .+6           ;CONTINUE IF NO
027442 000137 030152   TN22A: JMP   TN22EX        ;EXIT IF YES
027446 005204                 INC   R4            ;SUBTEST NUMBER=5(BOTH HDRS INVALID)
027450 012700 025462          MOV   #ISECRB,R0    ;GET ONE SECTOR READ BLOCK
027454 004537 022700          JSR   R5,DXFER      ;GO READ SECTOR
027460 005737 001322          TST   ERRFLG        ;ERRORS?
027464 001766                 BEQ   TN22A         ;EXIT IF NO
027466 005204                 INC   R4            ;SUBTEST NUMBER=6(ERROR DETECTED)
027470 017703 151320          MOV   @DERR,R3      ;GET ERROR REGISTER
027474 042703 177775          BIC   #177775,R3    ;EXTRACT ERRORS
027500 012702 000002          MOV   #2,R2         ;SET UP REFERENCE ERROR WORD
027504 020203                 CMP   R2,R3         ;ARE THEY EQUAL?
027506 001355                 BNE   TN22A         ;EXIT IF NO
027510 005204                 INC   R4            ;SUBTEST NUMBER=7(SECTOR ERROR DETECTED)
027512 005037 001322          CLR   ERRFLG        ;CLEAR GENERAL ERROR FLAG
027516 005037 044306          CLR   ODBUF-4       ;MAKE HEADER #1 VALID
027522 012700 027156          MOV   #WHSCCB,R0    ;GET TRANSFER BLOCK ADDRESS
027526 005237 001176          INC   SPECMD        ;SET SPECIAL COMMAND FLAG
027532 004537 022700          JSR   R5,DXFER      ;GO WRITE HEADERS
027536 005037 001176          CLR   SPECMD        ;CLEAR SPECIAL COMMAND FLAG
027542 005737 001322          TST   ERRFLG        ;ANY ERRORS?
027546 001335                 BNE   TN22A         ;EXIT IF YES
027550 005204                 INC   R4            ;SUBTEST NUMBER=10(WRITE HEADERS OK)
027552 012700 025462          MOV   #ISECRB,R0    ;GET ONE SECTOR READ BLOCK ADDR.
027556 004537 022700          JSR   R5,DXFER      ;GO READ SECTOR
027562 005737 001322          TST   ERRFLG        ;ANY ERRORS?
027566 001725                 BEQ   TN22A         ;EXIT IF NO
027570 005204                 INC   R4            ;SUBTEST NUMBER=11(ERRORS DETECTED)
027572 017703 151216          MOV   @DERR,R3      ;GET ERROR REGISTER
027576 042703 177776          BIC   #177776,R3    ;EXTRACT ERRORS
027602 012702 000001          MOV   #1,R2         ;SET UP REFERENCE ERROR WORD
027606 020203                 CMP   R2,R3         ;ARE THEY EQUAL
027610 001160                 BNE   TN22EX        ;EXIT IF NO
027612 005204                 INC   R4            ;SUBTEST NUMBER=12(TRK HDR DETECTED)
027614 017703 151174          MOV   @DERR,R3      ;GET ERROR REGISTER AGAIN
027620 042703 000001          BIC   #1,R3         ;MASK OUT TRACK HEADER ERROR
027624 005002                 CLR   R2            ;SET UP REFERENCE ERROR WORD
```

```
027626 020203                 CMP     R2,R3           ;ARE THEY EQUAL?
027630 001150                 BNE     TN22EX          ;EXIT IF NO
027632 005037 001322          CLR     ERRFLG          ;CLEAR ERROR FLAG
027636 005204                 INC     R4              ;SUBTEST NUMBER=13(NO OTHER ERRORS SET)
027640 005037 044310          CLR     ODBUF-2         ;MAKE TRACK HEADER VALID
027644 013702 001040          MOV     MAXSEC,R2       ;FORMAT BAD FIRST HEADER
027650 005202                 INC     R2              ;
027652 010237 044306          MOV     R2,ODBUF-4      ;LOAD INTO BUFFER AREA
027656 005237 001176          INC     SPECMD          ;SET SPECIAL COMMAND FLAG
027662 012700 027156          MOV     #WHSCCB,R0      ;GET  TRNSFER BLOCK
027666 004537 022700          JSR     R5,DXFER        ;GO WRITE HEADER
027672 005037 001176          CLR     SPECMD          ;CLEAR SPECIAL COMMAND FLAG
027676 005737 001322          TST     ERRFLG          ;ERRORS?
027702 001123                 BNE     TN22EX          ;EXIT IF YES
027704 005204                 INC     R4              ;SUBTEST NUMBER=14(HEADERS WRITTEN OK)
027706 012700 025462          MOV     #ISECRB,R0      ;GET ONE SECTOR READ BLOCK
027712 004537 022700          JSR     R5,DXFER        ; GO READ
027716 005737 001322          TST     ERRFLG          ;ERRORS DETECTED?
027722 001513                 BEQ     TN22EX          ;EXIT IF NO
027724 005204                 INC     R4              ;SUBTEST NUMBER=15(ERROR DETECTED)
027726 017703 151062          MOV     @DERR,R3        ;GET ERROR REGISTER
027732 042703 177775          BIC     #177775,R3      ;EXTRACT ERROR
027736 012702 000002          MOV     #2,R2           ;SET UP REFERENCE ERROR WORD
027742 020203                 CMP     R2,R3           ;ARE THEY EQUAL?
027744 001102                 BNE     TN22EX          ;EXIT IF NO
027746 005204                 INC     R4              ;SUBTEST NUMBER=16(HDR #1 ALONE ERROR DETECTED)
027750 017703 151040          MOV     @DERR,R3        ;GET ERROR REGISTER AGAIN
027754 042703 000002          BIC     #2,R3           ;MASK OUT CURRENT ERROR
027760 005002                 CLR     R2              ;SET UP REFERENCE ERROR WORD
027762 020203                 CMP     R2,R3           ;ARE THEY EQUAL?
027764 001072                 BNE     TN22EX          ;EXIT IF NOT
027766 005204                 INC     R4              ;SUBTEST NUMBER=17(NO OTHER ERRORS)
027770 005037 001322          CLR     ERRFLG          ;RESET ERRRO FLAG
027774 012737 000022 044312   MOV     #22,ODBUF       ;MAKE HEADER CRC WORD INVALID
030002 005037 044306          CLR     ODBUF-4         ;MAKE HEADERS VALID
030006 005037 044310          CLR     ODBUF-2         ;
030012 005237 001176          INC     SPECMD          ;SET SPECIAL COMMAND FLAG
030016 012700 027156          MOV     #WHSCCB,R0      ;POINT TO COMMAND BLOCK
030022 004537 022700          JSR     R5,DXFER        ;GO WRITE BAD CRC CHECK(HEADER)
030026 005037 001176          CLR     SPECMD          ;CLEAR SPECIAL COMMAND FLAG
030032 005737 001322          TST     ERRFLG          ;ANY ERRORS?
030036 001045                 BNE     TN22EX          ;EXIT IF ERROR
030040 005204                 INC     R4              ;SUBTEST NUMBER=20(REWRITE OK)
030042 012700 025462          MOV     #ISECRB,R0      ;GET ONE SECTOR READ BLOCK
030046 004537 022700          JSR     R5,DXFER        ;GO READ
030052 005737 001322          TST     ERRFLG          ;ERROR DETECTED?
030056 001435                 BEQ     TN22EX          ;EXIT IF NO
030060 005204                 INC     R4              ;SUBTEST NUMBER=21(ERROR DETECTED)
030062 017703 150726          MOV     @DERR,R3
030066 042703 177376          BIC     #177376,R3      ;CLEAR ERRORS NOT TESTED HERE
030072 012702 000401          MOV     #401,R2         ;SET UP REFERENCE ERROR WORD
030076 020203                 CMP     R2,R3           ;ARE THEY EQUAL?
030100 001024                 BNE     TN22EX          ;EXIT IF NO
030102 005204                 INC     R4              ;SUBTEST NUMBER=22(CRCHDR ERROR)
030104 017703 150704          MOV     @DERR,R3        ;GET ERROR REGISTER
030110 042703 000401          BIC     #401,R3         ;MASK OUT CRC ERROR
030114 005002                 CLR     R2              ;SET UP REFERENCE ERROR WORD
```

```
030116  005037  001322                CLR   ERRFLG          ;RESET ERROR FLAG
030122  020203                        CMP   R2,R3           ;ANY OTHER ERRORS?
030124  001012                        BNE   TN22EX          ;EXIT IF YES
030126  005237  001176                INC   SPECMD          ;SET SPECMD FLAG
030132  012700  025450                MOV   #ISECFB,R0      ;GET FORMAT BLOCK ADDRESS
030136  004537  022700                JSR   R5,DXFER        ;GO WRIT HEADERS
030142  005037  001176                CLR   SPECMD          ;CLEAR SPECIAL COMMAND FLAG
030146  004537  003342                JSR   R5,TSTCTL       ;RETURN TO TEST CONTROL
030152  005237  001322        TN22EX: INC   ERRFLG          ;SET ERROR FLAG
030156  000773                        BR    .-8.            ;AND RETURN

                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;       BAD SECTOR FLAG TESTS           TN23
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;       THIS TEST CHECKS THE BAD SECTOR LOGIC.
                                      ;  BAD SECTOR BITS ARE ADDED TO  HEADER WORD #1
                                      ; OF SECTOR 0,HEAD 0, TRACK 0 AND A READCOMMAND
                                      ; IS ISSUED.THE FIRST CHECK IS TO SEE IF THE BAD
                                      ; SECTOR BIT IS SET IN THE ERROR REGISTER(DERR)
                                      ; THEDATA IS THEN COMPARED WITH DATA
                                      ; PREVIOUSLY WRITTEN ON SECTOR ONE TO
                                      ; INSURE THAT THE READ TOOK PLACE FROM SECTOR ONE
                                      ; NOT SECTOR ZERO.
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
030160  005004                TN23:   CLR   R4              ;SUBTEST NUMBER=0
030162  004537  004276                JSR   R5,RQUEST       ;REQUEST 211
030166  005037  044306                CLR   ODBUF-4         ;MAKE FIRST HEADER VALID
030172  013737  001276  044310        MOV   BADBIT,ODBUF-2  ;ADD BAD SECTOR ID BITS
030200  012700  025450                MOV   #ISECFB,R0      ;GET FORMAT BLOCK ADDRESS
030204  013760  001046  000004        MOV   WPSEC,4(R0)     ;ADJUST WORD COUNT
030212  005237  001176                INC   SPECMD          ;FLAG SPECIAL TRANSFER
```

```
229 030216 004537 022700                 JSR  R5,DXFER          ;GO REFORMAT DATA
230 030222 005037 001176                 CLR  SPECMD            ;CLEAR SPECIAL TRANSFER FLAG
231 030226 005737 001322                 TST  ERRFLG            ;ANY ERRORS ON FORMATTING?
232 030232 001027                        BNE  TN23A             ;EXIT IF YES
233 030234 005204                        INC  R4                ;SUBTEST NUMBER=1(REFORMAT OK)
234 030236 012737 000001 024652          MOV  #1,ISECWB         ;MAKE DUSH =SECTOR ONE
235 030244 013701 001046                 MOV  WPSEC,R1          ;GET WORD COUNT
236 030250 012702 044312                 MOV  #ODBUF,R2         ;POINT R2 TO OUTPUT AREA
237 030254 012722 125252                 MOV  #125252,(R2)+     ;FORMAT DATA
238 030260 005201                        INC  R1                ;INCREMENT(DECREMENT) LOOP COUNT
239 030262 001374                        BNE  .-6               ;REPEAT IF NOT DONE
240 030264 012737 044312 024654          MOV  #ODBUF,ISECWB+2   ;LOAD OUTPUT BUFFER
241 030272 012700 024652                 MOV  #ISECWB,R0        ;GET ONE SECTOR WRITE BLOCK ORIGIN
242 030276 004537 022700                 JSR  R5,DXFER          ;GO WRITE KNOWN DATA
243 030302 005037 024652                 CLR  ISECWB            ;REMOVE SECTOR #1 ADDRESS
244 030306 005737 001322                 TST  ERRFLG            ;ERRORS?
245 030312 001104                 TN23A: BNE  TN23EX            ;EXIT IF YES
246 030314 005204                        INC  R4                ;SUBTEST NUMBER=2(WRITE ONE SECTOR ONE OK)
247 030316 012700 025462                 MOV  #ISECRB,R0        ;GET ONE SECTOR READ BLOCK
248 030322 005020                        CLR  (R0)+             ;MAKE DUSH =0
249 030324 013702 001234                 MOV  CURDSK,R2         ;GET CURRENT DISK NUMBER
250 030330 006302                        ASL  R2                ;MAKE VALID INDEX
251 030332 016210 001200                 MOV  IBFTBL(R2),(R0)   ;ENTER PROPER BUFFER ADDRESS
252 030336 005740                        TST  -(R0)             ;RELOCATE POINTER
253 030340 004537 022700                 JSR  R5,DXFER          ;GO ATTEMPT A READ ON SECTOR 0
254 030344 005737 001322                 TST  ERRFLG            ;ERROR?
255 030350 001003                        BNE  .+8.              ;CONTINUE IF NO
256 030352 005237 001322                 INC  ERRFLG            ;SET ERROR FLAG IF NO
257 030356 000462                        BR   TN23EX            ;EXIT
258 030360 005204                        INC  R4                ;SUBTEST NUMBER=3(ERROR OCCURRED)
259 030362 017703 150426                 MOV  @DERR,R3          ;GET ERROR REGISTER
260 030366 042703 157777                 BIC  #157777,R3        ;EXTRACT ERROR BIT
261 030372 012702 020000                 MOV  #20000,R2         ;SET UP ERROR REFERENCE
262 030376 020203                        CMP  R2,R3             ;ARE THEY EQUAL?
263 030400 001051                        BNE  TN23EX            ;EXIT IF NO
264 030402 005204                        INC  R4                ;SUBTEST NUMBER=4(BAD SECTOR FLAGGED)
265 030404 017703 150404                 MOV  @DERR,R3          ;GET ERROR REGISTER AGAIN
266 030410 042703 020000                 BIC  #20000,R3         ;MASK OUT ERROR BIT
267 030414 005002                        CLR  R2                ;SET UP REFERENCE ERROR WORD
268 030416 020203                        CMP  R2,R3             ;ARE THY EQUAL?
269 030420 001041                        BNE  TN23EX            ;EXIT IF NO
270 030422 005204                        INC  R4                ;SUBTEST NUMBER=5(NO OTHER ERRORS SET)
271 030424 017703 150352                 MOV  @DUSH,R3          ;GET DUSH
272 030430 042703 170000                 BIC  #170000,R3        ;EXTRACT SECTOR AND HEAD ADDRESSES
273 030434 005737 001246                 TST  SKPBAD            ;SKIP OVER BAD SECTOR?
274 030440 001033                        BNE  TN23C             ;LOOK FOR ABORT IF NO
275 030442 012702 000002                 MOV  #2,R2             ;SET UP REFERENCE ERROR WORD
276 030446 020203                        CMP  R2,R3             ;ARE THEY EQUAL?
277 030450 001025                        BNE  TN23EX            ;EXIT IF NO
278 030452 005204                 TN23B: INC  R4                ;SUBTEST NUMBER=6(DUSH OK)
279 030454 005037 001322                 CLR  ERRFLG            ;CLEAR GENERAL ERROR FLAG
280 030460 005037 044306                 CLR  ODBUF-4           ;MAKE BOTH HEADER WORDS VALID
281 030464 005037 044310                 CLR  ODBUF-2           ;
282 030470 012700 025450                 MOV  #ISECFB,R0        ;GET FORMAT TRANSFER BLOCK
283 030474 005237 001176                 INC  SPECMD            ;SET SPECIAL COMMAND FLAG
284 030500 004537 022700                 JSR  R5,DXFER          ;GO REFORMAT SECTOR
285 030504 005037 001176                 CLR  SPECMD            ;CLEARSPECIAL COMMAND FLAG
```

```
286 030510  005737  001322                  TST     ERRFLG          ;ANY ERRORS?
287 030514  001003                          BNE     TN23EX          ;EXIT IF YES
288 030516  005204                          INC     R4              ;SUBTEST NUMBER=7(REWRITE OK)
289 030520  005037  024652                  CLR     ISECWB          ;MAKE DUSH WRITE BLOCK=0
290 030524  004537  003342          TN23EX: JSR     R5,TSTCTL       ;EXIT TO TEST CONTROL
291                                         ;
292                                         ;
293                                         ;
294                                         ;
295 030530  012702  000001          TN23C:  MOV     #1,R2           ;SET UP REFERENCE
296 030534  020203                          CMP     R2,R3           ;OK?
297 030536  001372                          BNE     TN23EX          ;EXIT IF NO
298 030540  000744                          BR      TN23B           ;CONTINUE IF YES
299                                         ;
300                                         ;
301                                         ;
302                                         ;
303                                         ;
304                                         ;
305                                         ;
306                                         ;
307                                         ;
308                                         ;[
309                                         ;
310                                         ;
311                                         ;
312                                         ;
313                                 ;
314                                 ;
315                                 ;
316                                 ;
317                                 ;
318                                 ;
319                                 ;
320                                 ;
321                                 ;
322                                         ;       RTZ  SEEK TESTS
323                                         ;
324                                         ;
325                                         ;
326                                         ;       THESE TESTS CHECK THE RTZ COMMAND
327                                         ; FOR 2 AREAS.THE FIRST SECTION VERIFIES
328                                         ; THHAT THE DRIVE IS INDEED POSITIONED
329                                         ; ON TRACK 0 AFTER AN RTZ.SECONDLY THE CONDITIONS OF
330                                         ; THE SEEKING FLIP FLOPS ARE CHECKED (DSTAT) DURING
331                                         ; AND AFTER THE RTZ COMMAND.
332                                         ;
333                                         ;
334                                         ;
335                                         ;
336                                         ;
337                                         ;
338 030542  005004                  TN24:   CLR     R4              ;SUBTEST NUMBER=0
339 030544  004537  004276                  JSR     R5,RQUEST       ;REQUEST 211
340 030550  013777  001044  150232          MOV     MAXCYL,@DCYL    ;GET MAX CYLINDER VALUE
341 030556  105777  150216                  TSTB    @DCSR   ;FORMATTER READY
342 030562  100033                          BPL     TN24C           ;EXIT IF NO
```

```
030564  005204                        INC   R4              ;SUBTEST NUMBER-1
030566  013701  001234                MOV   CURDSK,R1       ;GET CURRENT DRIVE
030572  000301                        SWAB  R1              ;MAKE VALID SELECTION WORD
030574  006301                        ASL   R1              ;
030576  006301                        ASL   R1              ;
030600  006301                        ASL   R1              ;
030602  006301                        ASL   R1              ;
030604  010177  150172                MOV   R1,@DUSH        ;SELECT DISK DRIVE
030610  005777  150176                TST   @DSTAT  ;DRIVE READY?
030614  005777  150172                TST   @DSTAT          ;DEIVE READY?
030620  052777  000002  150152        BIS   #SEKCMD,@DCSR   ;ISSUE SEEK COMMAND
030626  052777  000001  150144        BIS   #GO,@DCSR       ;SET GO BIT
030634  105777  150152                TSTB  @DSTAT          ;WAIT SEEK DONE
030640  100375                        BPL   .-4             ;WAIT IF NOT
030642  005204                        INC   R4              ;SUBTEST NUMBER=2(SEEK DONE?)
030644  005777  150130                TST   @DCSR           ;ERRORS?
030650  100002                        BPL   .+6             ;CONTINUE IF NO
030652  000137  031352        TN24C:  JMP   TN24EX          ;EXIT IF ERROR
030656  005204                        INC   R4              ;SUBTEST NUMBER=3(NO ERROR)
030660  010177  150116                MOV   R1,@DUSH        ;RESELECT DISK DRIVE
030664  012702  000400                MOV   #400,R2         ;START REFERENCE WORD
030670  005005                        CLR   R5              ;START REFERENCE= DRIVE 0
030672  020537  001234        TN24A:  CMP   R5,CURDSK       ;FOUND IT?
030676  001403                        BEQ   TN24B           ;YES GO ON
030700  005205                        INC   R5              ;INCREMENT DISK COUNTER
030702  006302                        ASL   R2              ;SFLECT NEXT DRIVE
030704  000772                        BR    TN24A           ;TRY NEXT ONE
030706  010205                TN24B:  MOV   R2,R5           ;GET SEEK DONE COMPARE BIT
030710  000305                        SWAB  R5              ;PUT IN LOWER BYTE
030712  006305                        ASL   R5              ;SHIFT FOR REFERENCE
030714  006305                        ASL   R5              ;
030716  052702  100000        TN24D:  BIS   #BIT15,R2       ;OR IN DRIVE READY BIT
030722  053702  001234                BIS   CURDSK,R2       ;AND SEEK DONE ID BITS
030726  053705  001234                BIS   CURDSK,R5       ;OR IN INTERRUPT SEEK DONE ID BITS
030732  052705  100200                BIS   #100200,R5      ;R5=DSTAT IMAGE
030736  010177  150040                MOV   R1,@DUSH        ;RESELECT DRIVE
030742  042777  000136  150030        BIC   #FUNC,@DCSR     ;CLEAR FUNCTION BITS
030750  052777  000025  150022        BIS   #RECAL,@DCSR    ;ISSUE RECAL
030756  105777  150016                TSTB  @DCSR           ;WAIT FORMATTER READY
030762  100375                        BPL   .-4             ;
030764  017703  150022                MOV   @DSTAT,R3       ;LOAD DISK STATUS REG INTO R3
030770  042703  020000                BIC   #BIT13,R3       ;LOSE POSSIBLE WRITE PROTECT
030774  020203                        CMP   R2,R3           ;ANY SEEKING FLOPS SET?
030776  001165                        BNE   TN24EX          ;EXIT IF NO
031000  005204                        INC   R4              ;SUBTEST NUMBER=4(RIGHT SEEKING FLOPS SET)
031002  032703  000074                BIT   #SKDONE,R3      ;CHECK FOR SEEK DONE FLOPS
031006  001161                        BNE   TN24EX          ;EXIT IF ANY SET
031010  005204                        INC   R4              ;SUBTEST NUMBER=5(NO OTHER FLOPS SET)
031012  105777  147774                TSTB  @DSTAT          ;SEEK DONE?
031016  100375                        BPL   .-4             ;WAIT UNTIL SEEK DONE
031020  010502                        MOV   R5,R2           ;SET UP REFERENCE DATA WORD
031022  017703  147764                MOV   @DSTAT,R3       ;GET DISK STATUS REGISTER
031026  042703  020000                BIC   #BIT13,R3       ;LOSE POSSIBLE WRITE PROTECT
031032  042703  007400                BIC   #SKFLOP,R3      ;REMOVE SEEKING FLOPS IF ANY SET
031036  020203                        CMP   R2,R3           ;PROPER DISK DONE?
031040  001144                        BNE   TN24EX          ;EXIT IF NO
031042  005204                        INC   R4              ;SUBTEST NUMBER=6(SEEK DONE FLOPSET)
```

```
031044  032777  007400  147740          BIT     #SKFLOP,@DSTAT  ;ANY SEEKING FLOPS SET?
031052  001137                          BNE     TN24EX          ;EXIT IF YES
031054  005204                          INC     R4              ;SUBTEST NUMBER=7(NO SEEKING FLOPS SET)
031056  005077  147726                  CLR     @DCYL           ;LOAD CYLINDER ADDRESS 0 INTO DCYL
031062  010177  147714                  MOV     R1,@DUSH        ;RESELECT DISK
031066  042777  000136  147704          BIC     #FUNC,@DCSR     ;CLEAR FUNCTION BITS
031074  052777  000004  147676          BIS     #RDCMD,@DCSR    ;ISSUE READ
031102  013702  001234                  MOV     CURDSK,R2       ;GET CURRENT DISK AGAIN
031106  006302                          ASL     R2              ;MAKE VALID INDEX
031110  016277  001200  147666          MOV     IBFTBL(R2),@DCAR;LOAD INPUT AREA
031116  013777  001046  147662          MOV     WPSEC,@DWCNT    ;LOAD ONE SECTOR WORD COUNT
031124  053777  001302  147646          BIS     SEKINB,@DCSR    ;SET SEEK INHIBIT BIT
031132  005277  147642                  INC     @DCSR           ;SET GO BIT
031136  105777  147636                  TSTB    @DCSR           ;WAIT FORMATTER READY
031142  100375                          BPL     .-4             ;
031144  017703  147642                  MOV     @DSTAT,R3       ;LOAD DISK STATUS REGISTER
031150  042703  170377                  BIC     #170377,R3      ;EXTRACT SEEKING BITS
031154  005002                          CLR     R2              ;MAKE R2 VALID REFERENCE WORD
031156  020203                          CMP     R2,R3           ;ANY SEEKING FLOPS SET?
031160  001074                          BNE     TN24EX          ;EXIT IF YES
031162  005204                          INC     R4              ;SUBTEST NUMBER=10(SEEK INHIBIT OK)
031164  005777  147610                  TST     @DCSR           ;ERROR SUMMARY SET?
031170  100470                          BMI     TN24EX          ;EXIT IF YES
031172  005204                          INC     R4              ;SUBTEST NUMBER=11(DOES NOT SEEK ON SEEK INHIBIT)
031174  013702  001044                  MOV     MAXCYL,R2       ;LOAD CYLINDER ADDRESS
031200  006202                          ASR     R2              ;USE MAXCYL/2
031202  010277  147602                  MOV     R2,@DCYL        ;LOAD INTO CYLINDER REGISTER
031206  013702  001234                  MOV     CURDSK,R2       ;GET CURRENT DISK AGAIN
031212  006302                          ASL     R2              ;MAKE VALID INDEX
031214  016277  001200  147562          MOV     IBFTBL(R2),@DCAR;INPUT AREA
031222  042777  000136  147550          BIC     #FUNC,@DCSR     ;CLEAR FUNCTION BITS
031230  052777  000004  147542          BIS     #RDCMD,@DCSR    ;ISSUE READ
031236  013777  001046  147542          MOV     WPSEC,@DWCNT    ;LOAD WORD COUNT-ONE SECTOR
031244  053777  001302  147526          BIS     SEKINB,@DCSR    ;SET SEEK INHIBIT BIT
031252  005277  147522                  INC     @DCSR           ;AND SET GO BIT
031256  105777  147516                  TSTB    @DCSR           ;WAIT FORMATTER READY
031262  100375                          BPL     .-4             ;
031264  017703  147522                  MOV     @DSTAT,R3       ;GET DISK STATUS REGISTER
031270  042703  020000                  BIC     #BIT13,R3       ;LOSE POSSIBLE WRITE PROTECT
031274  012702  100000                  MOV     #100000,R2      ;SET UP REFERENCE WORD
031300  053702  001234                  BIS     CURDSK,R2       ;OR IN SEEK INTERRUPT ID BITS
031304  020203                          CMP     R2,R3           ;SEEK FLOPS SET?
031306  001021                          BNE     TN24EX          ;EXIT IF YES
031310  005204                          INC     R4              ;SUBTEST NUMBER=12(NO FLOPS SET)
031312  005777  147462                  TST     @DCSR           ;ERRORS?
031316  100015                  TEXA:   BPL     TN24EX          ;EXIT IF NO
031320  005204                          INC     R4              ;SUBTEST NUMBER=13(ERROR DETECTED)
031322  032777  000001  147464          BIT     #1,@DERR        ;TRACK HEADER ERROR?
031330  001410                          BEQ     TN24EX          ;EXIT IF NO
031332  042777  000136  147440          BIC     #FUNC,@DCSR     ;LOOSE FUNTION BITS
031340  052777  000001  147432          BIS     #SYSCLR,@DCSR   ;ISSUE FORMATTER CLEAR
031346  004537  003342                  JSR     R5,TSTCTL       ;EXIT
031352  005237  001322          TN24EX: INC     ERRFLG          ;SET ERROR FLAG
031356  000773                          BR      .-8.            ;AND EXIT
                                        ;
                                        ;
                                        ;
```

```
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       IMPLIED SEEK TESTS
                                        ;  ------------------------
                                        ;
                                        ;
                                        ;       THESE TESTS CHECK TWO SEPARATE
                                        ;  SEEK ORIENTED FUNCTIONS.FIRST THE IMPLIED
                                        ;  TYPE OF A SEEK IS GIVEN IN A READ COMMAND AND
                                        ;  A CHECK IS MADE TO SEE THAT THE SEEK DONE FLAG
                                        ;  IS NOT SET.SECONDLY, TWO READ COMMANDS ARE GIVEN
                                        ;  ONE WITH A PROPER CYLINDER LOCATION AND
                                        ;  SEEK INHIBIT SET TO INSURE THE ORIGINAL SEEK WAS
                                        ;  TO THE PROPER CYLINDER,AND THE OTHER
                                        ;  WITH A ERROEOUS CYLINDER ADDRESS AND SEEK INHIBITED
                                        ;  TO FORCE A CYLINDER HEADER COMPARE ERROR.
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
031360  005004                  TN25:   CLR     R4              ;SUBTEST NUMBER=0
031362  004537  004276                  JSR     R5,RQUEST       ;REQUEST FORMATTER
031366  004537  016332                  JSR     R5,DISKID       ;GO FORMAT DRIVE SELECTION BITS
031372  010277  147404                  MOV     R2,@DUSH        ;SELECT DISK DRIVE
031376  000240                          NOP
031400  000240                          NOP
031402  000240                          NOP
031404  005777  147402                  TST     @DSTAT          ;DRIVE READY?
031410  100126                          BPL     TN25EX          ;EXIT IF NO
031412  052777  000025  147360          BIS     #RECAL,@DCSR    ;RECALIBRTAE DRIVE
031420  105777  147366                  TSTB    @DSTAT          ;WAIT SEEK DONE
031424  100375                          BPL     .-4             ;WAIT UNTIL SEEK DONE
031426  005777  147346                  TST     @DCSR           ;ANY ERRORS?
031432  100003                          BPL     .+8.            ;CONTINUE IF NO
031434  005237  001322                  INC     ERRFLG          ;SET ERROR FLAG IF CONTROLLER ERROR
```

```
514 031440  000512                          BR      TN25EX          ;AND EXIT
515 031442  005204                          INC     R4              ;SUBTEST NUMBER=1(RTZ OK)
516 031444  012700  031712                  MOV     #IMPBLK,R0      ;GET IMPLIED SEEK BLOCK ADDRESS
517 031450  013760  001046  000004          MOV     WPSEC,4(R0)     ;LOAD WORD COUNT
518 031456  013760  001044  000006          MOV     MAXCYL,6(R0)    ;SEEK=MAXCYL
519 031464  006260  000006                  ASR     6(R0)           ;MAKE MAXCYL/2
520 031470  004537  022700                  JSR     R5,DXFER        ;SEEK TO MAXCYL/2
521 031474  005737  001322                  TST     ERRFLG          ;ANY ERRORS?
522 031500  001072                          BNE     TN25EX          ;EXIT IF YES
523 031502  005204                          INC     R4              ;SUBTEST NUMBER=2(FIRST SEEK OK)
524 031504  105777  147302                  TSTB    @DSTAT          ;IS A SEEK DONE FLAG SET?
525 031510  100003                          BPL     .+8.            ;CONTINUE IF NO
526 031512  005237  001322                  INC     ERRFLG          ;SET GENERAL ERROR FLAG IF YES
527 031516  000463                          BR      TN25EX          ;EXIT WITH ERROR
528 031520  005204                          INC     R4              ;SEEK DONE FLAG NOT SET,SUBTEST=3
529 031522  053737  001302  031722          BIS     SEKINB,IMPBLK+8.;ADD SEEK INHIBIT BIT
530 031530  012700  031712                  MOV     #IMPBLK,R0      ;GET IMPLIED SEEK BLOCK ADDRESS
531 031534  004537  022700                  JSR     R5,DXFER        ;GO READ SECTOR
532 031540  005737  001322                  TST     ERRFLG          ;ERRORS?
533 031544  001050                          BNE     TN25EX          ;EXIT IF YES
534 031546  005204                          INC     R4              ;NO ERRORS ON SEEK INHIBIT,SUBTEST =4
535 031550  013737  001044  031720          MOV     MAXCYL,IMPBLK+6 ;LOAD MAXCYL INTO DCYL POSITION
536 031556  012700  031712                  MOV     #IMPBLK,R0      ;GET IMPLIED SEEK BLOCK ADDRESS
537 031562  004537  022700                  JSR     R5,DXFER        ;GO TRY AND READ WITH SEEK INHIBIT
538 031566  005737  001322                  TST     ERRFLG          ;ERROR DETECTED?
539 031572  001003                          BNE     .+8.            ;CONTINUE IF YES
540 031574  005237  001322                  INC     ERRFLG          ;SET GENERAL ERROR FLAG IN NO ERROR
541 031600  000432                          BR      TN25EX          ;AND EXIT
542 031602  005204                          INC     R4              ;SUBTEST NUMBER=5(ERROR DETECTED)
543 031604  012702  000001                  MOV     #1,R2           ;SET UP REFERENCE ERROR WORD
544 031610  017703  147200                  MOV     @DERR,R3        ;GET ERROR REGISTER
545 031614  020203                          CMP     R2,R3           ;ARE THEY EQUAL?
546 031616  001023                          BNE     TN25EX          ;EXIT IF NOT
547 031620  005037  001322                  CLR     ERRFLG          ;CLEAR ERROR FLAG IF YES
548 031624  005204                          INC     R4              ;SUBTEST NUMBER=6(ERRORS CLEARED)
549 031626  042777  000136  147144          BIC     #FUNC,@DCSR     ;LOOSE FUNTION BITS
550 031634  052777  000001  147136          BIS     #SYSCLR,@DCSR   ;AND ISSUE SYSTEM CLEAR
551 031642  004537  004276                  JSR     R5,RQUEST       ;REQUEST 211
552 031646  105777  147126                  TSTB    @DCSR           ;WAIT FORMATTER READY
553 031652  100375                          BPL     .-4             ;
554 031654  005777  147120                  TST     @DCSR           ;ERRORS CLEARED?
555 031660  100002                          BPL     TN25EX          ;EXIT IF YES
556 031662  005237  001322                  INC     ERRFLG          ;SET ERROR FLAG IF NOT
557 031666  043737  001302  031722  TN25EX: BIC     SEKINB,IMPBLK+8.;CLEAR SEEK INHIBIT BIT
558 031674  013737  001044  031720          MOV     MAXCYL,IMPBLK+6 ;RESTORE CYLINDER VALUE
559 031702  006237  031720                  ASR     IMPBLK+6        ;
560 031706  004537  003342                  JSR     R5,TSTCTL       ;RETURN TO TEST CONTROL
561                                         ;
562                                         ;
563                                         ;
564                                         ;
565                                         ;
566                                         ;
567 031712  000000                  IMPBLK: .WORD 0                 ;DUSH IMAGE=0
568 031714  047314                          .WORD IDBUF1            ;DBAR=IDBUF1
569 031716  177400                          .WORD 177400            ;WORD COUNT = ONE SECTOR
570 031720  000000                          .WORD 0                 ;CYLINDER ADDRESS TO BE SETUP
```

```
    031722  000004          .WORD RDCMD             ;DCSR=READ COMMAND
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;       8M.ECC  TEST NUMBERS 26 >> 30   13-NOV-77
                            ;
                            ;
                            ;       SEEK LOOP TEST
                            ;
                            ;
                            ;
                            ;  THIS TEST SENDS THE DISK DRIVE INTO A WORST CASE
                            ; SEEKING LOOP.
                            ;  THE SEEKING STARTS AT THE EXTREME ENDS OF THE DISK
                            ; CARTRIDGE(TRACK0 AND TRACK 823.) AND ALTERNATES
                            ; BETWEEN AN INCREASING TRACK ADDRESS AND A DECREASING TRACK
                            ; ADDRESS.THIS PROCESS CONTINUES UNTIL EACH TRACK HAS BEEN
                            ;  SOUGHT TWICE.
                            ;
                            ;
                            ;       DISK ADDRESS TEST
                            ;   ************************
                            ;
                            ;
                            ;       THIS SECTION OF TEST 26  CHECKS THE THREE
                            ;  DISK ADDRESS FUNCTIONS: SECTOR,HEAD AND CYLINDER.
                            ;       STARTING AT DISK ADDRESS 0,A ONE SECTOR
                            ;   WRITE READ AND DATA COMPARE IS ISSUED UNTIL
                            ;   ALL SECTOR AND HEAD ADDRESSES ARE EXERCISED.
                            ;       THE SECOND PART USES A MAXIMUM SECTOR
                            ;  AND HEAD ADDRESS AND ISSUES THE SAME TRANSFER
                            ;/    SEQUENCE ON EACH CYLINDER UNTIL A DISK OVERRUN ERROR
                            ;   IS ENCOUNTERED.
                            ;       ENDING DISK PARAMETERS ARE CHECKED AFTER EACH TRANSFER.
                            ;
                            ;
                            ;
                            ;
                            ;
                            ;
```

```
                                          ;
                                          ;     ERROR CONDITIONS(SEEKING SECTION)
                                          ;
                                          ;             R1=DECREASING TRACK ADDRESS
                                          ;             R2=INCREASING TRACK ADDRESS
                                          ;             R3=LOOP COUNT ERROR OCCURRED ON
                                          ;             R5=1 FOR INCREASING ADDRESS
                                          ;                     OR 0 FOR DECREASING ADDR.
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
 031724  005004                  TN26:    CLR     R4                  ;SUBTEST NUMBER=0
 031726  004537  004276                   JSR     R5,RQUEST           ;REQUEST 211
 031732  105777  147042                   TSTB    @DCSR               ;FORMATTER READY?
 031736  100173                           BPL     TN26EX              ;EXIT IF NO
 031740  005204                           INC     R4                  ;SUBTEST NUMBER=1
 031742  004537  016332                   JSR     R5,DISKID           ;FETCH JUSTFIED DISK ID
 031746  010277  147030                   MOV     R2,@DUSH            ;SELECT DRIVE
 031752  010200                           MOV     R2,R0               ;USE R0 FOR SELECTING FROM NOW ON
 031754  000240                           NOP                         ;WAIT READY
 031756  000240                           NOP                         ;
 031760  000240                           NOP                         ;
 031762  000240                           NOP                         ;
 031764  005777  147022                   TST     @DSTAT              ;DRIVE READY?
 031770  100156                           BPL     TN26EX              ;EXIT IF NOT READY
 031772  005204                           INC     R4                  ;SUBTEST NUMBER=2
 031774  012703  000001                   MOV     #1,R3               ;ESTABLISH OUTER LOOP=1
 032000  013704  001044          TN26B:   MOV     MAXCYL,R4           ;ESTABLISH MAXIMUM CYLINDER COUNT
 032004  010401                           MOV     R4,R1               ;PUT LOOP COUNT IN R1
 032006  005002                           CLR     R2                  ;ESTABLISH INCREASING TRACK POINTER
                                          ;
 032010  010177  146774          TN26A:   MOV     R1,@DCYL            ;LOAD CYLINDER ADDRESS
 032014  010077  146762                   MOV     R0,@DUSH            ;SELECT DRIVE
 032020  052777  000002  146752           BIS     #SEKCMD,@DCSR       ;LOAD SEEK COMMAND
 032026  052777  000101  146744           BIS     #INTGO,@DCSR        ;SET GO AND INTERRUPTS
 032034  004537  017342                   JSR     R5,WAIT             ;WAIT FOR INTERRUPT
 032040  005737  001150                   TST     INTFLG              ;INTERRUPT OCCUR?
 032044  001530                           BEQ     TN26EX              ;EXIT IF NO
 032046  005037  001150                   CLR     INTFLG              ;INTERRUPT OCCURRED
 032052  105777  146734                   TSTB    @DSTAT              ;SEEK DONE?
 032056  100123                           BPL     TN26EX              ;EXIT IF NO
 032060  005777  146714                   TST     @DCSR               ;ERROR?
 032064  100520                           BMI     TN26EX              ;EXIT IF YES
 032066  010277  146716                   MOV     R2,@DCYL            ;LOAD INCREASING CYLINDER ADDR
 032072  010077  146704                   MOV     R0,@DUSH            ;SELECT DRIVE
 032076  052777  000002  146674           BIS     #SEKCMD,@DCSR       ;LOAD SEEK COMMAND
 032104  052777  000101  146666           BIS     #INTGO,@DCSR        ;SET GO AND INTERRUPT BITS
 032112  004537  017342                   JSR     R5,WAIT             ;WAIT FOR INTERRUPT
 032116  005737  001150                   TST     INTFLG              ;DID WE INTERRUPT?
 032122  001501                           BEQ     TN26EX              ;EXIT IF NO
 032124  005037  001150                   CLR     INTFLG              ;RESET INTERRUPT OCCURRENCE FLAG
 032130  105777  146656                   TSTB    @DSTAT              ;SEEK DONE?
 032134  100074                           BPL     TN26EX              ;EXIT IF NO
 032136  005777  146636                   TST     @DCSR               ;ERRORS?
```

```
032142  100471                    BMI     TN26EX          ;EXIT IF YES
032144  005202                    INC     R2              ;INCREMENT TRACK ADDRESS
032146  005301                    DEC     R1              ;DECREMENT DECREASING TRACK ADDRESS
032150  001317                    BNE     TN26A           ;REPEAT INNER LOOP IF NOT DONE
032152  005303                    DEC     R3              ;DECREMENT OUTER LOOP
032154  001311                    BNE     TN26B           ;REPEAT OUTER LOOP IF NOT DONE
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;
                                  ;       DISK ADDRESS TEST
                                  ;  **************************
                                  ;
                                  ;
                                  ;
                                  ;
032156  012704  000005   TN26C:   MOV     #5,R4           ;SET SUBTEST NUMBER=5(STARTING DISK ADDR TEST
032162  012700  035520            MOV     #WRCB,R0        ;POINT TO WRITE READ BLOCK
032166  005010                    CLR     (R0)            ;START DISK ADDR =0
032170  005060  000006            CLR     6(R0)           ;CYLINDER ALSO
032174  013760  001046  000004    MOV     WPSEC,4(R0)     ;WORD COUNT =ONE SECTOR
032202  012701  044312            MOV     #ODBUF,R1       ;POINT TO OUT PUT BUFFER
032206  004537  016736            JSR     R5,FORMAT       ;FORMAT DATA BUFFER
032212  004537  035356   TN26D:   JSR     R5,WRC          ;GO WRITE/READ AND DATA COMPARE
032216  005737  001322            TST     ERRFLG          ;ERRORS?
032222  001027                    BNE     TN26F           ;EXIT IF YES
032224  013710  001162            MOV     ENDSEC,(R0)     ;UPDATE DISK ADDRESSES
032230  053710  001164            BIS     ENDHD,(R0)      ;FIRST SECTOR AND HEAD
032234  013760  001166  000006    MOV     ENDCYL,6(R0)    ;THEN CYLINDER
032242  026027  000006  000001    CMP     6(R0),#1        ;HAVE WE REACHED FIRST CYLINDER?
032250  001360                    BNE     TN26D           ;CONTINUE WITH SECTOR - HEAD CHEACKOUT IF NO
032252  005204                    INC     R4              ;SUBTEST NUMBER=6(SECTOR HEAD ADDRESS OK)
032254  013710  001054            MOV     RDUSH,(R0)      ;MAKE SECTOR- HEAD ADDR. MAX
032260  004537  035356   TN26E:   JSR     R5,WRC          ;WRITE READ/ DATA COMPARE ALL CYLINDER ADDR.
032264  005737  001322            TST     ERRFLG          ;ERRORS?
032270  001004                    BNE     TN26F           ;CHECK OVERRUN ERROR IF YES
032272  013760  001166  000006    MOV     ENDCYL,6(R0)    ;UPDATE CYLINDER ADDRESS IF NO ERROR
032300  000767                    BR      TN26E           ;AND CONTINUE
032302  032777  000040  146504 TN26F: BIT #BIT5,@DERR     ;IS THE ERROR NONEXT CYLINDER
032310  001404                    BEQ     TN26OT          ;EXIT IF NOT THAT ERROR
032312  000240                    NOP
032314  000240                    NOP
032316  005037  001322            CLR     ERRFLG          ;CLEAR ERROR FLAG IF YES
032322  004537  003342   TN26OT:  JSR     R5,TSTCTL       ;EXIT ALL DONE
032326  005737  001322   TN26EX:  TST     ERRFLG          ;ERROR FLAG SET?
032332  001373                    BNE     TN26OT          ;EXIT IF YES
032334  005237  001322            INC     ERRFLG          ;SHOULD BE SET-SET IT
```

```
    032340  000770                             BR      TN260T          ;AND EXIT
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;
                                               ;               1. OVERRUN TEST
                                               ;       ****************************************
                                               ;
                                               ;          THIS TEST CHECKS THE OVERRUN ERROR LOGIC
                                               ;       BY ATTEMPTING A READ PAST MAXSEC,MAXHD
                                               ;       AND MAXCYINDER.
                                               ;
                                               ;               2. MEMORY EXTENSION TEST
                                               ;       ****************************************
                                               ;
                                               ;          THIS TEST CHECKS THAT THE MEMORY EXTENSION BITS
                                               ;       ON THE PHOENIX 211 ARE SET WHEN AN INCREMENT IS GENERATED ON
                                               ;       ON THE BUSS ADDRESS REGISTER AFTER THE REGISTER HAS BEEN
                                               ;       SET TO 177776.
                                               ;
                                               ;               3. BUSS TME OUT TEST
                                               ;
                                               ;       ****************************************
                                               ;
                                               ;          THIS TEST DOES A TRANSFER THAT EXTENDS INTO NON EXISTANT
                                               ;       CORE FORCING A BUSS TIME OUT ERROR.
                                               ;
                                               ;
                                               ;*********************************************************
                                               ;
    032342  000137  033334             STN27:  JMP     TN27EX          ;EXIT JUMP
                                               ;
                                               ;*********************************************************
                                               ;
                                               ;
                                               ;
                                               ;
    032346  005004                     TN27:   CLR     R4              ;SUBTEST NUMBER=0
    032350  004537  004276                     JSR     R5,RQUEST       ;REQUEST 211
    032354  012700  033350                     MOV     #OVBLK,R0       ;FETCH OVERRUN BLOCK ADDRESS
    032360  013737  001234  044310             MOV     CURDSK,ODBUF-2  ;SAVE CURRENT  DISK UNIT
    032366  013710  001054                     MOV     RDUSH,(R0)      ;LOAD DUSH AS MAXIMUM
    032372  013760  001046  000004             MOV     WPSEC,4(R0)     ;LOAD WORD COUNT
    032400  006360  000004                     ASL     4(R0)           ;MAKE TWO SECTORS
    032404  013760  001044  000006             MOV     MAXCYL,6(R0)    ;LOAD LAST CYLINDER ADDRESS
    032412  004537  022700                     JSR     R5,DXFER        ;ISSUE WRITE COMMAND
    032416  005737  001322                     TST     ERRFLG          ;ERROR DETECTED?
    032422  001747                             BEQ     STN27           ;AND EXIT TO TEST CONTROL
```

```
032424  005204                                 INC     R4              ;SUBTEST NUMBER=1(ERROR DETECTED)
032426  012702  000010                         MOV     #10,R2          ;SET UP REFERENCE ERROR WROD
032432  017703  146356                         MOV     @DERR,R3        ;GET ERROR REGISTER
032436  042703  177767                         BIC     #177767,R3      ;EXTRACT OVERRUN BIT
032442  020203                                 CMP     R2,R3           ;ARE THE RIGHT ERRORS SET?
032444  001336                                 BNE     $TN27           ;EXIT IF NO
032446  005204                                 INC     R4              ;SUBTEST NUMBER=2(OVERRUN ERROR DETECTED)
032450  017703  146340                         MOV     @DERR,R3        ;FETCH ERROR REGISTER AGAIN
032454  042703  000010                         BIC     #10,R3          ;MASK OUT OVERRUN ERROR
032460  005002                                 CLR     R2              ;SET UP REFERENCE ERROR WORD
032462  020203                                 CMP     R2,R3           ;ANY OTHER ERRORS SET?
032464  001326                                 BNE     $TN27           ;EXIT IF YES
032466  005204                                 INC     R4              ;SUBTEST NUMBER=3(NO OTHER ERRORS)
032470  042777  000136  146302                 BIC     #FUNC,@DCSR     ;LOSE FUNCTION BITS
032476  052777  000001  146274                 BIS     #SYSCLR,@DCSR   ;ISSUE FORMATTER CLEAR
032504  004537  004276                         JSR     R5,RQUEST       ;REQUEST 211
032510  105777  146264                         TSTB    @DCSR           ;WAIT FORMATTER READY
032514  100375                                 BPL     .-4             ;
032516  005777  146256                         TST     @DCSR           ;ERRORS CLEARED?
032522  100707                                 BMI     $TN27           ;EXIT WITH ERROR FLAG SET IF NO
032524  005204                                 INC     R4              ;SUBTEST NUMBER=4(ERROR CLEARED BY SYS CLEAR)
032526  005037  001322                         CLR     ERRFLG          ;CLEAR GENERAL ERROR FLAG IF YES
032532  012700  023610                 TN27A:  MOV     #WRTBLK+6,R0    ;FETCH WORD COUNT ENTRY OF WRITE BLOCK
032536  005010                                 CLR     (R0)            ;ENTER CYL
032540  012740  177777                         MOV     #-1,-(R0)       ;ENTER WORD COUNT OF ONE
032544  012740  177776                         MOV     #-2,-(R0)       ;ENTER MEMORY ADDRESS OF 177776
032550  005040                                 CLR     -(R0)           ;MAKE DUSH=0
032552  004537  022700                         JSR     R5,DXFER        ;TRY A WRITE (MAY BE OK DEPENDING ON SIZE OF MEMORY)
032556  005037  001322                         CLR     ERRFLG          ;RESET ERROR FLAG IF SET
032562  017703  146212                         MOV     @DCSR,R3        ;FETCH DCSR REGISTER
032566  042703  147777                         BIC     #147777,R3      ;EXTRACT MEMORY EXTENSION BITS
032572  012702  010000                         MOV     #10000,R2       ;SET UP REFERENCE WORD
032576  020203                                 CMP     R2,R3           ;DID MEMORY REGISTER INCRENMENT?
032600  001260                                 BNE     $TN27           ;EXIT IF NO
032602  005204                                 INC     R4              ;SUBTEST NUMBER=5(FIRST MEMORY BIT OK)
032604  005777  146170                         TST     @DCSR           ;IS THERE AN ERROR?
032610  100013                                 BPL     TN27C           ;CONTINUE IF NO
032612  042777  000136  146160                 BIC     #FUNC,@DCSR     ;LOSE FUNCTION BITS
032620  052777  000001  146152                 BIS     #SYSCLR,@DCSR   ;ISSUE SYSTEM CLEAR
032626  004537  004276                         JSR     R5,RQUEST       ;REQUEST 211
032632  005777  146142                         TST     @DCSR           ;RETEST FOT ERROR
032636  100641                                 BMI     $TN27           ;EXIT IF ERROR CONDITION PERSISTS
032640  052737  020000  023612         TN27C:  BIS     #20000,WRTBLK+8.;OR IN SECONF MEMORY EXTENSION
032646  012700  023602                         MOV     #WRTBLK,R0      ;POINT R0 TO WRTBLK ADDRESS
032652  004537  022700                         JSR     R5,DXFER        ;ATTEMPT ANOTHER WRITE
032656  005037  001322                         CLR     ERRFLG          ;RESET ERROR FLAG IIF SET
032662  012702  030000                         MOV     #30000,R2       ;SET UP REFERENCE WORD
032666  017703  146106                         MOV     @DCSR,R3        ;FETCH DCSR
032672  042703  147777                         BIC     #147777,R3      ;EXTRACT MEMORY EXTENSION BITS
032676  020203                                 CMP     R2,R3           ;ARE THEY EQUAL?
032700  001220                                 BNE     $TN27           ;EXIT IF NO
032702  005204                                 INC     R4              ;SUBTEST NUMBER=6(SECOND MEMORY BIT OK)
032704  005777  146070                         TST     @DCSR           ;ERROR IN DCSR?
032710  100016                                 BPL     TN27D           ;CONTINUE IF NO
032712  042777  000136  146060                 BIC     #FUNC,@DCSR     ;LOSE FUNCTION BITS
032720  052777  000001  146052                 BIS     #SYSCLR,@DCSR   ;ISSUE SYSTEM CLEAR
032726  004537  004276                         JSR     R5,RQUEST       ;REQUEST 211
```

