

# USER GUIDE

---

# DSD 440

**Data Systems Design**

**DSD 440**  
**FLEXIBLE DISK SYSTEM**

**USER'S MANUAL**

**Data Systems Design, Inc.**  
**2241 Lundy Avenue**  
**San Jose, CA 95131**  
**(408) 946-5800**  
**TWX 910-338-0249**

**Copyright ©1980**

**All rights reserved. No part of this manual  
may be reproduced in any form or by any  
means without prior written permission of  
Data Systems Design.**

**Data Systems Part No: 040009-01**

**Printed in USA**  
**7/81**

## PUBLICATION NOTICE

Changes in the text of this manual after its initial release are supplied in Manual Change Notices or in complete revisions to the manual. The history of any changes to this edition of the manual is listed under "PUBLICATION HISTORY".

Any changed pages are identified by the change number at the bottom of the page and a vertical line in the outer margin of the page. The line indicates the section of text that has changed.

### PUBLICATION HISTORY

<u>Revision Level</u>	<u>Date</u>
First Edition	September 1979
Change 1	June 1980
Second Edition	March 1981
Third Edition	July 1981

### WARRANTY STATEMENT

Data Systems Design's products are warranted against defects in materials and workmanship. For DSD products sold in the U.S.A., this warranty applies for ninety (90) days from date of shipment.\* DSD will, at its option, repair or replace either equipment or components which prove to be defective during the warranty period. This warranty includes labor, parts and surface travel costs of system modules or components. Freight charges for other than surface travel or for complete systems returned for repair are not included in this warranty. Equipment returned to DSD for repair must be shipped freight prepaid and accompanied by a Material Return Authorization number issued by DSD Customer Service. Repairs necessitated by shipping damage, misuse of the equipment, or by hardware, software, or interfacing not provided by DSD are not covered by this warranty.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. DSD SHALL NOT BE LIABLE FOR CONSEQUENTIAL DAMAGES.

For superior reliability and long service life, DSD recommends that you use DYSAN or MAXELL floppy diskettes.

\* For products sold outside the U.S.A., contact your local DSD distributor for warranty terms.

NOTE: ALL ORDERS SHOULD BE PLACED THROUGH THE NEAREST REGIONAL SALES OFFICE.

EASTERN REGION SALES  
51 Morgan Drive  
Norwood, MA 02062

TEL: (617) 769-7620  
TWX: 710-336-0120

WESTERN REGION SALES  
2560 Mission College Boulevard  
Suite 108  
Santa Clara, CA 95051

TEL: (408) 727-3163

## PREFACE

This manual describes the features, specifications, and register usage of the DSD 440 Flexible Disk System.

Instructions for DSD 440 installation, operation, hyperdiagnostic routines, troubleshooting and preventive maintenance are also included in this document.

The material in this manual is subject to change without notice. The manufacturer assumes no responsibility for any errors which may appear in this manual.

Please note that DEC, PDP, LSI-11, RT-11, OS/8, UNIBUS and OMNIBUS are registered trademarks of the DIGITAL EQUIPMENT CORPORATION.

## SAFETY

Operating and maintenance personnel must at all times observe sound safety practices. Do not replace components, or attempt repairs to this equipment with the power turned on. Under certain conditions, dangerous potentials may exist when the power switch is in the off position, due to charges retained by capacitors. To avoid injury, always remove power before attempting repair procedures.

Data Systems Design, Inc. will accept no responsibility or liability for injury or damage sustained as a result of operation or maintenance of this equipment with the covers removed and power applied.

## WARNING

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



## TABLE OF CONTENTS

	<u>Page</u>
<b>CHAPTER 1: <u>GENERAL INFORMATION</u></b>	1-1
Introduction	1-1
DSD 440 Features and Benefits	1-5
Components	1-7
Controller/Formatter Module	1-7
Disk Drives	1-9
Computer Interface Card	1-10
Power Supply	1-12
Interconnecting Cable	1-13
<b>CHAPTER 2: <u>INSTALLATION</u></b>	2-1
Unpacking and Inspection	2-1
Power Requirements	2-1
Operational Environment	2-2
Installing the DSD 440 Chassis	2-3
Installing the DSD 440 Interface Module and Cable	2-6
LSI-11 Based Systems	2-6
PDP-11 Based Systems	2-11
PDP-8 Based Systems	2-15
Configuring the DSD 440 for RX01 - RX02 Mode Operation	2-16
DSD 440 Chassis Connections	2-17
<b>CHAPTER 3: <u>OPERATION</u></b>	3-1
Applying AC Power	3-1
Inserting a Diskette	3-2
Initialization Response Check	3-3