```
 032732  105777  146042                  TSTB   @DCSR             ;WAIT FOR DONE
 032736  100375                          BPL    .-4               ;
 032740  005777  146034                  TST    @DCSR             ;RETEST DCSR
 032744  100573                          BMI    TN27EX            ;EXIT IF ERROR PERSISTS
 032746  052737  030000  023612  TN27D:  BIS    #30000,WRTBLK+8.  ;SET BOTH MEMORY ECTENSION BITS
 032754  012700  023602                  MOV    #WRTBLK,R0        ;POINT R0 TO WRTBLK
 032760  004537  022700                  JSR    R5,DXFER          ;ATTEMPT WRITE
 032764  005037  001322                  CLR    ERRFLG            ;RESET ERROR FLAG IF SET
 032770  005002                          CLR    R2                ;SET UP REFERENCE WORD
 032772  017703  146002                  MOV    @DCSR,R3          ;FETCH DCSR FOR R3
 032776  042703  147777                  BIC    #147777,R3        ;EXTRACT MEMORY EXTENSION BITS
 033002  020203                          CMP    R2,R3             ;DID BUSS ADDRESS OVERFLOW?
 033004  001153                          BNE    TN27EX            ;EXIT IF NO
 033006  005777  145766                  TST    @DCSR             ;ERROR CONDITION?
 033012  100016                          BPL    TN27E             ;CONTINUE IF NO
 033014  042777  000136  145756          BIC    #FUNC,@DCSR       ;LOSE FUNCTION BITS
 033022  052777  000001  145750          BIS    #SYSCLR,@DCSR     ;ISSUE SYS CLEAR
 033030  004537  004276                  JSR    R5,RQUEST         ;REQUEST 211
 033034  105777  145740                  TSTB   @DCSR             ;WAIT FOR DONE
 033040  100375                          BPL    .-4               ;
 033042  005777  145732                  TST    @DCSR             ;RETEST DCSR
 033046  100532                          BMI    TN27EX            ;EXIT IF ERROR PERSISTS
 033050  005204                  TN27E:  INC    R4                ;SUBTEST NUMBER=7(BOTH MEMORY BITS OK)
 033052  012700  023602                  MOV    #WRTBLK,R0        ;POINT TO WRITE BLOCK
 033056  052760  030000  000010          BIS    #MEMEX,8.(R0)     ;LOAD MEMORY EXTENSION BITS
 033064  012760  177776  000002          MOV    #-2,2(R0)         ;MAKE BUSS ADDRESS MAX
 033072  004537  022700                  JSR    R5,DXFER          ;ISSUE WRITE
 033076  005737  001322                  TST    ERRFLG            ;ERROR DETECTED?
 033102  001001                          BNE    TN27F             ;CONTINUE IF YES
                                 ;**********************************************************
                                 ;       NOTE:   AT THIS POINT PDP 11'S WITH LESS THAN 96 K CORE
                                 ;               WILL NOT HAVE GENERATED AN ERROR.IF AN ERROR HAS
                                 ;               OCCURRED TESTING WILL PICK UP AT TN27F.IF THERE IS
                                 ;               NO ERROR THE PROGRAM WILL DISABLE 22 BIT ADDRESSING
                                 ;               AND RETRY FORCING THE ERROR.
                                 ;**********************************************************
                                 ;       THIS IS A TEMPORARY PATCH UNTIL DIAGNOSTIC IS REWRITTEN TO
                                 ;       HANDLE CORE ABOVE96K
 033104  000500                          BR     TN27OT            ;EXIT HERE
                                 ;
                                 ;
                                 ;
                                 ;
                                 ;
 033106  005204                  TN27F:  INC    R4                ;SUBTEST NUMBER=10(ERROR DETECTED)
 033110  012702  100000                  MOV    #BIT15,R2         ;SET UP REFERENCE ERROR WORD
 033114  017703  145674                  MOV    @DERR,R3          ;FETCH ERROR REGISTER
 033120  042703  077777                  BIC    #77777,R3         ;EXTRACT BUSS TIME OUT ERROR
 033124  020203                          CMP    R2,R3             ;BUSS ERROR SET?
 033126  001102                          BNE    TN27EX            ;EXIT IF NO
 033130  005204                          INC    R4                ;SUBTEST NUMBER=11(BUSS ERROR DETECTED)
 033132  005002                          CLR    R2                ;SET UP REFERENCE WORD
 033134  017703  145654                  MOV    @DERR,R3          ;FETCH ERROR REGISTER AGAIN
 033140  042703  100000                  BIC    #BIT15,R3         ;CANCEL BUSS TIME OUT ERROR
 033144  020203                          CMP    R2,R3             ;ANY OTHER ERRORS SET?
 033146  001072                          BNE    TN27EX            ;EXIT IF YES
 033150  005204                          INC    R4                ;SUBTEST NUMBER=12(NO OTHER ERRORS DETECTED)
```

```
033152  042777  000136  145620          BIC     #FUNC,@DCSR     ;LOSE FUNCTION BITS
033160  052777  000001  145612          BIS     #SYSCLR,@DCSR   ;ISSUE SYSTEM CLEAR
033166  004537  004276                  JSR     R5,RQUEST       ;REQUEST 211
033172  105777  145602                  TSTB    @DCSR           ;WAIT DONE
033176  100375                          BPL     .-4             ;
033200  005777  145574                  TST     @DCSR           ;TEST FOR ERROR
033204  100453                          BMI     TN27EX          ;EXIT IF ERROR PERSISTS
033206  005037  001322                  CLR     ERRFLG          ;RESET ERROR FLAG (ALL DONE)
033212  005204                          INC     R4              ;SUBTEST NUMBER=13(ERROR CLEARED)
033214  012700  023604                  MOV     #WRTBLK+2,R0    ;POINT TO WRITE BLOCK
033220  012710  160000                  MOV     #160000,(R0)    ;THIS BUSS ADDRESS WILL DETECT
                                        ;       POSSIBILITY OF FCO #75  MISSING IN INTERFACE BD'S
                                        ;       THE INTERFACE BD'S AFFECTED BY THIS FCO
                                        ;       ARE LEVEL A3 AND UNDER(74177'S > 74271'S).
                                        ;
                                        ;       ********** ALL THESE BOARDS ARE WIRE WRAP********
                                        ;
                                        ;
                                        ;       IF THIS FCO IS MISSING CERTAIN ADDRESSES BOUNDARIES
                                        ;       IN THE 150000 TO 170000 AREA WILL ERRONEOUSLY
                                        ;       INCREMENT THE BUSS EXTENSION BITS.OPERATING SYSTEMS
                                        ;       LARGER THAN 28K WILL NOT RUN!!!!!!!!!!!!!!!
033224  042760  030000  000006          BIC     #MEMEX,6(R0)    ;CLEAR MEMORY EXT. BITS IF SET
033232  012700  023602                  MOV     #WRTBLK,R0      ;LOAD WRITE BLOCK ADDR
033236  012701  000020                  MOV     #20,R1          ;LOAD ATTEMPT LOOP COUNT
033242  004537  022700          TN27H:  JSR     R5,DXFER        ;TRY TO GENERATE THIS ERROR
033246  005037  001322                  CLR     ERRFLG          ;ERROR FLAG MAY BE SET - DONT CARE
033252  017703  145522                  MOV     @DCSR,R3        ;FETCH COMMAND AND STATUS REG
033256  005002                          CLR     R2              ;SET UP REFERENCE WORD
033260  042703  147777                  BIC     #147777,R3      ;EXTRACT MEMORY EXT.
033264  020203                          CMP     R2,R3           ;ANYTHING SET?
033266  001403                          BEQ     TN27G           ;CONTINUE IF NO
033270  005237  001322                  INC     ERRFLG          ;SET ERROR FLAG IF YES
033274  000417                          BR      TN27EX          ;AND EXIT
033276  005301                  TN27G:  DEC R1                  ;DECREMENT LOOP COUNT
033300  001360                          BNE     TN27H           ;AND TRY AGAIN
033302  000137  033362                  JMP     TN27J           ;CONTINUE TO MEMORY INC TEST
033306  042737  030000  023612  TN27OT: BIC     #MEMEX,WRTBLK+8.;CLEAR MEMORY EXTENSION BITS
033314  042737  030000  024662          BIC     #MEMEX,ISECWB+8.;IN BITH WRITE BLOCKS
033322  013737  044310  001234          MOV     ODBUF-2,CURDSK  ; RELOAD CURRENT DISK
033330  004537  003342                  JSR     R5,TSTCTL       ;EXIT TO TEST CONTROL
033334  005737  001322          TN27EX: TST     ERRFLG          ;ERROR FLAG SET?
033340  001362                          BNE     TN27OT          ;EXIT IF YES
033342  005237  001322                  INC     ERRFLG          ;SET IT IF NOT
033346  000757                          BR      TN27OT          ;AND EXIT
                                        ;
                                        ;
                                        ;       OVERRUN RUN  COMMAND BLOCK
                                        ;
                                        ;
                                        ;
033350  000000                  OVBLK:  .WORD 0                 ;DUSH IMAGE
033352  047314                          .WORD IDBUF1            ;INPUT BUFFER IS IDBUF1
033354  177000                          .WORD 177000            ;WORD COUNT IS TWO SECTORS
033356  000000                          .WORD 0                 ;CYLINDER ADDRESS IS MAXCYL
033360  000006                          .WORD WRTCMD            ;COMMAND IS WRITE
                                        ;
```

```
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       MEMORY INCREMENT TESTS
                                        ;    ****************************
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       THESE TESTS INSURE THAT THE
                                        ;  BUSS ADDRESS REGISTER INCREMENTS CORRECTLY
                                        ;   FROM ANY STARTING ADDRESS.
                                        ;       A ONE WORD WRITE COMMAND IS ISSUED AND A CHECK
                                        ;  OF THE ENDING BUSS ADDRESS IS MADE.
                                        ;
                                        ;       ERRORS MAY BE SHOWING IN THE ERROR REGISTER DUE TO
                                        ;   SEVERAL DIFFERENT REASONS WHICH ARE NOT PART OF
                                        ;
                                        ;       THIS TESTS PROCESS.
                                        ;
                                        ;
        001004                          DBAR=DCAR                       ;REDEFINE CORE ADDRESS REGISTER FOR THOSE
                                        ;                               PEOPLE WHO CALL IT BUSS ADDRESS REGISTER
                                        ;
 033362 005204                  TN27J:  INC     R4                      ;SUBTEST NUMBER=14(NOW TEST MEM INC)
 033364 012700  023602                  MOV     #WRTBLK,R0              ;POINT TO WRITE
 033370 013710  001040                  MOV     MAXSEC,(R0)             ;LOAD ILLEGAL SECTOR ADDR TO SPEED UP TEST
 033374 005210                          INC     (R0)                    ;
                                                                        ;NOTE: IF WE TEST ALL CORE LOCATIONS IN
                                                                        ;THIS TEST IT WOULD RUN VERY LONG
                                                                        ;BECAUSE WITH A LEGAL MEMORY ADDRESS
                                                                        ;WE WOULD BE WAITING FOR SECTOR COINCIDENCE
                                                                        ;USING AN ILLEGAL SECTOR ADDR. ABORTS
                                                                        ;THE TRANSFER EARLY.
 033376 012760  177777  000004          MOV     #-1,4(R0)               ;SUE ONE WORD TRANSFER
 033404 005060  000006                  CLR     6(R0)                   ;CYLINDER ADDRESS=0
 033410 012760  000006  000010          MOV     #WRTCMD,10(R0)          ;RELOAD WRITE COMMAND
 033416 005077  145362                  CLR     @DBAR                   ;START WITH BUSS=0
 033422 012700  023602          TN27K:  MOV     #WRTBLK,R0              ;INSURE TRANSFER BLOCK ADDR
 033426 017760  145352  000002          MOV     @DBAR,2(R0)             ;LOAD BUSS ADDRESS
 033434 000240                          NOP
 033436 000240                          NOP
 033440 000240                          NOP
 033442 000240                          NOP
 033444 000240                          NOP
 033446 004537  022700                  JSR     R5,DXFER                ;TRY A WRITE
 033452 005037  001322                  CLR     ERRFLG                  ;CLEAR ERRORS NOT INTERESTED
 033456 016002  000002                  MOV     2(R0),R2                ;FETCH ORIGINAL BUSS ADDRESS
 033462 020227  177776                  CMP     R2,#-2                  ;SHOULD EXTENSION BE SETTING NOW?
 033466 001425                          BEQ     TN27N                   ;GO CHECK MEMORY EXT. IF YES
 033470 016002  000010                  MOV     10(R0),R2               ;CHECK IF TRANSFER STARTED WITH MEM. EXT. SET
 033474 042702  147777                  BIC     #147777,R2              ;EXTRACT BITS
```

```
033500  005702                          TST     R2              ;ANY SET?
033502  001051                          BNE     TN27P           ;USE DIFFERENT ROUTINE IF YES
033504  017703  145270                  MOV     @DCSR,R3        ;IF NOT LETS CHECK THAT THEY DIDN'T
033510  042703  147777                  BIC     #147777,R3      ;EXTRACT ENDING DCSR REG
033514  020203                          CMP     R2,R3           ;SEE IF BIT SET
033516  001306                          BNE     TN27EX          ;EXIT IF SOMTHING SET
033520  016002  000002          TN27L:  MOV     2(R0),R2        ;LOAD STARTING BUSS ADDR.
033524  062702  000002                  ADD     #2,R2           ;UPDATE
033530  017703  145250          TN27M:  MOV     @DBAR,R3        ;FETCH ACTUAL BUSS ADDR. REG
033534  020203                          CMP     R2,R3           ;ARE THE EQUAL?
033536  001276                          BNE     TN27EX          ;EXIT IF NO
033540  000730                          BR      TN27K           ;CONTINUE IF LEGAL
033542  005002                  TN27N:  CLR     R2              ;SET UP REFERENCE WORD
033544  017703  145234                  MOV     @DBAR,R3        ;
033550  020203                          CMP     R2,R3           ;EQUAL?
033552  001270                          BNE     TN27EX          ;EXIT IF NO
033554  016002  000010                  MOV     10(R0),R2       ;LOAD STARTING ADDR AGAIN
033560  042702  147777                  BIC     #147777,R2      ;EXTRACT BITS
033564  020227  030000                  CMP     R2,#30000       ;OVER FLOW?
033570  001427                          BEQ     TN27R           ;EXIT IF YES
033572  062702  010000                  ADD     #10000,R2       ;UPDATE EXT BIT
033576  017703  145176                  MOV     @DCSR,R3        ;FETCH COMMAND REGISTER
033602  042703  147777                  BIC     #147777,R3      ;EXTRACT MEM. EXT. BITS
033606  020203                          CMP     R2,R3           ;BIT INCREMENT?
033610  001251                          BNE     TN27EX          ;EXIT IF NO
033612  042760  030000  000010          BIC     #30000,10(R0)   ;UP DATE STARTING POINT
033620  050360  000010                  BIS     R3,10(R0)       ;
033624  000676                          BR      TN27K           ;GO ON
033626  017703  145146          TN27P:  MOV     @DCSR,R3        ;MEM EXT SHOULD'T HAVE INCREMENTED HERE
033632  042703  147777                  BIC     #147777,R3      ;EXTRACT MEM EXT BITS
033636  020203                          CMP     R2,R3           ;DID IT INCREMENTED?
033640  001235                          BNE     TN27EX          ;EXIT IF YES
033642  016002  000002                  MOV     2(R0),R2        ;ASSEMBLE REFERENCE WORD
033646  000724                          BR      TN27L           ;NOW CHECK BUSS ADDR.
033650  005002                  TN27R:  CLR     R2              ;SET UP REFERENCE FOR LAST ADDR.
033652  017703  145122                  MOV     @DCSR,R3        ;FETCH COMMANDREG.
033656  042703  147777                  BIC     #147777,R3      ;EXTRACT MEM EXT BITS
033662  020203                          CMP     R2,R3           ;OVERFLOW?
033664  001223                          BNE     TN27EX          ;EXIT IF NO
033666  017703  145112                  MOV     @DBAR,R3        ;NOW CHECK BUSS ADDRESS
033672  020203                          CMP     R2,R3           ;OVERFLOW?
033674  001217                          BNE     TN27EX          ;EXIT IF NO
033676  012760  000006  000010          MOV     #WRTCMD,10(R0)  ;RESTORE COMMAND IN TRANSFER BLOCK
033704  005010                          CLR     (R0)            ;RESTORE DUSH ADDRESS
033706  000137  033306                  JMP     TN27OT          ;AND EXIT
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
```

```
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;       SEEK/WRITE/DATA COMPARE TEST   TN30
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;       THIS TEST IS A RELIABILITY TEST.
                                                ;  THE DRIVE IS MADE TO SEEK IN A WORST CASE
                                                ;  PATTERN AND WRITE DATA  GENERATED BY A
                                                ;  SUBROUTINE,PATGEN, THAT CREATES A UNIQUE
                                                ;  ARRAY OF DATA EACH TIME IT IS CALLED.
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;       RANDOM TRANSFER TEST
                                                ;  ++++++++++++++++++++++++++++
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;       THE SECOND SECTION OH THIS TEST
                                                ;   IS A WRTITE/READ DATA COMPARE EXERCISE
                                                ;  USING RANDOM DATA AND RANDOM DISK ADDRESSES.
                                                ;   THE FOLLOWING IS A LIST OF THE RANDOM PARAMETERS
                                                ;
                                                ;       1. SECTOR ADDRESS
                                                ;       2. HEAD ADDRESS
                                                ;       3. TRACK ADDRESS
                                                ;       4. UNIT ADDRESS(IF MORE THAN ONE ONLINE)
                                                ;       5. TRANSFER SIZE (UP TO 2 SECTORS)
                                                ;       6. DATA WORD
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
033712  013737  001044  034124  TN30:   MOV     MAXCYL,CYLDN    ;LOAD MAXIMUM CYLINDER
033720  042737  000001  034124          BIC     #BIT0,CYLDN     ;AVOID ODD ENING CYLINDER NUMBER
033726  004537  004276                  JSR     R5,RQUEST       ;REQUEST 301
033732  012737  000002  036142          MOV     #2,CNTINC       ;MAKE PATTERN GENERATOR INCREMENT=2
033740  005037  034122                  CLR     CYLUP           ;CLEAR INCREASING CYLINDER COUNTER
033744  005004                  TN30A:  CLR     R4              ;INITIALIZE SUBTEST NUMBER
```

```
033746  012700  035520                    MOV     #WRCB,R0        ;GET WRITE/READ BLOCK ADDRESS
033752  013710  001054                    MOV     RDUSH,(R0)      ;MAKE DUSH MAXIMUM
033756  013760  034124  000006            MOV     CYLDN,6(R0)     ;LOAD CYLINDER ADDRESS
033764  013701  034124                    MOV     CYLDN,R1        ;LOAD CYLINDER ADDRESS FOR PATTERN BIAS
033770  004537  036064                    JSR     R5,PATGEN       ;FORMAT DATA
033774  013760  001046  000004            MOV     WPSEC,4(R0)     ;GO WRITE DISK
034002  004537  035356                    JSR     R5,WRC          ;GO WRITE READ DATA COMPARE
034006  005737  001322                    TST     ERRFLG          ;ANY ERRORS?
034012  001041                            BNE     TN30EX          ;EXIT IF YES
034014  005204                            INC     R4              ;SUBTEST NUMBER=1,2
034016  005337  034124                    DEC     CYLDN           ;DECREMENT CYLINDER ADDRESS
034022  020427  000002                    CMP     R4,#2           ;WRITE 2 CYLINDERS?
034026  001347                            BNE     TN30A+2         ;DO IT AGAIN IF NO
034030  005204                            INC     R4              ;SUBTEST NUMBER=3
034032  012700  035520            TN30B:  MOV     #WRCB,R0        ;GET TRANSFER BLOCK AGAIN
034036  005010                            CLR     (R0)            ;MAKE DUSH MINIMUM
034040  013760  034122  000006            MOV     CYLUP,6(R0)     ;LOAD INCREASING TRACK ADDRESS
034046  013701  034122                    MOV     CYLUP,R1        ;USE CYLINDER ADDRESS FOR PATTERN BIAS
034052  004537  036064                    JSR     R5,PATGEN       ;FORMAT DATA
034056  004537  035356                    JSR     R5,WRC          ;GO WRITE DISK
034062  005737  001322                    TST     ERRFLG          ;ANY ERRORS?
034066  001013                            BNE     TN30EX          ;EXIT IF YES
034070  005204                            INC     R4              ;SUBTEST NUMBER=4,5
034072  005237  034122                    INC     CYLUP           ;INCREMENT CYLINDER ADDRESS
034076  020427  000005                    CMP     R4,#5           ;HAVE WE DONE 2 WRITES?
034102  001353                            BNE     TN30B           ;WRITE ANOTHER SECTOR IF NO
034104  005737  034124                    TST     CYLDN           ;HAVE WE DONE WHOLE DISK?
034110  001315                            BNE     TN30A           ;WRITE 4 MORE SECTIRS IF NOT
034112  000137  034126                    JMP     TN30E           ;NOW GO USE RANDOM DATA
034116  004537  003342            TN30EX: JSR     R5,TSTCTL       ;EXIT TO PROGRAM CONTROL IF NO
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
034122  000000                    CYLUP:  .WORD 0                 ;INCREASING CYLINDER COUNTER
034124  000000                    CYLDN:  .WORD 0                 ;DECREADSING CYLINDER COUNTER
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;       RANDOM TRANSFER TEST
                                          ;    **************************
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;       THIS SECTION OF TEST 30
```

```
                                                ;   IS THE RANDOM TRANSFER TEST MENTIONED
                                                ;   AT THE INTRODUCTION OF THE MAIN TEST
                                                ;
                                                ;
                                                ;
                                                ;
034126  012701  002000          TN30E:  MOV     #2000,R1        ;USR R1 TO HOLD TRANSFER COUNTER
034132  013737  001234  001254          MOV     CURDSK,DSKSAV   ;SAVE CURRENT DISK NUMBER
034140  012700  035520          TN30F:  MOV     #WRCB,R0        ;POINT R0 TO TRANSFER BLOCK ADDRESS
034144  004537  015524                  JSR     R5,RANBUF       ;FOMAT OUTPUT BUFF WITH RANDOM DATA
034150  004537  016130                  JSR     R5,RANBLK       ;SET UP TRANSFER WITH RANDOM DISK ADDRESS
034154  004537  035356                  JSR     R5,WRC          ;GO WRTIE/READ AND CHECK DATA
034160  013737  001254  001234          MOV     DSKSAV,CURDSK   ;RESTOR CURRENT UNIT
034166  005737  001322                  TST     ERRFLG          ;ERRORS?
034172  001351                          BNE     TN30EX          ;EXIT IF YES
034174  005301                          DEC     R1              ;DECREMET TRANSFER COUNT
034176  001360                          BNE     TN30F           ;DONE 1000 TRANSFERS?
034200  000137  034116                  JMP     TN30EX          ;EXIT IF DONE
                                                                ;GO ON IF NO
                                                                ;
                                                                ;
                                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;

                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
```

```
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;           9M.ECC  TEST NUMBERS 31 >> 32
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;           DISK DATA TEST
                                        ;    *************************
                                        ;
                                        ;
                                        ;    THIS TEST WRITES OVER THE ENTIRE DISK SURFACE
                                        ;  AND THEN VERIFIES THE DATA BY READING IT BACK AND
                                        ;  COMPARING IT WITH THE ORIGINAL DATA WRITTEN
                                        ;
                                        ;    DATA IS WRITTEN TWO SECTORS AT A TIME AND READ TWO SECTORSAT
                                        ;A TIME THROUGH ALL OF THE SURFACES BEFORE THE CYLINDER
                                        ;REGISTER IS INCREMENTED.
                                        ;
                                        ;    THE DATA PATTERN USED IS A ROTATING TEST PATTERN WITH
                                        ;THE EXCEPTION THAT THE FIRST THREE WORDS OF EVERY SECTOR
                                        ;CONTAINS THE ABSOLUTE DISK ADDRESS.
                                        ;
                                        ;
                                        ;
                                        ;           WRITE PROGRAM
                                        ;
                                        ;    THIS PROGRAM WRITES OVER ALL OF THE DISK SURFACES TWO
                                        ; SECTORS AT A TIME. IF AN ERROR IS DETECTED CURSEC,CURHD,
                                        ;AND CURCYL INDICATE THE DISK ADDRESS AT WHICH THE FAILURE
                                        ;OCCURRED.
                                        ;
                                        ;
                                        ;
034204  012701  044312          TN31:   MOV     #ODBUF,R1       ;POINT R1 TO OUTPUT DATA BUF ORIGIN
034210  004537  004276                  JSR     R5,RQUEST       ;REQUEST 211
034214  005004                          CLR     R4              ;RESET SUBTEST COUNT
034216  004537  016736                  JSR     R5,FORMAT       ;FORMAT OUTPUT DATA BUFFER
034222  063701  001050                  ADD     POSWP,R1        ;GET TWO SECTOR OUTPUT ADDRESS
034226  063701  001050                  ADD     POSWP,R1        ;
034232  004537  016736                  JSR     R5,FORMAT       ;FORMAT OUTPUT BUFFER FOR SECTOR 2
034236  005237  001136                  INC     CURSEC          ;START WITH ABS SECTOR 1
034242  012700  023602          T31WLP: MOV     #WRTBLK,RO      ;GET WRITE BLOCK ORIGIN  ADDRESS
034246  013710  001136                  MOV     CURSEC,(RO)     ;FORMAT DUSH VALUE
034252  053720  001140                  BIS     CURHD,(RO)+     ;OR IN HED FIELD
034256  012720  044312                  MOV     #ODBUF,(RO)+    ;POINT DBAR TO OUTPUT DATA BUFFER
034262  005720                          TST     (RO)+           ;POINT RO TO CYLINDERR ENTRY IN CONTROL BLOCK
034264  013710  001142                  MOV     CURCYL,(RO)     ;LOAD DISK ADDRESS VALUE
034270  012702  001136                  MOV     #CURSEC,R2      ;PUT ABS DISK ADR IN FIRST THREE BUFFER
034274  012701  044312                  MOV     #ODBUF,R1       ;GET OUTPUT BUFFER ADDRESS
```

```
034300  012221                          MOV     (R2)+,(R1)+     ;WORDS
034302  012221                          MOV     (R2)+,(R1)+     ;
034304  011211                          MOV     (R2),(R1)       ;SAVE CURRENT DISK ADDRESS VALUES
034306  012701  044312                  MOV     #ODBUF,R1       ;POINT R1 TO OUTPUT BUFFER
034312  063701  001050                  ADD     POSWP,R1        ;INCREASE TO SECOND BUFFER OUTPUT
034316  063701  001050                  ADD     POSWP,R1
034322  013740  001046                  MOV     WPSEC,-(R0)     ;SET WORD COUNT FOR ONE SECTOR
034326  004537  022436                  JSR     R5,ENDPAR       ;COMPUTE DISK ADDRESS FOR SECOND SECTOR
034332  013721  001162                  MOV     ENDSEC,(R1)+    ;PUT ABSOLUTE DISK ADDRESS IN SECOND BUFFER
034336  013721  001164                  MOV     ENDHD,(R1)+     ;
034342  013721  001166                  MOV     ENDCYL,(R1)+    ;
034346  013710  001046                  MOV     WPSEC,(R0)      ;SET WORD COUNT FOR 2 SECTORS
034352  006320                          ASL     (R0)+           ;
034354  004537  035000                  JSR     R5,DSKTST       ;REDUCE WORD COUNT BY 316 IF LAST
                                                                ;WRITE OPERATION ON DISK
034360  012700  023602                  MOV     #WRTBLK,R0      ;POINT R0 TO WRITE BLOCK ORIGIN
034364  004537  022700                  JSR     R5,DXFER        ;PERFORM TWO SECTOR WRITE ON DISK
034370  005737  001322                  TST     ERRFLG          ;ANY ERRORS?
034374  001144                          BNE     T31EX           ;EXIT IF YES
034376  005737  001144                  TST     DKEND           ;ARE WE DONE WRITING?
034402  001012                          BNE     RD              ;CONTINUE WITH NEXT 2 SECTORS IF NOT
034404  013737  001162  001136          MOV     ENDSEC,CURSEC   ;UPDATE CURRENT SECTOR
034412  013737  001166  001142          MOV     ENDCYL,CURCYL   ;AND CYLINDER
034420  013737  001164  001140          MOV     ENDHD,CURHD     ;AND HEAD
034426  000705                          BR      T31WLP          ;GO WRITE AGAIN
                                ;
                                ;
                                ;
                                ;       READ AND COMPARE PROGRAM
                                ;
                                ;
                                ;       THIS PROGRAM READS THE COMPLETE DISK SURFACE AND
                                ;       COMPARES THE DATA RECEIVED WITH THEREFERENCE DATA.
                                ;
034430  005204                  RD:     INC     R4              ;SUBTEST+1
034432  012737  000001  001136          MOV     #1,CURSEC       ;REINITIALIZE DISK ADDRESS PARAMETERS
034440  005037  001140                  CLR     CURHD           ;
034444  005037  001144                  CLR     DKEND           ;CLEAR END OF DISK FLAG
034450  005037  001142                  CLR     CURCYL          ;
034454  012701  044312          T31RLP:MOV      #ODBUF,R1       ;GET INPUT BUFFER ORIGIN
034460  012700  001136                  MOV     #CURSEC,R0      ;GETCURRENT SECTOR ADDRESS
034464  012021                          MOV     (R0)+,(R1)+     ;PUT ABS DISK ADDRESS IN BUFFER
034466  012021                          MOV     (R0)+,(R1)+     ;
034470  011011                          MOV     (R0),(R1)       ;
034472  012700  023614                  MOV     #RDBLK,R0       ;POINT R0 TO READ BLOCK ORIGIN
034476  013710  001136                  MOV     CURSEC,(R0)     ;FORMAT DUSH WORD
034502  053720  001140                  BIS     CURHD,(R0)+     ;OR IN HEAD FIELD
034506  013701  001234                  MOV     CURDSK,R1       ;GET CURRENT DISK
034512  006301                          ASL     R1              ;MAKE VALID INDEX
034514  016120  001200                  MOV     IBFTBL(R1),(R0)+;LOAD INPUT BUFFER AREA
034520  013720  001046                  MOV     WPSEC,(R0)+     ;SPECIFY ONE SECTOR READ
034524  013710  001142                  MOV     CURCYL,(R0)     ;SET DISK CYLINDER
034530  005740                          TST     -(R0)           ;POINT R0 TO WORD COUNT CB ENTRY
034532  004537  022436                  JSR     R5,ENDPAR       ;COMPUTE ENDING DISK ADDRESS
034536  012700  001162                  MOV     #ENDSEC,R0      ;GET CURRENT SECTOR ADDRESS
034542  012701  044312                  MOV     #ODBUF,R1       ;POINT R1 TO BUFFER AREA
034546  063701  001050                  ADD     POSWP,R1        ;ADJUS FOR SECOND SECTOR
```

```
034552	063701	001050			ADD	POSWP,R1
034556	012021				MOV	(R0)+,(R1)+	;PUT DISK ADRRESS IN BUFFER
034560	012021				MOV	(R0)+,(R1)+	;
034562	011011				MOV	(R0),(R1)	;
034564	012700	023620			MOV	#RDBLK+4,R0	;POINT R0 TO WORD COUNT CB ENTRY
034570	013710	001046			MOV	WPSEC,(R0)	;SET UP FOR TWO SECTOR READ
034574	006320				ASL	(R0)+		;
034576	004537	035000			JSR	R5,DSKTST	;REDUCE WORD COUNT BY 316
034602	012700	023614			MOV	#RDBLK,R0	;GET READ BLOCK ADDRESS
034606	004537	022700			JSR	R5,DXFER	;INITIATE TWO SECTOR READ
034612	005737	001322			TST	ERRFLG		;ANY ERRORS?
034616	001033				BNE	T31EX		;EXIT IF YES
034620	013700	001050			MOV	POSWP,R0	;LOAD WORD COUNT INTO R0 FOR COMPARE
034624	063700	001050			ADD	POSWP,R0	;
034630	005737	001144			TST	DKEND		;END OF DISK FLAG SET?
034634	001402				BEQ	.+6		;CONTINUE IF NO
034636	163700	001050			SUB	POSWP,R0	;SUBTRACT ONE SECTOR WORD COUINT IF YES
034642	004537	017106			JSR	R5,DATCMP	;GO COMPARE DATA
034646	005737	001322			TST	ERRFLG		;ERROR FLAG SET?
034652	001015				BNE	T31EX		;EXIT IF SET
034654	005737	001144			TST	DKEND		;ARE WE DONE READING?
034660	001012				BNE	T31EX		;READ 2 MORE SECTORS IF NOT DONE
034662	013737	001162	001136		MOV	ENDSEC,CURSEC	;UPDATE DISK ADDRESSES
034670	013737	001164	001140		MOV	ENDHD,CURHD	;
034676	013737	001166	001142		MOV	ENDCYL,CURCYL	;
034704	000663				BR	T31RLP		;REPEAT READ OPERATION
034706	004537	003342		T31EX:	JSR	R5,TSTCTL	;GO ON TO NEXT TEST
							;
							;
							;	DISK ADDRESS SAVE SUBROUTINE
							;
							;
							;	THIS SUBROUTINE SAVES THE CURRENT SECTOR,HEAD AND
							;	CYLINDER VALUES AT A TEMPORARY
							;	BUFFER ,TMPADR.
							;
034712	010046			ADRSAV:	MOV	R0,-(SP)	;SAVE R0,R1
034714	010146				MOV	R1,-(SP)	;
034716	012700	034772			MOV	#TMPADR,R0	;GET TEMPORARY BUF ADDRESS
034722	012701	001136			MOV	#CURSEC,R1	;GET CURRENT SECTOR ADDRESS
034726	012120				MOV	(R1)+,(R0)+	;SAVE CURRENT SECTOR
034730	012120				MOV	(R1)+,(R0)+	;SAVE CURRENT HEAD
034732	012120				MOV	(R1)+,(R0)+	;SAVE CURRENT CYLINDER
034734	012601				MOV	(SP)+,R1	;RESTORE R1,R0
034736	012600				MOV	(SP)+,R0	;
034740	000205				RTS	R5		;RETURN
							;
							;
							;
							;	DISK ADDRESS RESTORE SUBROUTINE
							;
							;
							;	THIS SUBROUTINE RESTORES THE DISK ADDRESS SECTOR,
							;	HEAD, AND CYLINDER VALUES CONTAINED IN TMPADR.
							;
							;
							;
```

```
034742  010046                   ADRRST: MOV     R0,-(SP)          ;SAVE R0,R1
034744  010146                           MOV     R1,-(SP)          ;
034746  012700  034772                   MOV     #TMPADR,R0        ;GET TEMPORARY BUF ADDRESS
034752  012701  001136                   MOV     #CURSEC,R1        ;GET CURRENT SECTOR ADDRESS
034756  012021                           MOV     (R0)+,(R1)+       ;RESTORE CURRENT SECTOR
034760  012021                           MOV     (R0)+,(R1)+       ;RESTORW CURRENT HEAD
034762  011011                           MOV     (R0),(R1)         ;RESTORE CURRENT CYLINDER
034764  012601                           MOV     (SP)+,R1          ;RESTORE R1,R0
034766  012600                           MOV     (SP)+,R0          ;
034770  000205                           RTS     R5                ;RETURN
                                         ;
                                         ;
034772  000000                   TMPADR: .WORD 0                   ;CURRENT SECTOR BUFFER
034774  000000                           .WORD 0                   ;CURRENT HEAD
034776  000000                           .WORD 0                   ;CURRENT CYLINDER
                                         ;
                                         ;
                                         ;
                                         ;       DISK END TEST SUBROUTINE
                                         ;
                                         ;  THIS SUBROUTINE DETERMINES WETHER THE
                                         ; PHYSICAL END OF DISK MEDIUM WILL BE REACHED
                                         ; IF THE OPERATIONWITH CURRENT WORD COUNT IS RUN
                                         ;
                                         ;
                                         ;  IF THE OPERATION EXCEEDS DISK CAPACITY THE WORD
                                         ; COUNT IS REDUCED BY 316 WORDS OR ON SECTOR
                                         ;
                                         ;
                                         ; AT ENTRY R0 POINTS TO THE DISK CYLINDER ADDRESS
                                         ; IN THE OPERATION CONTROL BLOCK.
                                         ;
                                         ;
                                         ;
035000  024000                   DSKTST: CMP     -(R0),R0          ;POINT R0 TO THE WORD COUNT REGISTER
035002  004537  022436                   JSR     R5,ENDPAR         ;COMPUTE ENDING ADDRESS PARAMETERS
035006  005737  001144                   TST     DKEND             ;DID WE REACH END OF DISK?
035012  001402                           BEQ     .+6               ;EXIT IF NO
035014  163710  001046                   SUB     WPSEC,(R0)        ;REDUCE WORD COUNT BY ONE SECTOR
035020  000205                           RTS     R5                ;RETURN
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;       TEMPORARY PATCH AREA
                                         ;
                                         ;
035022  000000                   PTCH:   .WORD 0
        035334                           .=.+200.
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;       ERROR   IMAGE   BLOCK
                                         ;
                                         ;
```

```
                                        ;       THIS BLOCK CONTAINS THE CONTENTS OF ALL OF THE
                                        ;DISK CONTROLLER REGISTERS AT THE TIME ANY ERROR IS
                                        ;DETECTED. ERROR INFORMATION IS PLACED IN THE BLOCK
                                        ;BY THE DXFER SUBROUTINE.
                                        ;
                                        ;
035334  000000                  ERRBLK: .WORD 0         ;CONTROL &STATUS REGISTER IMAGE
035336  000000                          .WORD 0         ;SECTOR,UNIT,HEAD REGISTER
035340  000000                          .WORD 0         ;BUS ADDRESS REGISTER
035342  000000                          .WORD 0         ;WORD COUNT REGISTER
035344  000000                          .WORD 0         ;CYLINDER ADDRESS REGISTER
035346  000000                          .WORD 0         ;DISK STATUS REGISTER
035350  000000                          .WORD 0         ;DISK ERROR REGISTER
035352  000000                          .WORD 0         ;ECC BIT REGISTER
035354  000000                          .WORD 0         ;ECC PATTERN REGISTER
                                        ;
                                        ;
                                        ;
                                        ;READ,WRITE AND CHECK SUBROUTINE
                                        ;
                                        ;       THIS SUBROUTINE WRITES THE SPECIFIED DATA BLOCK
                                        ;ONTO THE SPECIFIED DEVICE AT THE SPECIFIED ADDRESS.
                                        ;THE ROUTINE THEN READS THE WRITTEN DATA BACK FROM THE DISK
                                        ;AND COMPARES THE DATA READ WITH THE ORIGINAL DATA WRITTEN.
                                        ;
                                        ;       CALLING SEQUENCE:
                                        ;
                                        ;       JSR     R5,WRC
                                        ;
                                        ;AT THE TIME OF THE CALL R0 MUST CONTAIN THE ORIGIN ADDRESS
                                        ;OF THE CONTROL BLOCK DEFINING THE PARAMETERS OF THE
                                        ;OPERATIONS TO BE PERFORMED.
                                        ;
                                        ;REQUIRED CONTROL BLOCK FORMAT IS AS FOLLOWS:
                                        ;
                                        ;       ENTRY 1 =DESIRED DUSH IMAGE
                                        ;       ENTRY 2 = BLANK (USED BY WRC FOR DCAR IMAGE)
                                        ;       ENTRY 3 = DESIRED DWDCNT IMAGE
                                        ;       ENTRY 4 = DESIRED DCYL IMAGE
                                        ;       ENTRY 5 = BLANK (USED BY WRC FOR DCSR CMD IMAGE)
                                        ;       ;RETURN PARAMETERS:
                                        ;       1.NORMAL RETURN IS WITH GENERAL ERROR FLAG RESET.
                                        ;       2.IF AN ERROR IS DETECTED DURING ANY OPERATION
                                        ;CONTROL IS TRANSFERRED TO USER WITH THE ERROR FLAG SET.
                                        ;       IN THIS CASE ERROR INFORMATION IS CONTAINED IN THE
                                        ;       DXFER OR DATCMP ERROR FLAGS AS APPROPRIATE.
                                        ;
                                        ;
035356  010046                  WRC:    MOV     R0,-(SP)        ;SAVE REGISTERS
035360  010146                          MOV     R1,-(SP)        ;
035362  010446                          MOV     R4,-(SP)        ;
035364  010546                          MOV     R5,-(SP)        ;
035366  012760  044312  000002          MOV     #ODBUF,2(R0)    ;POINT DCAR TO OUTPUT DATA BUF
035374  012760  000006  000010          MOV     #WRTCMD,8.(R0)  ;SET UP DCSR FOR WRITE CMD
035402  005037  001174                  CLR     WRTFRR          ;RESET WRITE ERROR FLAG
035406  004537  022700                  JSR     R5,DXFER        ;PERFORM SPECIFIED WRITE OPERATION
035412  005737  001322                  TST     ERRFLG          ;ANY ERRORS?
```

```
035416  001403                          BEQ     .+8.                    ;CONTINUE IF NO
035420  005237  001174                  INC     WRTERR                  ;SET WRITE ERROR FLAG IF YES
035424  000430                          BR      WRCEX                   ;EXIT
035426  013701  001234                  MOV     CURDSK,R1               ;GET CURRENT DISK NUMBER
035432  006301                          ASL     R1                      ;MAKE VALID INDEX VALUE
035434  016160  001200  000002          MOV     IBFTBL(R1),2(R0);POINT DCAR TO APPROPRIATE INPUT BUF
035442  012760  000004  000010          MOV     #RDCMD,8.(R0)           ;SET UP DCSR FOR READ COMMAND
035450  005037  001172                  CLR     RDERR                   ;RESET READ ERROR FLAG
035454  004537  022700                  JSR     R5,DXFER                ;PERFORM DESIRED READ OPERATION
035460  005737  001322                  TST     ERRFLG                  ;ANY ERRORS?
035464  001403                          BEQ     .+8.                    ;CONTIUE IF NO
035466  005237  001172                  INC     RDERR                   ;SET READ ERROR FLAG IF YES
035472  000405                          BR      WRCEX                   ;EXIT
035474  016000  000004                  MOV     4(R0),R0                ;GET SPECIFIED WORD COUNT IN R0
035500  005400                          NEG     R0                      ;FORM POSITIVE NUMBER
035502  004537  017106                  JSR     R5,DATCMP               ;COMPARE READ AND WRITTEN DATA
035506  012605                  WRCEX:  MOV     (SP)+,R5                ;RESTORE REGISTERS
035510  012604                          MOV     (SP)+,R4                ;
035512  012601                          MOV     (SP)+,R1                ;
035514  012600                          MOV     (SP)+,R0                ;
035516  000205                          RTS     R5                      ;RETURN
                                ;
                                ;
                                ;
                                ;               READ WRITE CONTROL BLOCK
                                ;
                                ;               THIS CONTROL BLOCK IS USED FOR ALL OPERATIONS
                                ;INVOLVING THE WRC SUBROUTINE.
                                ;
                                ;
035520  000000                  WRCB:   .WORD 0                         ;DUSH IMAGE
035522  000000                          .WORD 0                         ;BLANK (DBAR IMAGE)
035524  000000                          .WORD 0                         ;DWDCNT IMAGE
035526  000000                          .WORD 0                         ;DCYL IMAGE
035530  000000                          .WORD 0                         ;BLANK (DCSR IMAGE)
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
```

```
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;
;		READ AND COMPARE TEST
;	-----------------------------
;
;
;
;
;
;
;
;
;
;
;		THIS PROGRAM READS THE COMPLETE DISK SURFACE AND COMPARES
;	DATA  RECEIVED WITH REFERENCE  DATA.
;		BEFROE ENTERING THIS TEST IT IS ASSUMED THAT THE
;	ENTIRE DISK AREA HAS BEEN WRITTEN OVER BY TEST 25.
;	TEST 32 IS ESSENTIALLY THE READ PORTION OF TEST 25
;	MODIFIED TO INCLUDE A DATA ERROR COUNTER FOR USE AS
;	RELIABILITY TEST.SEETING SWITCH 7 OF THE
;	SWITCH REGISTER UPON ENTERING THE PROGRAM INHIBITS
;	PROGRAM HALT WHEN A DTAT ERROR IS DETECTED.LOCATION
;	ERRCNT IS INCREMENTED EACH TIME AN ERROR IS FOUND.
;
;
```

```
1654                                           ;
1655                                           ;
1656                                           ;
1657                                           ;
1658                                           ;
1659                                           ;
1660                                           ;
1661                                           ;
1662                                           ;
1663                                           ;
1664                                           ;
1665                                           ;
1666 035532  005004                  TN32:    CLR     R4                  ;SUBTEST NUMBER=0
1667 035534  004537  004276                   JSR     R5,RQUEST           ;REQUEST 211
1668 035540  005037  001132                   CLR     SOFTRY              ;CLEAR SECTOR REREAD IF SET
1669 035544  005237  001136                   INC     CURSEC              ;SET CURRENT SECTOR TO OFFSET
1670 035550  012701  044312                   MOV     #ODBUF,R1           ;GET OUT PUT AREA ADDRESS
1671 035554  004537  016736                   JSR     R5,FORMAT           ;
1672 035560  063701  001050                   ADD     POSWP,R1            ;GET SECOND OUTPUT AREA ADDRESS
1673 035564  063701  001050                   ADD     POSWP,R1            ;
1674 035570  004537  016736                   JSR     R5,FORMAT           ;FORMAT BUFFERS
1675 035574  012701  044312          TN32LP:  MOV     #ODBUF,R1           ;GET OUTPUT AREA ADDRESS AGAIN
1676 035600  012700  001136                   MOV     #CURSEC,R0          ;GET CURRENT SECTOR ADDRESS
1677 035604  012021                           MOV     (R0)+,(R1)+         ;PUT ABS DISK ADDRESS IN BUFFER
1678 035606  012021                           MOV     (R0)+,(R1)+         ;
1679 035610  011011                           MOV     (R0),(R1)           ;
1680 035612  012700  023614                   MOV     #RDBLK,R0           ;POINT R0 TO READ BLOCK ORIGIN
1681 035616  013710  001136                   MOV     CURSEC,(R0)         ;
1682 035622  053720  001140                   BIS     CURHD,(R0)+         ;FORMAT DUSH WORD
1683 035626  013701  001234                   MOV     CURDSK,R1           ;GET CURRENT DISK
1684 035632  006301                           ASL     R1                  ;MAKE VALID INDEX
1685 035634  016120  001200                   MOV     IBFTBL(R1),(R0)+    ;LOAD INPUT AREA
1686 035640  013720  001046                   MOV     WPSEC,(R0)+         ;SPECIFY ONE SECTOR READ
1687 035644  013710  001142                   MOV     CURCYL,(R0)         ;SET DISK CYLINDE
1688 035650  005740                           TST     -(R0)               ;POINT R0 TO WORD COUNT ENTRY
1689 035652  004537  022436                   JSR     R5,ENDPAR           ;COMPUTE ENDING DISK ADDRESS
1690 035656  012700  001162                   MOV     #ENDSEC,R0          ;
1691 035662  012701  044312                   MOV     #ODBUF,R1           ;POINT R1 TO BUFFER AREA
1692 035666  063701  001050                   ADD     POSWP,R1            ;ADJUST FOR SECOND SECTOR
1693 035672  063701  001050                   ADD     POSWP,R1            ;
1694 035676  012021                           MOV     (R0)+,(R1)+         ;PUT DISK ADDRESS IN BUFFER
1695 035700  012021                           MOV     (R0)+,(R1)+         ;
1696 035702  011011                           MOV     (R0),(R1)           ;
1697 035704  012700  023620                   MOV     #RDBLK+4,R0         ;POINT R0 TO WORD COUNT ENTRY
1698 035710  013710  001046                   MOV     WPSEC,(R0)          ;SET UP FOR TWO SECTOR READ
1699 035714  006320                           ASL     (R0)+               ;
1700 035716  004537  035000                   JSR     R5,DSKTST           ;REDUCE WORD COUNT BY ONE SECTOR IF APPLICABLE
1701 035722  012700  023614                   MOV     #RDBLK,R0           ;GET READ BLOCK ADDRESS
1702 035726  004537  022700                   JSR     R5,DXFER            ;GO READ TWO SECTORS
1703 035732  005737  001132                   TST     SOFTRY              ;SKIP THIS SECTOR?
1704 035736  001405                           BEQ     TN32A               ;NO GO ON
1705 035740  005037  001132                   CLR     SOFTRY              ;CLEAR RETRY FLAG
1706 035744  005237  001312                   INC     SKIPER              ;INCREMENT SKIP ON ERROR FLAG
1707 035750  000431                           BR      UPD                 ;READ NEXT CONSECUTIVE SECTOR
1708 035752  005737  001322          TN32A:   TST     ERRFLG              ;ERRORS?
1709 035756  001040                           BNE     TN32EX              ;EXIT IF YES
1710 035760  005204                           INC     R4                  ;SUBTEST NUMBER=1(READ OK)
```

```
035762  013700  001050                    MOV     POSWP,R0        ;LOAD WORD COUNT
035766  063700  001050                    ADD     POSWP,R0        ;
035772  005737  001144                    TST     DKEND           ;END OF DISK FLAG SET?
035776  001402                            BEQ     .+6             ;CONTINUE IF NO
036000  163700  001050                    SUB     POSWP,R0        ;REDUCE COMPARE COUNT IF END OF DISK
036004  004537  017106                    JSR     R5,DATCMP       ;GO COMPARE DATA
036010  005737  001332                    TST     RTRYIN          ;ARE WE IN RETRY MODE?
036014  001267                            BNE     TN32LP          ;YES REREAD THE SAME SECTOR
036016  005737  001322                    TST     ERRFLG          ;ANY ERRORS?
036022  001016                            BNE     TN32EX          ;EXIT  IF YES
036024  005004                            CLR     R4              ;REINITIALIZE SUBTEST NUMBER
036026  005737  001144                    TST     DKEND           ;END OF DISK?
036032  001012                            BNE     TN32EX          ;EXIT IF YES
036034  013737  001162  001136  UPD:      MOV     ENDSEC,CURSEC   ;UPDATE DISK ADDRESSES
036042  013737  001164  001140            MOV     ENDHD,CURHD     ;
036050  013737  001166  001142            MOV     ENDCYL,CURCYL   ;
036056  000646                            BR      TN32LP          ;REPEAT READ SEQUENCE
036060  004537  003342          TN32EX:   JSR     R5,TSTCTL       ;EXIT TO TEST CONTROL
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;       PATTERN GENERATE SUBROUTINE
                                          ;    ------------------------------------
                                          ;
                                          ;
                                          ;
                                          ;       THIS SUBROUTINE FORMATS THE OUTPUT DATA FOR
                                          ;  FOR AN EVER INCREASING BINARY COUNT OR AN EVER
                                          ;  DECREASING NEGATIVE COUNT DEPENDING ON THE
                                          ;  STATE OF CNTFLG.
                                          ;       THE BASE FROM WHICH THE COUNT PROGRESSES IS
                                          ;  DETERMINEDB BY THE CONTENTS OF R1 AT THE TIME OF
```

```
                                         ;  ENTRY.THE INCREMENTAL VALUE BETWEEN SUCCESSIVE
                                         ;  LOCATIONS IS DETERMINED BY CNTINC.IF CNTINC
                                         ;  IS EQUAL TO 0  THE COUNT IS INCREASING ,WHILEIT
                                         ;  WILL BE DECREASING IF CNTINC IS EQUAL TO
                                         ;  1.
                                         ;      THE CONTENTS  OF THE PASS COUNTER IS ALSO ADDED
                                         ;   TO THE INITIAL VALUE IN R1 TO CHANGE ALL OF THE
                                         ;  DATA PATTERNS.       ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
 036064  010046                  PATGEN: MOV     R0,-(SP)        ;SAVE REGISTERS R0,R1,R2
 036066  010146                          MOV     R1,-(SP)        ;
 036070  010246                          MOV     R2,-(SP)        ;
 036072  063701  001236                  ADD     PASCNT,R1       ;CHANGE DATA PATTERN BIAS
 036076  005737  036140                  TST     CNTFLG          ;IS PATTERN NEGATIVE
 036102  001401                          BEQ     .+4             ;
 036104  005401                          NEG     R1              ;COMPLEMENT BASE VALUE IF NO
 036106  012700  044312                  MOV     #ODBUF,R0       ;GET OUTPUT BUFFER ORIGIN ADDRESS
 036112  012702  001000                  MOV     #512.,R2        ;ESTABLISH LOOP COUNT
 036116  010120                          MOV     R1,(R0)+        ;FORMAT ODBUF LOCATION
 036120  063701  036142                  ADD     CNTINC,R1       ;COMPUTE NEXT DATA WORD
 036124  005302                          DEC     R2              ;DECREMENT LOOP COUNTER
 036126  100373                          BPL     .-8.            ;REPEAT IF NOT DONE
 036130  012602                          MOV     (SP)+,R2                ;RESTORE REGISTERS
 036132  012601                          MOV     (SP)+,R1        ;
 036134  012600                          MOV     (SP)+,R0        ;
 036136  000205                          RTS     R5              ;
                                                                 ;EXIT TO CALLING PROGRAM
                                         ;
 036140  000000                  CNTFLG: .WORD 0
 036142  000001                  CNTINC: .WORD 1         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
```

```
                                                  ;
                                                  ;
                                                  ;       TEST 33          MULTIPLE CPU EXERCISE
                                                  ;
                                                  ;
                                                  ;
                                                  ;       THIS TEST IS AN EXERCISE FOR PHOENIX 211
                                                  ;   SYSTEMS THAT ARE OPERATING WITH THE MULTIPLE CPU
                                                  ;   OPTION.THIS IS THE ONLY 211 DIAGNOSTIC TEST THAT
                                                  ;   WILL ALLOW MORE THAN ONE CPU TO BE REQUESTING THE
                                                  ;   211.THE FORMAT OF THE THE TEST IS AS FOLLOWS:
                                                  ;
                                                  ;       1. THE FORMATTER IS REQUESTED.IF THIS CPU
                                                  ;          HAS ACCESS BEFORE THE REQUEST IS MADE THEN
                                                  ;          AN ERROR IS LOGGED.
                                                  ;
                                                  ;       2. A WRITE/READ AND DATA COMPARE ROUTINE IS
                                                  ;          STARTED AND NORMAL SUBROUTINES ARE USED
                                                  ;          TO DRIVE THIS SEQUENCE.
                                                  ;
                                                  ;       3. IF STEP #2 IS SUCESSFUL THE PROGRAM RELEASES
                                                  ;          FORMATTER AND IF ANOTHER CPU HAS A
                                                  ;          REQUEST SET THEN IT SHOULD RECEIVE ACCESS.
                                                  ;
                                                  ;       4. A COUNTER OF 1000. TRANSFERS IS DECREMENTED AND
                                                  ;          IF NOT DONE STEPS 1 THRU 3 ARE REPEATED.
                                                  ;
                                                  ;       ;      *** THIS TEST CAN BE RUN IF A CONTROLLER DOES NOT
                                                  ;          HAVE A MULTIPLE CPU SETUP,STEPS 1 AND 3
                                                  ;          ARE OMITTED.
                                                  ;
                                                  ;       THE DISK ADDRESS AND THE DATA PATTERN ARE REANDOM.
                                                  ;
                                                  ;       THE UNITS USED ARE ANY AND ALL THAT ARE ON LINE
                                                  ;       AND READY.(CHOSEN RANDOMLY ALSO)
036144  012737  001750  001272    TN33:     MOV       #1000.,LPCNT       ;SET TRANSFER COUNT TO 1000.
036152  004537  015564            $2:       JSR       R5,RANDOM          ;GENERATE RANDOM NUMBER
036156  004537  015524                      JSR       R5,RANBUF          ;FORMAT DATA BUFFER
036162  005737  001076                      TST       MULCPU             ;MULTIPLE CPU OPTION?
036166  001412                              BEQ       $4                 ;USE NORMAL DATA TRANSFER IF YES
036170  004537  036244            $222:     JSR       R5,REQUST          ;REQUEST AND GO
036174  005737  001322                      TST       ERRFLG             ;ERRORS?
036200  001003                              BNE       TN33EX             ;EXIT IF ERROR
036202  005337  001272            $3:       DEC       LPCNT              ;DECREMENT LOOP COUNT
036206  001361                              BNE       $2                 ;CONTINUE IF NOT DONE
036210  004537  003342            TN33EX:   JSR       R5,TSTCTL                  ;EXIT
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;
                                  ;****************************************************************
                                  ;****************************************************************
                                            ;
036214  012700  035520            $4:       MOV       #WRCB,R0           ;
```

```
036220  004537  016130                   JSR     R5,RANBLK        ;SET UP DISK ADDRESSES
036224  004537  035356                   JSR     R5,WRC           ;GO WRITE/READ
036230  005737  001322                   TST     ERRFLG           ;ERROR?
036234  001762                           BEQ     $3               ;CONTINUE IF NO
036236  013704  001152                   MOV     DTCNT,R4         ;FOCRE DXFER SUBCOUNT NUMBER
036242  000762                           BR      TN33EX           ;AND EXIT
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;       CONTROLLER REQUEST ROUTINE
                                         ;   ***********************************
                                         ;
                                         ;
                                         ;             THIS ROUTINE REQUESTS ACCESS TO THE
                                         ;       211 CONTROLLER AND WHEN COMMAND IS RECEIVED,
                                         ;       ISSUES A WRITE AND READ SEQUENCE.THESE COMMANDS
                                         ;       ARE ALL INTERRUPT DRIVENAND ERRORS ARE
                                         ;       CHECKED.
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;
                                         ;

036244  010046                   REQUST: MOV     R0,-(SP)         ;SAVE R0
036246  005004                           CLR     R4               ;RESET SUBTEST COUNTER
036250  005037  001150                   CLR     INTFLG           ;RESET INTERRUPT
036254  105777  142520                   TSTB    @DCSR            ;DO WE HAVE ACCESS NOW?
036260  100006                           BPL     $R1              ;CONTINUE IF NO
036262  005237  001322                   INC     ERRFLG           ;SET ERROR FLAG IF YES
036266  017703  142506                   MOV     @DCSR,R3         ;LOAD R3 WITH ERROR CONDITION
036272  005002                           CLR     R2               ;SET R2 FOR REFERENCE WORD
036274  000205                           RTS     R5               ;AND EXIT
036276  005204                   $R1:    INC     R4               ;SUBTEST NUMBER-1(FORMATTER DEAD)
036300  052777  000100  142472           BIS     #INTON,@DCSR     ;ENABLE INTERRUPTS AND REQUEST
036306  052777  000040  142464           BIS     #BIT5,@DCSR      ;REQUEST CONTROLLER
036314  004537  017342                   JSR     R5,WAIT          ;WAIT FOR INTERRUPT
036320  005737  001150                   TST     INTFLG           ;DID WE INTERRUPT?(COULD BE 6 SECONDS)
036324  001003                           BNE     $R2              ;CONTINUE IF YES
036326  005237  001322                   INC     ERRFLG           ;SET ERROR FLAG IF NO
036332  000205                           RTS     R5               ;AND EXIT
036334  005204                   $R2:    INC     R4               ;SUBTEST NUMBER=2(WE DID INTERRUPT)
036336  005037  001150                   CLR     INTFLG           ;RESET INTERRUPT FLAG
036342  017703  142432                   MOV     @DCSR,R3         ;SEE IF READY IS SET
036346  012702  000200                   MOV     #FMTRDY,R2       ;SET UP R2 WITH REFERENCE WORD
036352  052702  000100                   BIS     #INTON,R2        ;INCLUDE INTERRUPT ENABLE
036356  052702  000040                   BIS     #BIT5,R2         ;ALSO REQUEST BIT
036362  030203                           BIT     R2,R3            ;ARE THESE BITS SET?
```

```
036364  001003                          BNE     SR3             ;CONTINUE IF YES
036366  005237  001322                  INC     ERRFLG          ;SET ERROR FLAG IF NO
036372  000205                          RTS     R5              ;AND EXIT
036374  005204                  SR3:    INC     R4              ;SUBTEST NUMBER=3(READY SET,INTERRUPT
                                                                ;ENABLE SET AND NO ERRORS.
036376  012700  035520                  MOV     #WRCB,R0        ;POINT R0 TO WRITE/READ BLOCK
036402  004537  016042                  JSR     R5,FIUNIT       ;SEARCH FOR ACTIVE DISK DRIVES
036406  004537  016130                  JSR     R5,RANBLK       ;SET UP DISK ADDRESSES
036412  004537  035356                  JSR     R5,WRC          ;GO DO SOMETHING
036416  005737  001322                  TST     ERRFLG          ;ERRORS?
036422  001001                          BNE     SR4             ;CLEAR REQUEST IF YES
036424  005204                          INC     R4              ;SUBTEST NUMBER=4(NO ERRORS)
036426  042777  000040  142344  SR4:    BIC     #BIT5,@DCSR     ;LET GO OF CONTROLLER
036434  042777  000100  142336          BIC     #INTON,@DCSR    ;DISABLE INTERRUPTS
036442  017703  142332                  MOV     @DCSR,R3        ;FETCH DCSR
036446  005002                          CLR     R2              ;SET REFERENCE WORD=0
036450  020203                          CMP     R2,R3           ;BITS CLEARED?
036452  001402                          BEQ     SR5             ;EXIT IF YES
036454  005237  001322                  INC     ERRFLG          ;SET ERROR FLAG IF NO
036460  012600                  SR5:    MOV     (SP)+,R0        ;RESTORE R0
036462  000205                          RTS     R5              ;AND EXIT
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       10M.ECC TEST NUMBERS A16 >> A17 13-NOV-77
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       TNA16.PM        -NOV-77
                                        ;
                                        ;
                                        ;
                                        ;       ECC REGISTER CHECKS
                                        ;  --------------------------
                                        ;
                                        ;
                                        ;
                                        ;       THESE TESTS CHECK THE ECC REGISTER (PATTERN)
                                        ;  FOR LOADING THE ECC INHIBIT BIT AS WELL AS INITIALIZATION
                                        ;  OF BOTH THE ECC PATTERN REGISTER AND THE ECC BIT COUNT
                                        ;  REGISTER.
                                        ;
                                        ;
                                        ;
                                        ;
036464  005004                  TNA16:  CLR     R4              ;SUBTEST NUMBER=0
```

```
172 036466 004537 004276                 JSR    R5,RQUEST          ;REQUEST USE OF THE 211
173 036472 000005                        RESET                     ;ISSUE I/O RESET
174 036474 004537 004276                 JSR    R5,RQUEST          ;FETCH 211 AGAIN
175 036500 005002                        CLR    R2                 ;SET UP REFERENCE WORD
176 036502 017703 142310                 MOV    @ECCPB,R3          ;FETCH ECC POSITION REGISTER
177 036506 020203                        CMP    R2,R3              ;CLEARED?
178 036510 001404                        BEQ    CECC1              ;CONTINUE IF YES
179 036512 005237 001322                 INC    ERRFLG             ;SET ERROR FLAG IF NO
180 036516 000137 036770                 JMP    TNA16EX            ;AND EXIT
181 036522 005204                CECC1:  INC    R4                 ;SUBTEST NUMBER=1(ECCPB OK)
182 036524 017703 142270                 MOV    @ECCPW,R3          ;FETCH ECC BIT PATTERN REGISTER
183 036530 020203                        CMP    R2,R3              ;CLEARED?
184 036532 001404                        BEQ    CECC2              ;CONTINUE IF YES
185 036534 005237 001322                 INC    ERRFLG             ;SET ERROR FLAG IF NO
186 036540 000137 036770                 JMP    TNA16EX            ;AND EXIT
187 036544 005204                CECC2:  INC    R4                 ;SUBTEST NUMBER=(ECC BIT PATTERN OK)
188 036546 012777 100000 142244          MOV    #BIT15,@ECCPW      ;LOAD ECC INHIBIT BIT
189 036554 012702 100000                 MOV    #BIT15,R2          ;LOAD REFERENCE WORD
190 036560 017703 142234                 MOV    @ECCPW,R3          ;FETCH ECCPW>>R3
191 036564 020203                        CMP    R2,R3              ;BIT SEYT?
192 036566 001404                        BEQ    CECC3              ;CONTINUE IF YES
193 036570 005237 001322                 INC    ERRFLG             ;SET ERROR FLAG IF NO
194 036574 000137 036770                 JMP    TNA16EX            ;AND EXIT
195 036600 005204                CECC3:  INC    R4                 ;SUBTEST NUMBER=3(BIT 15 OK,ECCPW)
196 036602 005002                        CLR    R2                 ;SET UP REFERENCE WORD
197 036604 005077 142210                 CLR    @ECCPW             ;ATTEMPT TO CLEAR BIT 15
198 036610 017703 142204                 MOV    @ECCPW,R3          ;FETCH FOR COMPARE
199 036614 020203                        CMP    R2,R3              ;BIT CLEARED?
200 036616 001404                        BEQ    CECC4              ;CONTINUE IF YES
201 036620 005237 001322                 INC    ERRFLG             ;SET ERROR FLAG IF NO
202 036624 000137 036770                 JMP    TNA16EX            ;EXIT
203 036630 005204                CECC4:  INC    R4                 ;SUBTEST NUMBER=4(BIT 15 CLEARED)
204 036632 012777 100000 142160          MOV    #BIT15,@ECCPW      ;RELOAD BIT 15
205 036640 052777 000077 142132          BIS    #NOFCMD,@DCSR      ;ISSUE NOP COMMAND TO CLEAR BIT
206 036646 017703 142146                 MOV    @ECCPW,R3          ;SET UP FOR COMPARE
207 036652 012702 100000                 MOV    #BIT15,R2          ;SET UP REFERENCE IN R2
208 036656 020203                        CMP    R2,R3              ;BIT NO-OPPED OUT?
209 036660 001404                        BEQ    CECC5              ;CONTINUE IF NO
210 036662 005237 001322                 INC    ERRFLG             ;SET ERRROR FLAG IF NO
211 036666 000137 036770                 JMP    TNA16EX            ;AND EXIT
212 036672 005204                CECC5:  INC    R4                 ;SUBTEST NUMBER=5(GO DOESN'T CLEAR ERROR)
213 036674 012777 100000 142116          MOV    #BIT15,@ECCPW      ;LOAD BIT 15 AGAIN
214 036702 012777 000041 142070          MOV    #SYSCLR!BIT5,@DCSR;ISSUE SYSTEM CLEAR
215 036710 004537 004276                 JSR    R5,RQUEST          ;REQUEST 211 AGAIN
216 036714 017703 142100                 MOV    @ECCPW,R3          ;SET UP FOR COMPARE
217 036720 005002                        CLR    R2                 ;SET UP REFERENCE WORD
218 036722 020203                        CMP    R2,R3              ;CLEARED?
219 036724 001404                        BEQ    CECC6              ;CONTINUE IF YES
220 036726 005237 001322                 INC    ERRFLG             ;SET ERROR FLAG IF NO
221 036732 000137 036770                 JMP    TNA16EX            ;AND EXIT
222 036736 005204                CECC6:  INC    R4                 ;SUBTEST NUMBER=6(SYSCLR CLEARS ERROR BIT)
223 036740 012777 100000 142052          MOV    #BIT15,@ECCPW      ;LOAD BIT 15 ONE LAST TIME
224 036746 000005                        RESET                     ;ISSUE I/O RESET
225 036750 004537 004276                 JSR    R5,RQUEST          ;REQUEST FORMATTER AGAIN
226 036754 017703 142040                 MOV    @ECCPW,R3          ;SET UP FOR COMPARE
227 036760 020203                        CMP    R2,R3              ;CLEARED?
228 036762 001402                        BEQ    TNA16EX            ;EXIT IF YES
```

```
036764  005237  001322            INC     ERRFLG          ;SET ERROR FLAG IF NO
036770  004537  003342  TNA16EX:JSR     R5,TSTCTL       ;AND EXIT EITHER WAY
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;       TNA17.PM        1-NOV-77
                                ;
                                ;       ECC/211 HEADER FORMAT
                                ;    ***************************
                                ;
                                ;
                                ;       THE FOLLOWING DRAWING IS THE HEADER ARRANGEMENT
                                ;   FOR THE PHOENIX 211 CONTROLLER EQUIPTED WITH ECC LOGIC
                                ;       THESE CONTROLLERS HAVE A REDUNDANT SET OF HEADER WORDS
                                ;    ADDED TO THE SECTOR TO INSURE PERFECT DISK PACKS
                                ;       IF THERE IS A BAD SPOT(S) ON A DISK IN THE HEADER FIELD AREA
                                ;       THE CONTROLLER WILL ERROR ONLY IF BOTH HEADERS ARE DETECTED
                                ;   AS BAD.
                                ;
                                ;
                                ;
                                ;
                                ;       *****************************************************************
                                ;       *       *       *       *       *       *       *       *       *
                                ;       * WORD1 * WORD2 * CRC1  * CRC2  * WORD3 * WORD4 * CRC1  * CRC2  *
                                ;       * PAIR1 * PAIR1 * PAIR1 * PAIR1 * PAIR2 * PAIR2 * PAIR2 * PAIR2 *
                                ;       *       *       *       *       *       *       *       *       *
                                ;       *****************************************************************
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;       ECC/CRC DETECTION TEST
                                ;   ----------------------------
                                ;
                                ;       THESE TESTS CHECK THE HEADER CRC
                                ;   DETECTION AS WELL AS THE COMPATIBILITY
                                ;   OF THE CONTROLLER UNDER TEST.
```

```
    286                                        ;
    287                                        ;
    288 036774  005004                 TNA17:  CLR     R4              ;SUBTEST NUMBER=0
    289 036776  012700  040620                 MOV     #ECCWRT,R0      ;POINT TO WRITE BLOCK
    290 037002  004537  004276                 JSR     R5,RQUEST       ;REQUEST FORMATTER
    291 037006  005010                         CLR     (R0)            ;MAKE DUSH MIN
    292 037010  105777  141764                 TSTB    @DCSR           ;FORMATTER READY?
    293 037014  100404                         BMI     CRC1            ;CONTINUE IF YES
    294 037016  005237  001322                 INC     ERRFLG          ;
    295 037022  000137  040614                 JMP     CRCXEX          ;EXIT IF NO
    296 037026  005204                 CRC1:   INC     R4              ;SUBTEST NUMBER=1(FORMATTER READY)
    297 037030  004537  004034                 JSR     R5,CLRBUF       ;ZERO BUFFERS
    298 037034  016001  000002                 MOV     2(R0),R1        ;POINT R1 TO BUFFER
    299 037040  012761  000001  000004         MOV     #BIT0,4(R1)     ;ENTER SINGLE BITS INTO HEADERS 1 AND 2
    300 037046  012761  000001  000014         MOV     #BIT0,14(R1)    ;
    301 037054  005237  001176                 INC     SPECMD          ;SET SPECIAL COMMAND FLAG
    302 037060  004537  022700                 JSR     R5,DXFER        ;WRITE BAD HDR CRC
    303 037064  005037  001176                 CLR     SPECMD          ;RESET SPECIAL COMMAND FLAG
    304 037070  005737  001322                 TST     ERRFLG          ;ERRORS?
    305 037074  001402                         BEQ     CRC2            ;CONTINUE IF NO
    306 037076  000137  040614                 JMP     CRCXEX          ;EXIT IF YES
    307 037102  005204                 CRC2:   INC     R4              ;SUBTEST NUMBER=2(REFORMAT OK)
    308 037104  012700  023614                 MOV     #RDBLK,R0       ;POINT R0 TO READ BLOCK
    309 037110  005010                         CLR     (R0)            ;DUSH=0
    310 037112  005060  000006                 CLR     6(R0)           ;CYLINDER=0
    311 037116  013760  001046  000004         MOV     WPSEC,4(R0)     ;ONE SECTOR WORD COUNT
    312 037124  004537  022700                 JSR     R5,DXFER        ;GO READ
    313 037130  005737  001322                 TST     ERRFLG          ;ERRORS(SHOULD BE)
    314 037134  001004                         BNE     CRC3            ;CONTINUE IF YES
    315 037136  005237  001322                 INC     ERRFLG          ;SET ERROR FLAG IF NOT
    316 037142  000137  040614                 JMP     CRCXEX          ;AND EXIT
    317 037146  005204                 CRC3:   INC     R4              ;SUBTEST NUMBER=3(ERROR DETECTED)
    318 037150  017703  141640                 MOV     @DERR,R3        ;FETCH ERROR REGISTER
    319 037154  042703  177376                 BIC     #177376,R3      ;LOOSE ALL ERRORS EXCEPT HDR CRC
    320 037160  012702  000401                 MOV     #BIT8!BIT0,R2   ;SET UP REFERENCE
    321 037164  020203                         CMP     R2,R3           ;CRC ERROR SET?
    322 037166  001402                         BEQ     CRC4            ;CONTINUE IF YES
    323 037170  000137  040614                 JMP     CRCXEX          ;EXIT IF NOT
    324 037174  005204                 CRC4:   INC     R4              ;CRC ERROR DETECTED
    325 037176  017703  141612                 MOV     @DERR,R3        ;FETCH ERROR AGAIN
    326 037202  040203                         BIC     R2,R3           ;LOOSE CRC ERROR
    327 037204  005002                         CLR     R2              ;SET R2 FOR REFERENCE
    328 037206  020203                         CMP     R2,R3           ;ANY OTHER ERRORS SET?(SHOULDN'T BE)
    329 037210  001402                         BEQ     CRC5            ;CONTINUE IF NO
    330 037212  000137  040614                 JMP     CRCXEX          ;EXIT IF YES
    331 037216  005204                 CRC5:   INC     R4              ;SUBTEST NUMBER=5(NO OTHER ERRORS)
    332 037220  017703  141562                 MOV     @DWCNT,R3       ;FETCH WORD COUNT REGISTER
    333 037224  016002  000004                 MOV     4(R0),R2        ;FETCH ORIGINAL WORD COUNT REG.VALUE
    334 037230  020203                         CMP     R2,R3           ;SEE IF WORD COUNT MOVED
    335 037232  001402                         BEQ     CRC6            ;CONTINUE IF NO
    336 037234  000137  040614                 JMP     CRCXEX          ;EXIT IF YES
    337 037240  005204                 CRC6:   INC     R4              ;SUBTEST NUMBER=6(NO WORD COUNT INCREMENTATION)
    338 037242  017703  141536                 MOV     @DCAR,R3        ;FETCH CORE ADDRESS REGISTER
    339 037246  016002  000002                 MOV     2(R0),R2        ;FETCH ORIGINAL CORE ADDRESS VALUE
    340 037252  020203                         CMP     R2,R3           ;CORE ADDRESS MOVE?
    341 037254  001402                         BEQ     CRC7            ;CONTINUE IF NO
    342 037256  000137  040614                 JMP     CRCXEX          ;EXIT IF YES
```

```
343 037262 005204                 CRC7:   INC     R4              ;SUBTEST NUMBER=7(BUSS ADDR.-NO INCREMENTATION)
344 037264 017703 141512                  MOV     @DUSH,R3        ;FETCH UNIT-SECTOR-HEAD REGISTER
345 037270 011002                         MOV     (R0),R2         ;FETCH ORIGINAL VALUE
346 037272 042703 030000                  BIC     #30000,R3       ;LOOSE UNIT SELECT BITS IN BOTH
347 037276 042702 030000                  BIC     #30000,R2       ;
348 037302 005202                         INC     R2              ;MAKE R2 REFERENCE(ORIG+1)
349 037304 020203                         CMP     R2,R3           ;DID SECTOR ADDR BUMP?(SHOULD HAVE)
350 037306 001402                         BEQ     CRC10           ;CONTINUE IF YES
351 037310 000137 040614                  JMP     CRCXEX          ;EXIT IF NO
352 037314 005204                 CRC10:  INC     R4              ;SUBTEST=10(SECTOR ADDR =.+1)
353 037316 005037 001322                  CLR     ERRFLG          ;FINALLY RESET ERROR FLAG
354 037322 012700 025450                  MOV     #ISECFB,R0      ;POINT R0 TO FORMAT BLOCK
355 037326 005010                         CLR     (R0)            ;SECTOR,HEAD 0
356 037330 005037 044306                  CLR     ODBUF-4         ;MAKE VALID HEADER WORDS
357 037334 005037 044310                  CLR     ODBUF-2         ;
358 037340 005237 001176                  INC     SPECMD          ;SET SPECIAL COMMAND FLAG
359 037344 004537 022700                  JSR     R5,DXFER        ;REFORMAT SECTOR 0
360 037350 005037 001176                  CLR     SPECMD          ;RESET SPECIAL COMMAND FLAG
361 037354 005737 001322                  TST     ERRFLG          ;ERRORS DURING FORMAT
362 037360 001402                         BEQ     CRC11           ;CONTINUE IF NO
363 037362 000137 040614                  JMP     CRCXEX          ;EXIT IF YES
364 037366 005204                 CRC11:  INC     R4              ;SUBTEST NUMBER=11(REFORMAT SECTOR 0 OK)
365 037370 005210                         INC     (R0)            ;POINT TO SECTOR 1
366 037372 004537 004034                  JSR     R5,CLRBUF       ;CLEAR BUFFER AREA
367 037376 012737 000001 044306           MOV     #1,ODBUF-4      ;MAKE VALID HEADER WORD 1
368 037404 012737 000001 044312           MOV     #1,ODBUF        ;MAKE SECOND PAIR VALID
369 037412 005037 044310                  CLR     ODBUF-2         ;
370 037416 005237 001176                  INC     SPECMD          ;SET SPECIAL COMMAND FLAG
371 037422 004537 022700                  JSR     R5,DXFER        ;REFORMAT SECTOR
372 037426 005037 001176                  CLR     SPECMD          ;RESET SPECIAL COMMAND FLAG
373 037432 005010                         CLR     (R0)            ;SECTOR ADDR BACK TO ZERO
374 037434 005737 001322                  TST     ERRFLG          ;ERROR?
375 037440 001402                         BEQ     CRC12           ;CONTINUE IF NO
376 037442 000137 040614                  JMP     CRCXEX          ;EXIT IF YES
377 037446 005204                 CRC12:  INC     R4              ;SUBTEST NUMBER=12(FORMAT SECTOR 1 OK)
378 037450 012700 023614                  MOV     #RDBLK,R0       ;POINT TO READ BLOCK
379 037454 005210                         INC     (R0)            ;MAKE READ FROM SECTOR 0
380 037456 004537 022700                  JSR     R5,DXFER        ;READ
381 037462 005010                         CLR     (R0)            ;RESET SECTOR ADDRESS
382 037464 005737 001322                  TST     ERRFLG          ;ERRORS DURING READ?
383 037470 001402                         BEQ     CRC13           ;CONTINUE IF NO
384 037472 000137 040614                  JMP     CRCXEX          ;EXIT IF YES
385 037476 005204                 CRC13:  INC     R4              ;SUBTEST NUMBER=13(READ ON SECTOR 1 OK)
386 037500 012700 040632                  MOV     #RDCDE,R0       ;POINT TO READ EVERYTHING BLOCK
387 037504 005210                         INC     (R0)            ;MAKE IT SECTOR 1
388 037506 013702 001234                  MOV     CURDSK,R2       ;SET UP INPUT FIELD
389 037512 006302                         ASL     R2              ;
390 037514 016260 001200 000002           MOV     IBFTBL(R2),2(R0);FOR READ
391 037522 013760 001046 000004           MOV     WPSEC,4(R0)     ;ADJUST WORD COUNT
392 037530 162760 000012 000004           SUB     #10.,4(R0)      ;ADJUST WORD COUNT
393 037536 005237 001176                  INC     SPECMD          ;SET COMMAND FLAG
394 037542 004537 022700                  JSR     R5,DXFER        ;READ BALL GAME
395 037546 005010                         CLR     (R0)            ;RESET SECTOR ADDRESS
396 037550 005037 001176                  CLR     SPECMD          ;RESET COMMAND FLAG
397 037554 005737 001322                  TST     ERRFLG          ;ERRORS?
398 037560 001402                         BEQ     CRC14           ;CONTINUE IF NO
399 037562 000137 040614                  JMP     CRCXEX          ;EXIT IF YES
```

```
037566  005204                          CRC14:  INC     R4              ;SUBTEST NUMBER=14(READ CRC OK)
037570  016002  000002                          MOV     2(R0),R2        ;SET R2 FOR POINTER TO CRC WORD(S)
037574  016203  000004                          MOV     4(R2),R3        ;FETCH FIRST CRC WORD
037600  013702  001260                          MOV     HDRC1,R2        ;FETCH REFERENCE WORD
037604  020203                                  CMP     R2,R3           ;COMPATIBLE CRC GENERATION?
037606  001404                                  BEQ     CRC15           ;CONTINUE IF YES
037610  005237  001322                          INC     ERRFLG          ;SET ERROR FLAG IF NO
037614  000137  040614                          JMP     CRCXEX          ;AND EXIT
037620  005204                          CRC15:  INC     R4              ;SUBTEST NUMBER=15(FIRST CRC WORD OK)
037622  016001  000002                          MOV     2(R0),R1        ;COMPUTE ADDRESS FOR SECONC CRC WORD
037626  016103  000006                          MOV     6(R1),R3        ;LOAD SECOND CRC WORD
037632  013702  001262                          MOV     HDRC2,R2        ;FETCH REFERENCE WORD
037636  020203                                  CMP     R2,R3           ;NEW WORD COMPATIBLE?
037640  001404                                  BEQ     CRC16           ;CONTINUE IF YES
037642  005237  001322                          INC     ERRFLG          ;EXIT IF NO
037646  000137  040614                          JMP     CRCXEX          ;EXIT WITH ERROR
037652  005204                          CRC16:  INC     R4              ;SUBTEST NUMBER=16(SECOND CRC WORD COMPATIBLE
                                                                        ;THESE LAST TWO SUBTESTS(15+16)
                                                                        ;ARE USED TO DETERMINE IF THE CRC/ECC
                                                                        ;GENERATOR ON THE FORMATTER BEING
                                                                        ;TESTED IS COMPATIBLE WITH ALL OTHER
                                                                        ;OTHER PHOENIX 211/200 CONTROLLERS.
037654  016102  000004                          MOV     4(R1),R2        ;NOW SETUP TO CHECK SECOND PAIR OF HDR CRC'S
037660  016103  000014                          MOV     14(R1),R3       ;FOR COMPATIBILITY
037664  020203                                  CMP     R2,R3           ;SAME VALUE?
037666  001404                                  BEQ     CRC17           ;CONTINUE IF YES
037670  005237  001322                          INC     ERRFLG          ;SET ERROR FLAG IF NO
037674  000137  040614                          JMP     CRCXEX          ;AND EXIT
037700  005204                          CRC17:  INC     R4              ;SUBTEST NUMBER=17(FIRST WORD COMPATIBE)
037702  016102  000006                          MOV     6(R1),R2        ;FETCH SECOND CRC WORD
037706  016103  000016                          MOV     16(R1),R3       ;FOR SAME
037712  020203                                  CMP     R2,R3           ;SECOND WORD OK?
037714  001404                                  BEQ     CRC20           ;CONTINUE IF YES
037716  005237  001322                          INC     ERRFLG          ;SET ERROR FLAG IF NO
037722  000137  040614                          JMP     CRCXEX          ;AND EXIT
037726  005204                          CRC20:  INC     R4              ;SUBTEST NUMBER=20(SECOND WORD OK)
037730  012700  040620                          MOV     #ECCWRT,R0      ;POINT TO WRITE BLOCK
037734  005010                                  CLR     (R0)            ;SECTOR ADDRR.=0
037736  016001  000002                          MOV     2(R0),R1        ;POINT RO BUFFER AREA
037742  004537  004034                          JSR     R5,CLRBUF       ;ZERO BUFFERS
037746  012761  100000  000004                  MOV     #BIT15,4(R1)    ;MAKE LAST BIT OF FIRST CRC WORDS A 1
037754  012761  100000  000014                  MOV     #BIT15,14(R1)   ;BOTH MUST BE SET TO CAUSE AN ERROR
037762  013760  001046  000004                  MOV     WPSEC,4(R0)     ;ADJUST WORD COUNT FOR +ONE SECTOR
037770  162760  000012  000004                  SUB     #10.,4(R0)      ;INCLUDE HDRS+CRC+ECC WORDS
037776  005237  001176                          INC     SPECMD          ;SET SPECIAL COMMAND FLAG
040002  004537  022700                          JSR     R5,DXFER        ;WRITE NEW FIELD
040006  005037  001176                          CLR     SPECMD          ;CLEAR SPECIAL COMMAND FLAG
040012  005737  001322                          TST     ERRFLG          ;ERRORS?
040016  001402                                  BEQ     CRC21           ;CONTINUE IF NO
040020  000137  040614                          JMP     CRCXEX          ;EXIT IF YES
040024  005204                          CRC21:  INC     R4              ;SUBTEST  NUMBER=21(COMPATIBILE)
040026  012700  023614                          MOV     #RDBLK,R0       ;POINT TO READ BLOCK
040032  004537  022700                          JSR     R5,DXFER        ;ATTEMPT READ ON SECTOR 0
040036  005737  001322                          TST     ERRFLG          ;ERRORS?(SHOULD BE)
040042  001004                                  BNE     CRC22           ;CONTINUE IF YES
040044  005237  001322                          INC     ERRFLG          ;SET ERROR FLAG IF NO
040050  000137  040614                          JMP     CRCXEX          ;AND EXIT
```

```
040054  005204                   CRC22:  INC     R4              ;SUBTEST NUMBER=22(ERROR DETECTED)
040056  017703  140732                   MOV     @DERR,R3        ;FETCH ERROR REGISTER
040062  042703  177376                   BIC     #177376,R3      ;EXTRACT HDR CRC ERROR
040066  012702  000401                   MOV     #BIT8!BIT0,R2   ;SET UP REFERENCE WORD
040072  020203                           CMP     R2,R3           ;ARE THE EQUAL?
040074  001402                           BEQ     CRC23           ;CONTINUE IF YES
040076  000137  040614                   JMP     CRCXEX          ;EXIT IF NO
040102  005204                   CRC23:  INC     R4              ;SUBTEST NUMBER=23(CRCERROR DETECTED)
040104  005037  001322                   CLR     ERRFLG          ;RESET ERROR FLAG
040110  004537  004034                   JSR     R5,CLRBUF       ;ZERO BUFFERS
040114  012700  040620                   MOV     #ECCWRT,R0      ;POINT TO WRITE BLOCK
040120  016001  000002                   MOV     2(R0),R1        ;USE R1 TO POINT TO BUFFERS
040124  012761  000001  000006           MOV     #BIT0,6(R1)     ;MAKE FIRST BIT OF SECOND WORDS A 1
040132  012761  000001  000016           MOV     #BIT0,16(R1)    ;BOTH WORDS AGAIN
040140  005237  001176                   INC     SPECMD          ;SET SPECIAL COMMAND FLAG
040144  004537  022700                   JSR     R5,DXFER        ;REFORMAT
040150  005037  001176                   CLR     SPECMD          ;RESET COMMAND FLAG
040154  005737  001322                   TST     ERRFLG          ;ERRORS?
040160  001402                           BEQ     CRC24           ;CONTINUE IF NO
040162  000137  040614                   JMP     CRCXEX          ;EXIT IF ERROR
040166  005204                   CRC24:  INC     R4              ;SUBTEST NUMBER=24(REWRITE OK)
040170  012700  023614                   MOV     #RDBLK,R0       ;POINT TO READ BLOCK ADDRESS
040174  004537  022700                   JSR     R5,DXFER        ;READ SECTOR
040200  005737  001322                   TST     ERRFLG          ;ERROR?
040204  001004                           BNE     CRC25           ;CONTINUE IF YES
040206  005237  001322                   INC     ERRFLG          ;SET IT IF NOT
040212  000137  040614                   JMP     CRCXEX          ;AND EXIT
040216  005204                   CRC25:  INC     R4              ;SUBTEST NUMBER=25(ERROR DETECTED)
040220  017703  140570                   MOV     @DERR,R3        ;FETCH ERROR REGISTER
040224  012702  000401                   MOV     #BIT8!BIT0,R2   ;SET UP REFERENCE WORD IN R2
040230  020203                           CMP     R2,R3           ;CRC ERROR?
040232  001402                           BEQ     CRC26           ;CONTINUE IF YES
040234  000137  040614                   JMP     CRCXEX          ;EXIT IF NOT
040240  005204                   CRC26:  INC     R4              ;SUBTEST NUMBER=26(CRC DETECTED)
040242  005037  001322                   CLR     ERRFLG          ;RESET ERROR FLAG NOW
040246  012700  040620                   MOV     #ECCWRT,R0      ;FETCH WRITE BLOCK ADD AGAIN
040252  016001  000002                   MOV     2(R0),R1        ;USE R1 TO POINT TO WRITE BUFFER
040256  005061  000004                   CLR     4(R1)           ;CLEAR FIRST HEADER(VALIDATE)
040262  005061  000014                   CLR     14(R1)          ;BOTH SETS
040266  012761  100000  000006           MOV     #BIT15,6(R1)    ;MAKE LAST BIT OF CRC A 1
040274  012761  100000  000016           MOV     #BIT15,16(R1)   ;BOTH SETS
040302  005237  001176                   INC     SPECMD          ;SET SPECIAL COMMAND FLAG
040306  004537  022700                   JSR     R5,DXFER        ;REWRITE SECTOR
040312  005037  001176                   CLR     SPECMD          ;RESET FLAG
040316  005737  001322                   TST     ERRFLG          ;ERRORS?
040322  001402                           BEQ     CRC27           ;CONTINUE IF NO
040324  000137  040614                   JMP     CRCXEX          ;EXIT IF YES
040330  005204                   CRC27:  INC     R4              ;SUBTEST NUMBER=27(REWRITE OK)
040332  012700  023614                   MOV     #RDBLK,R0       ;POINT TO READ BLOCK ADDR
040336  004537  022700                   JSR     R5,DXFER        ;ATTEMPT READ
040342  005737  001322                   TST     ERRFLG          ;ERRORS?
040346  001004                           BNE     CRC30           ;CONTINUE IF YES
040350  005237  001322                   INC     ERRFLG          ;SET ERROR FLAG IF NOT
040354  000137  040614                   JMP     CRCXEX          ;AND EXIT
040360  005204                   CRC30:  INC     R4              ;SUBTEST NUMBER=30(ERROR DETECTED)
040362  017703  140426                   MOV     @DERR,R3        ;FETCH ERROR REGISTER
040366  012702  000401                   MOV     #BIT8!BIT0,R2   ;SET UP REFERENCE
```

```
    040372  020203                       CMP     R2,R3            ;CONTINUE IF YES
    040374  001402                       BEQ     CRC31            ;CONTINUE IF YES
    040376  000137  040614               JMP     CRCXEX           ;EXIT IF NO
    040402  005204              CRC31:   INC     R4               ;SUBTEST NUMBER=31(CRC ERROR)
    040404  005037  001322               CLR     ERRFLG           ;RESET ERROR FLAG
    040410  012700  025450               MOV     #ISECFB,R0       ;POINT TO FORMAT BLOCK
    040414  004537  004034               JSR     R5,CLRBUF        ;ZERO BUFFERS
    040420  005237  001176               INC     SPECMD           ;SET SPECIAL COMMAND FLAG
    040424  004537  022700               JSR     R5,DXFER         ;REFORMAT
    040430  005037  001176               CLR     SPECMD           ;RESET COMMAND FLAG
    040434  005737  001322               TST     ERRFLG           ;ERROR DURING FORMAT?
    040440  001402                       BEQ     CRC32            ;CONTINUE IF NO
    040442  000137  040614               JMP     CRCXEX           ;EXIT IF YES
    040446  005204              CRC32:   INC     R4               ;SUBTEST NUMBER=32(REFORMAT OK)
    040450  012700  040632               MOV     #RDCDE,R0        ;POINT TO READ BLOCK
    040454  005237  001176               INC     SPECMD           ;SET SPECIAL COMMAND FLAG
    040460  004537  022700               JSR     R5,DXFER         ;READ SECTOR
    040464  005037  001176               CLR     SPECMD           ;RESET COMMAND FLAG
    040470  005737  001322               TST     ERRFLG           ;ERRORS?
    040474  001402                       BEQ     CRC33            ;CONTINUE IF NO
    040476  000137  040614               JMP     CRCXEX           ;EXIT IF YES
    040502  005204              CRC33:   INC     R4               ;SUBTEST NUMBER=33(READ OK)
    040504  016001  000002               MOV     2(R0),R1         ;POINT TO BUFFER WITH R1
    040510  016103  000004               MOV     4(R1),R3         ;FETCH FIST CRC WORD
    040514  005002                       CLR     R2               ;SET UP REFERENCE
    040516  020203                       CMP     R2,R3            ;EQUAL?
    040520  001404                       BEQ     CRC34            ;CONTINUE IF YES
    040522  005237  001322               INC     ERRFLG           ;SET ERROR FLAG IF NO
    040526  000137  040614               JMP     CRCXEX           ;AND EXIT
    040532  005204              CRC34:   INC     R4               ;SUBTEST NUMBER=34(FIRST WORD OK)
    040534  016103  000006               MOV     6(R1),R3         ;FETCH SECOND WORD
    040540  020203                       CMP     R2,R3            ;SECOND WORD OK?
    040542  001404                       BEQ     CRC35            ;CONTINUE IF YES
    040544  005237  001322               INC     ERRFLG           ;SET ERROR FLAG IF NO
    040550  000137  040614               JMP     CRCXEX           ;AND EXIT
    040554  005204              CRC35:   INC     R4               ;SUBTEST NUMBER=35(SECOND WORD OK
    040556  016103  000014               MOV     14(R1),R3        ;FETCH THIRD WORD
    040562  020203                       CMP     R2,R3            ;OK?
    040564  001404                       BEQ     CRC36            ;CONTINUE IF YES
    040566  005237  001322               INC     ERRFLG           ;SET ERROR FLAG IF NO
    040572  000137  040614               JMP     CRCXEX           ;AND EXIT
    040576  005204              CRC36:   INC     R4               ;SUBTEST NUMBER=36(THIRD WORD OK
    040600  016103  000016               MOV     16(R1),R3        ;FETCH FOURTH WORD
    040604  020203                       CMP     R2,R3            ;OK?
    040606  001402                       BEQ     CRCXEX           ;EXIT IF YES
    040610  005237  001322               INC     ERRFLG           ;SET ERROR FLAG IF NO
    040614  004537  003342      CRCXEX:  JSR     R5,TSTCTL        ;EXIT HERE
                                         ;
                                         ;
                                         ;
                                         ;  CRC / ECC TEST COMMAND BLOCKS
                                         ;  ----------------------------
                                         ;
                                         ;
    040620  000000              ECCWRT:  .WORD 0          ;DUSH=0
    040622  044312                       .WORD ODBUF      ;OUTPUT BUFFER-ODBUF
    040624  177400                       .WORD 177400     ;WORD COUNT -FLEXIBLE
```

```
 040626  000000                    .WORD 0          ;CYLINDER ADDR=0
 040630  000014                    .WORD WHSCRC     ;WRITE THE BALL GAME COMMAND
                                   ;
                                   ;
                                   ;
                                   ;
 040632  000000           RDCDE:   .WORD 0          ;DUSH=0
 040634  047314                    .WORD IDBUFT     ;INPUT BUFFER-FLEXIBLE
 040636  177400                    .WORD 177400     ;WORD COUNT-FLEXIBLE
 040640  000000                    .WORD 0          ;CYLINDER ADDR=0
 040642  000012                    .WORD RDFMT      ;READ BALL GAME COMMAND
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;

                                   ;
                                   ;
                                   ;
                                   ;
                                   ;
                                   ;        ECC SECTOR FORMAT AND READ ROUTINE
                                   ;
                                   ;        THIS ROUTINE ISSUES 4 DISK COMMANDS DURING
                                   ;    THE ECC TEST SEQUENCE.THE GENERAL FUNCTION OF THE
                                   ;    EXERCISE IS TO FORCE ECC ERRORS BY DESTROYING A
                                   ;    SMALL SECTION OF A SECTOR BY USING THE WRITE
                                   ;    HEADER,CRC,DATA AND ECC COMMAND.THE OPERATOR CAN SELECT
                                   ;    THE AREA IN THE DATA FIELD WHERE HE DESIRES
                                   ;    TO CAUSE THE ERROR.THE STEPS TO ACCOMPLISH THIS ARE AS
                                   ;    FOLLOWS:
                                   ;
                                   ;        1. USING THE NORMAL WRITE COMMAND,WRITE A SINGLE SECTOR.
                                   ;           EXECUTING THIS COMMAND WILL CAUSE ECC WORDS
                                   ;           TO BE GENERATED.
                                   ;
                                   ;        2. READ THE SECTOR USING THE READ FORMAT COMMAND
                                   ;           THIS COMMAND READS THE ENTIRE SECTOR ,INCLUDING
                                   ;           HEADERS,HEADER CRC'S,DATA AND ECC WORDS
                                   ;
                                   ;        3. REWRITE THE SECTOR AFTER CHANGING DESIRED
                                   ;           DATA WORD(FROM1 TO 16 BITS).
                                   ;
                                   ;        4. ISSUE NORMAL READ ON THAT SECTOR TO CAUSE ECC ERROR.
                                   ;
                                   ;        5. EXIT ROUTINE.
                                   ;
                                   ;        CALLING SEQUENCE:
                                   ;
                                   ;        (X)     JSR     R5,WRTECC       ;PROGRAM CALL
                                   ;        (X+2)   .WORD 0                 ;DATA PATTERN DESIRED(ENTIRE SECTOR)
                                   ;        (X+4)   .WORD 0                 ;WORD POSITION(X2)
```