## TABLE OF CONTENTS (continued)

	<u>Page</u>
LSI-11 and PDP-11 System Bootstrapping (Initial Program Loader)	3-4
PDP-8 System Bootstrapping	3-5
Bootstrapping LSI-11 and PDP-11 Systems with Non-standard Device Adresses	3-5
Initializing LSI-11 and PDP-11 Systems with Read Only Memory (ROM) Installed as Main Memory	3-6
Bootstrapping Without System Test Functions	3-7
Acceptance Testing	3-7
<b>CHAPTER 4: <u>SOFTWARE</u></b>	<b>4-1</b>
Single Density Diskettes in a Double Density RX02-Compatible System	4-1
Formatting Double Density Diskettes	4-1
Generating an RX02 Compatible RT-11 System Diskette	4-1
Setting Media Density on the PDP-8	4-4
Generating a Double Density RX02 Mode OS/8 System Disk	4-5
Programming Interface	4-9
DEC 11 Family	4-10
Mode 1 (RX01 Compatible) Operation	4-11
Peripheral Device Registers	4-11
Mode 1 Protocols	4-16
Diskette Formatting	4-22
Typical Sequences of Operation	4-22
Interrupts	4-23

## TABLE OF CONTENTS (continued)

	<u>Page</u>
Mode 2 (RX02 Compatible) Operation	4-27
Peripheral Device Registers	4-27
Mode 2 Protocols	4-33
Diskette Format Programming	4-38
Typical Sequence of Operations	4-39
Interrupts	4-43
DEC PDP-8 Programming Interface	4-43
Instruction Set	4-45
Register Descriptions	4-48
Function Code Descriptions	4-52
Diskette Formatting in Selected Density	4-57
CHAPTER 5: <u>TESTING</u>	5-1
DSD 440 Self-Tests	5-1
Automatic Self-Testing	5-1
User-Selectable Self-Tests	5-2
Test Selection Indicator Lights and DIP Switch Location	5-2
Reading the Indicator Lights	5-3
Normal Mode	5-6
DIP-Switch Settings for Normal Operation	5-6
Indicator Light (LED) Definitions	5-8
During Normal Operation	
"HYPERDIAGNOSTIC" Mode	
DIP-Switch Settings for "HYPERDIAGNOSTIC" Operation	5-10

## TABLE OF CONTENTS (continued)

	<u>Page</u>
Indicator Light (LED) Definitions During "HYPERDIAGNOSTIC" Operation	5-11
Test Strategy Using "HYPERDIAGNOSTICS"	5-12
Individual "Hyperdiagnostic" Tests	5-14
Computer Resident Diagnostics	5-19
FRD440 Diagnostic Program for LSI-11 and PDP-11 Systems	5-19
Program Loading and Monitor Protocol	5-19
FRD440 Program Functional Modes of Operation	5-21
Mode Setting Commands	5-26
Format Initialization Commands	5-28
Dump and Copy Utility Commands	5-29
FRD440 Program Status and Error Displays	5-30
DSD 440-8 Floppy Disk Diagnostic (VEP 210)	5-34
Interrupt Restart Commands	5-37
Effects of Switch Register Settings on the Diagnostic Tests	5-38
Interpretation of VEP210 Error Messages	5-38

## APPENDICES

### APPENDIX A: CONTROLLER AND INTERFACE MODULE SCHEMATICS

Controller Board Assembly Drawing	A-1
Controller Board Schematic	A-2
LSI-11 Interface Board Assembly Drawing	A-8
LSI-11 Interface Schematic	A-9
PDP-11 Interface Board Assembly Drawing	A-12

TABLE OF CONTENTS (continued)