```
                                        ;       (X+6)   .WORD 0                 ;BIT(S) CLEARED TO CAUSE ERROR
                                        ;
                                        ;
                                        ;       NOTE: THIS ROUTINE DOES NOT MODIFY THE ECC INHIBIT
                                        ;          BIT (BIT 15 OF THE ECC BIT PATTERN REG).IT IS  THEREFORE
                                        ;          POSSIBLE TO USE USE THIS CALL TO CHECK ECC CORRECTION
                                        ;          AS WELL AS DECTECTION LOGIC.
                                        ;
                                        ;
                                        ;
                                        ;
040644  012700  024652          WRTECC: MOV     #ISECWB,R0      ;POINT TO WRITE BLOCK
040650  016001  000002                  MOV     2(R0),R1        ;POINT TO WRITE BUFFER
040654  013702  001050                  MOV     POSWP,R2        ;GETCOUNT FOR BUFFER FORMAT
040660  011503                          MOV     (R5),R3         ;R3=DATA PATTERN FOR SECTOR
040662  010321                  WRT1:   MOV     R3,(R1)+        ;FORMAT BUFFER WORD
040664  005302                          DEC     R2              ;DECRMENN WORD COUNTER
040666  001375                          BNE     WRT1            ;CONNINUE IF NOT DONE
040670  004537  022700                  JSR     R5,DXFER        ;GO FORMAT SECTOR
040674  005737  001322                  TST     ERRFLG          ;ERRORS?
040700  001061                          BNE     ECCEXT          ;EXIT IF ERROR(S)
040702  012700  041064                  MOV     #RDECC,R0       ;POINT TO READ BLOCK
040706  013760  001046  000004          MOV     WPSEC,4(R0)     ;ENTER WORD COUNT
040714  162760  000012  000004          SUB     #10.,4(R0)      ;ADJUST FOR EXTENDED TRANSFER
040722  005237  001176                  INC     SPECMD          ;
040726  004537  022700                  JSR     R5,DXFER        ;GO READ SECTOR U
040732  005037  001176                  CLR     SPECMD          ;RESET FLAG
040736  005737  001322                  TST     ERRFLG          ;ERRORS
040742  001040                          BNE     ECCEXT          ;EXIT IF YES
040744  016001  000002                  MOV     2(R0),R1        ;USE R1 TO INDEX WORD POSITION
040750  066501  000002                  ADD     2(R5),R1        ;ADJUST FOR DESIRED WORD
040754  062701  000020                  ADD     #20,R1          ;INCLUDE HEADERS IN POSITIONING
040760  016511  000004                  MOV     4(R5),(R1)      ;ENTER DESIRED WORD
040764  012700  041052                  MOV     #ECCWBK,R0      ;POINT TO WRITE BLOCK
040770  013760  001046  000004          MOV     WPSEC,4(R0)     ;ENTER WORD COUNT
040776  162760  000012  000004          SUB     #10.,4(R0)      ;ADJUST FOR HEADERS,CRC,ECC
041004  005237  001176                  INC     SPECMD          ;SET TRANSFER FLAG
041010  004537  022700                  JSR     R5,DXFER        ;REWRITE SECTOR
041014  005037  001176                  CLR     SPECMD          ;RESET FLAG
041020  005737  001322                  TST     ERRFLG          ;ERRORS?
041024  001007                          BNE     ECCEXT          ;EXIT IF YES
041026  012700  025462                  MOV     #ISECRB,R0      ;POINT TO READ BLOCK
041032  013760  001046  000004          MOV     WPSEC,4(R0)     ;ENTER WORD COUNT
041040  004537  022700                  JSR     R5,DXFER        ;READ SECTOR
041044  062705  000006          ECCEXT: ADD     #6,R5           ;UPDATE R5 FOR RETURN
041050  000205                          RTS     R5              ;AND EXIT
                                                                ;WE DONT'T CARE ABOUT ERRORS AT THIS POINT
                                        ;
                                        ;
                                        ;
                                        ;
041052  000000                  ECCWBK: .WORD 0                 ;SEC,HEAD=0
041054  044312                          .WORD ODBUF             ;OUTPUT BUFFER ADDRESS
041056  177400                          .WORD 177400            ;WORDS PER TRANSFER
041060  000000                          .WORD 0                 ;CYLINDER ADDRESS
041062  000014                          .WORD WHSCRC            ;WRITE HEADER,CRC,DATA,ECC COMMAND
                                        ;
```

```
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;       READ ENTIRE SECTOR COMMAND BLOCK
                                             ;
                                             ;
041064  000000                      RDECC:   .WORD 0               ;SECTOR,HEAD=0
041066  044312                               .WORD ODBUF           ;INPUT BUFFER
041070  177400                               .WORD 177400          ;WORD COUNT
041072  000000                               .WORD 0               ;CYLINDER ADDRESS
041074  000012                               .WORD RDFMT           ;READ FORMAT COMMAND
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;       TIM.ECC TEST NUMBERS 21 >> 23   13-NOV-77
                                             ;
                                             ;
                                             ;       TNA21.PM        8-NOV-77
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;          ECC DECTECTION TESTS
                                             ;       ****************************
                                             ;
                                             ;
                                             ;
                                             ;       THIS TEST SEQUENCE CHECKS THE ECC DETECTION
                                             ;    LOGIC OF THE PHOENIX 211.ALL THESE SUBTESTS USE THE
                                             ;    ECC INHIBIT OPTION THAT STOPS ALL CORRECTION ACTIONS
                                             ;    THAT WOULD NORMALLY OCCUR IF ONE ENCOUNTERS
                                             ;    AN ECC ERROR.THE PURPOSE OF INHIBITNG ECC IS THAT
                                             ;    AT THIS TIME IN TESTING THE CORRECTION LOGIC HAS
                                             ;    NOT BEEN CHECKED.
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
041076  005004                      TNA21:   CLR     R4            ;SUBTEST NUMBER=0
041100  004537  004276                       JSR     R5,RQUEST     ;REQUEST FORMATTER
041104  012777  100000  137704               MOV     #BIT15,@ECCPB ;SET ECC INHIBIT BIT
041112  004537  040644                       JSR     R5,WRTECC     ;FIRST CHECK ALL ZERO DATA FIELD
                                                                   ;THE THREE LOCATIONS FOLLOWING AN ECC
                                                                   ;WRITE CALL(WRTECC) ARE PARAMETERS THAT
                                                                   ;DEFINE THE DATA SETUP:
                                                     ;WORD 1 ; SECTOR PATTERN
```

```
                                                    ;
                                                    ;WORD 2 ;POSITION IN SECTOR OF WORD TO CHANGE FROM PATTERN
                                                    ;
                                                    ;WORD 3 ;DESIRED BIT CHANGE
                                                    ;
041116  000000                      .WORD   0               ;SECTOR PATTERN
041120  000000                      .WORD   0               ;POSITION 0
041122  000000                      .WORD   0               ;NO BIT CHANGE
041124  005737  001322              TST     ERRFLG          ;ERROR SET? -SHOULD'T BE
041130  001402                      BEQ     ECCD1           ;CONTINUE IF NO
041132  000137  041642              JMP     ECCDEX          ;EXIT IF ERROR
041136  005204              ECCD1:  INC     R4              ;SUBTEST NUMBER=1(NO ERROR ON CORRECT SECTOR)
041140  004537  040644              JSR     R5,WRTECC       ;NOW FORCE ERROR
041144  000000                      .WORD   0               ;SECTOR PATTERN=0
041146  000000                      .WORD   0               ;POSITION 0
041150  000001                      .WORD   1               ;FIRST BIT WRONG
041152  005737  001322              TST     ERRFLG          ;ERROR?
041156  001002                      BNE     ECCD2           ;CONTINUE IF YES
041160  000137  041642              JMP     ECCDEX          ;AND EXIT
041164  005204              ECCD2:  INC     R4              ;SUBTEST NUMBER=2(ERROR DETECTED)
041166  004537  041652              JSR     R5,ECCERR       ;CHECK IF ECC ERROR(CRC ERROR BIT SET)
041172  005737  001322              TST     ERRFLG          ;OK?
041176  001402                      BEQ     ECCD3           ;CONTINUE IF YES
041200  000137  041642              JMP     ECCDEX          ;EXIT IF NO
041204  005204              ECCD3:  INC     R4              ;SUBTEST NUMBER=3(ERROR BIT SET)
041206  004537  040644              JSR     R5,WRTECC       ;TRY LAST BIT OF SECTOR
041212  000000                      .WORD 0                 ;STILL SECTOR OF ZEROES
041214  000776                      .WORD 776               ;LAST WORD
041216  100000                      .WORD 100000            ;LAST BIT
041220  005737  001322              TST     ERRFLG          ;ERROR?
041224  001002                      BNE     ECCD4           ;CONTINUE IF ERROR
041226  000137  041642              JMP     ECCDEX          ;AND EXIT
041232  005204              ECCD4:  INC     R4              ;SUBTEST NUMBER=4(ERROR AGAIN)
041234  004537  041652              JSR     R5,ECCERR       ;CHECK ERROR BITS
041240  005737  001322              TST     ERRFLG          ;OK?
041244  001402                      BEQ     ECCD5           ;GO ON IF YES
041246  000137  041642              JMP     ECCDEX          ;EXIT IF NOT
041252  005204              ECCD5:  INC     R4              ;SUBTEST NUMBER=5(ERROR BITS OK)
041254  004537  040644              JSR     R5,WRTECC       ;TRY MIDDLE OF SECTOR
041260  000000                      .WORD 0                 ;STILL ZEROES
041262  000376                      .WORD 376               ;MIDDLE WORD
041264  000200                      .WORD 200               ;MIDDLE OF WORD(WELL,ALMOST)
041266  005737  001322              TST     ERRFLG          ;ERRORS?
041272  001004                      BNE     ECCD6           ;CONTINUE IF YES
041274  005237  001322              INC     ERRFLG          ;SET IT IF NOT
041300  000137  041642              JMP     ECCDEX          ;AND EXIT
041304  005204              ECCD6:  INC     R4              ;SUBTEST NUMBER=6(ERROR SET)
                                                            ;ISN'T THIS FUN!!
041306  004537  041652              JSR     R5,ECCERR       ;CHECK ERROR BITS
041312  005737  001322              TST     ERRFLG          ;ERROR SET?
041316  001402                      BEQ     ECCD7           ;CONTINUE IF NO
041320  000137  041642              JMP     ECCDEX          ;AND EXIT IF NOT
041324  005204              ECCD7:  INC     R4              ;SUBTEST NUMBER=7(MIDDLE OF SECTOR OK)
041326  004537  040644              JSR     R5,WRTECC       ;NOW LETS TRY ALL ONES(NO ERROR)
041332  177777                      .WORD   177777          ;SECTOR ALL ONES
041334  000000                      .WORD   0               ;FIRST WORD
041336  177777                      .WORD   177777          ;KEEP ALL BITS
```

```
799 041340  005737  001322          TST     ERRFLG          ;ERRROR SET?
800 041344  001402                  BEQ     ECCD10          ;CONTINUE IF NO
801 041346  000137  041642          JMP     ECCDEX          ;EXIT IF YES
802 041352  005204          ECCD10: INC     R4              ;SUBTEST NUMBER=10(NO ERROR ALL ONES)
803 041354  004537  040644          JSR     R5,WRTECC       ;NOW FIRST BIT A ZERO
804 041360  177777                  .WORD   177777          ;SECTOR ALL ONES
805 041362  000000                  .WORD   0               ;FIRST WORD
806 041364  177776                  .WORD   177776          ;LOSE BIT 0
807 041366  005737  001322          TST     ERRFLG          ;ERRORS?
808 041372  001002                  BNE     ECCD11          ;CONTINUE IF YES
809 041374  000137  041642          JMP     ECCDEX          ;AND EXIT
810 041400  005204          ECCD11: INC     R4              ;SUBTEST NUMBER=11(PICKED UP FIRST ZERO
811 041402  004537  041652          JSR     R5,ECCERR       ;NOW CHECK ERROR BIT
812 041406  005737  001322          TST     ERRFLG          ;ERROR OK?
813 041412  001402                  BEQ     ECCD12          ;CONTINUE IF YES
814 041414  000137  041642          JMP     ECCDEX          ;EXIT IF NO
815 041420  005204          ECCD12: INC     R4              ;SUBTEST NUMBER=12(ERROR BIT OK)
816 041422  004537  040644          JSR     R5,WRTECC       ;NOW TRY LAST BIT
817 041426  177777                  .WORD   177777          ;SECTOR ALL ONES
818 041430  000776                  .WORD   776             ;LAST WORD
819 041432  077777                  .WORD   77777           ;LAST BIT
820 041434  005737  001322          TST     ERRFLG          ;ERROR?
821 041440  001002                  BNE     ECCD13          ;CONTINUE IF YES
822 041442  000137  041642          JMP     ECCDEX          ;EXIT IF NO
823 041446  005204          ECCD13: INC     R4              ;SUBTEST NUMBER=13(LAST BIT ERRORED)
824 041450  004537  041652          JSR     R5,ECCERR       ;CHECK ERROR BIT
825 041454  005737  001322          TST     ERRFLG          ;ERRROS OK?
826 041460  001402                  BEQ     ECCD14          ;CONTINUE IF YES
827 041462  000137  041642          JMP     ECCDEX          ;EXIT IF NO
828 041466  005204          ECCD14: INC     R4              ;SUBTEST NUMBER=14(ERROR BIT OK)
829 041470  004537  040644          JSR     R5,WRTECC       ;TRY MIDDLE OF SECTOR
830 041474  177777                  .WORD   177777          ;SECTOR ALL ONES
831 041476  000376                  .WORD   376             ;MIDDLE WORD
832 041500  177377                  .WORD   177377          ;LOSE BIT 7
833 041502  005737  001322          TST     ERRFLG          ;ERROR SET?
834 041506  001002                  BNE     ECCD15          ;CONTINUE IF YES
835 041510  000137  041642          JMP     ECCDEX          ;EXIT IF NOT
836 041514  005204          ECCD15: INC     R4              ;SUBTEST NUMBER=15(MIDDLE ZERO ERR)
837 041516  004537  041652          JSR     R5,ECCERR       ;CHECKERROR BIT
838 041522  005737  001322          TST     ERRFLG          ;BIT OK?
839 041526  001402                  BEQ     ECCD16          ;GO NO IF YES
840 041530  000137  041642          JMP     ECCDEX          ;LEAVE THIS PLACE IF NOT
841 041534  005204          ECCD16: INC     R4              ;SUBTEST NUMBER=16(CRC BIT AGAIN OK)
842 041536  004537  040644          JSR     R5,WRTECC       ;NOW LET'S FOOL WITH A CHECKER
843                                                         ;BOARD PATTERN.
844 041542  125252                  .WORD   125252          ;SECTOR PATTERN(CHECKER BOARD)
845 041544  000376                  .WORD   376             ;MIDDLE OF SECTOR
846 041546  125052                  .WORD   125052          ;LOSE BIT 7(TRICKY)
847 041550  005737  001322          TST     ERRFLG          ;DID IT ERROR?
848 041554  001002                  BNE     ECCD17          ;CONTINUE IF YES
849 041556  000137  041642          JMP     ECCDEX          ;EXIT IF NO
850 041562  005204          ECCD17: INC     R4              ;SUBTEST NUMBER=17(CHECKERBD ERROR)
851 041564  004537  041652          JSR     R5,ECCERR       ;CHECK ECC ERROR BIT
852 041570  005737  001322          TST     ERRFLG          ;ERROR?
853 041574  001402                  BEQ     ECCD20          ;CONTINUE IF YES
854 041576  000137  041642          JMP     ECCDEX          ;EXIT IF NOT CORRECT
855 041602  005204          ECCD20: INC     R4              ;SUBTEST NUMBER=20(CRC BIT OK)
```

```
041604	004537	040644		JSR	R5,WRTECC	;DO SAME TEST WITH REVERSE CHECKERBD.
041610	052525			.WORD	052525		;HERE SI SECTOR PATTERN
041612	000376			.WORD	376		;POSITION-MIDDLE OF SECTOR
041614	052125			.WORD	052125		;LOSE BIT 8
041616	005737	001322		TST	ERRFLG		;ERROR?
041622	001002			BNE	ECCD21		;CONTINUE IF YES
041624	000137	041642		JMP	ECCDEX		;EXIT IF ERROR
041630	005204		ECCD21:	INC	R4		;SUBTEST NUMBER=21(SECOND CHECKER OK)
041632	004537	041652		JSR	R5,ECCERR	;ERROR BIT OK?
041636	004537	003342	ECCD22:	JSR	R5,TSTCTL	;EXIT TO TEST CONTROL HERE
					;
					;
					;
041642	012737	000001	001322	ECCDEX:	MOV	#1,ERRFLG	;INSURE ERROR FLAG=1
041650	000772			BR	ECCD22		;AND EXIT
					;
					;
					;
					;
					;
					;
					;
					;
					;
					;
					;	ECC ERROR BIT CHECK ROUTINE
					;    ********************************
					;
					;	   THIS SHORT SUROUTINE HELPS REDUCE CODE
					;	WHEN CHECKING ERROR DETECTION OF THE ECC LOGIC
					;
					;	AT EXIT R2=REFERENCE ERROR WORD(CRC ERROR BIT)
					;		R3= ACTUAL ERROR REGISTER
					;
					;	AND THE ERROR FLAG "ERRFLG" WILL BE
					;	SET IF R2 DOES NOT EQUAL R3
					;
					;
					;
041652	012702	000400	ECCERR:	MOV	#BIT8,R2	;SET UP REFERENCE WORD (CRC ERROR)
041656	017703	137132		MOV	@DERR,R3	;FETCH ERROR REGISTER
041662	020203			CMP	R2,R3		;EQUAL?
041664	001002			BNE	ECEROT		;EXIT IF NO
041666	005037	001322		CLR	ERRFLG		;RESET ERROR FLAG IF YES
041672	000205		ECEROT:	RTS	R5		;AND RETURN
					;
					;
					;
					;
					;
					;
					;
					;
					;
					;
					;
					;	TNA22.PM		8-NOV-77
```

```
                                                ;
                                                ;
                                                ;     ECC DETECTION/CORRECTION TLOGIC TEST
                                                ;     ******************************************
                                                ;
                                                ;
                                                ;     THESE TESTS ARE THE FIRST TO CHECK THE
                                                ;  CAPABILITIES OF THE ECC LOGIC IN CORRECTING
                                                ;  BAD DATA AS A RESULT OF NONWRITABLE SPOTS
                                                ;  ON A DISK PACK .
                                                ;
                                                ;  THE PHOENIX 211 HAS COMMANDS THAT ENABLE THE
                                                ;  PROGRAMMER TO WRITE DATA ON ANY PART OF THE
                                                ;  SECTOR FIELD INCLUDING THE HEADER CRC WORDS AND
                                                ;  THE DATA ECC WORDS.THIS OPTION MAKES IT VERY
                                                ;  EASY TO CHECK OTHER CONTROLLER FUNCTIONS
                                                ;  LIKE THE ECC GENERATION.
                                                ;
                                                ;     THE FOLLOWING TESTS WILL TRY TO FORCE ECC
                                                ;  ERRORS IN THE BELOW MANNER:
                                                ;
                                                ;  1.ISSUE A WRITE WITH PRESLECTED DATA FIELD
                                                ;     THIS COMMAND IS A NORMAL WRITE FUNCTION.
                                                ;
                                                ;  2.ISSUE A READ COMMAND TO READ ECC WORDS
                                                ;    THIS COMMAND READS ALL FIELDS OF THE SECTOR.
                                                ;
                                                ;  3. ISSUE A SECOND WRITE COMMAND AFTER HAVING CHANGED
                                                ;     THE DATA FIELD TO FORCE THE ECC ERROR.
                                                ;  THIS COMMAND WRITE ALL FIELDS OF THE SECTOR.
                                                ;
                                                ;  4.ISSUE A SECOND NORMAL READ TO FORCE THE ECC ERROR
                                                ;
                                                ;  5.LATER IN THE TEST PROCEDURE THE ECC POSITION
                                                ;    AND PATTERN REGISTERS WILL BE CHECKED FOR
                                                ;    PROPER  VALUES.
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
                                                ;
    041674  005004                      TNA22:  CLR     R4              ;SUBTEST NUMBER=0
    041676  004537  004276                       JSR     R5,RQUEST       ;GET FORMATTER
    041702  052777  100000  137110               BIS     #BIT15,@ECCPW   ;SET ECC INHIBIT BIT
    041710  004537  040644                       JSR     R5,WRTECC       ;FIRST WRITE A SINGLE BIT ERROR
    041714  000000                               .WORD   0               ;SECTOR PATTERN=0
    041716  000000                               .WORD   0               ;FIRST DATA WORD
    041720  000001                               .WORD   1               ;FIRST BIT BAD
    041722  005737  001322                       TST     ERRFLG          ;ERROR BIT SET?(SHOULD BE)
    041726  001002                               BNE     ECDC1           ;CONTINUE IF YES
    041730  000137  042504                       JMP     EDCEX           ;EXIT IF NO
    041734  005204                       ECDC1:  INC     R4              ;SUBTEST NUMBER=1(FIRST ERROR DETECTED)
    041736  004537  041652                       JSR     R5,ECCERR       ;CHECK ERROR BITS
    041742  005737  001322                       TST     ERRFLG          ;ERROR RIGHT?
    041746  001402                               BEQ     ECDC2           ;CONTINUE IF YES
    041750  000137  042504                       JMP     EDCEX           ;EXIT IF NO
```

```
 970 041754  005204                      ECDC2:  INC     R4              ;SUBTEST NUMBER=2(ECC ERROR)
 971 041756  017703  137024                      MOV     @DWCNT,R3       ;TRANSFER STOP?
 972 041762  005002                              CLR     R2              ;SET UP REFERENCE
 973 041764  020203                              CMP     R2,R3           ;WORD COUNT OVERFLOW?
 974 041766  001002                              BNE     ECDC3           ;CONTINUE IF NO
 975 041770  000137  042504                      JMP     EDCEX           ;EXIT IF OVERFLOWED
 976 041774  005204                      ECDC3:  INC     R4              ;SUBTEST NUMBER=3
 977 041776  017703  137014                      MOV     @ECCPB,R3       ;FETCH BIT POSITION REGISTER
 978 042002  005002                              CLR     R2              ;SET UP REFERENCE WORD
 979 042004  020203                              CMP     R2,R3           ;BIT ZEROED?
 980 042006  001402                              BEQ     ECDC4           ;CONTINUE IF YES
 981 042010  000137  042504                      JMP     EDCEX           ;EXIT IFNO
 982 042014  005204                      ECDC4:  INC     R4              ;SUBTEST NUMBER=4(BIT POSITION DIDN'T CHANGE)
 983 042016  042777  100000  136774              BIC     #BIT15,@ECCPW   ;RESET ECC INHIBIT
 984 042024  004537  040644                      JSR     R5,WRTECC       ;TRY ECC CORRECTION OF 1 ERROR BIT
 985 042030  000000                              .WORD   0               ;PATTERN=0
 986 042032  000000                              .WORD   0               ;POSITION 1
 987 042034  000001                              .WORD   1               ;FIRST BIT IS BAD
 988 042036  005737  001322                      TST     ERRFLG          ;ERROR SET(SHOULD BE)
 989 042042  001002                              BNE     ECDC5           ;CONTINUE IF YES
 990 042044  000137  042504                      JMP     EDCEX           ;EXIT IF NO ERROR
 991 042050  005204                      ECDC5:  INC     R4              ;SUBTEST NUMBER=5(ERROR DETECTED)
 992 042052  004537  041652                      JSR     R5,ECCERR       ;CHECK ERROR BIT(CRC)
 993 042056  005737  001322                      TST     ERRFLG          ;OK?
 994 042062  001402                              BEQ     ECDC6           ;CONTINUE IF YES
 995 042064  000137  042504                      JMP     EDCEX           ;EXIT IF NO
 996 042070  005204                      ECDC6:  INC     R4              ;SUBTEST NUMBER=6(ECC ERROR DETECTED)
 997 042072  017703  136720                      MOV     @ECCPB,R3       ;FECTH ECC BIT POSITION REG
 998 042076  012702  000001                      MOV     #BIT0,R2        ;SET UP REFERENCE WORD
 999 042102  020203                              CMP     R2,R3           ;BITS SET?
1000 042104  001402                              BEQ     ECDC7           ;CONTINUE IF YES
1001 042106  000137  042504                      JMP     EDCEX           ;EXIT IF NO
1002 042112  005204                      ECDC7:  INC     R4              ;SUBTEST NUMBER=7(COUNT=1)
1003 042114  017703  136700                      MOV     @ECCPW,R3       ;FETCH PATTERN WORD
1004 042120  020203                              CMP     R2,R3           ;SET IF PATTERN WORD OK
1005 042122  001402                              BEQ     ECDC10          ;CONTINUE IF YES
1006 042124  000137  042504                      JMP     EDCEX           ;EXIT IF NO
1007 042130  005204                      ECDC10: INC     R4              ;SUBTEST NUMBER=10(PATTERN OK)
1008 042132  004537  040644                      JSR     R5,WRTECC       ;TRY 11 BIT ERROR
1009 042136  000000                              .WORD   0               ;FIELD=0
1010 042140  000000                              .WORD   0               ;POSTION=1
1011 042142  003777                              .WORD   3777            ;ERROR IS FIRST 11 BITS
1012 042144  005737  001322                      TST     ERRFLG          ;ERROR?
1013 042150  001002                              BNE     ECDC11          ;CONTINUE IF YES
1014 042152  000137  042504                      JMP     EDCEX           ;EXIT IF NO
1015 042156  005204                      ECDC11: INC     R4              ;SUBTEST NUMBER=11(ERROR OCCURED)
1016 042160  004537  041652                      JSR     R5,ECCERR       ;CHECK ECC ERROR BIT
1017 042164  005737  001322                      TST     ERRFLG          ;OK?
1018 042170  001402                              BEQ     ECDC12          ;CONTINUE IF YES
1019 042172  000137  042504                      JMP     EDCEX           ;EXIT IF NOT
1020 042176  005204                      ECDC12: INC     R4              ;SUBTEST NUMBER=12(ERROR BIT OK)
1021 042200  012702  000001                      MOV     #BIT0,R2        ;SET UP REFERENCE FOR BIT POINTER
1022 042204  017703  136606                      MOV     @ECCPB,R3       ;FETCH BIT POSTION REG
1023 042210  020203                              CMP     R2,R3           ;OK?
1024 042212  001402                              BEQ     ECDC13          ;CONTINUE IF YES
1025 042214  000137  042504                      JMP     EDCEX           ;EXIT IF NO
1026 042220  005204                      ECDC13: INC     R4              ;SUBTEST NUMBER=13(COUNTER OK)
```

```
1027 042222  017703  136572                  MOV     @ECCPW,R3          ;FETCH ERROR PATTERN
1028 042226  012702  003777                  MOV     #3777,R2           ;SET UP DESIRED RESULT
1029 042232  020203                          CMP     R2,R3              ;EQUAL?
1030 042234  001402                          BEQ     ECDC14             ;CONTINUE IF YES
1031 042236  000137  042504                  JMP     EDCEX              ;EXIT IF NO
1032 042242  005204                  ECDC14: INC     R4                 ;PATTERN OK(NO HARD ERROR)
1033 042244  004537  040644                  JSR     R5,WRTECC          ;FORCE HARD ERROR #>11
1034 042250  000000                          .WORD   0                  ;SECTOR=0
1035 042252  000000                          .WORD   0                  ;FIRST WORD
1036 042254  007777                          .WORD   7777               ;12 BIT ERROR
1037 042256  005737  001322                  TST     ERRFLG             ;ERROR SET?
1038 042262  001002                          BNE     ECDC15             ;CONTINUE IF YES
1039 042264  000137  042504                  JMP     EDCEX              ;EXIT IF NO
1040 042270  005204                  ECDC15: INC     R4                 ;SUBTEST NUMBER=15(AN ERROR IS SET)
1041 042272  004537  041652                  JSR     R5,ECCERR          ;CHECK ERROR BIT SET
1042 042276  005737  001322                  TST     ERRFLG             ;OK?
1043 042302  001402                          BEQ     ECDC16             ;CONTINUE IF YES
1044 042304  000137  042504                  JMP     EDCEX              ;EXIT IF NO
1045 042310  005204                  ECDC16: INC     R4                 ;SUBTEST NUMBER=16(ERROR SET)
1046 042312  012702  110041                  MOV     #110041,R2         ;SET UP REFERENCE FOR BIT COUNT
1047 042316  017703  136474                  MOV     @ECCPB,R3          ;FETCH COUNT REGISTER
1048 042322  020203                          CMP     R2,R3              ;EQUAL?
1049 042324  001402                          BEQ     ECDC17             ;CONTINUE IF YES
1050 042326  000137  042504                  JMP     EDCEX              ;EXIT IF NOT
1051 042332  005001                  ECDC17: CLR     R1                 ;SUBTEST NUMBER=17(COUNTER OVERFLOWED)
1052                                                                    ;AND HARD ECC ERROR DETECTED
1053 042334  012704  000017                  MOV     #17,R4             ;FORCE SUBTEST NUMBER=17
1054 042340  016137  002612  042362          MOV     S1ORG(R1),ECD17    ;LOAD BIT PATTERN
1055 042346  005721                          TST     (R1)+              ;UPDATE POINTER
1056 042350  010146                          MOV     R1,-(SP)           ;SAVE R1
1057 042352  004537  040644                  JSR     R5,WRTECC          ;WRITE A SLIDING BIT  ERROR
1058 042356  000000                          .WORD   0                  ;SECTOR=0
1059 042360  000000                          .WORD   0                  ;POSITION =1
1060 042362  000001                  ECD17:  .WORD   1                  ;BIT PATTERN
1061 042364  005737  001322                  TST     ERRFLG             ;ERROR SET?(SHOULD BE)
1062 042370  001002                          BNE     ECDC20             ;CONTINUE IF YES
1063 042372  000137  042504                  JMP     EDCEX              ;EXIT IF NO
1064 042376  005204                  ECDC20: INC     R4                 ;SUBTEST NUMBER=20 ERROR DETECTED
1065 042400  013702  042362                  MOV     ECD17,R2           ;SET UP REFERENCE FOR PATTERN COMPARE
1066 042404  017703  136410                  MOV     @ECCPW,R3          ;FETCH PATTERN WORD
1067 042410  004537  041652                  JSR     R5,ECCERR          ;CHECK ERROR BIT
1068 042414  005737  001322                  TST     ERRFLG             ;ERROR SET?
1069 042420  001406                          BEQ     ECDC21             ;CONTINUE IF YES
1070 042422  000137  042504                  JMP     EDCEX              ;EXIT IF ERROR
1071 042426  020203                          CMP     R2,R3              ;PATTERN OK?
1072 042430  001402                          BEQ     ECDC21             ;CONTINUE IF YES
1073 042432  000137  042504                  JMP     EDCEX              ;EXIT IF NO
1074 042436  005204                  ECDC21: INC     R4                 ;SUBTEST NUMBER=21(PATTERN OK)
1075 042440  012702  000001                  MOV     #BIT0,R2           ;SET UP FOR COUNT CHECK
1076                                                                    ;
1077                                                                    ; NOTE: UNTIL A SINGLE BIT ECC ERROR
1078                                                                    ;       PASSES THE 11 BIT POSITION
1079                                                                    ;       IN THE SECTOR FIELD,THE COUNT
1080                                                                    ;       REGISTER WILL BE THE VALUE 1.
1081                                                                    ;
1082 042444  017703  136346                  MOV     @ECCPB,R3          ;FETCH COUNT REGISTER
1083 042450  020203                          CMP     R2,R3              ;OK?
```

```
042452  001402                            BEQ     ECDC22          ;CONTINUE IF YES
042454  000137  042504                    JMP     EDCEX           ;EXIT IF NO
042460  005204                  ECDC22:   INC     R4              ;SUBTEST NUMBER=22(COUNT OK)
042462  012601                            MOV     (SP)+,R1        ;RESTOR R1
042464  010137  042514                    MOV     R1,TMPCNT       ;SAVE IN TEMP.
                                                                  ;AT THIS POINT WE WILL
                                                                  ;REUSE THE PRECEEDING CODE ,FROM
                                                                  ;SUBTEST 17 TO SUBTEST22, TO
                                                                  ;SLIDE THE SINGLE BIT TROUGH 11
                                                                  ;POSITIONS.IF AN ERROR SHOULD OCCUR,
                                                                  ;LOCATION "TMPCNT" WILL BE A
                                                                  ;AN INDEX ON TABLE "S1ORG" TO AID THE
                                                                  ;THE TECHNICIAN IN DEBUGGING.ADDING TMPCNT TO
                                                                  ;ADDRESS S1ORG WILL POINT TO THE
                                                                  ;BIT POSITION FAILING.
                                                                  ;
042470  027727  136324  002000            CMP     @ECCPW,#BIT10   ;SEE IF WE ARE ON LAST BIT
042476  002716                            BLT     ECDC17+2        ;DO NEXT BIT IF LESS THAN
042500  004537  003342          ECDC24:   JSR     R5,TSTCTL       ;GO HOME
                                                                  ;
                                                                  ;
042504  012737  000001  001322  EDCEX:    MOV     #1,ERRFLG       ;SET ERROR FLG=1
042512  000772                            BR      ECDC24          ;AND EXIT
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
                                          ;
042514  000000                  TMPCNT:   .WORD 0                 ;TEMPORARY LOCATION
                                          ;
                                          ;
                                          ;

                                          ;
                                          ;
                                          ;       TNA23           9-NOV-77
                                          ;
                                          ;
                                          ;
                                          ;       ECC COMPATIBILITY/EXERCISE TESTS
                                          ;       ********************************
                                          ;
                                          ;          THESE TESTS CHECK THAT THE CONTROLLER
                                          ;       UNDER TEST IS GENERATING THE SAME CRC/ECC PATTERNS
                                          ;       THAT PREVIOUS 211 CONTROLLERS HAVE GENERATED.
                                          ;
                                          ;         A SECOND PART OF THIS EXERCISE IS TO
                                          ;       FORCE  CRC ERRORS ON VARIOUS AREAS ON THE DISK
                                          ;       AND CHECK THAT THE ECC INFORMATION GENERATED
                                          ;       BY THE ECC LOGIC IS CORRECT.TESTS TNA21 AND TNA22
                                          ;       DO SIMILAR CHECKS HOWEVER,THOSE TESTS CENTER ON
                                          ;       THE FIRST WORD OF THE DATA FIELD.
                                          ;
                                          ;
                                          ;
042516  005004                  TNA23:    CLR     R4              ;SUBTEST NUMBER=0
```

```
1141 042520 004537 004276                  JSR     R5,RQUEST         ;REQUEST THE FORMATTER
1142 042524 012700 024652                  MOV     #ISECWB,R0        ;USE ONE SECTOR WRITE BLOCK
1143 042530 005010                         CLR     (R0)              ;INSURE SECTOR ZERO
1144 042532 023727 001046 177400           CMP     WPSEC,#177400     ;256 WORDS PER SECTOR
1145 042540 001402                         BEQ     ECOMM             ;GO ON IF YES
1146 042542 000137 042752                  JMP     ECCOMEX           ;EXIT IF NO
1147 042546                         ECOMM:
1148 042546 013760 001046 000004           MOV     WPSEC,4(R0)       ;INSURE WORD COUNT
1149 042554 004537 004034                  JSR     R5,CLRBUF         ;CLEAR BUFFER AREA
1150 042560 012737 000001 044312           MOV     #1,ODBUF          ;MAKE DATA FOR SECTOR=1 BIT
1151 042566 004537 022700                  JSR     R5,DXFER          ;GO WRITE SECTOR
1152 042572 005737 001322                  TST     ERRFLG            ;ERRORS?
1153 042576 001402                         BEQ     ECCOM1            ;CONTINUE IF NO
1154 042600 000137 042752                  JMP     ECCOMEX           ;EXIT IF ERROR
1155 042604 005204                  ECCOM1: INC     R4                ;SUBTEST NUMBER=1(SECTOR WRITTEN OK)
1156 042606 012700 040632                  MOV     #RDCDE,R0         ;FETCH READ ALL COMMAND
1157 042612 013702 001234                  MOV     CURDSK,R2         ;ENTER PROPER BUFFER AREA
1158 042616 006302                         ASL     R2                ;
1159 042620 016260 001200 000002           MOV     IBFTBL(R2),2(R0);
1160 042626 013760 001046 000004           MOV     WPSEC,4(R0)       ;SET UP WORD COUNT
1161 042634 162760 000012 000004           SUB     #10.,4(R0)        ;INCLUDE HEADERS,CRC'S AND ECC WORDS
1162 042642 005237 001176                  INC     SPECMD            ;FLAG WEIRD  TRANSFER
1163 042646 004537 022700                  JSR     R5,DXFER          ;GO READ ECC WORDS
1164 042652 005037 001176                  CLR     SPECMD            ;CLEAR WEIRD XFER FLAG
1165 042656 005737 001322                  TST     ERRFLG            ;ERROR SET?
1166 042662 001402                         BEQ     ECCOM2            ;CONTINUE IF NO
1167 042664 000137 042752                  JMP     ECCOMEX           ;EXIT IF ERROR
1168 042670 005204                  ECCOM2: INC     R4                ;SUBTEST NUMBER=2(READ SUCESSFUL)
1169 042672 013701 001050                  MOV     POSWP,R1          ;SET UP POINTER TO LOOK AT ECC WORDS
1170 042676 006301                         ASL     R1                ;MAKE WORDS PER SECTOR AS CORE ADDR
1171 042700 016002 000002                  MOV     2(R0),R2          ;FIND OSTARTING ADDR.
1172 042704 060201                         ADD     R2,R1             ;COMPUTE ECC WORD POSITION
1173 042706 062701 000020                  ADD     #20,R1            ;INCLUDE HEADERS AND HEADER CRC'S
1174 042712 011103                         MOV     (R1),R3           ;FETCH FIRST ECC WORD
1175 042714 013702 001264                  MOV     ECCW1,R2          ;FETCH REFERENCE WORD
1176                                                                  ;  THE REFERENCE WORDS USED HERE
1177                                                                  ; WERE READ FROM THE FIRST ECC
1178                                                                  ;211 CONTROLLER AND WERE INCLUDED
1179                                                                  ;IN THIS ASSEMBLY.
1180 042720 020203                         CMP     R2,R3             ;EQUAL?
1181 042722 001402                         BEQ     ECCOM3            ;CONTINUE IF YES
1182 042724 000137 042752                  JMP     ECCOMEX           ;EXIT IF NOT
1183 042730 005204                  ECCOM3: INC     R4                ;FIRST ECC WORD COMPATIBLE
1184 042732 016103 000002                  MOV     2(R1),R3          ;FETCH SECOND WORD
1185 042736 013702 001266                  MOV     ECCW2,R2          ;SET UP SECOND REFERENCE
1186 042742 020203                         CMP     R2,R3             ;EQUAL?
1187 042744 001002                         BNE     ECCOMEX
1188 042746 004537 003342          ECCOM4: JSR     R5,TSTCTL         ;EXIT HERE
1189                                       ;
1190                                       ;
1191                                       ;
1192                                       ;
1193 042752 012737 000001 001322   ECCOME: MOV     #1,ERRFLG         ;SET ERROR FLAG
1194 042760 000772                         BR      ECCOM4            ;AND EXIT
1195                                       ;
1196                                       ;
1197                                       ;
```

```
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       TN45.ECC        ;14-NOV-77
                                        ;
                                        ;
                                        ;       FORMAT PROGRAM ECC AND NON ECC CONTROLLERS
                                        ; ***********************************************
                                        ;
                                        ;          THIS PROGRAM WILL FORMAT ANY SIZE
                                        ;       STORAGE MODULE DISK DRIVE.16K OF CORE IS REQUIRED TO
                                        ;       RUN THIS PROGRAM.
                                        ;
                                        ;
042762  005004                  TN45:   CLR     R4              ;SUBTEST NUMBER=0
042764  004537  004276                  JSR     R5,RQUEST       ;REQUEST FORMATTER
042770  105777  136004                  TSTB    @DCSR           ;AHVE IT?
042774  100402                          BMI     FT1             ;CONTINUE IF YES
042776  000137  043270                  JMP     FTEX            ;EXIT IF NO
043002  005204                  FT1:    INC     R4              ;SUBTEST NUMBER=1(FORMATTER READY)
043004  004537  016332                  JSR     R5,DISKID       ;SELECT DISK DRIVE
043010  010277  135766                  MOV     R2,@DUSH        ;SELECT
043014  000240                          NOP
043016  005777  135770                  TST     @DSTAT          ;DRIVE READY?
043022  100402                          BMI     FT2             ;CONTINUE IF YES
043024  000137  043270                  JMP     FTEX            ;EXIT IF NO
043030  005204                  FT2:    INC     R4              ;SUBTEST NUMBER=2(DRIVE READY
043032  005001                  FT3:    CLR     R1              ;INITIALIZE OFFSET
043034  013761  001136  044312  FT4:    MOV     CURSEC,ODBUF(R1);LOAD CURRNT SECTOR #
043042  053761  001140  044312          BIS     CURHD,ODBUF(R1) ;AND HEAD
043050  005721                          TST     (R1)+           ;POINT TO CYLINDER ENTRY
043052  013761  001142  044312          MOV     CURCYL,ODBUF(R1);AND CYLINDER
043060  005721                          TST     (R1)+           ;LOOK AT NEXT WORD
043062  005737  001074                  TST     ECC             ;ECC CONTROLLER?
043066  001413                          BEQ     FT5             ;CONTINUE IF YES
043070  013761  001136  044312          MOV     CURSEC,ODBUF(R1);ENTER SECOND SET
043076  053761  001140  044312          BIS     CURHD,ODBUF(R1) ;
043104  005721                          TST     (R1)+           ;POINT TO CYLINDER ENTRY
043106  013761  001142  044312          MOV     CURCYL,ODBUF(R1);
043114  005721                          TST     (R1)+           ;LOOK TO NEXT WORD
043116  005237  001136          FT5:    INC     CURSEC          ;LOOK AT NEXT SECTOR
043122  023727  001136  000037          CMP     CURSEC,#37      ;FORMATTED AT LEAST 1 SURFACE?
043130  003011                          BGT     FT6             ;STOP IF YES
043132  023737  001136  001040          CMP     CURSEC,MAXSEC   ;LAST SECTOR ANYWAY?
043140  003005                          BGT     FT6             ;GET OUT THE
043142  013702  001050                  MOV     POSWP,R2        ;CMPUTE NEXT HEADER ADDRESS
043146  006302                          ASL     R2              ;MAK'WORDS PER SECTOR USEFUL
043150  060201                          ADD     R2,R1           ;ADD TO PRESENT POINTER
043152  000730                          BR      FT4             ;AND GO ON
043154  012700  043302          FT6:    MOV     #FMTBK,R0       ;POINT TO FORMAT BLOCK
043160  006201                          ASR     R1              ;SHIFT REMAINDER FOR WORD COUNT
043162  005101                          COM     R1              ;1'S COMPLEMENT (PICK UP ONE WORD)
043164  010160  000004                  MOV     R1,4(R0)        ;ENTER INTO WRITE BLOCK
```

```
043170  013710  001162                      MOV     ENDSEC,(R0)         ;LOAD UP DISK ADDRESS
043174  013760  001166  000006              MOV     ENDCYL,6(R0)        ;
043202  053710  001164                      BIS     ENDHD,(R0)          ;
043206  005237  001176                      INC     SPECMD              ;SET SPECIAL XFER FLAG
043212  004537  022700                      JSR     R5,DXFER            ;GO FORMAT
043216  005037  001176                      CLR     SPECMD              ;CLEAR SPECIAL XFER FLAG
043222  005737  001322                      TST     ERRFLG  ;ERRORS?
043226  001020                              BNE     FTEX                ;EXIT IF YES
043230  013702  001044                      MOV     MAXCYL,R2           ;SET UP TO CHECK ALL DONE
043234  005202                              INC     R2                  ;
043236  020237  001166                      CMP     R2,ENDCYL           ;END OF DISK?
043242  001415                              BEQ     FMTOUT              ;EXIT IF YES
043244  013737  001162  001136              MOV     ENDSEC,CURSEC       ;LOOK TO NEXT RANSFER IF NO
043252  013737  001164  001140              MOV     ENDHD,CURHD         ;
043260  013737  001166  001142              MOV     ENDCYL,CURCYL       ;
043266  000661                              BR      FT3                 ;AND SET BUFFERS UP AGAIN
                                            ;
                                            ;
043270  012737  000001  001322      FTEX:   MOV     #1,ERRFLG           ;SET ERROR FLAG=1
043276  004537  003342          FMTOUT:     JSR     R5,TSTCTL           ;EXIT
                                            ;
                                            ;
                                            ;
043302  000000                  FMTBK:      .WORD 0                     ;SECTOR HEAD=0
043304  044312                              .WORD ODBUF                 ;OUTPUT BUFFER
043306  177400                              .WORD 177400                ;WORD COUTN WILL BE SET
043310  000000                              .WORD 0                     ;CYLINDER ADDR=0
043312  000010                              .WORD FMTCMD                ;FORMAT COMMAND(NO GO)
                                            ;
                                            ;
                                            ;
                                            ;
                                            ;       14M.ECC TEST NUMBERS 46 >> 50   13-NOV-77
                                            ;
                                            ;       FORMAT CHECK TEST
                                            ;
                                            ;
                                            ;
                                            ;       THIS TEST  CHECKS THE FORMAT
                                            ;       OF ALL THE HEADER WORDS BY
                                            ;       COMMANDING A COMPLETE SURFACE WRITE
                                            ;       AND USING THE FORMATTERS LOGIC TO
                                            ;       CHECK THE HEADER WORDS.
                                            ;
                                            ;
043314  005004                      TN46:   CLR     R4                  ;INITIALIZE SUBTEST NUMBER
043316  004537  004276                      JSR     R5,RQUEST           ;REQUEST 211
043322                              TN46A:
043322  012700  035530                      MOV     #WRCB+8.,R0         ;GET COMMAND ENTRY CB  ADDRESS
043326  012710  000006                      MOV     #WRTCMD,(R0)        ;SPECIFY WRITE COMMAND
043332  013740  001142                      MOV     CURCYL,-(R0)        ;INITIALIZE CYLINDER
043336  013701  001040                      MOV     MAXSEC,R1           ;COMPUTE ONE SURFACE WORD COUNT
043342  013702  001050                      MOV     POSWP,R2            ;LOAD R2 WITH WORD COUNT/ONE SECTOR
043346  010203                              MOV     R2,R3               ;LOAD INTO R3 ALSO
043350  005301                      TN46B:  DEC     R1                  ;START LOOP
043352  100402                              BMI     TN46C               ; EXIT IF DONE
043354  060203                              ADD     R2,R3               ; ADD ANOTHER SECTOR COUNT HERE
```

```
043356  000774                          BR      TN46B           ; GO ON
043360                          TN46C:
043360  005403                          NEG     R3              ;MAKE 2'S COMPLEMENT FOR WORD COUNT
043362  010340                          MOV     R3,-(R0)        ;ENTER INTO TRANSFER BLOCK
043364  012740  036144                  MOV     #TN33,-(R0)     ; SPECIFY OUTPUT BUFFER
043370  013740  001136                  MOV     CURSEC,-(R0)    ;SPECIFY SECTOR
043374  053710  001140                  BIS     CURHD,(R0)      ;OR IN CURRENT HEAD
043400  004537  022700                  JSR     R5,DXFER        ;READ HEADER
043404  005737  001322                  TST     ERRFLG          ;ANY ERRORS?
043410  001031                          BNE     TN46EX          ;EXIT IF YES
043412  012760  000026  000010          MOV     #WRCHK,10(R0)   ;LOAD WRITE CHECK COMMAND
043420  004537  022700                  JSR     R5,DXFER        ;GO WRITE CHECK LAST DATA TRANSFER
043424  012760  000006  000010          MOV     #WRTCMD,10(R0)  ;RELOAD WRITE COMMAND
043432  005737  001322                  TST     ERRFLG          ;ERRORS?
043436  001016                          BNE     TN46EX          ;EXIT IF YES
043440  005737  001144                  TST     DKEND           ;ARE WE AT THE END OF THE DISK MEDIUM
043444  001013                          BNE     TN46EX          ;EXIT IF ALL DONE
043446  013737  001162  001136          MOV     ENDSEC,CURSEC   ;UPDATE DISK ADDRESS
043454  013737  001164  001140          MOV     ENDHD,CURHD     ;
043462  013737  001166  001142          MOV     ENDCYL,CURCYL   ;
043470  005204                          INC     R4              ;UPDATE SUBTEST COUNT
043472  000713                          BR      TN46A           ;GO READ NEXT FORMAT
043474  004537  003342          TN46EX: JSR     R5,TSTCTL       ;GO TO TEST CONTROL
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       12M.ECC TEST NUMBER A47 13-NOV-77
                                        ;
                                        ;
                                        ;       FORMATTING PROGRAM #2
                                        ;    *****************************
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;       THIS PROGRAM FORMATS THE DISK
                                        ;  WITHOUT THE OPERATOR ENTERING ANY PARAMETERS.
                                        ;  THE NUMBER OF SECTORS AND THE NUMBER OF DATA
                                        ;   WORDS IN EACH SECTOR IS SPECIFIED IN THE FIRST PORTION
                                        ;  OF THE DIAGNOSTIC.IF THE OPERATOR WISHES TO FORMAT
                                        ;   A SECTION OR ONE SECTOR OF THE DISK SURFACE  TEST50
                                        ;  ALLOWS HIM TO DO SO.(BAD SECTOR BITS,WRITE PROTECT ETC.)
                                        ;
                                        ;
                                        ;
043500  000000                  FSTHLF: .WORD 0                 ;REVOLUTION FLAG
043502  000000                  ESECTA: .WORD 0                 ;=MAXSEC+1
043504  000000                  ESECTB: .WORD 0                 ;=MAXSEC+2
                                        ;
                                        ;
                                        ;
043506  005004                  TN47:   CLR     R4              ;SUBTEST NUMBER=0
043510  004537  004276                  JSR     R5,RQUEST       ;REQUEST 211
043514  105777  135260                  TSTB    @DCSR           ;CONTROLLER READY?
043520  100402                          BMI     F2T1            ;CONTINUE IF YES
043522  000137  044176                  JMP     F2TEX           ;EXIT IF NO
```

```
043526  005204                          F2T1:   INC     R4              ;SUBTEST NUMBER=1(FORMATTER READY)
043530  004537  016332                          JSR     R5,DISKID       ;POSITION DRIVE SELECT BITS
043534  010277  135242                          MOV     R2,@DUSH        ;SELECT UNIT
043540  010237  044300                          MOV     R2,FMTDSK       ;SAVE FOR LATER USE
043544  005777  135242                          TST     @DSTAT          ;DRIVE READY?
043550  100402                                  BMI     F2T2            ;ECONTINUE IF YES
043552  000137  044176                          JMP     F2TEX           ;EXIT WITH ERROR
043556  005204                          F2T2:   INC     R4              ;SUBTEEST NUMBER=2(DRIVE READY)
043560  013737  001040  043502                  MOV     MAXSEC,ESECTA   ;SET UP REV FLAG
043566  005237  043502                          INC     ESECTA          ;
043572  005037  043500                          CLR     FSTHLF          ;CLEAR REVOLUTION FLAG
043576  013703  001050                          MOV     POSWP,R3        ;LOAD NUMBER OF WORDS PER SECTOR
                                                                        ;INTO REGISTER R3
043602  006303                                  ASL     R3              ;MULTIPLY BY 2 FOR MEMORY USE
043604  062703  000004                          ADD     #4,R3           ;INCLUDE HEADERS IN COUNT
043610  005737  001074                          TST     ECC             ;ECC CONTROLLER?
043614  001402                                  BEQ     F2T3            ;GO NO IF YES
043616  062703  000004                          ADD     #4,R3           ;ADJUST FOR ECC WORDS
043622  062703  044312                  F2T3:   ADD     #ODBUF,R3       ;COMPUTE ADDRESS FOR SECOND SET
043626  005213                                  INC     (R3)            ;MAKE FIRST HEADER WORD VALID
043630  005263  000004                          INC     4(R3)           ;SET SECOND SET(DOESN'T MATTER IF NON ECC
043634  012704  044312                          MOV     #ODBUF,R4       ;ENTER ADDRESS OF FIRST HEADER
043640  013701  001050                          MOV     POSWP,R1        ;LOAD R1 WITH #WORDS PER SECTOR
043644  062701  000010                          ADD     #10,R1          ;INCLUDE HEADERS IN WORD COUNT
043650  005401                          F2T4:   NEG     R1              ;MAKE 2'S COMP
043652  013705  001010                  F2T5:   MOV     DCYL,R5         ;SET R5 AS POINTER TO DISK ADDR'S
043656  013715  001142                          MOV     CURCYL,(R5)     ;LOAD WORD COUNT
043662  010145                                  MOV     R1,-(R5)        ;LOAD WORD COUNT
043664  010445                                  MOV     R4,-(R5)        ;LOAD CORE ADDRESS
043666  011445                                  MOV     (R4),-(R5)      ;LOAD DUSH
043670  053715  044300                          BIS     FMTDSK,(R5)     ;OR IN UNIT SELECT BITS
043674  052777  000010  135076          F2T6:   BIS     #FMTCMD,@DCSR   ;SET DCSR FOR FORMAT
043702  052777  000001  135070                  BIS     #GO,@DCSR       ;SET GO BIT
043710  053737  001302  043676                  BIS     SEKINB,F2T6+2   ;SET SEEK INHIBIT BIT
043716  062737  000004  001136                  ADD     #4,CURSEC               ;INCREMENT SECTOR COUNTER
043724  005737  043500                          TST     FSTHLF          ;FIRST TIME AROUND
043730  001012                                  BNE     F2T7            ;NO,CHECK IF SURFACE DONE
043732  023737  001136  043502                  CMP     CURSEC,ESECTA   ;DONE WITH FIRST PASS
043740  001057                                  BNE     F2T10           ;NO GO FORMAT AGAIN
043742  005237  043500                          INC     FSTHLF          ;YES SET FLAG AND SPIN AGAIN
043746  012737  000002  001136                  MOV     #2,CURSEC       ;SET CURSEC TO 2
043754  000451                                  BR      F2T10           ;AND GO FORMAT AGAIN
043756  023737  001136  001040          F2T7:   CMP     CURSEC,MAXSEC   ;DONE WITH SECOND PASS?
043764  002445                                  BLT     F2T10           ;NO,GO FORMAT AGAIN
043766  005037  043500                          CLR     FSTHLF          ;RSET PASS COUNTER
043772  005037  001136                          CLR     CURSEC          ;RESET CURSEC
043776  023737  001140  001042                  CMP     CURHD,MAXHD     ;DONE WITH SURFACES?
044004  001404                                  BEQ     F2T11           ;YES GO INCREMENT CYLINDER
044006  062737  000200  001140                  ADD     #HDINC,CURHD    ;INCREMENT HEAD COUNTER
044014  000431                                  BR      F2T10           ;AND FORMAT AGAIN
044016  005037  001140                  F2T11:  CLR     CURHD           ;RESET HEAD COUNTER
044022  105777  134752                          TSTB    @DCSR           ;WAIT FORMATTER READY?
044026  100375                                  BPL     .-4             ;
044030  023737  001142  001044                  CMP     CURCYL,MAXCYL   ;END OF DISK
044036  001462                                  BEQ     F2TOUT          ;EXIT IF YES
044040  005237  001142                          INC     CURCYL          ;INCREMENT TRACK COUNTER IF NO
044044  043777  001302  134726                  BIC     SEKINB,@DCSR    ;CLEAR SEEK INHIBIT
```

```
044052 043737 001302 043676           BIC    SEKINB,F2T6+2    ;CLEAR SEEK INHIBIT BIT
044060 022324                         CMP    (R3)+,(R4)+      ;INDEX POINTERS TO TRACK ADDRESS
044062 005213                         INC    (R3)             ;INCRENEBT BOTH TRACKS ADDR
044064 005214                         INC    (R4)             ;
044066 005263 000004                  INC    4(R3)            ;SET SECOND HEADER WORDS
044072 005264 000004                  INC    4(R4)            ;
044076 024344                         CMP    -(R3),-(R4)      ;RESET POINTERS TO FIRST
                                                              ;HEADER WORD ADDRESS
044100 105777 134674         F2T10:   TSTB   @DCSR            ;WAIT FORMATTER READY
044104 100375                         BPL    .-4              ;
044106 013714 001136                  MOV    CURSEC,(R4)      ;ENTER NEW SECTOR NUMBER
044112 013764 001136 000004           MOV    CURSEC,4(R4)     ;ENTER NEW SECTOR NUMBER
044120 053714 001140                  BIS    CURHD,(R4)       ;AND HEAD
044124 053764 001140 000004           BIS    CURHD,4(R4)      ;
044132 013713 001136                  MOV    CURSEC,(R3)      ;
044136 013763 001136 000004           MOV    CURSEC,4(R3)     ;
044144 053713 001140                  BIS    CURHD,(R3)       ;ENTER HEAD
044150 053763 001140 000004           BIS    CURHD,4(R3)      ;
044156 005213                         INC    (R3)             ;INCREMENT SECOND SECTOR
044160 005263 000004                  INC    4(R3)            ;
044164 005777 134610                  TST    @DCSR            ;ERRORS?
044170 100230                         BPL    F2T5             ;CONTINUE IF NO
044172 005237 001322                  INC    ERRFLG           ;SET ERROR FLAG IF YES
044176 012737 000001 001322  F2TEX:   MOV    #1,ERRFLG        ;SET ERROR FLAG TO 1
044204 004537 003342         F2TOUT:  JSR    R5,TSTCTL        ;EXIT HERE
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;
                                      ;       XOR.PM            9-NOV-77
                                      ;
                                      ;
                                      ;       ECC WORD CORRECTION ROUTINE
                                      ;      ********************************
                                      ;
                                      ;
                                      ;
                                      ;       THIS ROUTINE USES THE ECC PATTERN AREGISTER
                                      ;       AND ONE ENTRY POINT TO FORM A CORRECTED CORE
                                      ;       WORD DURING THE ECC TESTS.AT EXIT REGISTER R3 WILL
                                      ;       CONTAIN THE CORRECTED WORD.
                                      ;
                                      ;
                                      ;       CALL:    JSR       APCORR,PC
                                      ;
                                      ;       AT ENTRY           4(R5)=CORE WORD TO BE CORRECTED
                                      ;
                                      ;
                                      ;
                                      ;
044210 016546 000004         APCORR:  MOV    4(R5),-(SP)      ;SAVE CORE IMAGE
```

```
     044214  011603                          MOV     (SP),R3         ;USE R3 FOR WORK
     044216  017702  134576                  MOV     @ECCPW,R2       ;R2=PATTERN WORD
     044222  032702  000001         APR1:    BIT     #BIT0,R2        ;RIGHT JUSTIFY PATTERN REISTER
     044226  001002                          BNE     APR2            ;GO ON IF DONE
     044230  006202                          ASR     R2              ;SHIFT RIGHT IF  BIT 0=0
     044232  000773                          BR      APR1            ;CHECK IT
     044234  040203                 APR2:    BIC     R2,R3           ;./NOT. PATTERN WITH DATA
     044236  042602                          BIC     (SP)+,R2        ;.NOT DATA WITH PATTERN
     044240  050203                          BIS     R2,R3           ;OR PATTENR+DATA
     044242  000207                          RTS     PC              ;LEAVE ALL DONE
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
     044244  000000                 BSECT:   .WORD 0                 ;BEGINNING SECTOR
     044246  000000                 ESECT:   .WORD 0                 ;ENDING SECTOR ADDRESS
     044250  000000                 BSURF:   .WORD 0                 ;BEGINNING SURFACE
     044252  000000                 ESURF:   .WORD 0                 ;ENDING SURFACE NUMBER
     044254  000000                 BTRK:    .WORD 0                 ;BEGINNING TRACK
     044256  000000                 ETRK:    .WORD 0                 ;ENDING TRACK
     044260  000000                 BDSEC:   .WORD 0                 ;BAD SECTOR FLAG
     044262  000000                 WRPT:    .WORD 0                 ;WRITE PROTECT FLAG
     044264  000000                 CSECT:   .WORD 0                 ;CURRENT SECTOR
     044266  000000                 CSURF:   .WORD 0                 ;CURRENT SURFACE
     044270  000000                 CTRK:    .WORD 0                 ;CURRENT TRACK
     044272  000000                 WRD1:    .WORD 0                 ;HEADER WORD ONE
     044274  000000                 WRD2:    .WORD 0                 ;HEADER WORD 2
     044276  000000                 NORM:    .WORD 0                 ;NORMAL SECTOR FLAG
     044300  000000                 FMTDSK:  .WORD 0                 ;CURRENT DISK DUSH BITS
                                             ;
                                             ;
                                             ;
                                             ;;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;
                                             ;       OUTPUT DATA BUFFER
                                             ;
                                             ;
                                             ;       THIS BUFFER CONTAINS DATA FOR ALL DISK WRITE OPERATIONS
                                             ;    THIS BUFFER DATA IS FORMATTED PRIOR TO ANY WRITE OPERATIONS
                                             ;    AS PART OF THE TESTING SEQUENCE.
                                             ;
             044312                          .=.+10         ;
     044312  000000                 ODBUF:   .WORD 0        ;ORIGIN ADDRESS
                                             ;
             047314                          .=.+1536.      ;BUFFER IS 768. WORDS LONG
                                             ;
```

```
                                        ;
                                        ;       INPUT BUFFERS
                                        ;
                                        ;
                                        ;   THESE DATA BUFFERS ARE ALWAYS USED DURING A READ COMMAND
                                        ;   AND MAY VARY IN SIZE DEPENDING ON SIZE OF CORE.
                                        ;
                                        ;
047314  000000                 IDBUF1: .WORD 0         ;DRIVE 0 INPUT BUFFER AREA
        050314                         .=.+510.        ;EACH AREA IS 512. WORDS LONG
050314  000000                 IDBUF2: .WORD 0         ;DRIVE 1 INPUT BUFFER AREA
        051314                         .=.+510.        ;
051314  000000                 IDBUF3: .WORD 0         ;DRIVE 2 INPUT BUFFER AREA
        052314                         .=.+510.        ;
052314  000000                 IDBUF4: .WORD 0         ;DRIVE 3 INPUT BUFFER AREA
        053314                         .=.+510.        ;
053314  000000                 IDBUF5: .WORD 0         ;
        054314                         .=.+510.        ;
                                        ;   512. WORDS OF BUFFER AREA ARE ASSIGNED TO EACH OF 4 DRIVES
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
        000001'                         .END
```

SYMBOL TABLE

ACTMSG	012477	ADR	015666	ADREX	016034
ADRRST	034742	ADRSAV	034712	ADRSTP	015644
ADR1	016004	ADR2	016012	ADR3	016030
APCORR	044210	APR1	044222	APR2	044234
ASCII	006754	ASRMSG	013525	AUTMSG	002241
AUTO	001036	AUTOSW	001100	BADBIT	001276
BADNUM	006750	BDSEC	044260	BIT0 =	000001
BIT1 =	000002	BIT10 =	002000	BIT11 =	004000
BIT12 =	010000	BIT13 =	020000	BIT14 =	040000
BIT15 =	100000	BIT2 =	000004	BIT3 =	000010
BIT4 =	000020	BIT5 =	000040	BIT6 =	000100
BIT7 =	000200	BIT8 =	000400	BIT9 =	001000
BOOT	003500	BSECT	044244	BSURF	044250
BTRK	044254	BUFCLR	001300	CASCI	007014
CECC1	036522	CECC2	036544	CECC3	036600
CECC4	036630	CECC5	036672	CECC6	036736
CERP	021072	CEXIT	006740	CHKERR	022132
CIMODE	005522	CI1	005352	CI2	005366
CI3	005372	CI4	005374	CI5	005412
CI6	005414	CI7	005420	CI8	005424
CLI	005346	CLRBUF	004034	CLRB1	004052
CLRCNT=	007012	CLRERR	021030	CLRFTR	021510
CMDTB	005434	CMPCNT	001134	CMPLP	022510
CMSSG	010514	CNSTNT	002712 G	CNTFLG	036140
CNTINC	036142	COCEX	006734	COCTDN	006716
COCTER	006726	COCTI	006660	COEMSG	007425
COMMSG	011456	CONBL	005522	CONC	005732
CONTST	001230	CONTU	015342	CONU	005734
CORE	003010	CORE1	003022	CR =	000015
CRCODE	005742	CRCXEX	040614	CRC1	037026
CRC10	037314	CRC11	037366	CRC12	037446
CRC13	037476	CRC14	037566	CRC15	037620
CRC16	037652	CRC17	037700	CRC2	037102
CRC20	037726	CRC21	040024	CRC22	040054
CRC23	040102	CRC24	040166	CRC25	040216
CRC26	040240	CRC27	040330	CRC3	037146
CRC30	040360	CRC31	040402	CRC32	040446
CRC33	040502	CRC34	040532	CRC35	040554
CRC36	040576	CRC4	037174	CRC5	037216
CRC6	037240	CRC7	037262	CSECT	044204
CSURF	044266	CSWR =	177570	CSWTCH	001250
CTLERR	023474	CTRK	044270	CURCYL	001142
CURDSK	001234	CURHD	001140	CURSEC	001136
CURTN	001232	CYLADR	002542	CYLDN	034124
CYLMSG	001732	CYLMSK=	176000	CYLUP	034122
CYMSG	011612	DARMSG	012644	DATA	006752
DATBUF	011222	DATCMP	017106	DATE	010676
DATERR	001320	DATEW	011222	DATMSG	013034
DBAR =	001004	DCAR	001004	DCCMSG	012560
DCMPA	017146	DCMSG	002066	DCRMSG	012602
DCSMSG	002054	DCSR	001000	DCSRA	001070
DCYL	001010	DCYMSG	012706	DECGSM	002060
DERECC	023402	DERMSG	012750	DERR	001014
DFMSG	013456	DISKHD	010032	DISKID	016332
DISKIN	007740	DISKP	001642	DISKTP	013446
DKDNEX	017332	DKEND	001144	DKINIT	016352
DKI1	016370	DKI2	016506	DKI3	016426

SYMBOL TABLE

DKI4	016440	DSCAR	001034	DSERR	023410
DSERRA	023462	DSKDN	017304	DSKMSG	012727
DSKSAV	001254	DSKTST	035000	DSTAT	001012
DTADR	002024	DTCMER	017172	DTCMEX	017162
DTCMLP	017136	DTCNT	001152	DTE	002045
DTERR	023320	DTERRA	023342	DTEXIT	023306
DT1	010716	DT10	011140	DT2	010770
DT3	011006	DT4	011024	DT5	011042
DT6	011066	DT7	011102	DUMSG	011510
DUNFT	013452	DUSH	001002	DUSMSG	012623
DWCMSG	012665	DWCNT	001006	DXFER	022700
DXF1	022774	EBMSG	013176	ECBMSG	012771
ECC	001074	ECCBB	013214	ECCDEX	041642
ECCD1	041136	ECCD10	041352	ECCD11	041400
ECCD12	041420	ECCD13	041446	ECCD14	041466
ECCD15	041514	ECCD16	041534	ECCD17	041562
ECCD2	041164	ECCD20	041602	ECCD21	041630
ECCD22	041636	ECCD3	041204	ECCD4	041232
ECCD5	041252	ECCD6	041304	ECCD7	041324
ECCERR	041652	ECCEXT	041044	ECCMSG	002137
ECCCME	042752	ECCOM1	042604	ECCOM2	042670
ECCOM3	042730	ECCOM4	042746	ECCPB	001016
ECCPW	001020	ECCWBK	041052	ECCWRT	040620
ECCW1	001264	ECCW2	001266	ECDC1	041734
ECDC10	042130	ECDC11	042156	ECDC12	042176
ECDC13	042220	ECDC14	042242	ECDC15	042270
ECDC16	042310	ECDC17	042332	ECDC2	041754
ECDC20	042376	ECDC21	042436	ECDC22	042460
ECDC24	042500	ECDC3	041774	ECDC4	042014
ECDC5	042050	ECDC6	042070	ECDC7	042112
ECD17	042362	ECEROT	041672	ECHOEM	005534
ECOMM	042546	ECPMSG	013012	ECRMSG	012576
EDCEX	042504	EINMSG	007365	EMULAT	001274
ENDCYL	001166	ENDFLG	001146	ENDHD	001164
ENDMSG	001456	ENDPAR	022436	ENDSEC	001162
ENDTST	022576	ENTMSG	001400	ERCNTM	010544
ERDATA	013220	EREXIT	023554	ERIMGE	001170
ERMSG	012352	ERPRIN	001226	ERRBLK	035334
ERRCNT	001314	ERRFLG	001322	ERRID	023502
ERROR	001324	ERRTAB	001336	ERRWRD	001160
ERSET	017272	ESECT	044246	ESECTA	043502
ESECTB	043504	ESURF	044252	ETRK	044256
EXIT6	022136	FORMSG	012367	FIEXIT	016114
FIUNIT	016042	FI1	016056	FI2	016066
FLTCLR=	000022	FMTBK	043302	FMTCLR	003614
FMTCMD=	000010	FMTDK	013240	FMTDSK	044300
FMTD1	013334	FMTD2	013406	FMTK1	013260
FMTOUT	043276	FMTRDY=	000200	FMTTR	007072
FOLP	017032	FORMAT	016736	FSTHLF	043500
FTEX	043270	FTTMSG	012417	FT1	043002
FT2	043030	FT3	043032	FT4	043034
FT5	043116	FT6	043154	FUNC =	000136
F1LP	016756	F2LP	016774	F2TEX	044176
F2TOUT	044204	F2T1	043526	F2T10	044100
F2T11	044016	F2T2	043556	F2T3	043622
F2T4	043650	F2T5	043652	F2T6	043674
F2T7	043756	F3LP	017016	F4LP	017042

```
F5LP    017060          GO     = 000001          GOTEST  003676
HALT8   003776          HDERR    017264          HDINC = 000200
HDMSG   001703          HDMSK =  170177          HDRC1   001260
HDRC2   001262          HELP     013652          HERCNT  001316
HLPMSG  014046          HLP1     013726          HLP2    013742
HLP3    013760          HPMSG    014004          IBFTBL  001200
ICNT    001156          IDBUF1   047314          IDBUF2  050314
IDBUF3  051314          IDBUF4   052314          IDBUF5  053314
IDERR   023570          IDLP     023536          IMPBLK  031712
INCHD   022670          INITLP   003562          INTCAL= 000125
INTCNT= 000033 G        INTFLG   001150          INTGO = 000101
INTMSG  002114          INTON =  000100          INTSAV  001052
INTTBL  001122          INTVEC   001072          INWAIT  004446
IOERR   005744          IPFLG    001154          ISECFB= 025450
ISECRB= 025462          ISECWB=  024652          LDEX    007630
LDRD    017530          LD1      007566          LD1A    007562
LD10    007726          LD2      007604          LD3     007632
LD4     007640          LD5      007660          LD6     007706
LD7     007714          LF     = 000012          LFCODE  005740
LOADIN  007502          LPCNT    001272          LRERR   017646
LREX    017634          LRLP1    017554          LRLP2   017606
LRLP3   017602          LRLP4    017612          LSTTST  003702
MAXCYL  001044          MAXHD    001042          MAXHED  001060
MAXSEC  001040          MAXTFR   001270          MCPU    001102
MDRIVE  001106          MEMEX =  030000          MICRO   001066
MONTH   011154          MSGADR   005526          MSG1    014131
MSG10   014754          MSG11    015021          MSG12   015046
MSG13   015101          MSG14    015152          MSG15   015177
MSG16   015245          MSG17    015301          MSG2    014210
MSG3    014271          MSG4     014347          MSG5    014436
MSG6    014513          MSG7     014574          MSG8    014634
MSG9    014705          MULCPU   001076          MULMSG  002204
NCR     005530          NINPUT   001242          NONCON  001244
NOPCMD= 000077          NORM     044276          NOSTOP  001222
NUMBER  005540          OCTAL    006640          ODBUF   044312
OLDHDR  001064          OUTBUF   010574          OVBLK   033350
PARINT  003532          PARMB    005544          PASCNT  001236
PASINC  003720          PASINH   001224          PASMSG  010403
PASS    011640          PATGEN   036064          PINC1   003736
PINMSG  007320          PMGR     004446          PMG1    004506
PM1     004460          PM2      004540          PM3     004544
PM4     004562          PM5      004624          PNTMSG  010334
POLY    001256          POSWP    001050          POUT    013104
PPARMB  005546          PRDATA   010306          PRINT   010126
PRMSG   010322          PROMPT   007062          PS    = 177776
PSMSG   012006          PSS1     011766          PSS2    011762
PSS3    012000          PSS4     011744          PSWORD  003032
PTCH    035022          PT1      013120          P2MSG   006424
P3MSG   006603          QDUMY =  000336          QON   = 000100
QPRDY = 000400          QUEST    007060          RAMBLE  001252
RANB    015550          RANBLK   016130          RANBUF  015524
RANDOM  015564          RAN1     016166          RAN10   016324
RAN2    016172          RAN3     016216          RAN4    016246
RAN5    016142          RAN6     016274          RAN7    016320
RAN8    016222          RD       034430          RDBLK   023614
RDCDE   040632          RDCMD =  000004          RDECC   041064
RDERR   001172          RDFMT =  000012          RDHDBK  025436
```

SYMBOL TABLE

RDUSH	001054	REBOOT	004020	RECAL =	000025
REEPSK	002552	REGSAV	015440	REGO	002722
REG1	015460	REPBIT=	100000	REPFLG	001220
REPPSG	001536	REQFLG	005532	REQUST	036244
RERMSG	001574	REST	015510	RFRMSG	012447
RGR	013636	RGREAD	013564	RQEX	004374
RQUEST	004276	RQ1	004404	RQ2	004330
RQ3	004370	RTRYIN	001332	RTZCMD=	000024
RUB	005736	RUN	010064	RUN1	010122
RWCYL	001056	RWHCMD=	000016	SAVE	015474
SCBLK	011420	SCBLTP	011434	SCOADR	011446
SCOPE	011236	SC1	011270	SC2	011312
SC2A	011342	SC2B	011354	SC2C	011370
SC2D	011400	SC2H	011402	SC5	011406
SECINC	022612	SECMSG	001652	SECMSK=	177600
SEKBK	004240	SEKCMD=	000002	SEKCMP	001306
SEKER	004234	SEKFLP=	000010	SEKINB	001302
SEKOUT	004172	SEKOU1	004216	SEKOU2	004212
SEKTIM	004066	SELA	021502	SELECT	021406
SKCNT	001112	SKDON	004260	SKDONE=	000074
SKFLOP=	007400	SKIPER	001312	SKLP	004152
SKLP1	004144	SKPBAD	001246	SLERR	020236
SLEX	020226	SLHMSG	007066	SLIDE	020144
SLIDE1	020212	SLIDE2	020216	SL1	020164
SOFERR	001330	SOFER1	001326	SOFTRY	001132
SPACE =	000040	SPECMD	001176	SREPFL	003330
SRTMSG	001426	SRTTN	001210	START	003000
STARTA	003072	STCLR	003260	STDMSG	001505
STOP	003412	STOP8	003474	STPDSK	001214
STPTN	001210	STPTST	003504	STRBUF=	****** GX
STRCNT	001062	STRMSG	006507	STRRD =	000032
STRTDS	001216	STRTN	001212	SUBMSG	012527
SYSCLR=	000001	S0ORG	002652 G	S1ORG	002612 G
TDATE	001110	TER1	012042	TER2	012072
TER3	012100	TER4	012156	TER5	012210
TER6	012346	TER7	012236	TER8	012334
TER9	012260	TER9A	012302	TESTP	007176
TESTP1	007230	TEXA	031316	TIME =	177777
TIMLP	001310	TKB	005724	TKS	005722
TMPADR	034772	TMPCNT	042514	TNA16	036464
TNA16E	036770	TNA17	036774	TNA20	025740
TNA21	041076	TNA22	041674	TNA23	042516
TNINT	004006	TNINTA	004000	TNORG	002276
TNORGA	002420	TNTST	003516	TN0	016530
TN0A	016612	TN0B	016676	TN0C	016712
TN0ERR	016726	TN0OUT	016732	TN1	017412
TN1A	017434	TN1B	017452	TN1C	017470
TN1D	017506	TN1ERR	017524	TN10	023626
TN11	023670	TN12	023760	TN13	024042
TN13EX	024124	TN13LP	024056	TN14	024130
TN15	024220	TN15LP	024244	TN16	024314
TN16EX	024646	TN17	024664	TN17A	024746
TN17EX	025416	TN17FB	025450	TN17RB	025462
TN2	017654	TN2A	017672	TN2B	017710
TN2C	017726	TN2D	017744	TN2E	017762
TN2ERR	017776	TN2OUT	020002	TN20	025474
TN20EX	026324	TN21	026350	TN21A	026636

```
TN21EX  027152      TN22    027170      TN22A   027442
TN22EX  030152      TN23    030160      TN23A   030312
TN23B   030452      TN23C   030530      TN23EX  030524
TN24    030542      TN24A   030672      TN24B   030706
TN24C   030652      TN24D   030716      TN24EX  031352
TN25    031360      TN25EX  031666      TN26    031724
TN26A   032010      TN26B   032000      TN26C   032156
TN26D   032212      TN26E   032260      TN26EX  032326
TN26F   032302      TN26OT  032322      TN27    032346
TN27A   032532      TN27C   032640      TN27D   032746
TN27E   033050      TN27EX  033334      TN27F   033106
TN27G   033276      TN27H   033242      TN27J   033362
TN27K   033422      TN27L   033520      TN27M   033530
TN27N   033542      TN27OT  033306      TN27P   033626
TN27R   033650      TN3     020006      TN3A    020024
TN3B    020046      TN3C    020064      TN3D    020102
TN3E    020120      TN3ERR  020136      TN3OUT  020132
TN30    033712      TN30A   033744      TN30B   034032
TN30E   034126      TN30EX  034116      TN30F   034140
TN31    034204      TN32    035532      TN32A   035752
TN32EX  036060      TN32LP  035574      TN33    036144
TN33EX  036210      TN34  = ****** GX   TN35  = ****** GX
TN36  = ****** GX   TN37  = ****** GX   TN4     020244
TN4A    020354      TN4B    020374      TN4C    020464
TN4D    020500      TN4E    020566      TN4ERR  021020
TN4EX   020326      TN4F    020602      TN4FIN  021024
TN4G    020712      TN4H    021014      TN40  = ****** GX
TN41  = ****** GX   TN42  = ****** GX   TN43  = ****** GX
TN44  = ****** GX   TN45    042762      TN46    043314
TN46A   043322      TN46B   043350      TN46C   043360
TN46EX  043474      TN47    043506      TN5     021110
TN5A    021200      TN5AA   021146      TN5AB   021156
TN5B    021244      TN5C    021302      TN5D    021336
TN5ERR  021400      TN5OUT  021374      TN50  = ****** GX
TN6     021544      TN6C    021636      TN6CHK  022112
TN6D    021660      TN6E    021674      TN6ERR  022104
TN6F    021704      TN6J    021764      TN6K    022004
TN6OUT  022100      TN7     022216      TN7A    022244
TN7B    022304      TN7C    022336      TN7D    022406
TN7ERR  022430      TN7OUT  022424      TOS1    006146
TOS2    006166      TPB     005730      TPS     005726
TRASH   005542      TRSH    006270      TRSH1   006302
TSTACT  005524      TSTADR  001356      TSTART  003044
TSTBDR  001366      TSTCA   003222      TSTCB   003254
TSTCON  003106      TSTCTL  003342      TSTDSK  021100
TSTINT  005750      TST1    003404      TST2    003366
TST3    003466      TS1     006000      TS2     006026
TS3     006030      TS3A    006060      TS4     006112
TS5     006120      TTBAD   005336      TTBUF   005550
TTERR   012030      TTISEV  005770      TTOINT  005716
TTOSEV  006136      TT010   005266      TT011   005274
TT012   005302      TT013   005310      TT014   005314
TT015   005324      TT02    005150      TT03    005156
TT05    005174      TT06    005200      TT07    005210
TT08    005226      TT09    005240      TTSTAR  006170
TTWORK  005746      TTYACT  005536      TTYIN   004650
TTYINT  005720      TTYOUT  005134      TTYPUT  001104
```

```
TTYSET  015414       TTYSEX  015436       TTZ     004672
TTZZ    004710       TT04    005164       TT1     004734
TT10    005122       TT2     004760       TT3     004766
TT4     005002       TT5     005012       TT6     005014
TT6A    005020       TT7     005026       TT8     005104
TT9     005114       T1MSG   010434       T15EX   024310
T16WB   024652       T2MSG   010464       T31EX   034706
T31RLP  034454       T31WLP  034242       T6DONE  022140
UNIFLD= 030000       UNIMSK= 147777       UNIT    001240
UPD     036034       WAIT    017342       WAIT1   022164
WAIT1A  022174       WHSCCB  027156       WHSCRC= 000014
WORCMD= 000020       WPMSG   001765       WPORCB  026336
WPRBIT  001304       WPSEC   001046       WRC     035356
WRCB    035520       WRCEX   035506       WRCHK = 000026
WRD1    044272       WRD2    044274       WRPT    044264
WRTBLK  023602       WRTCMD= 000006       WRTECC  040644
WRTERR  001174       WRT1    040662       WSMSG   011550
WTEXIT  017374       WTPL    017356       WTPLA   017352
WTPL1   017410       $RN1    015612       $RN2    015614
$RN3    015620       $R1     036276       $R2     036334
$R3     036374       $R4     036426       $R5     036460
$TN20   026114       $TN20A  026120       $TN20B  026236
$TN27   032342       $TST1   003124       $T61    021612
$T62    021622       $$CON = ****** G     $2      036152
$222    036170       $3      036202       $4      036214

. ABS.  054314    000
        000024    001
ERRORS DETECTED:  1
FREE CORE:  3283. WORDS
PHOENX,PHOENX<SMD1.P11,SMD2.P11
```

XYLOGIC OEM COMPONENTS GROUP, INC.

42 Third Avenue

Burlington, Massachusetts 01803

617-272-8140

Phoenix 200 Disk Controller

User Interface Manual

Dwg. No. 1043-04

Revision B

Date June 3, 1976

TABLE OF CONTENTS

USER INTERFACE MANUAL

TABLE OF CONTENTS (CONT'D)

1.0 Introduction

It is the intent of this manual to provide the user with sufficient technical information to enable him to successfully design an adapter to interface the Phoenix 200 Disk Formatter to current generation computers with a minimum of difficulty.

The information contained in this manual is principally concerned with the physical, electrical, and timing interface provided to the user by the Phoenix 200 Disk Formatter, and the use thereof by the user in any computer adapter design.

The functional capabilities, performance characteristics, and programming interface of the Phoenix 200 Disk Formatter are explained in great detail in the Phoenix 211 Disk Controller Programming Reference Manual. The user should be equally familiar with the contents of the programming manual, as well as this manual before attempting the detailed design of any computer adapter for the formatter.

This manual will be expanded and updated as required to provide the user with complete, accurate, interfacing information for the Phoenix 200 Formatter.

Any questions relating to the content of this manual should be directed to:

Engineering Department
Xylogic OEM Component Group, Inc.
42 Third Avenue
Burlington, Mass. 01803
617-272-8140

4.0 User Interface Signal Definitions

Note: All formatter interface signals are low active.

4.1 Data Buss: DBOL-DB15L (Bidirectional)

The 16 Bidirectional DATA lines are used to transmit Register Data and DMA Data from/to the computer Interface Card from/to the Formatter. Data Buss timing is explained in detail in the Timing Diagram Section.

4.2 Register Control: RCO(L) - RC3(L) (to Formatter)

The four Register control lines are used to select a register located in the formatter for purposes of reading or writing. The information on these lines along with the appropriate strobe define which Register will be written or read.

4.3 Word Count Overflow: WCROVFL (to Formatter)

A pulse or a High to Low Transition On Line indicates to the formatter that the Word Count Register in the user interface card has overflowed meaning then the most significant bit of that register has made a TRUE to FALSE transition. DMA Requests will no longer be issued by the 211 formatter when the overflow condition occurs. Refer to DMA timing section.

4.4 Computer Initilize: INIT (L) (to Formatter)

This is a pulse which causes the formatter to revert to the quiescent state; i.e. reset. This line may be used to disable the formatter if the computer power is lost, or to clear the formatter in the case of illegal commands or unusual conditions.

4.5 Direct Memory Data In: DMDATI (L) (to Formatter)

When Active, this signal causes the Formatter to place DMA data on the bidirectional data bus for use by the user computer interface. The data placed on the bus interface is data that the formatter has read from the disk. The user may strobe data 200NS after DMDATI(1) is activated by user. Refer to DMA Timing section.

4.6 Direct Memory Data Out: DM DATO (L) (to Formatter)

This is a pulse which is used by the formatter to strobe data supplied by the computer interface during a DMA write operation. The user puts the DMA data to be read by the formatter on the bus and waits a minimum of 150NS before activiating DMDATO(L).

The formatter normally strobes data into one of the internal FIFO memories on the leading edge (High to Low) of DMDATO(L). DMDATO(L) must be a minimum of 100NS wide and maximum frequency is 2.0 to MHZ. Refer to DMA timing section.

4.7 Interrupt Acknowledge: INTRACKL (to Formatter)

This is a pulse which is issued by computer interface to inform the formatter that the host computer has acted upon the Formatter's Interrupt request. This causes the Interrupt Request signal to be cleared. Interrupt Request is also cleared when any new command is issued to the formatter.

4.8 Direct Memory Direction: DMDIRL (from Formatter)

This is a level which when active (Low) indicates that the DMA Transfer taking place is from the formatter to the user Interface, i.e. Read. When this signal is false the direction is defined as computer interface to Formatter, i.e. WRITE. This signal state is defined when any data transfer command is initiated and the formatter becomes "busy."

4.9 Load "Q" Buss Address Register: LDQBAR (L) (from Formatter)

This is a pulse which is used by the user interface to load Data via the Formatter Data Buss into the user Interface Buss address Register. This signal is only used in "Q" mode of operation.

4.10 Load "Q" Memory Extension Register: LDMEX (L) (from Formatter)

This is a pulse which is used by the user interface to load data via the formatter Data Buss into the user interface memory extension of the Command Status Register. This signal is only used in the "Q" mode of operation.

4.11 Load "Q" Word Count Register: LDQWCR (L) (from Formatter)

This is a pulse which is used by the user interface to load data via the formatter Data Buss into the user Interface Word Count Register. This signal is used only in the "Q" mode of operation.

4.12 Data In Strobe: DSDATIL (to Formatter)

This signal gates the register data as defined by the register control lines to the computer interface on the Data Buss. Refer to Register timing section for data validity.

4.13 Formatter Error Condition: FECOND (L) (to Formatter)

This is a level from the computer interface signifying that an error condition exists in the computer interface. Any command currently in progress (Read, Write) will be aborted. The general error bit will be set in the formatter and an interrupt generated.

4.14 Data Out Strobe: DSDATOL (to Formatter)

This is a pulse which causes the formatter to copy the data from the Data Buss into the register selected by the Register Control lines. Refer to Register timing for further information.

4.15 Enable MMu Control: ENBMM (L) (to Formatter)

A pulse (200ns min. 500ns max) on this line causes the "Q" mode control to execute one instruction sequence. This must only be issued if the host computer will not be accessing the formatter or the computer interface for the next 800ns. This includes DMA, Interrupt, Register reading and writing. This signal is used only in Command Queue Mode.

4.16 System Initialize: SYSINITL (from Formatter)

This is a pulse generated by the user to initialize all logic within the formatter to a defined quiescent state. This signal is also generated by the Formatter Clear Command.

4.17 Interrupt Request: INTRL (from Formatter)

When active this signal informs the computer interface that the formatter has completed the commanded operation. User software should interrogate formatter registers to determine if the commanded operation was performed successfully. This signal, once active, is reset when the leading edge of the user generated Interrupt Acknowledge Signal is generated.

4.18 Load Low Byte Of Command and Status Register: LDLCSRL (to Formatter)

This is a pulse which causes the formatter to copy Data from the low order Byte of the Data Bus (DBØ-DB7) into the low order byte (Bits Ø-7) of the Command and Status Register. (Refer to Register Timing Diagram).

4.19 Load High Byte of Command and Status Register: LDHCSRL (to Formatter)

This is a pulse which causes the formatter to copy data from the high order Byte of the Data Bus (DB8-DB15) into the high order Byte (Bits 8-15) of the Command and Status Register. (Refer to Register Timing Diagram).

4.20 DMA Request Signal DMR(L) (FROM Formatter)

1. This signal is used by the formatter to initiate DMA transfers via the user interface, DMR(L).

2. DMR(L) is associated with the state of the Formatter internal FIFO memory.

3. In a Write operation the request signal will be active when the controller is busy, the associated FIFO memory buffer is not full, and the Word Count Overflow (WCROVFL) Signal has not been received from the user interface.

4. In a Read operation the request signal will be active when the controller is busy, the associated FIFO memory buffer contains any disk data, and the Word Count Overflow (WCROVFL) Signal has not been received from the user interface.

5. The user interface must sample the state of the DMA request signal.

6. The DMACK(L) signal is utilized to enable the user interface DMA cycle look-a-head capability required for multiple cycle DMA transfers on certain computers.

4.21 DMA Acknowledge Signal: DMACK(L) (To Formatter)

1. This signal acknowledges the current DMA Request.

2. The DMACK(L) Signal is used to detect the state of the formatter's FIFO memory and modify the DMA Request Signal accordingly.

3. DMA Request can be sampled 200ns min. after the leading edge of the DMACK Signal.

5.0 Loading and Reading of Internal Formatter Registers

The Phoenix 200 Formatter contains a number of internal 16 bit registers which are necessary to be loaded with parameters or interrogated under direct program control by the user. The following sections are devoted to defining the user procedures to be followed in register addressing, loading and reading program control operations.

Note that some of the internal registers provided in the Phoenix 200 Formatter are applicable only to formatters equipped with special extra cost options.

5.1 Phoenix 200 Formatter Register Addressing

5.1.1 A register is selected by the user for reading or writing under direct program control by controlling the state of four register select signals [RCO(L) - RC3(L)] provided.

5.1.2 Specific internal registers contained in the Phoenix 200 Formatter and the associated assigned register select signal state required to address them are contained in Table 1.

Specific functional read, write capabilities of each register is also indicated, as well as those registers that are applicable to specific optional configurations of the formatter.

5.2 Register Loading

5.2.1 Register Load Sequence Overview

Internal registers in the Phoenix 200 Formatter are loaded by:

1. Selecting the register to be loaded by appropriately controlling the states of Register Select Signals RCO(L) - RC3(L).

2. Gating onto the 16 bit formatter data bus (DBOL-DB15L) the data that is to be loaded into the selected register.

3. Supplying a 250NS minimum strobe pulse (DSDATOL) to be used by the formatter to strobe the data on the formatter data bus into the selected register.

The user interface controls completely the register loading operation, and the associated timing. The only timing requirement placed upon the user is to guarantee the validity of the data and register select buss during the DSDATOL strobe pulse.

5.2.2 Command & Status Register Loading

The Command and Status Register is loaded by:

1. Gating onto the 16 Bit Formatter data bus (DBØL-DB15L) the data that is to be loaded into the command and status register.

2. Supplying a 250ns minimum strobe pulse (LDLCSRL or LDHCSRL) to be used by the formatter to strobe data into the command and Status Register.

3. The LDLCSRL Signal is used to load the low order byte of the Command and Status Register. The LDHCSRL Signal is used to load the high order byte of the Command and Status Register.

4. The LDLCSRL and LDHCSRL signals may be issued separately or together. The DSDATOL signal should be suppressed when the loading of the Command and Status Register takes place.

TABLE 1

PHOENIX 200 DISK FORMATTER

INTERNAL REGISTER SELECT STATE ASSIGNMENTS

	RC3(L)	RC2(L)	RC1(L)	RCØ(L)	REGISTER SELECTED	READ/WRITE USAGE
	0	0	0	0	Control & Status Register**	:READ ONLY
	0	0	0	1	Unit, Select, Head Register	:READ/WRITE
	0	0	1	0	Not Used	:READ/WRITE
	0	0	1	1	Disk Word Count Register	:WRITE ONLY
	0	1	0	0	Cylinder Address Register	:READ/WRITE
	0	1	0	1	Disk Status Register	:READ/WRITE
	0	1	1	0	Error Register	:READ ONLY
*	0	1	1	1	Queue Mode Control & Status	:READ/WRITE
*	1	0	0	0	Queued Control & Status For Drive N	:READ/WRITE
*	1	0	0	1	Queued Unit, Sector Head Reg. for Drive N	:READ/WRITE
*	1	0	1	1	Queued Word Count Reg. For Drive N	:READ/WRITE
*	1	1	0	0	Queued Cylinder Address Reg. For Drive N	:READ/WRITE
*	1	1	0	1	Spare	:READ/WRITE
*	1	1	1	0	Spare	:READ/WRITE
*	1	1	1	1	Queued Strip Buss Address Reg. For Drive N	:READ/WRITE

**The Control and Status Register is loaded by using the LDLCSRL or LDHCSRL Signals appropriately.

*Refers to optional features of Formatter.

Note: A logical '1' is defined as active in this table. (Signals at interface are actually active when Low.)

5.2.3 Register Load Sequence Timing

A detailed definition of the timing requirements and relationships of all signals involved in a formatter register load operation is given in Figure 5.1.

5.3.1 Register Read Sequence Overview

Internal registers in the Phoenix 200 Formatter are interrogated or read by:

1. Selecting the register to be read by appropriately controlling the status of Register Select Signals RCO(L)-RC3(L).

2. Commanding the formatter to gate the contents of the selected register onto the 16 bit data bus (DBOL-DB15L) by enabling the DSDATIL signal.

3. Waiting for the formatter data bus contents to be valid, and then gating the register data into the user interface logic with an internal strobe signal.

In the formatter register read sequence depicted above, the user completely controls the operations and associated timing. Any timing requirements imposed upon the user are only to guarantee register select and data validity.

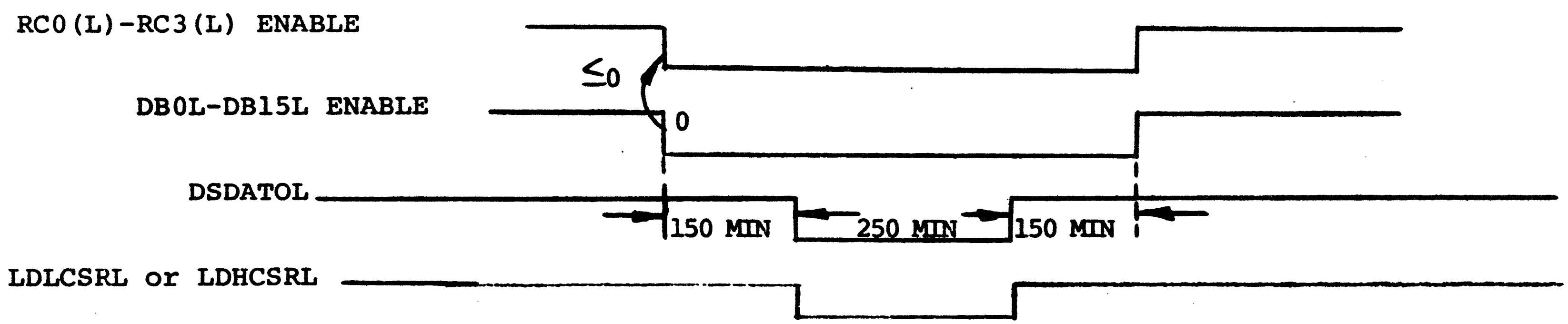
5.3.2 Register Read Sequence Timing

A detailed definition of the timing requirements and relationships of all signals involved in a formatter register read operation is given in Figure 5.2.

FIGURE 5.1

XYLOGIC PHOENIX 200 FORMATTER

REGISTER LOAD TIMING DIAGRAM



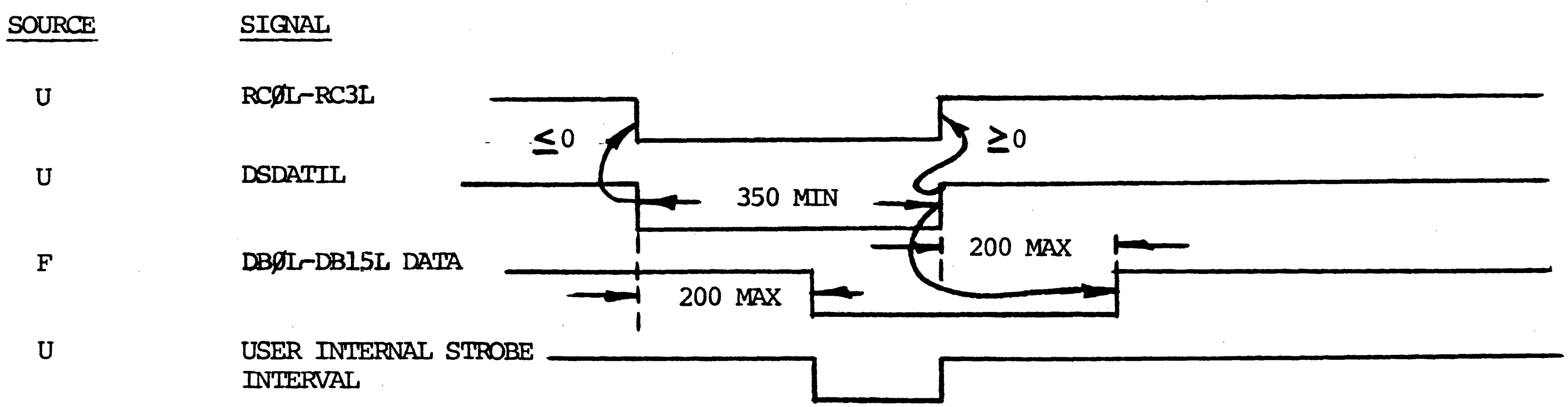
NOTES

1. Timing is at Formatter Interface Connector
2. All signals are active when Low (Negative Logic) at Interface.
3. All times are in nanoseconds.
4. RCØ(L) - RC3(L) and DSDATOL not applicable on Command & Status Register Loading.

FIG 5.2

XYLOGIC PHOENIX 200 FORMATTER

REGISTER READ TIMING DIAGRAM



NOTES:

1.) Timing is at Formatter interface connector.

2.) All signals are low active (Negative Logic)

3.) All times are in nanoseconds.

6.0 Interrupt Request Timing Sequence

6.1 Interrupt Operation Definition

When the Phoenix 200 Formatter completes any commanded disk operation (except Clear, or Release,) the formatter will generate a signal to the user interface. This signal is normally used by the user interface to generate an interrupt to the operating computer program to enable it to reallocate the disk subsystem resource to some new task.

The interrupt request signal (INTRL) will be generated at the end of a disk operation, independently of whether the operation just completed was completed successfully or not. It is up to user software to test the error summary bit to determine whether an error did occur during the previous commanded operation and to take appropriate action.

6.2 Interrupt Timing Reference

All timing given is at the user interface 10' from the Phoenix 200 Formatter.

6.3 Interrupt Sequence Overview

Two signals are involved in the interrupt operation INTRL and INTRACKL. At the end of a commanded disk operation the formatter initiates the interrupt sequence by activating INTRL. The user then completes the sequence by activating the interrupt acknowledge signal INTRACKL.

Upon detection of the INTRACKL Signal the formatter removes the INTRL signal. The only timing requirement imposed by the formatter is that the INTRACKL Signal generated by the user be at least 100NS wide.

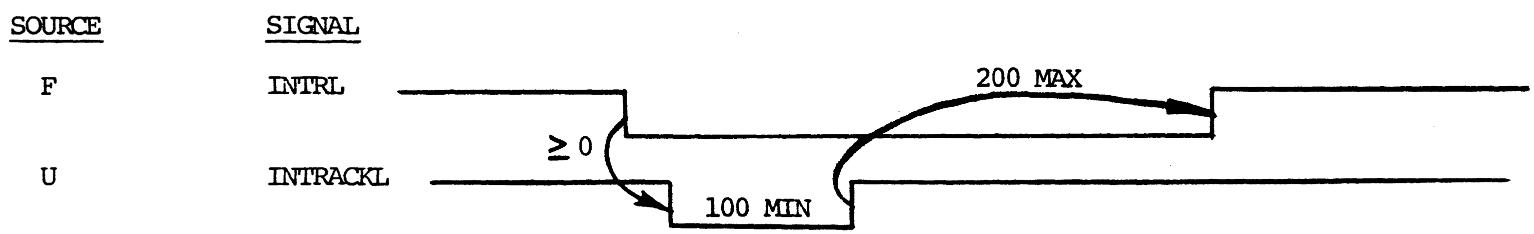
6.4 Detailed Interrupt Sequence Timing Information

Figure 6.0 shows all applicable interrupt timing relationships and requirements.

FIG[illegible] 6.0

XYLOGIC PHOENIX 200 FORMATTER

INTERRUPT TIMING DIAGRAM



NOTES:

1.) Timing is at Formatter interface connector.

2.) All signals are active when low (Negative Logic) at interface.

3.) Formatter becomes "Ready" during INTRL.

4.) All times are in nanoseconds.

7.0 Write Direct Memory Access Timing Sequence

7.1 Write Operation Definition

A "Write" disk formatter operation, for purposes of this discussion, is defined as being a transfer of 16 bit data words from the user interface to one of the formatter internal FIFO memories and thence to the selected disk drive.

7.2 Interface Timing Reference

All timing information given, unless otherwise specifically noted, is at the Formatter cable interface.

7.3 Write Operation Initiation

A "Write" operation is initiated by activating the Formatter "Go" bit after properly loading the Formatter registers with appropriate parameters defining the specific operation to be performed.

Note: A comple disk controller consisting of the Phoenix 200 Formatter and a user computer interface actually contains two Word Count Registers, one within the user interface and one within the 200 Formatter.

The user interface Word Count Register controls the number of words transferred between the Phoenix 200 Formatter and computer memory, while the formatter word count register controls the number of words transferred between the formatter and the selected disk drive.

Both Word Count Registers are loaded at the same time.

7.4 DMA Direction Establishment by DMDIR

When the write operation is specified, the formatter immediately establishes the direction of all DMA data transfers required by putting formatter signal DMDIR in the inactive (high) state.

This signal will remain at this high level for the duration of any write operation.

7.5 DMA Transfer to Formatter FIFO Memory's Overview

1. At the start of a writer operation the internal formatter FIFO memories are empty.

2. The formatter will request data transfers from the user interface to fill the buffer memories before disk synchronization is accomplished.

   Once the FIFO buffer memories are full, DMA requests are made only on a demand basis to keep the buffers full until the Word Count Overflow Signal (WCROVFL) is detected.

3. The formatter contains an internal timer which guarantees that the internal FIFO memories have had time to be sufficiently loaded with data before data transfers between the FIFO memories and the disk drive are allowed to begin.

Note: The guaranteed buffer load period is variable and may have to be increased if the maximum DMA data transfer rate possible through the user interface is not fast enough.

The factory and interval setting should be adequate for all current generation computers.

7.6 DMA Write Data Transfer Cycle Sequence (Reference Fig.7)

7.6.1 Each DMA write cycle data transfer is initiated by DMR (L) being active, indicating that the associated buffer is not full and is requesting more data.

7.6.2 Upon detection of DMR (L) (and the availability of data from computer) the user should put the data for the next word to be transferred on the formatter bidirectional data bus.

7.6.3 A minimum of 150NS. later the user initiates the strobing of the data into one of the internal formatter FIFO memories by activating the DMDATO signal.

(The 150NS delay from data enabling to leading edge of data strobe allows for data skew, propagation, and settling time.)

7.6.4 The DMACK Signal may be issued at any time during the current cycle or at some convenient time previous to the current cycle. (DMA Look Ahead).

7.6.5 The DMDATO Data Strobe Signal should be a minimum of 100NS wide.

7.6.6 The user must maintain data on the bus 150NS minimum after the leading edge of the DMDATO Signal.

7.6.7 The DMA request signal associated with the FIFO memory loaded by the data transfer will become valid a minimum of 200NS from the leading edge of the DMACK Signal providing the FIFO memory is not yet full.

7.6.8 After the data for a given cycle has been transferred per the above, subsequent DMA transfers may take place by repeating the sequence as appropriate.

DMDATA Strobe Signal frequency must not exceed 2.0MHZ (500NS period).

Note: The user effectively controls the Write DMA data transfer rate and timing by controlling the DMDATA and DMACK Signals.

7.6.9 The user interface logic should increment the Word Count Register at the beginning of each DMA transfer cycle so that a logic decision to terminate upon Word Count Register Overflow can be reached before the end of the last cycle.

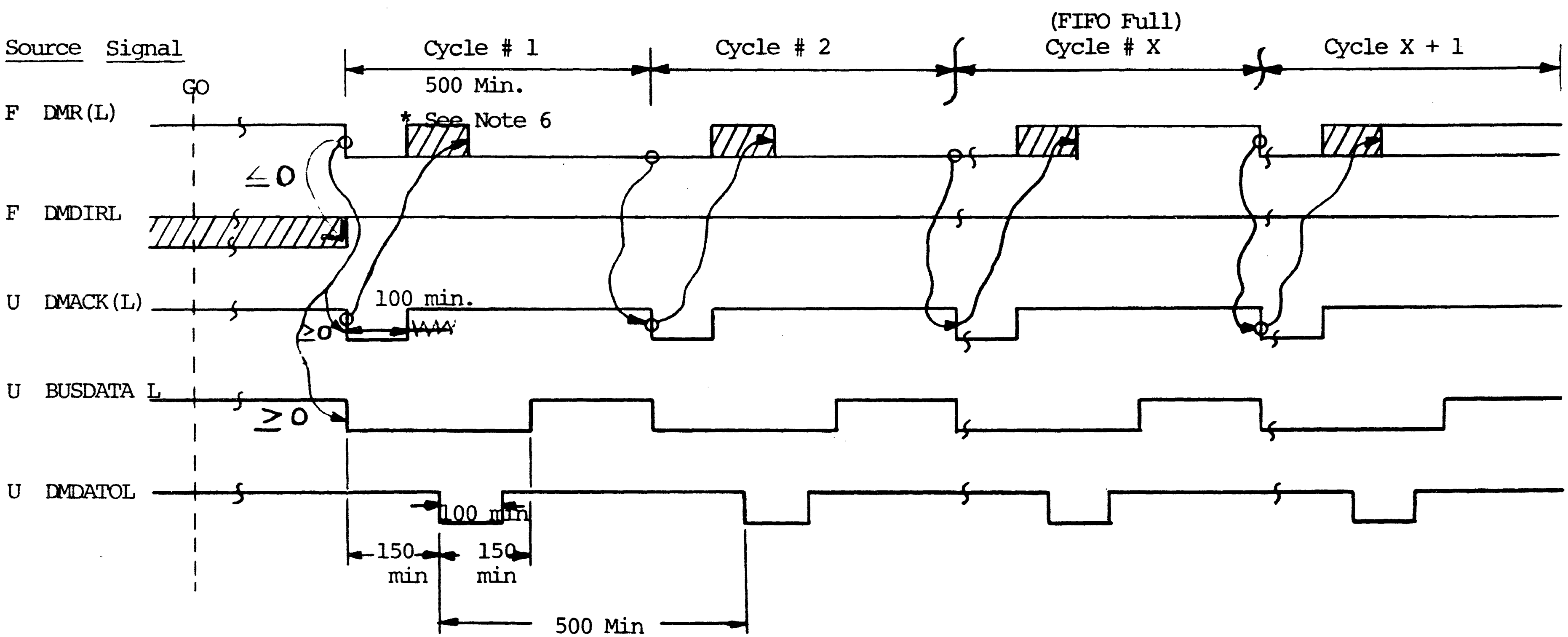
7.6.10 Write DMA data transfers from the user interface to the formatter internal FIFO memories terminate when the Word Count Overflow Signal (WCROVFL) is received from the user interface. DMR (L) if active will be returned to the inactive state within 200NS of the leading edge of the WCROVFL Signal.

7.6.11 The Word Count Register Overflow Signal generated by the user interface may be either a pulse or a level but must be a minimum of 100NS wide.

7.6.12 If the Word Count Overflow Signal is issued prematurely by the user, DMA transfers terminate immediately, but the controller will continue to write alternately the last two words transferred to the FIFO memories onto the selected disk until the formatter internal Word Count Register Overflows and the end of the last addressed sector is reached.

FIGURE

Phoenix 200 Formatter
Write DMA Timing Diagram (3 Full Cycles)



Notes:

1. Timing is at Formatter Interface Connector
2. All Signals are active when low (Negative Logic) at Interface
3. Data On Buss is valid during DMDATO Pulse Interval
4. Transfer rate is controlled by user by controlling frequency of DMDATO and DMACK
5. All times are in nanoseconds
6. DMR(L) is not valid until 200 ns min. after leading edge of DMACK(L)

8.0 Read Direct Memory Access Timing Diagram

8.1 Read Operation Definition

A "Read" disk formatter operation for purposes of this discussion is defined as being a transfer of 16 bit data words from the selected disk to the Phoenix 200 Formatter and thence to the user interface.

8.2 Interface Timing Reference

All timing information given, unless otherwise specified, is at the Formatter interface connector.

8.3 Read Operation Initiation

A "Read" operation is initiated by activating the formatter "GO" bit after properly initializing the formatter registers with appropriate transfers defining the specific operation to be performed.

8.4 DMA Direction Establishment by DMDIR Signal

When the read operation is specified the formatter immediately establishes the direction of all DMA data transfers required by forcing the DMDIR Signal to the active state (low).

This signal will remain at this state for the duration of any read operation.

8.5 DMA Transfer from Formatter to User Interface Overview

8.5.1 At the start of a read operation, the formatter internal FIFO memories are empty.

8.5.2 Data transfers from the selected disk to the FIFO memories will begin when the disk is on the correct cylinder and sector.

8.5.3 Words received from the selected disk are alternately temporarily stored in one of the two internal FIFO memories. All odd words are normally stored in FIFO memory #1 while all even words are stored in FIFO memory #2.

8.5.4 DMA Data transfers to the user interface are initiated when data is present in the FIFO memory and will persist until no data is contained in either FIFO memory.

8.5.5 Data flow from the selected disk to the FIFO memories terminates when the formatter internal Word Count register overflows.

8.5.6 Data flow from the FIFO memories to the user interface terminates when the Word Count Register in the user interface overflows and activates the WCROVFL Signal.

8.6 DMA Read Data Transfer Cycle Sequence (Reference Fig. 8)

8.6.1 Each DMA read cycle is initiated by DMR (L) being active, indicating that the associated FIFO memory buffer contains data to be transferred to computer memory via the user interface.

8.6.2 Upon detection of DMR (L) the user, when his interface is ready to accept data, should activate the DMDATI Signal.

8.6.3 The formatter, upon detection of the DMDATI Signal, will put the data to be transferred on the formatter data bus. A maximum of 200NS from the leading edge of a DMDATI pulse, data is guaranteed to be valid at the Formatter interface connector.

8.6.4 The user should strobe data from the bus into the user interface 300ns minimum from the leading edge and during the DMDATI Signal Interval.

8.6.5 The DMDATI Signal pulse must be a minimum of 350NS wide.

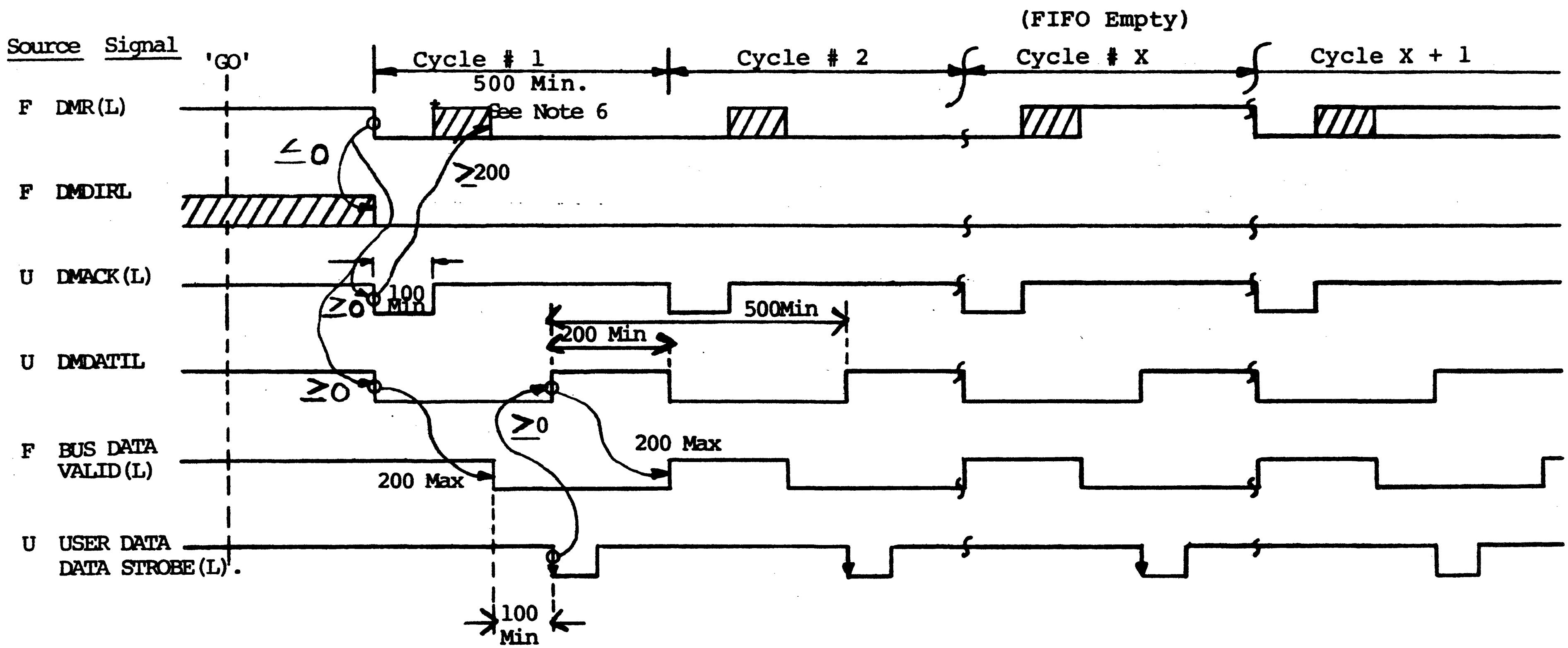
8.6.6 The formatter, upon detection of the trailing edge of the DMDATI Signal Pulse, will immediately remove data from the formatter bus. Data, then, is not valid after DMDATI is removed.

8.6.7 The DMDATI Signal, once removed at the end of a read DMA data transfer, must remain inactive for a minimum of 200 NS before it can be activated again for the next read DMA transfer cycle.

8.6.8 The DMACK Signal may be issued at any time during the current cycle or at some convenient time previous to the current cycle. (DMA Look Ahead).

8.6.9 The DMA request signal will become vaid a minimum of 200NS from the leading edge of the DMACK Signal providing there is another data word available in FIFO memory.

FIGURE 8
Phoenix 200 Formatter
Read DMA Timing Diagram (Two Cycles)



Notes:

1. Timing is at Formatter Interface Connector.
2. All signals are active when low (negative logic) at interface.
3. Data On Buss is valid 200 ns from leading edge of DMDATI to end of DMDATI Pulse.
4. Transfer rate is controlled by user by controlling frequency of DMDATI and DMACK.
5. All times are in nanoseconds.
6. DMR(L) is not valid until 200 ns min. after leading edge of DMACK(L).

8.6.10 After the read data for a given cycle has been transferred from the formatter to the user interface per above, subsequent read DMA transfers may take place as required by repeating the above sequence.

Note: The user effectively controls the read DMA Data Transfer Timing and data transfer rate by the DMDATI and DMACK Signals.

8.6.11 The user interface should increment the Word Count Register at the beginning of each read DMA data transfer cycle so that a logical decision to terminate upon word count overflow can be reached before the end of the last cycle.

8.6.12 Read DMA data transfers from the formatter to the user interface terminate when the Word Count Register overflow signal (WCROVFL) is activated by the user interface.

DMR (L) if active will be returned to the inactive state within 200NS of the leading edge of the WCROVFL Signal.

8.6.13 The WCROVFL signal may be either a pulse or a level but must be at least a minimum of 100NS wide.

9.0 DMA Data Transfer Considerations

9.1.1 Two First in First Out (FIFO) memories are contained in the Phoenix 200 Formatter. All data to be transferred to or from disk drives connected to the formatter is buffered in one of the two FIFO memories in the formatter.

9.2.1 The two FIFO memories in the formatter are alternately used for temporary data storage on a word basis in order to maximize DMA data transfer throughout rate.

Normally all odd words are buffered in FIFO1 while all even words are buffered in FIFO2.

9.3.1 Maximum theoretical DMA transfer 16 bit word transfer rate is 2.0 MHZ which is more than ample for interfacing to current generation computers.

9.4.1 The actual DMA transfer rate in a disk subsystem using the Phoenix 200 Formatter is determined by the user.

9.5.1 All formatter interface signals are negative active at the user connector.

9.6.1 DMA data transfers normally utilize the same 16 bit bidirectional data bus used for direct program control register loading and reading.

DMA data transfers may be optionally made over a separate 16 bit bidirectional data used exclusively for DMA.

Such an interface is available on all printed circuit versions of the Phoenix 200 Formatter and requires a separate cable and associated connectors.

9.7.1 If the standard data buss is used for DMA data transfers, the user interface must resolve any formatter bus usage arbitration necessary to insure that no attempt is made to use the formatter bus for DMA and direct program control operations during the same physical bus cycle.

9.7.2 The DMACK Signal must be issued only once for each DMA data transfer.

USER INTERFACE CONNECTOR SIGNAL/PIN ASSIGNMENT

	PIN #	SIGNAL NAME	DIRECTION
	1	GND	-
	2	GND	-
	3	GND	-
	4	DBØL	BI
	5	DB1L	BI
	6	DB2L	BI
	7	DB3L	BI
	8	DB4L	BI
	9	DB5L	BI
	10	DB6L	BI
	11	DB7L	BI
	12	DB8L	BI
	13	DB9L	BI
	14	DB10L	BI
	15	DB11L	BI
	16	DB12L	BI
	17	DB13L	BI
	18	DB14L	BI
	19	DB15L	BI
	20	RCØ(L)	TO FORMATTER
	21	RC1(L)	TO FORMATTER
	22	RC2(L)	TO FORMATTER
*	23	RC3(L)	TO FORMATTER
	24	WCROVFL	TO FORMATTER
	25	INIT(L)	TO FORMATTER
	26	DMDATI(L)	TO FORMATTER
	27	DMDATO(L)	TO FORMATTER
	28	INTRACKL	TO FORMATTER
	29	LDHCSRL	TO FORMATTER
	30	DSDATIL	TO FORMATTER
	31	FECOND(L)	TO FORMATTER
	32	DSDATOL	TO FORMATTER
*	33	ENBMM(L)	TO FORMATTER
	34	SYSINITL	FROM FORMATTER
	35	INTRL	FROM FORMATTER
	36	DMRL	FROM FORMATTER
	37	DMDIRL	FROM FORMATTER
*	38	LDQBAR(L)	FROM FORMATTER
*	39	LDMEX(L)	FROM FORMATTER
*	40	LDQWCR(L)	FROM FORMATTER
*	41	ENQDATAL	FROM FORMATTER
	42	DMACK(L)	TO FORMATTER
**	43	No Strip(L)	FROM FORMATTER
	44	LDLCSRL	TO FORMATTER
	45	SPare	-
	46	Spare	-
	47	Spare	-
	48	GND	-
	49	GND	-
	50	GND	-

CONNECTOR TYPE USED ON BOARD 1 of FORMATTER:

3M # 3433-1002

Notes:
1. BI Denotes Bidirectional signal

** Used in strip option only

RECOMMENDED INTERFACE

Drivers, Receivers, Termination, and Cable

SIGNAL LEVEL DEFINITION:

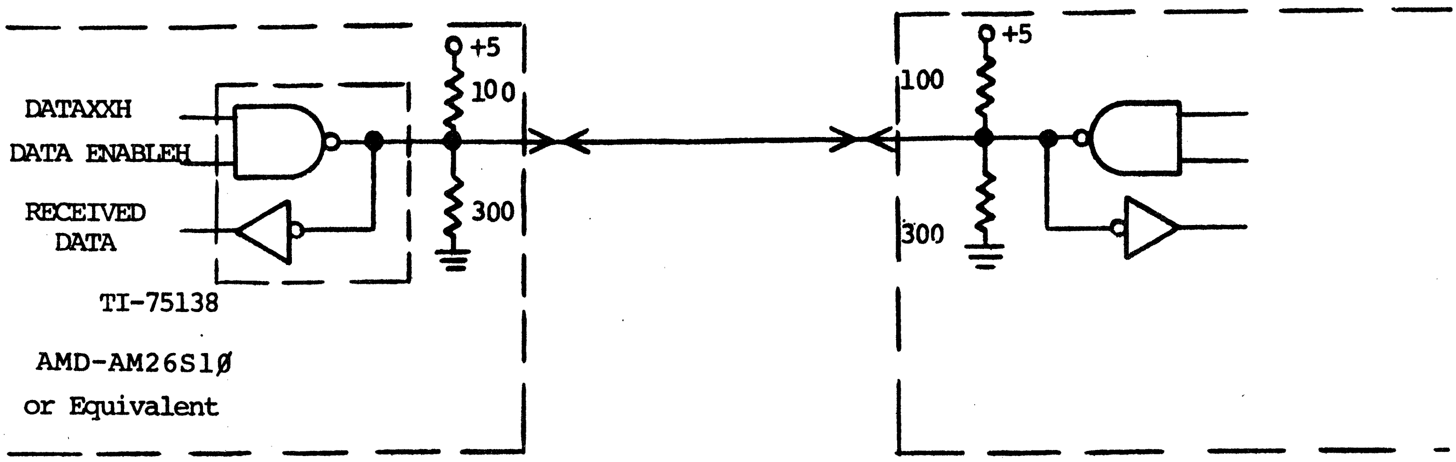
ALL SIGNALS ARE LOW, ACTIVE TRUE

LEVELS: TRUE= LOW= 0.7 VDC @ 100MA

FALSE= HIGH= 3.5VDC

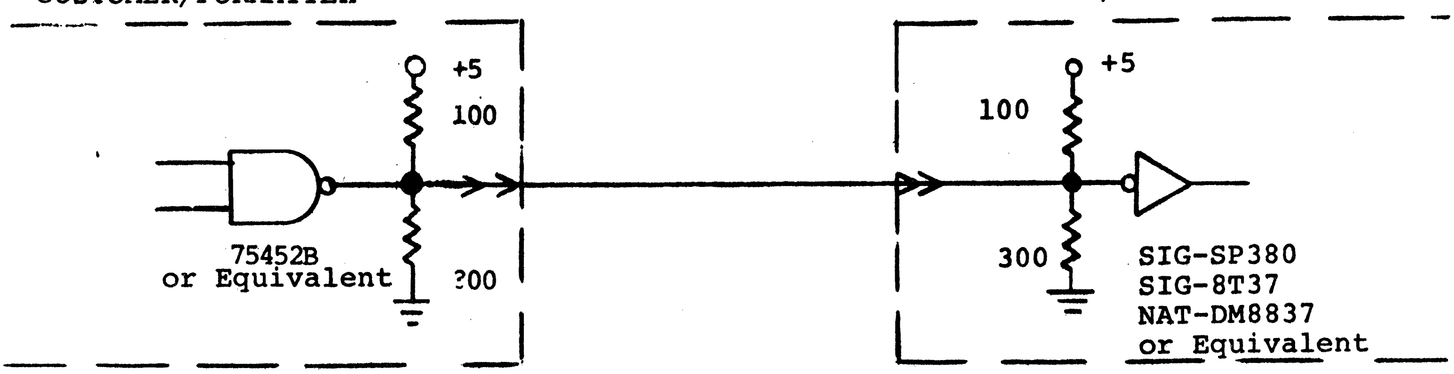
DATA BUS DRIVERS/ RECEIVERS/ TERMINATION (BIDIRECTIONAL)

CUSTOMER INTERFACE FORMATTER



CONTROL SIGNALS DRIVERS/RECEIVERS/TERMINATION

CUSTOMER/FORMATTER CUSTOMER/FORMATTER



RECOMMENDED CABLE

3M Part No. 3476/50; Flat Cable with Ground Plane and Drain Wire

XYLOGIC OEM COMPONENTS GROUP, INC.

42 Third Avenue

Burlington, Massachusetts 01803

617-272-8140

Phoenix 211 Disk Controller

Optional Data Port

User Interface Manual

Dwg. No. 1043-09

Revision A

Date August 1, 1976

TABLE OF CONTENTS

OPTIONAL DATA PORT

1.0 Introduction

It is the intent of this manual to provide the user with sufficient technical information to successfully interface external equipment to the optional data port interface provided by the Xylogic Phoenix 211 Disk Controller with a minimum of difficulty.

The information contained in this manual is principally concerned with the physical, electrical, and timing interface provided to the user by the optional data port of the Phoenix 211 Disk Controller.

The functional capabilities, performance characteristics, and programming interface of the Phoenix 211 Disk Controller are explained in greater detail in the Phoenix 211 Disk Controller Programming Reference Manual. The user should be equally familiar with the contents of the programming manual, as well as this manual before attempting to interface external equipment to the Phoenix 211 optional data port.

This manual will be expanded and updated as required to provide the user with complete, accurate, interfacing information for the Phoenix 211 Optional Data Port.

Any questions relating to the contents of this manual should be directed to:

Engineering Department
Xylogic OEM Components Group, Inc.
42 Third Avenue
Burlington, Mass. 01803
617-272-8140

2.0 Optional Data Port Description (Reference Figure 1)

2.1 General Description

The Optional Data Port provided by the Phoenix 211 Disk Controller enables the user to conduct disk data transfers directly between external equipment and disk drive(s), under control of programs being executed in a PDP11 Computer.

In effect, this option separates the data transfer path from the control and status functions within the Phoenix 200 Formatter. The computer interface is utilized to initiate, control, and monitor the results of disk data transfers, while the actual data transfer takes place between user external equipment and the disk drives, via the Phoenix 200 Formatter.

2.2 Physical Description

The Optional Data Port physically is provided by a 50 pin ribbon cable connector located on board 2 of the Phoenix 200 Disk Formatter. This 50 pin ribbon cable connector contains a 16 bit bidirectional data bus over which data is to be transmitted to and received from during disk data transfers, as well as the control signals and ample grounds. Detailed descriptions of the signals provided on this interface are contained in subsequent sections of this manual.

3.0 Programming Definition of Optional Data Port

The use of the optional data port interface does not affect the programming of the Phoenix 211 Disk Controller. Operation of a Phoenix 211 disk controller utilizing the optional data port interface is identical to a standard Phoenix 211 disk controller, as defined in the Programmers Reference Manual, except that the data transfer occurs over the optional data port interface.

This assumes, however, that the external equipment will always be ready and able to respond appropriately to disk transfer operations initiated by the controlling computer. Any control signals needed for communication between the controlling computer and the external equipment required to quarantee the validity of this assumption must be provided by the user via some other means.

4.0 Optional Data Port Enabling

Any standard Phoenix 211 Disk Controller is equipped with the optional data port interface. The optional data port configuration is achieved by removing staples DØH-D15H and three additional control staples on board 2 of the Phoenix 200 Formatter. Said staples are shown on Logic Drawing D1033-01 sheet 1.

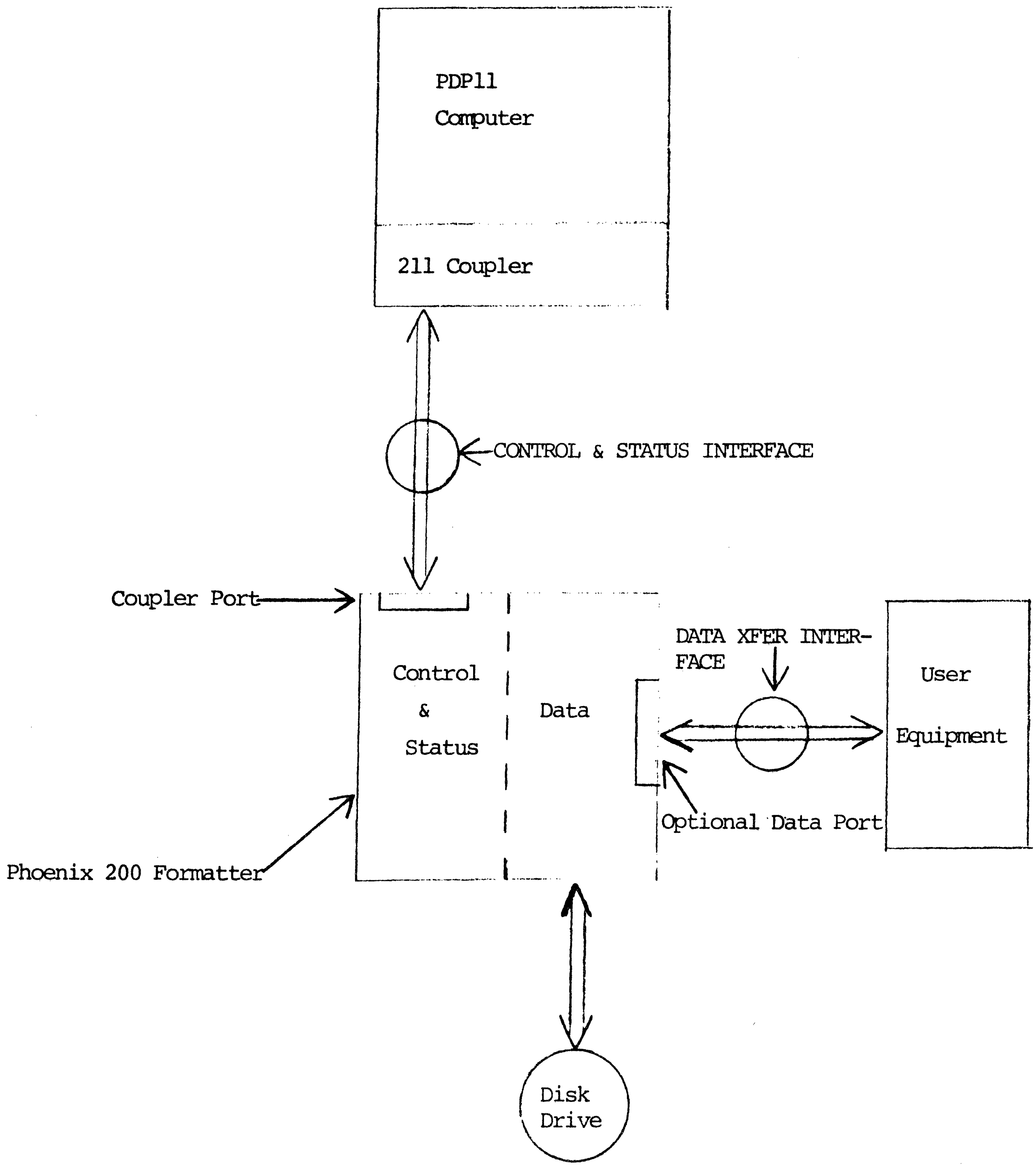
5.0 Optional Data Port Interface Testing

The optional data port interface, once enabled may be readily tested by connecting the computer coupler cable to the optional data port interface connector, and to the normal formatter coupler cable connector.

In this configuration data transfers will take place between the computer and disk drive via the optional port interface logic.

FIGURE 1

Phoenix 211 Optional Data Port Configuration



Note: 1. The optional data port interface connector and coupler connector have the same functional signal/pin assignments to facilitate this operation.

2. The required coupler cable can be readily configured by adding one additional coupler ribbon cable connector to the coupler cable 6" from the end of the formatter end of the coupler cable.

6.0 Data Transfer Mode Switching

Disk transfers may be conducted between the controlling computer and disk or external equipment connected to the optional data port and disk. Switching between these two modes of operation normally requires manual connection of the appropriate coupler or external equipment cable to the optional data port interface connector.

Automatic program controlled mode switching requires the addition of one "Autoswitch" printed circuit board assembly to the disk controller at additional cost.

7.0 Optional Data Port User Interface Signal Definitions

Note: All interface signals are low active.

7.1 Bidirectional Data Bus = DMDATAØL-DMDATA15L

The 16 bit bidirectional data lines are used to transmit disk data between the internal disk formatter memory and user equipment normally connected to the optional data port interface connector. Timing relationships required in transferring data over the data bus is given in detail in section

7.2 Direct Memory Access Direction Control - DMDIRL (From Formatter)

This signal is a level controlled by the Phoenix formatter and sent to the user external equipment to define the direction in which data is to move over the data bus during any disk data transfers.

This signal is active (low) when data is to be transferred from the formatter to the user during any disk read data transfer operations.

This signal is inactive (high) when data is to be transferred from the user to the formatter during a disk write data transfer.

7.3 <u>Data Transfer Request - DMREQ(L) (From Formatter)</u>

.1 This signal is used by the formatter to initiate data transfers between it and user equipment connected to the optional data port.

.2 The DMREQ(L) signal is associated with the state of the formatter internal FIFO memory.

.3 In a Disk Write operation the DMREQ(L) signal will be active when the formatter is busy, the internal FIFO memory buffers are <u>not</u> full, and the Word Count Overflow (WCROVFL) Signal has not been received from the user interface.

.4 In a disk read operation the DMREQ(L) Signal will be active when the formatter is busy, the internal FIFO memory buffer contains any data (is not empty) and the Word Count Overflow (WCROVFL) Signal has not been received from the user interface.

.5 The user interface must sample the state of the DMA Request Signal.

.6 The DMACK(L) Signal is utilized to facilitate DMA look-ahead capability for multiple cycle transfers.

7.4 <u>Data Transfer Acknowledge Signal - DMACK(L) (To Formatter)</u>

.1 This signal is issued by the user interface to acknowledge receipt of a DMREQ(L) signal from the formatter.

.2 Upon receipt of the DMACK(L) signal the formatter interrogates the FIFO memory content and will modify the state of the DMREQ(L) signal accordingly.

.3 The DMREQ(L) signal may be sampled 200 NS after the leading edge of DMACK signal to facilitate multiple cycle look-ahead transfer implementations.

7.5 <u>Write Data Transfer Strobe- DMLDSTB(L) (To Formatter)</u>

This signal is pulsed by the user interface to the formatter during a Write Data Transfer to the formatter when the data on the bidirectional data bus is valid.

This signal is used by the formatter to strobe data from the bidirectional bus into internal FIFO memory, and may be activated a minimum of 150 NS after data is gated onto the bidirectional data bus.

7.6 Read Data Transfer Strobe - DMRDSTB(L) (To Formatter)

1. This signal is activated by the user interface during a read data transfer from the formatter to the user interface, when the user is ready to accept data from the formatter.

2. Upon receipt of the DMRDSTB(L) signal the formatter will put data on the bidirectional data bus.

3. Data will be guaranteed to be valid 150 NS after receipt of DMRDSTB(L) by formatter.

4. User may strobe data from bidirectional data bus any time after the 150 NS period.

5. The DMRDSTB(L) signal may then be removed. User may in fact use the trailing edge of the DMRDSTB(L) signal to strobe data from bus into his own logic.

7.7 Word Count Overflow - WCROVFL (To Formatter)

A pulse or high-to-low transition indicates to the formatter that the word count register in the user interface has overflowed, and that no more data transfers will take place.

7.8 Disk Transfer Done - DSKXFRD(L) (From Formatter)

This signal, when true, signifies that the disk word count register has overflowed and the formatter has reached the End of Sector mark. This signal is reset by issuing a new command to the formatter. The user interface may use this signal to detect data transfer errors. The error conditions are:

WRITE OPERATION:	Disk Transfer Done occurs before the user word count overflows.
READ OPERATION:	The user word count overflows before Disk Transfer Done is generated.

8.0 Disk Write Data Transfer Timing Sequence

8.1 Write Operation Definition

A "Write" disk formatter operation, for purposes of this discussion, is defined as being a transfer of 16 bit date words from the user interface to one of the formatter internal FIFO memories and thence to the selected disk drive.

8.2 Interface Timing Reference

All timing information given, unless otherwise specifically noted, is at the Formatter cable interface.

8.3 Write Operation Initiation

A "Write" operation is initiated by activating the Formatter "Go" bit after properly loading the Formatter registers with appropriate parameters defining the specific operation to be performed.

Note: A complete disk controller consisting of the Phoenix 200 Formatter and a user external equipment interface actually contains two Word Count Registers, one within the user interface and one within the 200 Formatter.

The user interface Word Count Register controls the number of words transferred between the Phoenix 200 Formatter and external equipment, while the formatter word count register controls the number of words transferred between the formatter and the selected disk drive.

Both Word Count Registers should be pre-set to the same count before any data transfer operation is initiated.

8.4 Data Transfer Direction Establishment by DMDIR

When the write operation is specified, the formatter immediately establishes the direction of all DMA data transfers required by putting formatter signal DMDIR in the inactive (high) state.

This signal will remain at this high level for the duration of any write operation.

8.5 Data Transfer to Formatter FIFO Memory Overview

1. At the start of a writer operation the internal formatter FIFO memories are empty.

2. The formatter will request data transfers from the user interface to fill the buffer memories before disc synchronization is accomplished.

   Once the FIFO buffer memories are full, data transfer requests are made only on a demand basis to keep the buffers full until the Word Count Overflow Signal (WCROVFL) is detected.

3. The formatter contains an internal timer which guarantees that the internal FIFO memories have had time to be sufficiently loaded with data before data transfers between the FIFO memories and the disk drive are allowed to begin.

Note: The guaranteed buffer load period is variable and may have to be increased if the maximum data transfer rate possible through the user interface is not fast enough.

The factory interval setting should be adequate for all current generation computers.

8.6 Write Data Transfer Cycle Sequence (Reference Fig. 7)

8.6.1 Each write cycle data transfer is initiated by DMREQ (L) being active, indicating that the associated buffer is not full and is requesting more data.

8.6.2 Upon detection of DMREQ(L) (and the availability of data from user equipment) the user should put the data for the next word to be transferred on the formatter optional data port bidirectional data bus.

8.6.3 A minimum of 150NS. later the user initiates the strobing of the data into one of the internal formatter FIFO memories by activating the DMLDSTB(L).

(The 150NS delay from data enabling to leading edge of data strobe allows for data skew, propagation, and settling time.)

8.6.4 The DMACK signal may be issued at any time during the current cycle or at some convenient time previous to the current cycle. (Transfer Look Ahead).

8.6.5 The DMLDSTB(L) Data Strobe Signal should be a minimum of 100NS wide.

8.6.6 The user must maintain data on the bus 150NS minimum after the leading edge of the DMLDSTB(L) signal.

8.6.7 The transfer request signal associated with the FIFO memory loaded by the data transfer will become valid a minimum of 200NS from the leading edge of the DMACK Signal, providing the FIFO memory is not yet full.

8.6.8 After the data for a given cycle has been transferred per the above, subsequent data transfers may take place by repeating the sequence as appropriate.

DMLDSTB(L) Strobe Signal frequency must not exceed 2.OMHZ (500NS period).

Note: The user effectively controls the Write DMA data transfer rate and timing by controlling the DMLDSTB(L) and DMACK Signals.

8.6.9 The user interface logic should increment the Word Count Register at the beginning of each data transfer cycle so that a logic decision to terminate upon Word Count Register Overflow can be reached before the end of the last cycle.

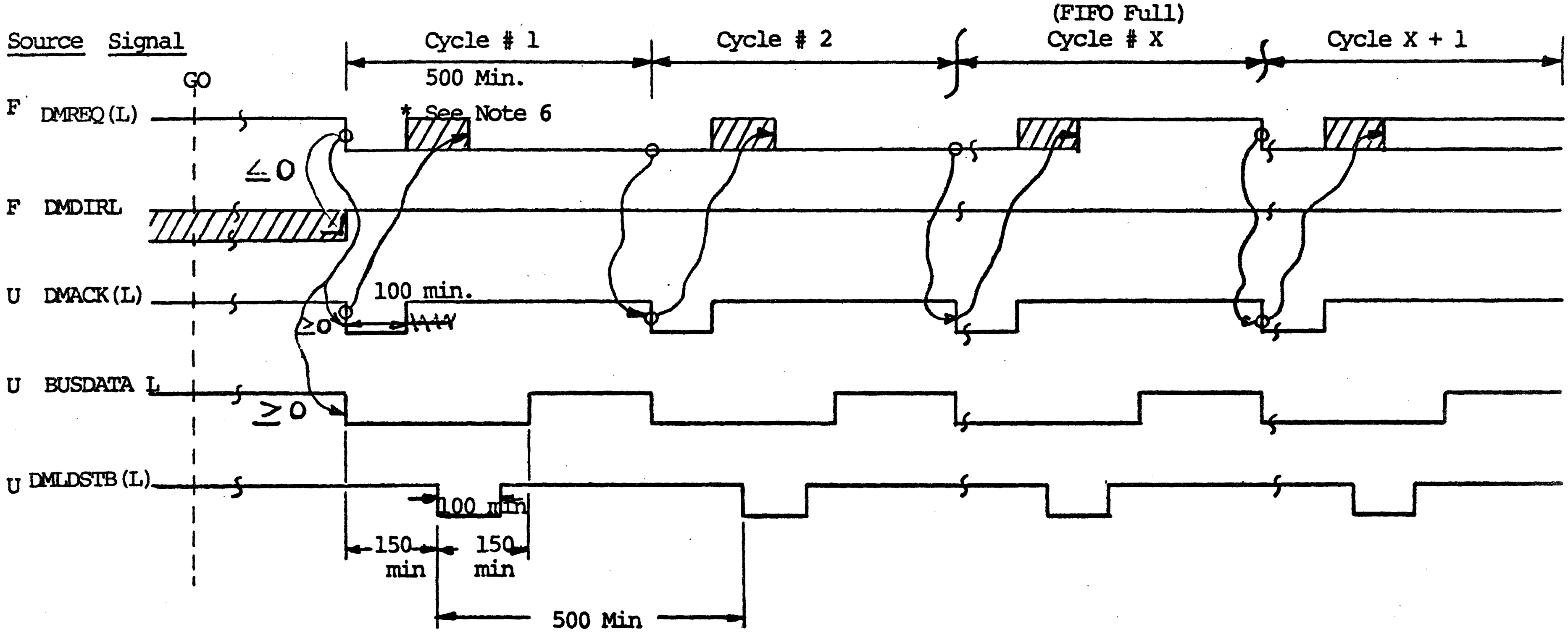
8.6.10 Write DMA data transfers from the user interface to the formatter internal FIFO memories terminate when the Word Count Overflow Signal (WCROVFL) is received from the user interface. DMREQ(L) if active will be returned to the inactive state within 200NS of the leading edge of the SWROVFL Signal.

8.6.11 The Word Count Register Overflow Signal generated by the user interface may be either a pulse or a level but must be a minimum of 100NS wide.

8.6.12 If the Word Count Overflow Signal is issued prematurely by the user, DMA transfers terminate immediately, but the controller will continue to write alternately the last two words transferred to the FIFO memories onto the selected disk until the formatter internal Word Count Register Overflows and the end of the last addressed sector is reached.

FIGURE 7

OPTIONAL DATA PORT
PHOENIX 200 FORMATTER
WRITE DATA TRANSFER TIMING DIAGRAM (3 FULL CYCLES)



Notes:

1. Timing is at Formatter Interface Connector
2. All Signals are active when low (Negative Logic) at Interface
3. Data On Buss is valid during DMLDSTB(L) Pulse Interval
4. Transfer rate is controlled by user by controlling frequency of DMLDSTB(L) and DMACK
5. All times are in nanoseconds
6. DMREQ(L) is not valid until 200 ns min. after leading edge of DMACK(L)

9.0 Read Data Transfer Timing Diagram

9.1 Read Operation Definition

A "Read" disk formatter operation for purposes of this discussion is defined as being a transfer of 16 bit data words from the selected disk to the Phoenix 200 Formatter and thence to the user interface.

9.2 Interface Timing Reference

All timing information given, unless otherwise specified, is at the Formater Optional Data Port interface connector.

9.3 Read Operation Initiation

A "Read" operation is initiated by activatingthe formatter "GO" bit after properly initializing the formatter registers with appropriate transfers defining the specific operation to be performed.

9.4 Data Transfer Direction Establishment by DMDIR Signal

When the read operation is specified the formatter immediately establishes the direction of all data transfers required by forcing the DMDIR Signal to the active state (low).

This signal will remaian at this state for the duration of any read operation.

9.5 Data Transfer from Formatter to User Interface Overview

9.5.1 At the start of a read operation, the formatter internal FIFO memories are empty.

9.5.2 Data transfers from the selected disk to the FIFO memories will begin when the disk is on the correct cylinder and sector.

9.5.3 Words received from the selected disk are alternately temporarily stored in one of the two internal FIFO memories. All odd words are normally stored iN FIFO memory #1 while all even words are stored in FIFO memory #2.

9.5.4 Data transfers to the user interface are initiated when data is present in the FIFO memory and will persist until no data is contained in either FIFO memory.

9.5.5 Data flow from the selected disk to the FIFI memories terminates when the formatter internal Word Count register overflows.

9.5.6 Data flow from the FIFO memories to the user interface terminates when the Word Count Register in the user interface overflows and activates the WCROVFL Signal.

9.6 <u>Read Data Transfer Cycle Sequence (Reference Fig. 8)</u>

9.6.1 Each read data transfer cycle is initiated by DMREQ(L) being active, indicating that the associated FIFO memory buffer contains data to be transferred to computer memory via the user interface.

9.6.2 Upon detection of DMREQ(L) the user, when his interface is ready to accept data, should activate the DMRDSTB(L) Signal.

9.6.3 The formatter, upon detection of the DMRDSTB(L) Signal, will put the data to be transferred on the formatter data bus. A maximum of 200 NS from the leading edge of a DMRDSTB(L) pulse, data is guaranteed to be valid at the Formatter interface connector.

9.6.4 The user should strobe data from the bus into the user interface 300NS minimum from the leading edge and during the DMRDSTB(L) Signal Interval.

9.6.5 The DMRDSTB(L)Signal pulse must be a minimum of 350NW wide.

9.6.6 The formatter, upon detection of the trailing edge of the DMRDSTB Signal Pulse, will immediately remove data from the formatter bus. Data, then, is not valid after DMRDSTB(L) is removed.

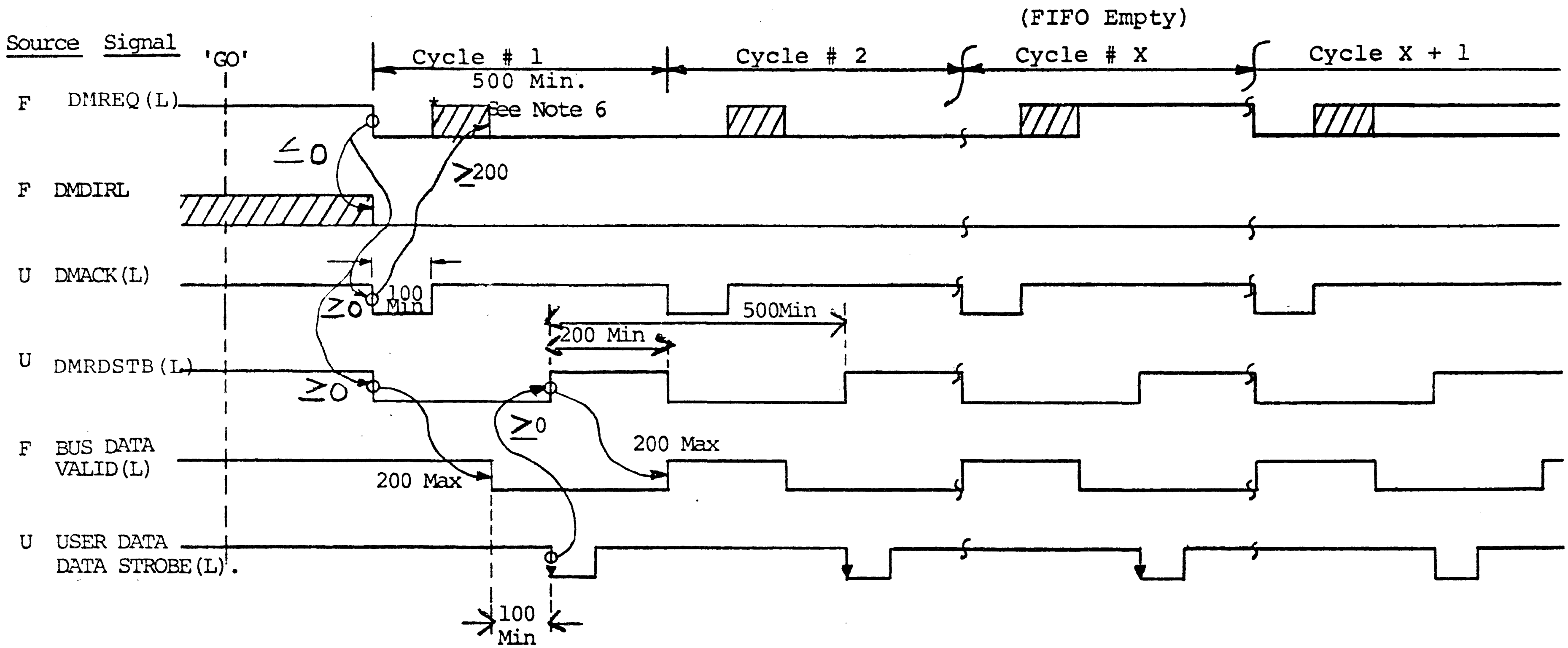
9.6.7 THE DMRDSTB(L) Signal, once removed at the end of a read DMA data transfer, must remain inactive for a minimum of 200 NS before it can be activated again for the next read data transfer cycle.

9.6.8 The DMACK Signal may be issued at any time during the current cycle or at some convenient time previous to the current cycle. (Transfer Cycle Look Ahead).

9.6.9 The data transfer request signal will become valid a minimum of 200 NS from the leading edge of the DMACK Signal providing there is another data word available in FIFO memory.

FIGURE 8
Phoenix 200 Formatter
Read Timing Diagram (Two Cycles)

Optional Data Port Interface



Notes:

1. Timing is at Formatter Interface Connector.
2. All signals are active when low (negative logic) at interface.
3. Data On Buss is valid 200 ns from leading edge of DMRDSTB(L) to end of DMRDSTB(L) Pulse.
4. Transfer rate is controlled by user by controlling frequency of DMRDSTB(L) & DMACK.
5. All times are in nanoseconds.
6. DMREQ(L) is not valid until 200 ns min. after leading edge of DMACK(L).

9.6.10 After the read data for a given cycle has been transferred from the formatter to the user interface per above, subsequent read data transfers may take place as required by repeating the above sequence.

Note: The user effectively controls the read Data Transfer Timing and data transfer rate by the DMRDSTAB(L) and DMACK Signals.

9.6.11 The user interface should increment the Word Count Register at the beginning of each read data transfer cycle so that a logical decision to terminate upon word count overflow can be reached before the end of the last cycle.

9.6.12 Read data transfers from the formatter to the user interface terminate when the Word Count Register overflow signal (WCROVFL) is activated by the user interface.

DMREQ(L) if active will be returned to the inactive state within 200NS of the leading edge of the WCROVFL Signal.

9.6.13 The WCROVFL signal may be either a pulse or a level but must be at least a minimum of 100NS wide.

10.0 Data Transfer Considerations

10.1.1 Two First in First Out (FIFO) memories are contained in the Phoenix 200 Formatter. All data to be transferred to or from disk drives connected to the formatter is buffered in one of the two FIFO memories in the formatter.

10.2.1 The two FIFO memories in the formatter are alternately used for temporary data storage on a word basis in order to maximize data transfer throughout rate.

Normally all odd words are buffered in FIFO1 while all even words are buffered in FIFO2.

10.3.1 Maximum theoretical data transfer (16 bit words) rate is 2.0 MHZ which is more than ample for interfacing to current generation computers and associated peripheral equipment.

10.4.1 The actual data transfer rate in a disk subsystem using the Phoenix 200 Formatter is determined by the user.

10.5.1 All formatter interface signals are negative active at the user connector.

10.6.1 Data transfers normally utilize the same 16 bit bidirectional data bus used for direct program control register loading and reading.

Data transfers may be optionally made over a separate 16 bit bidirectional data used exclusively for data.

The optional data port interface is available on all printed circuit versions of the Phoenix 200 Formatter and requires a separate cable and associated connectors.

10.7.1 If the standard data buss is used for DMA data transfers, the user coupler interface must resolve any formatter bus usage arbitration necessary to insure that no attempt is made to use the formatter bus for DMA and direct program control operations during the same physical bus cycle.

10.7.2 The DMACK signal must be issued only once for each data transfer.

RECOMMENDED INTERFACE

OPTIONAL DATA PORT
Drivers, Receivers, Termination, and Cable

SIGNAL LEVEL DEFINITION:

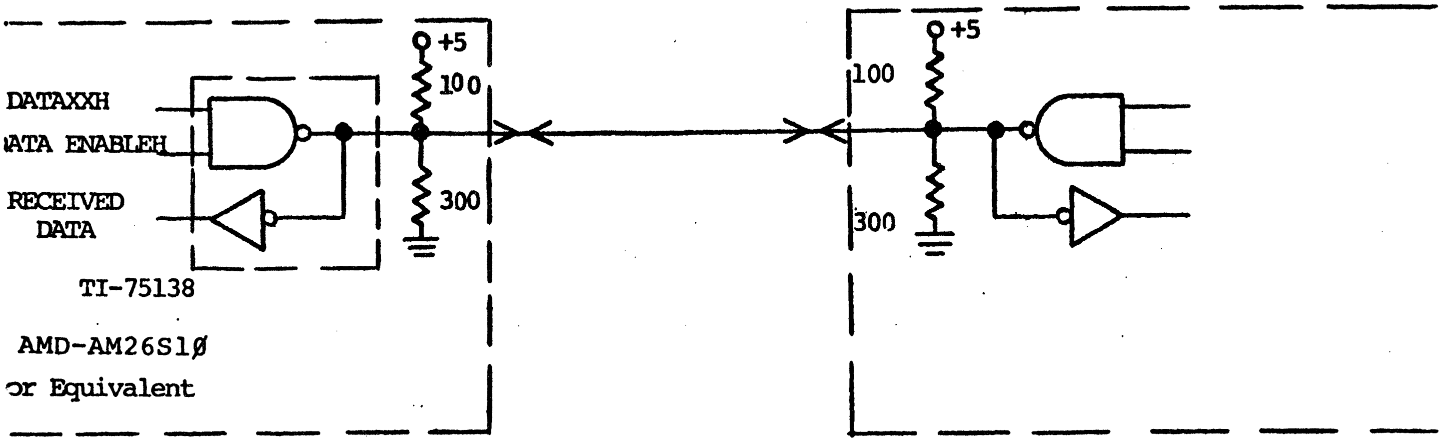
ALL SIGNALS ARE LOW, ACTIVE TRUE

LEVELS: TRUE= LOW= 0.7 VDC @ 100MA

FALSE= HIGH= 3.5VDC

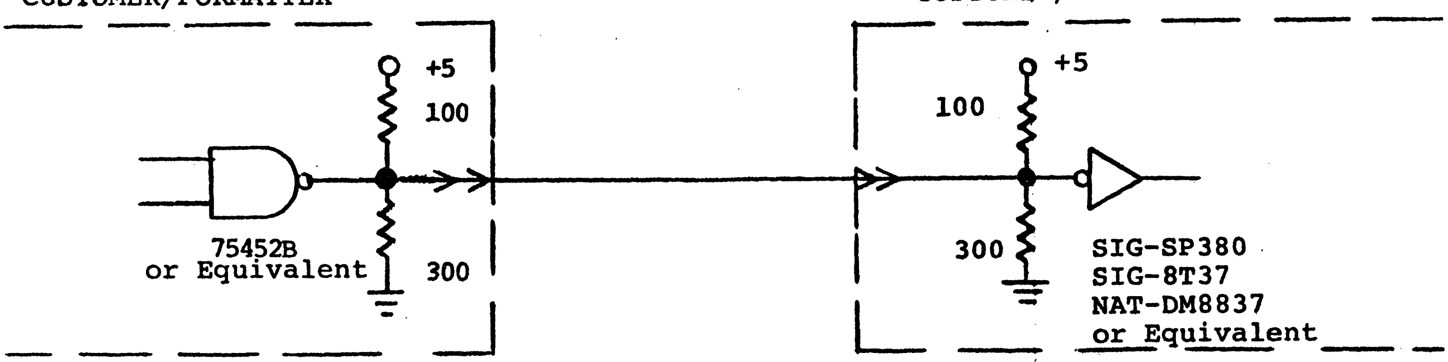
DATA BUS DRIVERS/ RECEIVERS/ TERMINATION (BIDIRECTIONAL)

CUSTOMER INTERFACE FORMATTER



CONTROL SIGNALS DRIVERS/RECEIVERS/TERMINATION

CUSTOMER/FORMATTER CUSTOMER/FORMATTER



RECOMMENDED CABLE

3M Part No. 3476/50; Flat Cable with Ground Plane and Drain Wire

PHOENIX 200 FORMATTER

OPTIONAL DATA PORT

USER INTERFACE CONNECTOR SIGNAL/PIN ASSIGNMENT

PIN #	SIGNAL NAME	DIRECTION
1	GND	-
2	GND	-
3	GND	-
4	DBØL	BI
5	DB1L	BI
6	DB2L	BI
7	DB3L	BI
8	DB4L	BI
9	DB5L	BI
10	DB6L	BI
11	DB7L	BI
12	DB8L	BI
13	DB9L	BI
14	DB10L	BI
15	DB11L	BI
16	DB12L	BI
17	DB13L	BI
18	DB14L	BI
19	DB15L	BI
20	-	-
21	-	-
22	-	-
23	-	-
24	WCROVFL	TO FORMATTER
25	-	-
26	DMRDSTB(L)	TO FORMMATER
27	DMLDSTB(L)	TO FORMATTER
28	-	-
29	-	-
30	-	-
31	-	-
32	-	-
33	-	-
34	-	-
35	-	-
36	DMREQ(L)	FROM FORMATTER
37	DMDIRL	FROM FORMATTER
38	-	-
39	-	-
40	-	-
41	-	-
42	DMACK(L)	TO FORMATTER
43	DSKXFRD(L)	FROM FORMATTER
44	-	-
45	-	-
46	-	-
47	-	-
48	GND	-
49	GND	-
50	GND	-

CONNECTOR TYPE USED ON BOARD 2 OF FORMATTER: 3M # 3433 - 1002

NOTE: BI denotes BIDIRECTIONAL signal

XYLOGIC OEM COMPONENTS GROUP, INC.
42 Third Avenue
Burlington, Massachusetts 01803
617-272-8140

Phoenix 211 Disk Controller

Autoswitch Manual

Dwg. No. 1043-08
Revision Original
Date November 1, 1976

TABLE OF CONTENTS

AUTOSWITCH MANUAL

1.0 Introduction

This manual is intended to provide the user all of the information necessary to utilize the Autoswitch option to the Phoenix 211 Disk Controller.

2.0 Autoswitch Functional Definition/Description

2.1 The Autoswitch Assembly Option is used in conjunction with the Optional Data Port Interface of the Phoenix 211 Disk Controller. (Reference: Phoenix 211 Optional Data Port Manual)

2.2 The Autoswitch option enables the Optional Data Port Interface of the Phoenix 211 Disk Controller to conduct data transfers with either computer memory or user supplied external equipment under direct program control.

2.3 The Autoswitch Option is in effect a programmable digital electronic switch, with two positions; Computer Port Enable and External Port Enable.

2.3.1 Computer Port Mode. When the Autoswitch is in the Computer Port Mode, data transfers can be conducted between the computer memory adn the Data Port Interface of the Phoenix 211 Disk Controller.

This permits the user to access data from and store data on any disks connected to the Phoenix 211 Disk Controller.

2.3.2 External Equipment Mode. When the Autoswitch is in the External Equipment Port Mode, data transfers can be set up by the computer program, but all data transfers will take place between the user supplied external equipment and the Data Port Interface of the Phoenix 211 Disk Controller.

This capability maximizes the total system throughput by eliminiating the necessity to transfer large amounts of bulk data from user external equipment to computer memory and thence from computer memory to the disk drive(s).

2.3.3 Switch Position Control

2.3.3.1 Program Control. The position or mode of the Autoswitch is programmable and may be changed by the execution of one computer instruction. (See Section 4.)

2.3.3.2 Manual Control. A Manual Control Switch is physically provided on each Autoswitch Assembly to permit direct manual control of the Autoswitch position or mode.

3.0 General Specifications

3.1 Physical Size = 8½" X 15" Standard

1.) 8½ X 15" Standard Hex Printed Circuit Board with Notch cut at connector locations A & B to permit board to be mounted in outside slots (1 & 4) of standard DEC DD11 Systems Unit Assembly.

3.2 Weight: 1.5 Lbs. excluding cables

3.3 Power Requirements: 2.0 amps +5V±10% VDC (Supplied by computer or systems unit backplane)

3.4 Environmental: Temperature and humidity tolerances exceed those required by the host computer system.

3.5 Connector Cabling Requirements

3.5.1 External Equipment Port = Connector J2

3.5.2 Computer Data Port = Connector J1

3.5.3 Phoenix 211 Data Port = Connector J3

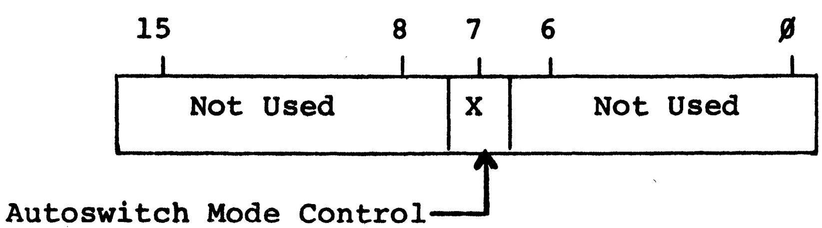
4.0 Programming Interface

4.1 General

The programming of the Autoswitch is as simple as possible. One bit is provided, the state of which is controlled directly by the programmer, and which also determines the mode of the Autoswitch.

This bit can be both loaded and read. Further specifics are given in the following subsections.

4.2 Autoswitch Control Register (Typical Address - 164600)



4.2.2. Register Format Explanation

4.2.2.1 The Autoswitch Mode Control (Bit 07) is the only bit used in the word. All other bits when the register is read will be zeros.

4.2.2.2 The state of the Autoswitch Mode Control Bit directly determines the state of the Autoswitch Mode per the following:

Autoswitch Mode Control Bit State	Autoswitch Mode Selected
Ø	Computer Port
1	External Equipment Port

4.2.3 Initialization Mode. The Austswitch Mode Control bit is initialized to the reset or Computer Port Mode state at system initialization.

4.3 Timing Considerations - None

The Autoswitch switches mode immediately upon command.

5.0 Autoswitch Register Address Selection (Reference Drawing D1-016-02A, Sheet 2)

5.1 The address of the Autoswitch Mode Control Register is completely manually selectable by staples at component locations D8 and D9 on the Autoswitch printed circuit board assembly.

5.2 These staples are labeled A0-A12 and shown on Sheet 2 of logic drawing D1016-02A.

5.3 The nominal staple settings shown represent a base address of 164600.

5.4 Table 1 shows the staples used to control the Autoswitch Register address and the corresponding assigned address bits.

5.5 Appendix A shows the block of addresses to be used for multiple Autoswitch Assignments.

TABLE 1

Phoenix 211 Autoswitch

Mode Control Register Address Selection

(Reference D1016-02A Sheet 2)

Address Bit	Controlling Staple Designation
A12	A12
A11	A11
A10	A10
A09	A09
A09	A09
A08	A07
A06	A06
A05	A05
A04	A04
A03	A03
A02	A02
A01	A01

NOTE:

1.) Staple Removed = logic 1

2.) Staple Inserted = logic Ø

6.0 Manual Mode Control of Autoswitch Mode

6.1 A manually controlled switch is provided at the top left hand side of the Autoswitch printed circuit board assembly.

This switch is provided to facilitate debugging and maintenance of the Autoswitch by enabling an operator to manually force the Autoswitch into the External Equipment Port Mode.

6.2 External Equipment Switch Position

When the mode control switch is in the External Equipment position, data will be transferred between external equipment supplied by the user and the Optional Data Port Interface on Board 2 of the Phoenix 200 Formatter irrespective of program commands.

This is accomplished by unconditionally forcing control element C11 shown on Sheet 2 of D1016-01B to the set or External Equipment state.

6.3 Computer Port Switch Position

When the mode control switch is in the Computer Port Position, data will normally be transferred between the computer memory and the Optional Data Port Interface on Board 2 of the Phoenix 200 Formatter, unless the Autoswitch is specifically commanded to the External Equipment via a user program.

In effect the Computer Port Switch position is the "normal" switch position and enables the Autoswitch mode to be controlled via user program commands.

NOTE:

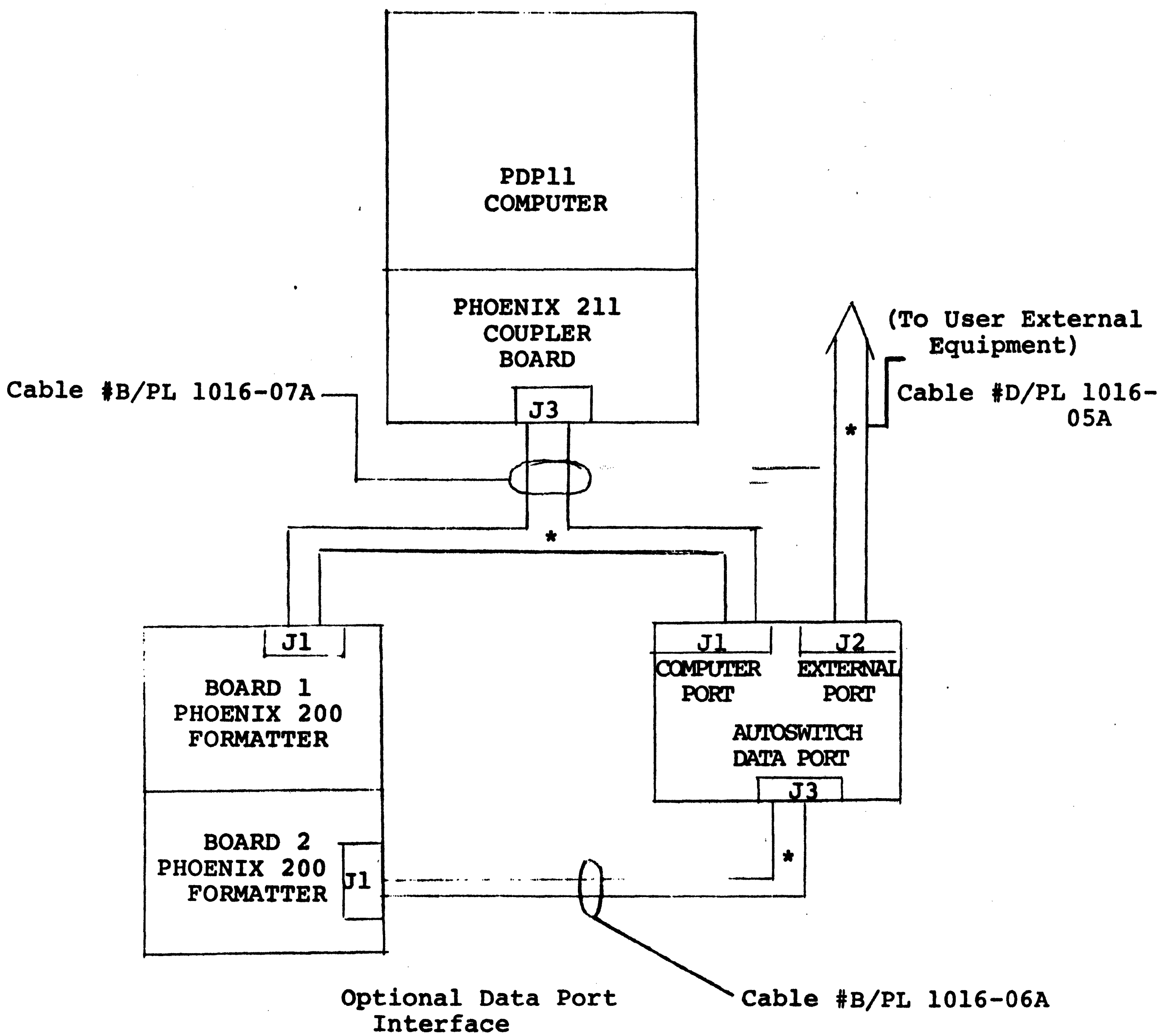
If the Autoswitch Mode Control Switch is put into the "External Equipment Port" position and is then returned to the "Computer Port" position, the Autoswitch will remain in the External Equipment Port Position until a initialization Signal is issued by the host computer or the Autoswitch is specifically commanded to the "Computer Port" position via a user program.

7.0 Phoenix 211 Autoswitch System Configuration and Installation (Reference Figure 1)

Figure 1 shows the physical relationship between the Autoswitch and other elements of a Phoenix 211 disk subsystem.

FIGURE 1

AUTOSWITCH SYSTEM CONFIGURATION



* Shielded Ribbon Cable

Appendix B

Computer Port Connector (J1) Signal/Pin Assignment

PIN #	SIGNAL NAME	DIRECTION
J1	GND	-
J2	GND	-
J3	GND	-
J4	DBØL	BI
J5	DB1L	BI
J6	DB2L	BI
J7	DB3L	BI
J8	DB4L	BI
J9	DB5L	BI
J10	DB6L	BI
J11	DB7L	BI
J12	DB8L	BI
J13	DB9L	BI
J14	DB10L	BI
J15	DB11L	BI
J16	DB12L	BI
J17	DB13L	BI
J18	DB14L	BI
J19	DB15L	BI
J20	-	-
J21	-	-
J22	-	-
J23	-	-
J24	WCROVFL	TO FORMATTER
J25	-	-
J26	DMRDSTB(L)	TO FORMATTER
J27	DMLDSTB(L)	TO FORMATTER
J28	-	-
J29	-	-
J30	-	-
J31	-	-
J32	-	-
J33	-	-
J34	-	-
J35	-	-
J36	DMREQ(L)	FROM FORMATTER
J37	DMDIRL	FROM FORMATTER
J38	-	-
J39	-	-
J40	-	-
J41	-	-
J42	DMACK(L)	TO FORMATTER
J43	DSKXFRD(L)	FROM FORMATTER
J44	-	-
J45	-	-
J46	-	-
J47	-	-
J48	GND	-
J49	GND	-
J50	GND	-

NOTE: BI denotes BIDIRECTIONAL signal

Appendix C

Data Port Connector (J3) Signal/Pin Assignment

PIN #	SIGNAL NAME	DIRECTION
J3-1	GND	-
J3-2	GND	-
J3-3	GND	-
J3-4	DBØL	-
J3-5	DB1L	BI
J3-6	DB2L	BI
J3-7	DB3L	BI
J3-8	DB4L	BI
J3-9	DB5L	BI
J3-10	DB6L	BI
J3-11	DB7L	BI
J3-12	DB8L 1	BI
J3-13	DB9L	BI
J3-14	DB10L	BI
J3-15	DB11L	BI
J3-16	DB12L	BI
J3-17	DB13L	BI
J3-18	DB14L	BI
J3-19	DB15L	BI
J3-20	-	-
J3-21	-	-
J3-22	-	-
J3-23	-	-
J3-24	WCROVFL	TO FORMATTER
J3-25	-	-
J3-26	DMRDSTB(L)	TO FORMATTER
J3-27	DMLDSTB(L)	TO FORMATTER
J3-28	-	-
J3-29	-	-
J3-30	-	-
J3-31	-	-
J3-32	-	-
J3-33	-	-
J3-34	-	-
J3-35	-	-
J3-36	DMREQ(L)	FROM FORMATTER
J3-37	DMDIRL	FROM FORMATTER
J3-38	-	-
J3-39	-	-
J3-40	-	-
J3-41	-	-
J3-42	DMACK(L)	TO FORMATTER
J3-43	DSKXFRD(L)	FROM FORMATTER
J3-44	-	-
J3-45	-	-
J3-46	-	-
J3-47	-	-
J3-48	GND	-
J3-49	GND	-
J3-50	GND	-

NOTE: BI denotes BIDI-RECTIONAL signal

Appendix D

External Equipment Connector (J2) Signal/Pin Assignment

PIN #	SIGNAL NAME	DIRECTION
J2-1	GND	-
J2-2	GND	-
J2-3	GND	-
J2-4	DBØL	BI
J2-5	DB1L	BI
J2-6	DB2L	BI
J2-7	DB3L	BI
J2-8	DB4L	BI
J2-9	DB5L	BI
J2-10	DB6L	BI
J2-11	DB7L	BI
J2-12	DB8L	BI
J2-13	DB9L	BI
J2-14	DB10L	BI
J2-15	DB11L	BI
J2-16	DB12L	BI
J2-17	DB13L	BI
J2-18	DB14L	BI
J2-19	DB15L	BI
J2-20	-	-
J2-21	-	-
J2-22	-	-
J2-23	-	-
J2-24	WCROVFL	TO FORMATTER
J2-25	-	-
J2-26	DMRDSTB(L)	TO FORMATTER
J2-27	DMLDSTB(L)	TO FORMATTER
J2-28	-	-
J2-29	-	-
J2-30	-	-
J2-31	-	-
J2-32	-	-
J2-33	-	-
J2-34	-	-
J2-35	-	-
J2-36	DMREQ(L)	FROM FORMATTER
J2-37	DMDIRL	FROM FORMATTER
J2-38	-	-
J2-39	-	-
J2-40	-	-
J2-41	-	-
J2-42	DMACK(L)	TO FORMATTER
J2-43	DSKXFRD(L)	FROM FORMATTER
J2-44	-	-
J2-45	-	-
J2-46	-	-
J2-47	-	-
J2-48	GND	-
J2-49	GND	-
J2-50	GND	-

NOTE:

1.) CONNECTOR USED ON AUTOSWITCH IS BERG 65268-011.

2.) REQUIRED USER MATCHING CONNECTOR IS 3M3433-1002 OR EQUIVALENT.

3.) BI denotes BIDIRECTIONAL signal

Appendix E

External Equipment Port Required User Specifications

RECOMMENDED INTERFACE

OPTIONAL DATA PORT
Drivers, Receivers, Termination, and Cable

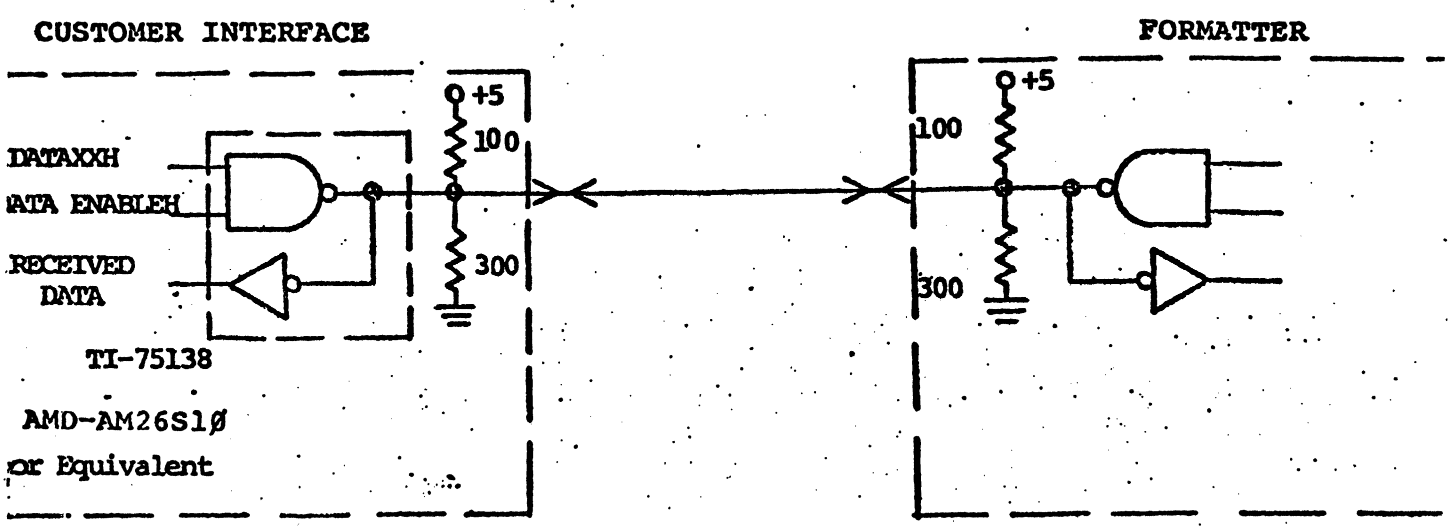
SIGNAL LEVEL DEFINITION:

ALL SIGNALS ARE LOW, ACTIVE TRUE

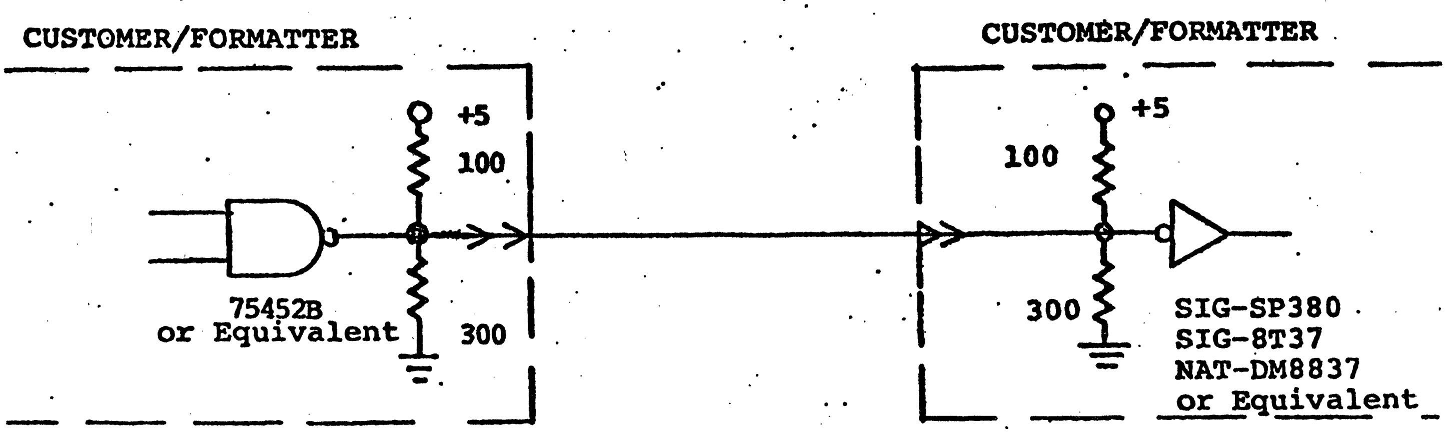
LEVELS: **TRUE= LOW= 0.7 VDC @ 100MA**

FALSE= HIGH= 3.5VDC

DATA BUS DRIVERS/ RECEIVERS/ TERMINATION (BIDIRECTIONAL)



CONTROL SIGNALS DRIVERS/RECEIVERS/TERMINATION



RECOMMENDED CABLE

3M Part No. 3476/50; Flat Cable with Ground Plane and Drain Wire

7.1 Autoswitch Board Installation

The Autoswitch Board is physically a standard "hex" printed circuit board with a notch in the A & B connector section. The board may be physically plugged into a standard small peripheral controller slot (SPC) in any PDP11 computer or BA11K Expansion Box backplane. Installation of the Autoswitch Board simply consists of physically plugging it in to any available backplane slot after having first determined the mode control register address via staple selection as outlined in Section 5.

7.2 Computer Port Cabling (J1)

7.2.1 The Computer Port on the Autoswitch is physically provided by connector J1 on the Autoswitch Board. (Connector J1 is the rightmost connector on the top of the board. Reference Assembly Drawing D1016-03)

7.2.2 J1, the Computer Port Connector is physically cabled to J3 of the Phoenix 211 Coupler Board and J1 of board 1 of the Phoenix 200 Board 1 as shown on Figure 1., via Cable B/PL 1016-07A.

7.2.3 Installation of the Computer Port Cable is accomplished by simply plugging the labelled cable connectors. into the matching connector receptables on the Autoswitch Board, 211 Coupler Board and Board 1 of the formatter.

7.3 Data Port Cabling (J3)

7.3.1 The common Data Port Interface of the Autoswitch is physically provided by connector J3 on the Autoswitch Board. (Connector J3 is the leftmost connector on the top of the board. Reference Assembly Drawing D1016-03)

7.3.2 J3, the Data Port Connector is physically cabled to J1 of board 2 of the Phoenix 200 Formatter as shown on Figure 1 via cable

7.3.3 Installation of the Data Port Cable is accomplished by simply plugging the labelled cable connectors into the matching connector receptacles located on the Autoswitch and board 2 of the Phoenix 200 Formatter.

7.4 External Equipment Port Cabling (J3)

7.4.1 The user External Equipment Port of the Autoswitch is physically provided by connector J2 on the Autoswitch Board. (Connector J2 is the center connector on the top of the board. Reference Assembly Drawing D1016-03)

7.4.2 J2, the External Equipment Connector is to be physcially connected to user supplied external equipment as shown on Figure 1.

NOTE:

1.) The user designed external equipment interface logic must conform to the Phoenix 211 Optional Data Port Interface electrical and signal timing specifications as outlined in the Phoenix 211 Optional Data Port Interface Manual.

2.) Appendix D defines the required physical connector, cable and signal pin assignments of user equipment to be connected to this interface.

8.0 Autoswitch Testing (Reference Figure 2)

The Autoswitch may be tested completely using the standard supplied Phoenix 211 Disk Diagnostics and two cabling configurations as shown in Figure 2.

8.1 Test Position 1 - Computer Port Data Path Test

8.1.1 In the Computer Port Data Path Test, the Coupler cable is connected to the Autoswitch Computer Port Connector and the Autoswitch is electrically switched to the Computer Port position.

NOTE:

This is the default mode of the Autoswitch.

8.1.2 All data transferred between the disk and the computer is thus constrained to flow through the Computer Port of the Autoswitch and the Optional Data Port Interface of the Phoenix 200 Formatter.

8.1.3 The integrity of the Autoswitch Computer Port Data Path is verified by running Diagnostic Tests 0 - 25 as a single group.

8.2 Test Position 2 - External Port Data Path Test

8.2.1 In the External Data Path Test the coupler cable is connected to the Autoswitch External Port Connector and the Autoswitch is electrically switched to the External Port position under direct program control.

NOTE:

1.) The Autoswitch may be forced to the External Equipment Mode by putting the Manual Mode Control Switch located on the top of the Autoswitch Board to the External Equipment Position.

2.) The Autoswitch may be switched to the External Equipment Mode under control of the supplied Phoenix 211 Diagnostic Program by depositing ___1___ in location ___AUTOSW___ of the program.

8.2.2 All data transferred between the disk and the computer is thus constrained to flow through the External Port of the Autoswitch and the Optional Data Port Interface of the Phoenix 200 Formatter.

8.2.3 The integrity of the Autoswitch External Port Data Path is verified by running Diagnostic Tests 0 - 25 as a single group.

8.2.4 The Autoswitch External Port position is selected by depositing a "1" into the "Autoswitch" control memory location of the diagnostic program prior to initiating the test.

Phoenix 211 Autoswitch Manual

Appendix A

Autoswitch Unibus Address Assignments

Autoswitch Unit No.	Assigned Base Address
1	164600
2	164602
3	164604
4	164606
5	164610
6	164612
7	164614
8	164616
9	164620
10	164622
11	164624
12	164626
13	164630
14	164632
15	164634
16	164636
17	164640
18	164642
20	164646
21	164650
22	164652
23	164654
24	164656

XYLOGIC OEM COMPONENTS GROUP, INC.
42 THIRD AVENUE
BURLINGTON, MASS. 01803
617-272-8140

PHOENIX 211 DISK CONTROLLER

MULTIPLE COMPUTER PORT OPTION SPECIFICATION

Revision B

October 15, 1977

Phoenix 211 Disk Controller
Multiple Computer Port Option

1. Purpose

The multiple computer port option enables up to four PDP11 Computers to retrieve data from and store data on common storage module type disk drives connected to one Phoenix 200 Disk Formatter. Access to the common formatter is provided on a selectable equal priority or controlled priority basis.

2. Overview - (Reference Figure 1)

Each PDP11 Computer in a multiple computer disk subsystem contains a Phoenix 211 Coupler Board which physically interfaces the computer to the common formatter bus.

The common formatter bus is "daisy-chained" from the formatter to each of the 211 couplers associated with each computer to be included in the disk system. The formatter interface is terminated at the last 211 coupler board.

Arbitration logic is contained within the formatter to insure that the common formatter resource is allocated to the connected computers on a request basis with failsafe timeout provisions.

3.0 Operation

3.1 General

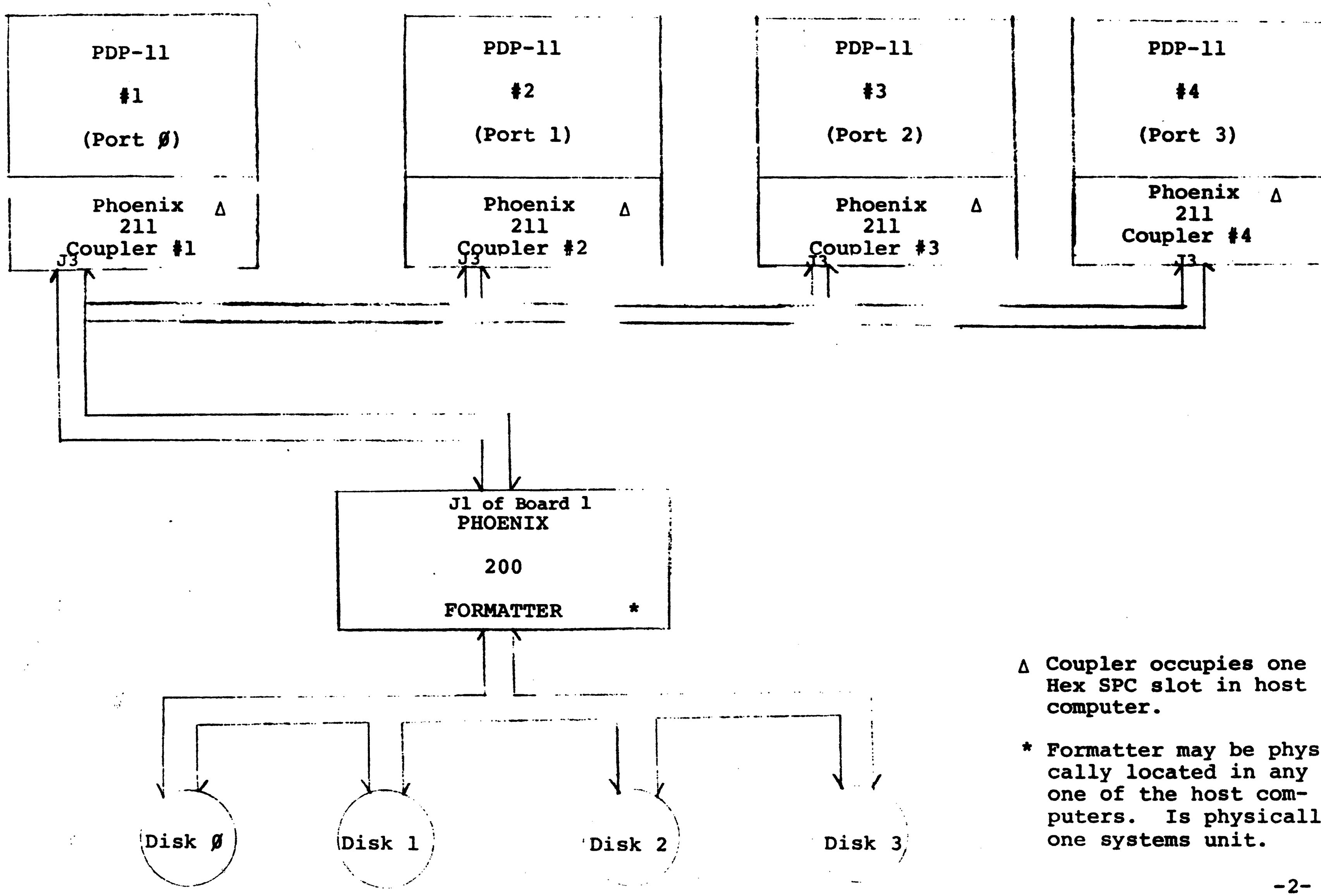
The common formatter is "shared" between all computers in the system on a mutual exclusive basis. Requests are made to the computer port arbitrator by each of the computers whenever disk operations are to be performed.

The arbitrator fields all requests for the use of the formatter and assigns the formatter to the requesting computer ports on a equal or selectable controlled priority basis.

Once the formatter has been assigned to a given port, the associated computer may use the formatter to conduct as many disk operations as desired. When the formatter is so assigned to one computer port in the system all

Figure 1

XYLOGIC PHOENIX MULTIPLE PDP-11 COMPUTER DISK SYSTEM CONFIGURATION



Δ Coupler occupies one Hex SPC slot in host computer.

* Formatter may be physically located in any one of the host computers. Is physically one systems unit.

other computer ports are prohibited from using the formatter.

The formatter is normally released for use by other ports when the current assigned computer port removes its "request" signal at the conclusion of its disk operations.

However, the arbitrator also monitors actual formatter usage and will automatically reassign the formatter to another port if no activity is detected on the currently assigned port within a selectable, predefined time interval.

When the formatter is assigned to a requesting computer port by the arbitrator an interrupt is automatically generated to the associated computer by the 211 coupler board to facilitate system software.

3.2 Detailed Operation

3.2.1 Formatter Request Bit (Bit 5 of the 211 DCSR)

Requests for use of the formatter by any computer port are initiated by setting the "Formatter Request" Bit.

This Bit sets an internal control memory element which sends a unique request signal to the arbitrator.

No further action can be taken by the programmer until the arbitrator assigns the formatter to his port, at which time the controller becomes ready and will generate an interrupt, if enabled.

The Formatter Request bit must be set and the port be assigned as per above before any disk operations can be initiated.

3.2.2 Computer Port Assignment

1. The arbitrator utilizes a four state scanner and associated control logic to allocate the common formatter to the maximum of four computer ports.

2. When the formatter is unassigned, the arbitrator sequentially scans the request signals from the computer ports. When a request signal is detected the scanner is stopped and the binary state is issued to all computer ports over two signal interfaces.

3. Each computer port is assigned a strappable port number. (Ø-3). See Appendix A for staple selection.

   When the assigned port number is issued by the arbitrator, all requesting ports automatically compare this value with the associated preassigned physical port number.

The requesting port associated with the matching preassigned port number then becomes ready, generates an interrupt to the host computer, and is free to initiate any and all appropriate disk transfers via the formatter.

3.2.3 Formatter Release

When all desired disk operations have been completed by the current assigned computer port, the formatter is released for use by other computer ports by resetting the Formatter request bit.

This removes the associated port request signal to the formatter arbitrator, and causes the arbitrator to reassign the formatter to some other requesting computer port.

3.2.4 Formatter Reassignment Priority Options

Computer Port Formatter Requests may be serviced on a selectable equal priority or a controlled priority basis.

If the equal priority option is chosen, the arbitrator scanner will always sequentially scan all of the computer ports for formatter requests. Hence, all computer ports have an equal opportunity to obtain formatter access.

If the controlled priority option is selected, the arbitrator scanner will always be reset to state Ø when released, and then sequentially scan the associated ports for new requests.

Hence, this option assigns the highest priority to port Ø and the least priority to port 3.

3.2.5 Automatic Timeout Port Reassignment

The arbitrator contains a automatic timeout feature which will automatically reassign the formatter to another requesting port if no activity is detected on the currently assigned port within a selectable, predefined interval.

4.0 Physical Configuration

4.1 Phoenix 211 PDP11 Coupler

One Phoenix 211 PDP11 Coupler Board is required for each PDP11 Computer that is to utilize the Phoenix 200 Formatter.

4.2 Phoenix 200 Formatter

The common Phoenix 200 Formatter is packaged in one PDP11 systems unit and may reside in any one of the system PDP11 computers.

The formatter physically interfaces to the system disk drives.

4.3 Coupler Cabling

A shielded ribbon cable is required between the formatter and each cpu coupler card.

APPENDIX A

MULTIPLE CPU STAPLE SELECTION

1) To enable the Multi-CPU option

   A) Install W36 on Phoenix 200 Board 1
   B) Remove W22 on Phoenix 211 coupler card

2) 211 Coupler Port Selection

Port	W54	W55	W56	W57	W36	W37	W38	W39
Ø	IN	OUT	OUT	OUT	IN	OUT	OUT	OUT
1	OUT	IN	OUT	OUT	OUT	IN	OUT	OUT
2	OUT	OUT	IN	OUT	OUT	OUT	IN	OUT
3	OUT	OUT	OUT	IN	OUT	OUT	OUT	IN

NOTE: On a given system each coupler must be assigned a different port number.

3) To enable the controlled priority option install W41 on Phoenix 200 Board 1.

XYLOGIC OEM COMPONENTS GROUP, INC.

42 THIRD AVENUE

BURLINGTON, MASSACHUSETTS

PHOENIX 211 ECC SPECIFICATION

Rev. C

November 30, 1977

1. Introduction

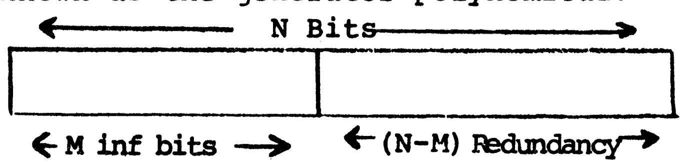
Storage Module media is not as perfect as Disk Drive Vendors would like it to be. The bit recording density of about 6000 bpi is state of the art and requires a head flying height and coating thickness of about 20 micro-inches. (The figure for 2314 type packs is 60 micro-inches.)

Practical testing has revealed that 100% of all unrecoverable errors are one, two or three bit bursts. Disk Drive Vendors specify that a track shall not have more than one burst error and it, in turn, shall be no longer than 11 bits long. Cylinder 0 (heads 0 and 1) shall have no errors. Also, there shall be no more than 30 correctable error tracks per 80 MB pack.

Based on the above, a Storage Module Disk System has the ability to either bypass bad tracks or to perform error correction on them. One solution to the potential problem is an error correction scheme.

2. Principles of Error Detection

A checkword is appended to the message (data field) when the information is written on the disk. The checkword is obtained as the remainder when the message is divided by a carefully chosen number known as the generator polynomical.



The above figure shows a message of M information bits having added to it (N-M) redundancy bits. The redundancy bits are achieved as follows:

Let M(X) equal the message polynomial.
Let P(X) equal the generator polynomial.

Hence, $M(X) = Q(X) + R(X)$ where $Q(X)$ and $R(X)$ are respectively the quotient and remainder following division. The quotient has no real significance and is discarded but R(X) becomes the (N-M) redundancy bits shown above. Hence, the total message (information plus redundancy) can be expressed as:

$$N(X) = X^{(N-M)} M(X) + R(X)$$
$$= Q(X)\ P(X)$$

If the received message can be divided by the polynomial P(X) without a remainder, then it has been received correctly and further action is not needed.

3. Choice of Polynomial

The strength or weakness of a redundancy scheme is intimately tied to the choice of polynomial and this depends on the type of errors to be expected. Data transmission suffers from long burst errors. High-speed solid-state memories suffer from isolated single-bit errors. Rotating magnetic memories compromise between these two extremes and suffer from short bursts (1-3 bits). The polynomial chosen was a Fire Code and thus is of the following form:

$$P(X) = P1(X)\ (X^{C} + 1)$$

Where P(X) is the generator polynomial for a Fire Code; this must have two properties. (1) P1(X) is a primitive (irreducible) polynomial of degree M and order E. (Note, the degree of a polynomial is defined to be the greatest power of X in which the coefficient is non-zero and E is defined to be $(2^{M}-1)$. (2) The parameter C must not be divisable by E.

The above Fire code polynomial will have the following properties: (1) The length of the code, N, is equal to the least common multiple of E and C. This works out to be $(2^{M}-1)C$. (2) The number of redundancy bits is equal to (M+C). (3) The number of information bits, M, is equal to $(2^{M}-1)C-(M+C)$.

The polynomial chosen for the Phoenix 211 Disk Controller is as follows:

$$P(X) = (X^{11} + X^{2} + 1)\ (X^{21} + 1)$$

The degree of the P1 (X) portion is 11 and E is therefore equal to $(2^{11}-1)$ or 2047. The length of the code is equal to (E-C) where C equals 21. Hence, code length equals (2047-21) bits. The number of redundancy bits is equal to (M + C) or (11 + 21).

In summary, the above polynomial will support a record length up to 2680 words and each record will be followed by a 32 bit checkword.

4. Error Detection and Correction Performance

The polynomial chosen was $P(X) = (X^{11} + X^{2} + 1)(X^{21} + 1)$

$$= X^{32} + X^{23} + X^{21} + X^{11} + X^{2} + 1$$

This will <u>detect</u> two error burst of combined length 22, one error burst of length 32 and any odd number of errors. This will <u>correct</u> any single burst up to 11 bits long.

5. Programming Details

The Phoenix 211 has an ECC (error correction code) capability which will detect and correct an error by reconstructing a portion of the data within the specified code word length, which is fixed. The burst ECC code will correct an error which must fall within the specified length of the burst. The actual location of the burst within the code word (data field of a sector) is irrelevant.

Any errors outside the specified burst length will be detected but not corrected. The ECC hardware, in this case, will yield an ECC uncorrectable error. The Phoenix 211 logic contains the hardware to find the burst within which the read error is included and determine the exact location of the burst within the data field.

The ECC pattern register contains the actual error burst and the ECC positon register contains the address for determining the actual location of the error burst within the data field.

NOTE: The actual correction of the data field is done by the software with the help of the ECC position and ECC position and ECC pattern registers.

6. ECC Register Formats

(A) ECC Bit Position Register (764016)

UCERR	Ø	Ø	BPR 4096	BPR 2048	BPR 1024	BPR 512	BPR 256	BPR 128	BPR 64	BPR 32	BPR 16	BPR 8	BPR 4	BPR 2	BPR 1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Name	Function
0-12	Bit Position Address (Read Only)	Contains the exact location of the error burst within the data field following the completion of the error correction procedure. Upon completion of the ECC process the formatter will load this register with the physical location of the first bit of the 11-bit error burst. The position register will only be loaded if the ECC logic is enabled by having the ECC Inhibit Bit reset.

Bit	Name	Function
13-14	Not Used (Read Only)	Always zeros
15	Uncorrectable ECC Error (Read Only)	Set when the conclusion of the error correction procedure indicates that the error was an uncorrectable ECC error. Cleared by UNIBUS INIT, Formatter Clear, or a "GO" Bit.

(B) ECC Pattern Register (764020)

ECC INHB	Ø	Ø	Ø	Ø	ECC PAT 10	ECC PAT 9	ECC PAT 8	ECC PAT 7	ECC PAT 6	ECC PAT 5	ECC PAT 4	ECC PAT 3	ECC PAT 2	ECC PAT 1	ECC PAT Ø
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Ø

Bit	Name	Function
0-10	ECC Pattern (Read Only)	Contains the actual error burst available at the completion of the ECC internal to the formatter logic error correction process. These bits will be exclusive OR'd with the eleven bits in error by software. The software will use the contents of the ECC Position Register to find the actual location of the error burst in the data field, then the error burst itself will determine the bits in error within the 11-bit field.
11-14	Not Used (Read Only)	Always Zeros
15	ECC Inhibit (Read/Write)	Set when the software desires to inhibit error correction. If ECC Inhibit is set, error correction code is disallowed; if ECC Inhibit is reset, the error correction process is allowed. Cleared by Unibus INIT, formatter clear, or by writing a zero. If a data error is detected at the end of the data transmission in the read mode with the ECC Inhibit Bit reset, the formatter will immediately go into the ECC correction process. Prior to beginning the correction process, the formatter will also set the CRC Error

Function(Cont'd.)

Bit (Bit 8 of the Error Status Register), which will remain set until a UNIBUS Init, formatter Clear, or a "GO" Bit is received.

APPENDIX A

ECC DUAL HEADER FOR CDC SMD'S 9760, 9762, 9764, 9766

Preamble #1 240 Bits	HDR SYNC 2 Bits	HDR 1 32 Bits	HDR 1 CRC 32 Bits	GAP 16 Zeros	HDR 2 32 Bits	HDR 2 CRC 32 Bits	Preamble #2 96 Bits	Data Sync 2Bits	DATA 4096 Bits	DATA ECC(CRC) 32 Bits	POSTAMBLE 428 Bits

APPENDIX B

ECC Board I Switch Settings (300-031-90X)

Normal Operation

K7-1 ON	E5-3 ON	
K7-8 ON	E5-7 ON	All other switches OFF
K7-2 ON	E5-4 ON	
K7-3 ON	E5-5 ON	
K7-7 ON		

For Normal Operation with ECC with 256 Words/Sector

K7-1 ON	E5-3 ON	C2-1 ON
K7-8 ON	E5-4 ON	C2-7 ON
K7-2 ON	E5-5 ON	
K7-3 ON	E5-8 ON	
K7-7 ON	E5-6 ON	

D6-2 ON	D7-1 ON	
D6-5 ON	D7-3 ON	
D6-7 ON	D7-4 ON	All other switches OFF
D6-8 ON	D7-5 ON	
	D7-8 ON	

XYLOGIC OEM COMPONENTS GROUP, INC.

42 Third Avenue

Burlington, Massachusetts 01803

ROM Bootstrap Board

&

Phoenix 211 Disk Controller Bootstrap Program

1. General Description

The Xylogic ROM Bootstrap Board is a quad sized printed circuit board which may be plugged into any standard small peripheral controller (SPC) I/O slots in any standard PDP11 Unibus backplane.

The ROM Bootstrap Board contains sockets to accommodate up to a maximum of 512 words of PROM program storage.

2. Phoenix 211 Disk Controller Bootstrap Program

2.1 The Bootstrap Program for the Phoenix 211 Disk Controller is contained in two 32 x 8 Proms on the bootstrap printed circuit board.

2.2 The Phoenix 211 Disk Controller Program starting address is 173700.

2.3 The boot program utilizes disk drive unit Ø of a Phoenix 211 Disk Drive Subsystem.

2.4 The program reads data absolute from sectors Ø and 1 (sector Ø, head Ø, cylinder Ø) of unit Ø and transfers data obtained to computer memory starting at absolute memory location zero.

2.5 When the boot operation is successfully completed program control is transferred to absolute memory location Ø.

2.6 If an error is detected, the operation will be repeated.

3. Phoenix 211 Disl Controller Bootstrap Program Installation Procedure

3.1 Plug the XYLOGIC ROM Bootstrap Board Assembly into any available standard small peripheral controller I/O slot in the PDP11 computer or Expansion Box.

Caution: Insure that all system power is off before plugging or removing any boards in system.

3.2 Turn power on system.

3.3 Verify proper operation of Bootstrap Board by checking proper ROM content via computer console.

3.3.1 Set 173700 in Console Switch Register

3.3.2 Momentarily Depress the Console "Load Address" Register Switch

3.3.3 Momentarily depress the "Examine" Console Switch

3.3.4 Observe that the contents of the Phoenix 211 ROM Bootstrap Program memory location is the same as that specified on the Phoenix 211 Bootstrap Program Listing.

3.3.5 Repeat steps 3.3.3 and 3.3.4 until all of bootstrap memory locations have been verified for proper content.

4. Phoenix 211 Disk Controller Bootstrap Program Operating Procedures

4.1 Insure that disk drive unit Ø contains proper operating system disk pack, has power applied, and is enabled for use.

4.2 Set 173700 in Console Switch Register

4.3 Put Console "Enable/Halt" Switch in "Enable" position

4.4 Momentarily depress the console "Load Address" Switch

4.5 Momentarily depress the console "Start" Switch

4.6 Absolute Sectors Ø and 1 of disk drive Ø will then be read, transferred to memory, and control transferred to absolute memory location zero.

XYLOGIC OEM COMPONENTS GROUP, INC.

42 Third Avenue

Burlington, Massachusetts 01803

617-272-8140

Phoenix 211 Disk Controller

Maintenance Manual

Dwg. No. 900-006-301

Revision A

Date March 10, 1977

TABLE OF CONTENTS

MAINTENANCE MANUAL

Phoenix 211 Disk Controller

1.0 MAINTENANCE PHILOSOPHY

The maintenance philosophy for the Phoenix 211 Disk Controller that appears to make the most sense for Xylogics and the majority of its customers is:

1. Troubleshoot to the printed circuit board level

2. Swap out any defective boards

3. Have any defective boards repaired at the factory for a nominal charge.

Execution of such a maintenance philosophy should result in minimizing customer down time and facilitate third party or customer service of equipment in the field. Accordingly all Xylogic documentation is geared towards isolation of any problems to the board level.

2.0 PHOENIX 211 DISK SUBSYSTEM USER MAINTENANCE ALTERNATIVES

2.1.0 GENERAL

A Phoenix 211 Disk Subsystem consists of the Phoenix 211 Disk Controller and at least one disk drive. The disk controller is solid state and requires no preventitive maintenance, while the disk drive requires regularly scheduled preventitive maintenance in order to keep it operating.

Three options are open to the user with respect to maintenance of a Phoenix 211 Disk Subsystem.

- Contracted Third Party Maintenance (ICE)
- On Call Maintenance (ICE/XYLOGICS)
- User Maintenance

2.1.1 THIRD PARTY MAINTENANCE

Third Party Maintenance of a Phoenix 211 Disk Subsystem is normally available through Iomec Customer Engineering, Incorporated (ICE). ICE has offices nationwide, and can provide service on a reasonable monthly contract basis for both the Phoenix 211 Disk Controller and associated disk drives. Typically, a maintenance contract with ICE covers scheduled preventitive maintenance calls and parts and labor on repair calls. Third Party Maintenance is strongly recommended for small OEM and end users. Response time for contract maintenance is less than 8 hours.

2.1.2 ON CALL MAINTENANCE

On Call Maintenance is sometimes opted in lieu of contracted maintenance. On Call Maintenance can be obtained according to the following groundrules.

1. Response Time:

   On Call Maintenance requests for service to ICE will be honored only after all outstanding requests for contracted service have been taken care of. On Call Maintenance requests for service to Xylogics will be honored only in the event that service from ICE is not available in the customer area. Such service will be scheduled at Xylogic's convenience. A user opting for On Call Maintenance must be willing to live with possible response times of 24-96 hours, since he is at the bottom of ICE and Xylogic priority.

2. Charges:

   Charges for On Call Maintenance visits are based on a time, materials, and expense basis. Hence, a customer will pay for all material, any and all travel expenses, and direct labor at a predefined rate. A Purchase Order number, to cover all such charges, will be obtained from any user prior to making the service visit.

2.3 Considerations:

A user who chooses to utilize On Call Maintenance, rather than Contract Maintenance is effectively betting that the equipment will be reliable to the point that the On Call costs will be less than the Contract Maintenance costs.

Such a user must be willing to live with the slow response time and be willing to take the risk that it may cost him more if a catastrophic disk drive failure occurs.

Thirdly, such a user must also assume the responsibility for seeing that the proper disk drive preventitive maintenance procedures are followed or he can be guaranteed of a catastrophic disk drive failure.

2.1.3 USER MAINTENANCE

In User Maintenance the user assumes complete maintenance responsibility for the Phoenix 211 Disk Subsystem. To perform maintenance properly, the user must have both capable trained personnel and proper maintenance equipment.

2.1.3.1 SPARES AND MAINTENANCE EQUIPMENT REQUIREMENTS

A list of the required equipment necessary for the maintenance of both the Phoenix 211 Disk Controller and a Control Data Corp. 9762, 80 Megabyte Disk Drive are listed in Appendix A. Note that the capital outlay for spare parts and maintenance equipment is in excess of $8000.00.

2.1.3.2 TRAINED PERSONNEL REQUIREMENTS

The most important ingredient in a user maintenance program is the availability of capable personnel who have a systematic approach to problem solving, and who have been properly trained on, and are familiar with both the Phoenix 211 Disk Controller and user associated disk drives.

Training of personnel on the disk controller and associated disk drives requires two weeks, with one week being spent on each. Use of untrained user personnel, no matter how capable, to maintain a Phoenix 211 Disk Subsystem is not recommended. Such individuals will not be able to get the job done properly, even if provided with a 100% complement of spares and maintenance equipment, and are likely to induce failures into a subsystem, by abusing the equipment physically, and/or electrically. An untrained field service engineer is worse than no field service engineer. We are all too familiar with the true war stories of personnel who are "experts" at taking apart equipment, but who, under no circumstances can put it back together again and make it work.

2.1.3.3 CONSIDERATIONS

Assumption of maintenance responsibility by the user is not to be taken lightly or without serious consideration of the economics involved. If the field service engineer were to be dedicated to Phoenix 211 Subsystems, it would take a total of 12 Disk Drives and 6 Phoenix 211 Disk Drives to break even, in terms of paying for labor, parts, and equipment.

3.0 MAINTENANCE TOOLS

3.1 Disk Diagnostic Program

The chief maintenance tool provided by Xylogics is the Phoenix 211 Disk Diagnostic Program. This program is designed with the Philosophy that nothing in the disk controller operates properly, and it attempts to prove just that.

The Disk Diagnostic initially exercises the smallest amount of disk controller logic possible to determine proper operation. Once proper operation has been verified, this logic is then used to exercise additonal logic. This process is then repeated until all of the disk controller logic has been tested.

A natural byproduct of this approach is that the logic being exercised in any test or subtest can usually be readily identified.

3.2 Disk Diagnostic Program Manual

The Phoenix 211 Disk Diagnostic Program Manual provides the user with complete instructions on how to operate the Phoenix 211 Disk Diagnostic Program in addition to specific details on what functional logic is being exercised by each test, and on what board(s)the logic is physically contained on are provided.

The intent is to enable the user to determine what functional logic is inoperative, and what board(s) are suspect by running the diagnostic program and noting what specific test and subtest an error occurs on.

3.3 Functional Logic Partitioning Guide

For those users who desire to become involved in troubleshooting to the component level a functional logic partitioning guide is provided which identifies each functional logic block by name, physical board where it is located, and a drawing sheet reference.

3.4 Documentation Conventions Glossary

A documentation convention glossary is also provided, which explains in detail all documentation conventions and symbology utilized by Xylogics in the Phoenix 211 logic drawings.

This glossary should go a long way towards minimizing the amount of user time required to familiarize himself with the hardware for maintenance purposes.

3. 5. Hardware Documentation Package

A complete hardware documentation package is supplied with each Phoenix 211 Disk Controller. This hardware documentation package includes:

- Logic Drawings
- Assembly Drawings
- Parts Lists
- Wiring Lists
- Drawing Directory

Care has been taken to maximize the amount of information provided to the user by functionally partitioning, labeling, and adding notes wherever appropriate to all logic drawings.

3. 6. Operators Manual

An operator's manual is provided which covers all topics from physical unpacking, installation, and checkout to providing detailed descriptions of each user defined parameters, and instructions on how to change such parameters.

The operator's manual provides an insight into the myriad of options provided by the Phoenix 211 Disk Controller to the user, as well as an insight into how the parameters are logically implemented in the disk controller.

The Phoenix 211 operator's manual is recommended reading for those users intending to perform their own component level maintenance.

3. 7. Phoenix 200 Formatter Interface Manual

The Phoenix 211 Disk Controller physically consists of the Phoenix 200 Disk Formatter and the Phoenix 211 Interface Board.

The Phoenix 200 Disk Formatter is equipped with a standard interface which is adapted to the PDP-11 Unibus by the Phoenix 211 Interface Board. (The Phoenix 200 Formatter has been interfaced to many other computers as well)

The Phoenix 200 Formatter Interface Manual describes in detail the signal, electrical, and timing relationships required to cause the Phoenix 200 Disk Formatter to perform the required disk data transfer operations.

This manual is likewise, recommended reading material for those users interested in performing their own component level maintenance.

4.0 PHOENIX 211 DISK SUBSYSTEM MAINTENANCE DON'TS

1. Do not touch any hardware associated with the disk controller or disk drive without first analizing fault symptom, as defined by supplied diagnostic program.

2. Do not remove any printed circuit board of disk controller or disk drive without first turning power off.

3. Do not, under any circumstances, take down another working system in order to attempt to isolate a fault in a defective system. More often than not, two defective systems will result.

4. Do not, under any circumstances, randomly substitute spare boards into a defective system. More often than not defective system will not be fixed and disposition of spare boards will be compromised.

5. Do not substitute more than one spare board at a time into a Phoenix 211 Disk Subsystem. Changing more than one board at a time does not facilitate fault isolation to the board level and increases the probability of inducing additional faults into the system, by inserting boards into the wrong backplance locations (particularly on the disk drives).

6. Do not attempt to diagnose a disk subsystem problem using any type of operating system or with any valued data pack on any of the disk drives connected to the disk controller. Use of any data pack on a suspect disk subsystem will more likely than not, result in the loss of data contained on any such pack, and will not provide any useful diagnostic information.

7. Do not call Xylogics for assistance if a disk subsystem problem occurs, until after the supplied disk diagnostic program has been run and the following information can be supplied to Xylogics personnel.

    - Failing Test Number
    - Failing Subtest Number
    - Actual Erroneous Test Data Obtained
    - Reference Test Data
    - Failing Program Count Value
    - System Configuration
        - Computer Model
        - Memory Capacity
        - Other peripherals on System
        - Position on Bus of Phoenix 211 Interface Board
    - Disk Drive Type and Model
    - Number of Disk Drives

Telephone calls indicating such system failure symptoms as "FAILED TO BOOT", "HOME BLOCK READ ERROR", etc. have no significance to Xylogics personnel and are a waste of time and money.

Phoenix 211 Disk Controller

Maintenance Manual

5.0 Controller Physical Organization and Logic Partitioning

5.10 General

The Phoenix 211 Disk Controller physically consists of a total of five hex 8½" x 15" printed circuit boards compatible with any standard DEC backplane assembly.

5.20 Phoenix 200 Formatter

5.2.1 Four of the printed circuit boards are mounted in a supplied four slot systems unit type backplane assembly.

5.2.2 The systems unit backplane assembly can be mounted directly in any standard computer or BA11K Expansion Chassis which physically has room.

5.2.3 Power for the backplane assembly and associated printed circuit boards is provided by the host computer or expansion box, and a power harness to connect to the power distribution connector of the host is also supplied.

5.2.4 The four boards and backplane assembly constitute the Phoenix 200 Formatter which is also sold as a separate product by Xylogics.

5.2.5 The four printed circuit boards comprising the Phoenix 200 Formatter are identified as simply "Phoenix 200 Formatter Board 1,2, 3, or 4" in all associated documentation.

5.2.6 The Phoenix 200 Formatter is not physically tied to the host PDP11 Computer Unibus.

5.2.7 Each pin of all connectors on a given I/O slot of formatter backplane assembly is bussed to the same pin on all of the other three I/O slots of the backplane assembly.

Hence any Formatter board can be plugged into any I/O slot of the backplane assembly to facilitate maintenance operations.

5.30 Phoenix 211 Coupler (PDP11 Interface Board)

5.3.1 The fifth printed circuit board assembly, the Phoenix 211 Coupler, is used to interface the Phoenix 200 Formatter to the PDP11 Unibus.

5.3.2 The Phoenix 211 Coupler Board occupies one standard hex Small Peripheral Controller (SPC) I/O slot available in any DEC PDP11 computer or expansion chassis, and is connected to the Unibus or said computer.

5.3.3 The hex I/O slot used for the Phoenix 211 Coupler Board must be altered to allow the DMA Non Processor Grant Signal to pass serially through the board as specified in the installation instructions.

NOTE: In most cases the alteration consists of cutting one wire on the I/O slot backplane.

5.3.4 The Phoenix 211 Coupler Board is physically connected to the Phoenix 200 Formatter via a 50 conductor shielded ribbon cable provided for that purpose.

5.4.0 Physical Logic Partitioning

The total logic of the disk controller has been physically partitioned on a functional basis to achieve modularity and flexibility.

The following sheets provide a tabulation of the functional logic blocks contained on each of the five controller printed circuit board assemblies, as well as logic drawing sheet reference to facilitate controller maintenance.

Phoenix 211 Disk Controller

Physical Logic Partitioning

Phoenix 211 PDP11 Interface Board

The following functional logic blocks are physically contained on the Phoenix 211 PDP11 Interface Board. (Reference Logic Drawing D1025)

Functional Logic Block	Drawing Sheet Reference
1) Unibus Data Bus Buffering Logic	1
2) Unibus Address Bus Buffering Logic	1
3) Interrupt Vector Logic	1
4) Interrupt Priority Selection Logic	1
5) Unibus Control Signal Buffering Logic	1
6) Unibus DMA Control Logic	2
7) Unibus Interrupt Control Logic	2
8) DMA Throttle Logic	2
9) Main Bus Address Register	3
10) Strip Option Address Register	3
11) Strip Control Logic	3
12) DMA Word Count Register	4
13) Register Address Decode Logic	4
14) Formatter Interface Logic	5

Phoenix 211 Disk Controller

Physical Logic Partitioning

Phoenix 200 Formatter Board 1

(Non Queue)

The following functional logic blocks are physically contained on Board 1 of the Pheonix 200 Formatter. (Reference Logic Drawing D1032)

Functional Logic Block	Drawing Sheet Reference
1) Formatter Coupler Interface Logic	1
2) Formatter Register Address Control Logic	1
3) Formatter Interrupt Request Logic	1
4) Command & Status Register	2
5) General Error Logic	2
6) Error Register	2
7) Multiple Computer Arbitration Logic	3

Pheonix 211 Disk Controller

Physical Logic Partitioning

Phoenix 200 Formatter Board 2

The following functional logic blocks are physically contained on the Pheonix 200 Formatter Board 2. (Reference Logic Drawing D1033-01)

Functional Logic Block	Drawing Sheet Reference
1) Seek Logic	3
2) Disk Status Register	4
3) FIFO Transfer Request Delay Logic	4
4) Data Late Error Detection Logic	4
5) DMA Request Control Logic	4
6) Even, Odd FIFO Memories	5
7) Optional Data Port Interface	1
8) FIFO Read Data Multiplexing Logic	1
9) FIFO Write Data Multiplexing Logic	1
10) FIFO Write/Read Control Logic	1
11) Serial to Parallel Data Conversion	2
a.) Data Shift Register #1	
b.) Data Shift Register #2	
c.) Shift Register Data Multiplexer	
12) Data Shift Register Control Logic	2
13) Write Check Error Logic	2

Phoenix 211 Disk Controller

Physical Logic Partitioning

Phoenix 200 Formatter Board 3

The following functional logic blocks are physically contained on the Pheonix 200 Formatter Board 3. (Reference Logic Drawing D1034-01)

	Functional Logic Block	Drawing Sheet Reference
1)	CRC Generation & Check Logic	1
2)	Self Test Logic	1
3)	Bad Sector Ident Detect & Skip Logic	1
4)	Sector Write Protect Logic	1
5)	Header Check Logic	1
6)	Disk Sequencer Logic	3
7)	Disk Word Count Register	4
8)	Sequencer Word Count Logic	4
9)	Controller Busy/Done Logic	4
10)	Seek Start Logic	4
11)	Servo Offset Done Delay Logic	4
12)	Linkage Strip Word Count Logic	4
13)	Unit, Sector, Head Register	5
14)	Maximum Sector Detect Logic	5
15)	Maximum Head Detect Logic	5
16)	Head and Cylinder Incrementation Logic	5
17)	Command Decode Logic	2
18)	End of Sequence Control Logic	2
19)	Sector Coincidence Enable Logic	2
20)	Maintenance Panel Switch Write Protect Logic	2
21)	Maintenance Panel Indicator Logic	2

Phoenix 211 Disk Controller

Physical Logic Partitioning

Phoenix 200 Formatter Control Data Corp. Compatible Board 4

The following functional Logic Blocks are physically contained on Board 4 of the Phoenix 200 Formatter. (Reference Logic Drawing D1035)

Functional Logic Block	Drawing Sheet Reference
1) Power Fail Detect Logic	1
2) Disk Control Bus Logic	1
a.) Control Data Bus Drivers	
b.) Tag Bus Drivers	
c.) Cylinder Tag Logic	
d.) Head Tag Logic	
e.) Control Tag Logic	
f.) Tag Timing Logic	
3) Miscellaneous A Cable Disk Interface Logic	1
a.) Fault	
b.) Seek Error	
c.) On Cylinder	
d.) Unit Ready	
e.) Unit Select Ø, 1	
f.) Port Busy	
4) Physical Port Ø Radial Cable Disk Interface	2
5) Physical Port 1 Radial Cable Disk Interface	2
6) Physical Port 2 Radial Cable Disk Interface	2
7) Physical Port 3 Radial Cable Disk Interface	2
8) Sector Coincidence Detector	3
9) Cylinder Address Register	4
10) Maximum Cylinder Detect Logic	4
11) Header Control Logic	4
12) Port Ø-3 Sector Count Logic	5
13) Sector Count Multiplexing Logic	5
14) Sector Interleaving Logic	5
15) Data and Clock Multiplexing Logic	5

XYLOGIC OEM Components Group, Inc.

42 Third Avenue
Burlington, Massachusetts 01803
617-272-8140

6.0 Documentation Standards and Conventions

6.1.0 General

Standard documentation standards are utilized on all of the logic drawings provided in this manual in order to convey as much information as possible to the user.

The following sections of this document are devoted to defining the standards used.

6.2.0 Backplane Signal Identification Symbol

6.2.1 All logic signals coming from or going from a printed circuit board to the backplane assembly via any of edge connectors are identified on the logic drawings with diamond symbol (<>).

6.2.2 The connector and pin number by which any such signal is to be interfaced to the backplane assembly is contained above the diamond symbol enclosed in square brackets.

Example: ——◇ [DS1] ←Connector and pin No.

6.3.0 Ribbon Cable Connector Identification Symbol

6.3.1 Ribbon Cable Connectors, if used, are normally contained on the top of a printed circuit board assembly.

6.3.2 If more than one such connector is used on a printed circuit board, the rightmost connector (when viewing the board from the component side) is labeled J1. Subsequent connectors moving to the left are labeled J2, J3, etc.

6.3.3 Pine 1 of any ribbon cable connector is located on the left side of the connector when viewed from the component side of the board.

6.3.4 Any signal going to or coming from a ribbon cable connector is identified by the symbols >>, >, or >< in conjunction with the connector number, pin number, and signal name.

An example of a signal going to a ribbon cable connector is as follows:

>> J3-34 READH

6.3.4.1 If a ribbon cable edge connector is the source of a signal, the connector symbol, connector number, pin number and signal name will appear to the left of the point where it is used.

6.3.4.2 If a ribbon cable edge connector is a load for a signal generated on a board, the connector symbol, connector number, pin number, and signal name will appear to the right of the source for the signal.

6.4.0 Signal Name Conventions

6.4.1 Signal Active State Designation

The active state of any named signal appearing on any supplied logic drawings may be readily determined by examining the signal name associated with the signal.

The last character of the signal indicates the active state of the signal per following:

6.4.1.1 Case 1 - Signal Source is Non Memory Logic Element

If the source of a signal is a gating logic element the last character of the signal name will be an H or L depending on whether the signal is active when in the logic 1 state or logic Ø state respectively.

6.4.1.2 Case 2 - Signal Source is Memory Logic Element

If the source of a signal is a memory element such as a flip flop, the last character of the signal name will be (1) or (Ø) depending on whether the signal is active when the memory element is set or reset respectively.

6.4.2 Signal Name Philosophy

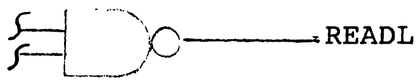
6.4.2.1 A signal will normally be named if it is used on more than one sheet of logic drawings, or if it is used so many times on one sheet such that it cannot be interconnected without "cluttering up" the drawing.

6.4.2.2 Signal names, when used, will normally be limited to 7 characters or less, and will reflect as close as possible, the operation or function of the associated signal.

6.5.0 Signal Sheet Cross Referencing

Signals appearing on the supplied logic drawings fit into three categories not previously discussed. By noting the conventions used in signal cross referencing, the user may quickly determine whether a signal has additional loads on the same or other sheets of the logic drawings.

6. 5.1 If a signal is used elsewhere on the same sheet, the signal name is appended to the end of a straight line at the signal source. An example of this is:



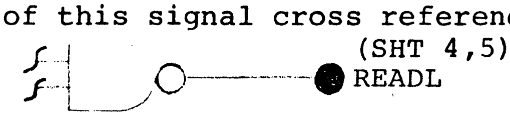
6. 5.2 If a signal is not used anywhere else on the current logic sheet, but is used on other sheets, a circle is appended to the end of a line at the right of signal source, and all other sheets where the signal is used are identified in parenthesis immediately above the circle.

An example of this signal cross reference is:

READL is not used anywhere else on sheet, but is used on sheets 4 & 5.

6. 5.3 If a signal is used elsewhere on the same sheet, and also used on other sheets, a solid circle is appended to the end of a line at the right of the signal source, and all other sheets where the signal is used are identified in parenthesis immediately above the solid circle.

An example of this signal cross reference is:



READL is used elsewhere on logic sheet as well as on sheets 4 & 5.

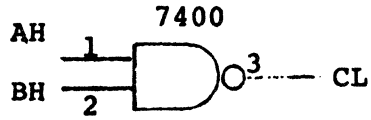
6.6.0 Logic Symbology

6.6.1 Logic symbols used in all supplied documentation are industry standard and are used in a manner conducive to making the logic prints easy to understand.

6.6.2 AND/OR Interpretation

Logic elements, particularly gating elements may be drawn as either an "AND" or as an "OR" function. In all supplied documentation, a gating element will be drawn as either an "AND" or "OR" function as appropriate, to best convey the logical function being implemented. Hence a standard 7400 NAND Gate may be drawn as:

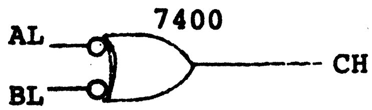
A.)



$CL = \overline{AH \cdot BH}$

or

B.)



CH = AL + BL

L = Active State = Logic Ø

H = Active State = Logic 1

6.6.3 MSI, LSI Symbology

Medium scale and large scale integrated circuit devices used are symbolically shown as a rectangular box with the suppliers functional notation inside the box next to the associated circuit pin.

6.6.4 Manufacturer's Part Number Listing

The common manufacturer's part number is located prominently inside or outside any logical device used to facilitate cross referencing and user understanding of any questionable device function.

6.7.0 Printed Circuit Board Component Orientation

Integrated circuit components for most boards supplied are organized on a grid basis.

6.7.1 The component grid supplied consists of both rows and columns.

6.7.2 A component is assigned a grid address based on the row column coordinates associated with its location per the following:

6.7.2.1 Row Coordinate

The integrated circuit components are normally organized horizontally into 10 rows on a hex sized printed circuit board. These rows are assigned A,B,C,D,E,F,G,H,J,& K with the "A" row being just above and covering the six backplane edge connectors.

6.7.2.2 Column Coordinate

The integrated circuit components are normally organized vertically into 12 columns on a hex sized printed circuit board identified as 1-12. The columns are assigned consectutively right to left with column 1 being at the right hand side of the board when viewing the component side of the board with the board vertical and resting on the backplane edge connectors.

6.7.2.3 Component Location Designation

The location designation of any component appearing on all documentation is determined by the row coordinate followed by the column designation.

Hence, the location designation of the component appearing on the extreme lower right hand corner of printed circuit board is "A1" while the location designation of the component at the extreme top left hand corner of the printed circuit board is "K12".

7.0 Required Test Equipment

The following equipment is required to test and maintain the Phoenix 211 Disk Controller.

Item	Quantity	Equipment Description
1	1	PDP11 Computer with a mimimum of 8192 words of memory
2	1	DEC W984 Quad Extended Height Extender Board
3	1	DEC W987 Double Extended Height Extender Board
4	3	Pomona 3916 Dip Clip - 16 pin dual inline integrated circuit test clip or equivalent
5	1	Tektronix 453 Dual Channel Oscilloscope with external trigger, X10 Probes, or equivalent
6	1	WTCPN Weller Controller Temperature Soldering Iron or equivalent
7	1	"Micro-Shear" 175 Flush Safety Cutter or equivalent
8	1	Xcelite 41CG Needle Nose Pliers or equivalent
9	1	Precista T-2C Solder Removal Tool or equivalent
10	A/R	Alpha 815 Flux or equivalent
11	A/R	Service Chemical D-Sol F-13 Defluxer or equivalent
12	1	Solder Removal Co. Insertic #880 Insertion Tool or equivalent
13	A/R	Solder, 63/37 Rosin Core

8.0 RECOMMENDED SPARES

8.1.0 PHOENIX 211 SPARES LIST

	PRICE
Interface	$1000
Board 1 Standard	1150
Board 1 with Command Queue	2000
128 Word Buffer Board 2	1000
512 Word Buffer Board 2	1300
Board 3	1000
CDC Board 4	1200
Calcomp Board 4	1200
Cables & Terminator	500

8.2.0 CDC 9762 DISK DRIVE SPARES LIST

Field Exerciser for Disk Unit	3200
First Level Spares for CDC 9762 1 head, 1 set of boards, tools	3500
Alignment Pack	1750

9.0 MBTF Prediction Calculations

9.1 Reference Used

Reliability calculations were made using the MIL-HDBK 217B, Parts Count Prediction Method.

9.2 Reliability Prediction Formula

The formula used to predict reliability MBTF is:

$$\lambda Equip = \sum_{i=1}^{i=n} Ni \; (\lambda_G \; ¶_Q) \; i$$

Where: $\lambda Equip$ = Total rate of failure ($/10^6$ hr)

λ_G = Generic failure rate for ith part

$¶_Q$ = Quality factor for ith part

Ni = Quantity of ith part

n = Number of catagories

NOTE: Only Failures which cause non-operation were included.

9.3 Disk Controller Configuration

The Phoenix 211 Disk Controller configuration used in the reliability prediction calculations was as follows:

- Phoenix 211 Interface Board
- Phoenix 200 Board 1
- Phoenix 200 Board 2
- Phoenix 200 Board 3
- Phoenix 200 Board 4
- Phoenix 211 Autoswitch Board

9.4 Calculation Summary

For the 211, with interface and autoswitch, the calculations are as follows:

9.4.1	Component Failures	$F/10^6$ hours
9.4.2	Integrated Circuits	88.98
9.4.3	Capacitors	12.48
9.4.4	Resistors	.85
9.4.5	Connectors, edge	3.13
9.4.6	Connectors, right angle	.625
9.4.7	Other	.509
	Total:	106.074 Failures/10^6 hours

9.5 Predicted MBTF

The net result is a predicted MTBF of 9500 hours.

10.0 Data Flow Sequence

10.1.0 Normal Disk Write Sequence

10.1.1 When write operation is initiated data is transferred a word at a time via DMA through 2:1 Multiplexer into Latch register preceding FIFO memories.

10.1.2 All odd data words are transferred from the latch register into FIFO Memory #1.

10.1.3 All even data words are transferred from the latch register into FIFO Memory #2.

Such DMA transfers will continue until both FIFO memories are full or until the DMA Word Counter overflows.

10.1.4 When disk controller is in synchronism with the disk drive and the beginning of the sector data field is reached, data is transferred in parallel from the two FIFO memories to one of the two shift registers.

10.1.5 Data is then shifted serially from one of the shift registers to the disk.

10.1.6 Data Words are alternately loaded into Shift Register #1 and then Shift Register #2.

10.1.7 When the contents of one shift register are being transferred to the disk, the other shift register is loaded from the FIFO memories with the next word to be transferred.

10.1.8 The Disk Word Count Register is incremented once for each word transferred to the disk.

10.2.0 Normal Disk Read Sequence

10.2.1 Data is received serially from the disk drive and into one of the two serial shift registers.

10.2.2 When a 16 bit data word has been accumulated, the serial data bit stream is diverted to the other shift register and the data word collected is transferred in parallel to one of the two FIFO Memories.

The Disk Word Count Register is incremented once for each word that is received from the disk.

10.2.3 Normally all odd data words are formed in Shift Register #1 and stored temporarily in FIFO Memory #1 while all even data words are formed in Shift Register #2 and stored temporarily in FIFO Memory #2.

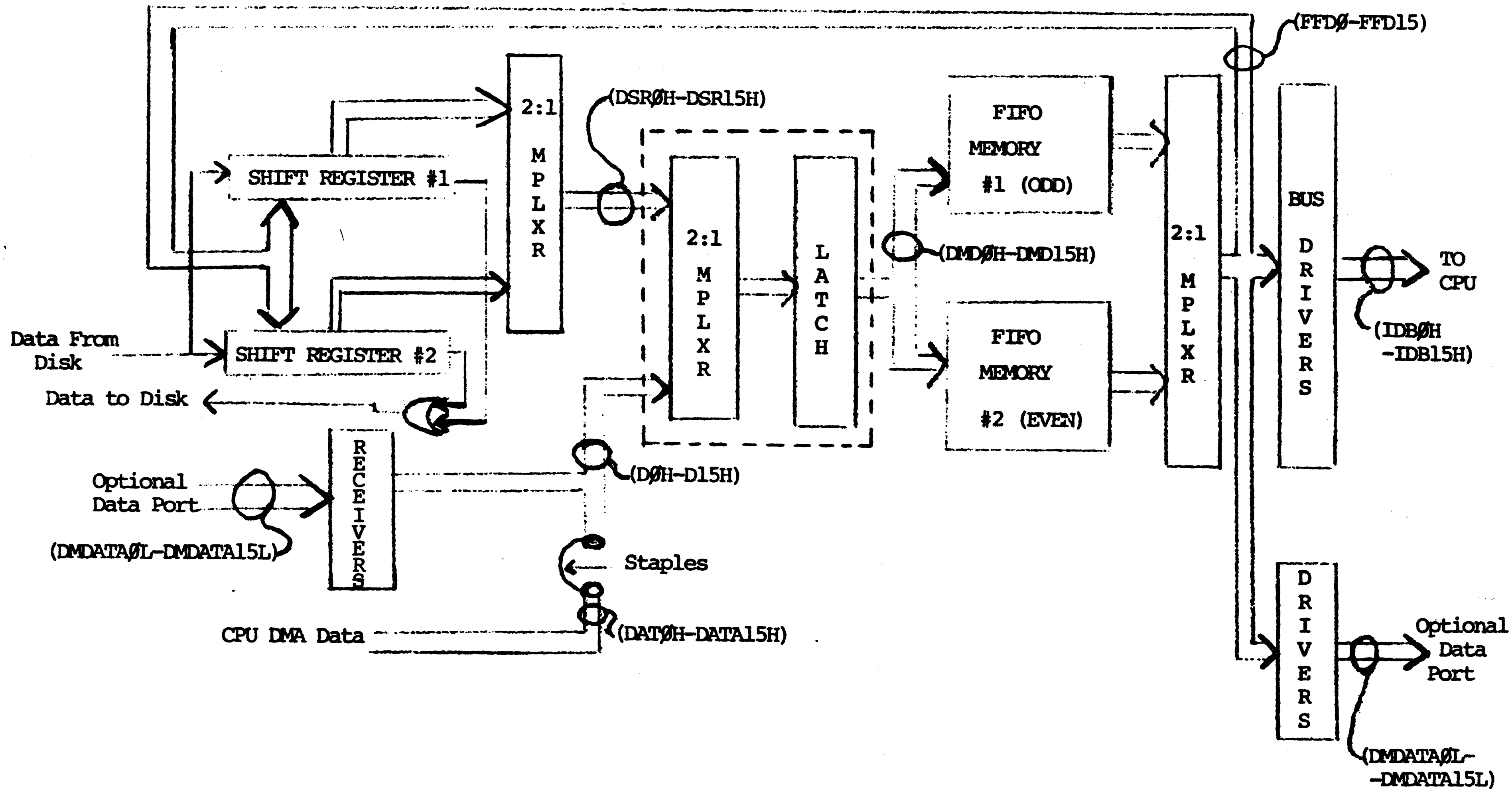
10.2.4 When the presence of data is detected in either FIFO, DMA requests are initiated to the host computer.

Data Words are extracted alternately from the two FIFO memories until either the DMA word Count Register overflows or the FIFO memories are empty of all disk data.

PHOENIX 211 DISK CONTROLLER

Data Paths Block Diagram

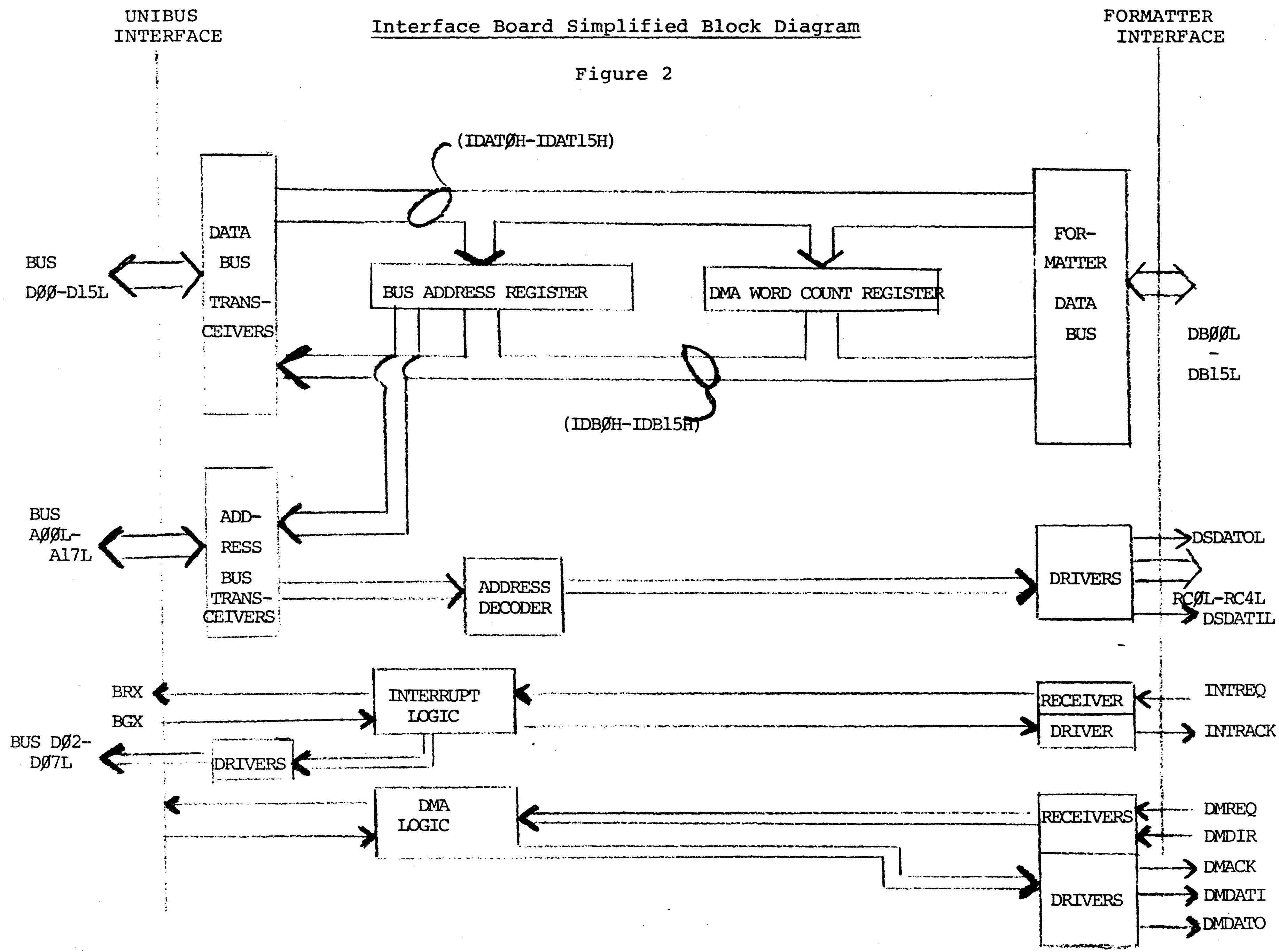
Figure 1



Phoenix 211 Disk Controller

Interface Board Simplified Block Diagram

Figure 2



11.0 Hardware Documentation

11.1 The following sections of this manual contain all of the hardware documentation for the Phoenix 211 Disk Controller. For each major Phoenix 211 Sub Assembly the following documentation is provided:

- Parts List
- Assembly Drawing
- Wiring Lists*
- Logic Drawings*

* Where appropriate.

11.2 The first section contains an overall drawing Directory or "Phoenix 211 Family Tree", for the Phoenix 211 Disk Controller.

11.3 The major Phoenix 211 Subassemblies organized in order of presented in this manual are as follows:

1. Phoenix 211 Interface Board
2. Phoenix 200 Board 1
3. Phoenix 200 Board 2
4. Phoenix 200 Board 3
5. Phoenix 200 Board 4
6. Phoenix 200 Systems Unit
7. Phoenix 211 Interconnect Cable
8. Disk Drive A Cable
9. Disk Drive B Cable
10. Phoenix 211 Autoswitch Board

ECC LOGIC DESCRIPTION

I Error Correction Code Logic Block Diagram Discussion

Figure 1 is a block diagram of the circuits used to implement error correction processing within the Formatter. The circuits on this diagram can be looked at as performing three separate functions, all related to error correction code processing.

1. Generation of a 32-bit ECC redundancy code during write operations. This information is written serially on the disk immediately following the 256 word data field (Figure 2).

2. Checkout of the ECC redundancy field during read operations to detect the possible presence of a data check error. The check is made by ANDing together the low order 21 bits of the ECC register and looking for a zero status.

3. Entry into the error correction process on detection of a data check error indicated by the fact that the low order 21 bits of the ECC register are not zeros. This has two immediate effects with respect to the Phoenix 211 Interface lines:

   . The DONE (INTRL) signal normally inserted at the end of a transfer is inhibited or delayed until completing error correction processing.

   . The CRC error (CRCERR) bit in the Error Register is set causing the setting of the composite error bit in the control and status register.

Following the error correction process, the Formatter detects one of two conditions:

1. Detects the 11-bit error burst and its position (physical location within the data field). The burst pattern and position information are supplied to the software operating system via the appropriate registers.

2. Determines that the error is non-correctable and sets the error correction hard error bit in the ECC Bit Location Count Register.

II Generating and Writing the ECC Field (Write Operation) - Generating and writing the ECC redundancy code in the ECC field (Figure 1) occurs when the Formatter executes either of the write commands. The code is formed within the ECC shift register during the time that the data field is written onto the disk. Each bit shifted from the shift register (for transfer to the disk) is also entered into the ECC shift register. This is accomplished as follows:

a. Signal (WGH) WRITE GATE H (applied to the ECC register feedback control logic) is asserted since a write operation is in progress.

b. At the start of the data field, signal (WCCGH) WRITE CHECK CHARACTER GATE is negated. This in turn causes A7-6 to switch high and enable the ECC register feedback gating.

ONEs and ZEROs coming from the shift register (signal DATAOUTL) now enter the ECC shift register throughout the data field transfer. In this way, the ECC redundancy code (to be written onto the disk following the data field) is formed. At the end of the data field, signal WCCGH asserts because it is now time to write the ECC field associated with the data just written. Signal WCCGH inhibits the ECC Register feed back at A6-2, the ECC Register is now shifted to the NRZ data latch (on Board 2 SH2 of logics) where the serial pulse train of write data is generated (DATAOUT)

III Checkout of the ECC Redundancy Code (Read Operation) - During read operations, the ECC redundancy code is formed again by applying each data field bit read from the disk to the ECC shift register. When reading from the disk, the enabling of the ECC register feedback gating is effected in the following way:

a. With a read operation undertaken, signal (RDGATEH) READ GATE (applied to the ECC register feedback control logic) is asserted.

b. At the end of the Preamble #2, signal DATAH asserts to define the start of the data field.

c. Signal WCCGH is negated to enable the ECC register feedback gating and thereby allows each bit coming from the disk (READDATAH) to enter the ECC register.

Applying each bit read from the disk to the ECC register (with the feedback loop enabled) reconstructs the same code that was attached to the ECC field when the data was written.

When the end of the data field is reached, signal DATAH negates; however, signal WCCGH is negated to maintain signal A7-6 at the asserted level allowing the ECC field bits coming from the disk (READDATAH) to enter the ECC register while the feedback loops are still enabled.

When the ECC redundancy code read from the disk and clocked into the ECC register matches that developed (in the ECC register) at the close of reading the ECC field (signal RCCGATE - read check character gate) then the 21 low order bits of the ECC register all contain zeros. This means that the data field has checked out OK. As a result, signal E1-4 (zero's detect) asserts and sector processing is terminated normally by raising the EOSH signal.

NOTE

The 21 low order bits of the ECC pattern register are ANDed together. When all bits are zeros, signal E1-4 (zero detect) asserts.

When the 21 low order bits of the ECC register fail to contain all zeros, it means that there is an error in the data read from the disk. The actual location and the nature (soft or hard) of that error is not known at this time. The Formatter now enters the error correction process (provided it is not inhibited from doing so) as described in the subsequent paragraphs.

IV Error Correction Processing - When signal E1-4 (zero detect) fails to assert at the end of the ECC envelope, the data check error detect logic issues three outputs that are used as follows:

1. Signal ECCERROR (1) asserts to set bit 8 (data check error) in the Error Register.
2. Signal CRCERR (1) asserts. This is used in the Busy/Done logic to inhibit generation of the DONE (Formatter Ready) signal.

NOTE

If the Error Correction Inhibit (ECI) bit in the ECC pattern register is set, INTRL signal is sent to the Interface. The error correction process in this case, is inhibited

3. CRCERR (1) is applied to enable the ECC correction enable logic. The latter circuit now asserts A9-5 (correction enable) provided that the error correction enable signal (ECCENB (1) is asserted.

   Assertion of A9-5 together with CRCERR (1) begins the error correction process and has two immediate effects:

1. It negates signal A4-8 via the ECC register feedback control logic to force zero's as data. (This is essential for the error correction process).

2. Enables clocking of the leading zero's word counter to maintain a count of each bit serially shifted within the ECC register.

NOTE

> The polynomial used for the error correction process is capable of accommodating a field much larger than the 256 word data field of each sector. For this reason, the Formatter goes through a process of shifting leading zeros within the ECC register and feedback paths. The zero's word counter maintains a count of the leading zeros. This is done so the time at which error correction code processing enters the data field can be decoded and the task of detecting the 11-bit error burst can begin.

The number of leading zeros shifted within the pattern register depends on the Data Field format that is being used. The leading zero value is shown below: for a data field of 256, 16 bit words.

	LEADING ZEROS	DATA FIELD	ECC FIELD
16-bit	38,859		

When the applicable number of leading zeros has been counted, the data field entry detect logic asserts signal B5-9 (Leading zero's counter overflow.) This acts as an enabling signal to the position register shift lock enable gating. The position register keeps a count of data field bits shifted (in the ECC register) until such time as the 11-bit error burst is located. A second use of signal B5-9 is its application to the error burst detect clock circuits. Here it acts as an enabling signal (i.e., in an anticipation of de-detecting the 11-bit error burst) because the shifting of bits is now within the data field. Conditions are now set up for detecting the presence of the error burst in the 11 high order bits of the ECC register. That portion of the ECC register is also called the "ECC pattern" register. Design is such that the location of the 11 bit error burst is detected as being identified when the 21 low order bits all contain zeros (i.e., as a result of the continuous shifting/feedback process). An all zero condition is sampled in the zero detect gating and asserts signal zero (Ø). On application to the ECC correction enable logic, signal Zero (Ø) produces the following results:

a. Inhibits the position register shift clock enable gating to stop the count of the position register at that point in the data field (or ECC field). The count stored in the "ECC position" register identifies the physical location within the data field, of the first bit of the 11-bit error burst.

b. Causes the error burst detect logic to assert signal ECCABOL. This, in turn, has a double effect:

1. Inhibits the ECC register shift clock enable gating so that no further shifting of bits occurs in the ECC register. This is necessary because the 11 high order bits ("ECC pattern" containing the error burst) must now be sent to the central processor.

2. Forces the DONE generation logic to send an INTR signal to the Interface. This is done to indicate that error correction processing is complete and that the CPU may now take the contents of the ECC position and pattern registers.

c. Inhibits further counting by the leading zero's word counter.

This completes Formatter error correction processing for those cases where the location of the error burst is detected within the data field and the error is correctable. If the error correction logic fails to detect an error burst within the data (or ECC) field, the Formatter notifies the CPU of a "hard error" condition. This condition is indicated to the logic by the fact that the Bit Location counter has exceeded the maximum size of the entire ECC code length without having found an all zeros condition in the low order 21 flip-flops of the ECC register.

NOTE

> By definition, "hard error" means that the Formatter failed to detect a correctable error burst within the data or ECC fields.

The error correction logic keeps a count of the bits being shifted in the pattern register after the shifting process enters the data field. Consequently, when the Bit Location counter reaches a value of 4128_{10} bits* (following entry into the data field) it means that no error burst has been detected and the uncorrectable error bit must be set in the bit location count register. This occurs when the data field entry/hard error detect logic determines that the count from the Bit Location counter has gone past the ECC field and asserts signal UNCORER (1).