	<u>Page</u>
PDP-11 Interface Schematic	A-13
PDP-8 Interface Board Assembly Drawing	A-19
PDP-8 Interface Schematic	A-20
Power Distribution System Schematic	A-22
 <b>APPENDIX B: STANDARD JUMPER POSITIONS ON INTERFACE AND CONTROLLER MODULES</b>	
DSD 440 Controller Module	B-1
LSI-11 Interface Card (RX02 MODE)	B-2
LSI-11 Interface Card (RX01 MODE)	B-3
PDP-11 Interface Card (RX02 MODE)	B-4
PDP-11 Interface Card (RX01 MODE)	B-5
 <b>APPENDIX C: DISK DRIVE MAINTENANCE MANUAL</b>	
	C-1
 <b>APPENDIX D: STANDARD JUMPER POSITIONS FOR DRIVE ELECTRONICS</b>	
Drive Jumper Configuration for DSD 440	D-1
Drive 0 Configuration Drawings	D-2
Drive 1 Configuration Drawings	D-4
Single Drive Configuration Drawings	D-6
 <b>APPENDIX E: POWER SUPPLY MANUAL</b>	
Specifications	E-1
Parts List	E-2
Trouble Shooting Guide	E-4
Schematic	E-6

TABLE OF CONTENTS (continued)

	<u>Page</u>
<b>APPENDIX F: FLEXIBLE DISKETTE DESCRIPTION AND MAINTENANCE</b>	
Diskette Description	F-1
Industry Recording and Media Standards	F-2
Diskette Care Guidelines	F-3
<b>APPENDIX G: LSI-11 AND PDP-11 BOOTSTRAP PROM PROGRAM</b>	G-1
<b>APPENDIX H: PDP-8 BOOTSTRAP PROGRAMS</b>	
Part I - PDP-8/A Bootstrap PROM Procedure	H-1
PDP-8/A Bootstrap Program	H-2
Part II - Minimal PDP-8 Bootstrap Programs	H-3
<b>APPENDIX I: CONTROLLER ERROR CODE DESCRIPTIONS                   AND CAUSES</b>	I-1
<b>APPENDIX J: FLPEXR DIAGNOSTIC PROGRAM USER'S MANUAL</b>	J-1

## TABLES

<u>Table No.</u>	<u>Description</u>	<u>Page</u>
1-1	DSD 440 Specifications and Environmental Requirements	1-2
2-1	Temperature Range for Diskettes	2-3
2-2	Rack Installation Hardware	2-4
2-3	Register and Boot PROM Address	2-7
4-1	Mode 1 Command and Status Register	4-12
4-2	Error Register Codes (RXER)	4-21
4-3	Fill/Empty RX01 Sector Buffer Example	4-24
4-4	Read/Write/Write Deleted Data RX01 Sector Example	4-26
4-5	Fill/Empty RX01 Sector Buffer Example	4-40
4-6	Read/Write RX02 Sector Example	4-42
4-7	Definitive Error Register Codes for the RXER	4-44
4-8	Input/Output Transfer Instructions	4-45
4-9	Command Register Function Codes	4-48
4-10	Operating Mode, Word Length, XDR Instructions Relationship	4-53
5-1	Error Class Code in NORMAL Mode (Both Green LEDs On)	5-7
5-2	Error Register Codes	5-9
5-3	"HYPERDIAGNOSTIC" Error Code Interpretation	5-11
5-4	Interpretation of LEDs 5, 6 and 8 During Execution of "Hyperdiagnostics" While Run LED is On	5-13
5-5	Error Register Codes (RX2ER)	5-33
 Appendices		
H-1	Setting the Bootstrap Address	H-1

## FIGURES

<u>Figure No.</u>	<u>Description</u>	<u>Page</u>
1-1	The DSD 440 System	1-1
1-2	Components of the DSD 440 System	1-8
1-3	440 Controller/Formatter Module	1-9
1-4	LSI-11 Computer Interface Card	1-10
1-5	PDP-11 Computer Interface Card	1-11
1-6	PDP-8 Computer Interface Card	1-11
1-7	DSD 440 Power Supply	1-12
1-8	Interconnecting Cable	1-13
2-1	AC Power Assembly	2-2
2-2	Installing Chassis Mounts	2-5
2-3	Securing the DSD 440 in a Rack	2-5
2-4	Front View of Chassis With Front Panel Removed to Show Securing Holes	2-6
2-5	LSI-11 Computer Interface Card	2-7
2-6	LSI-11 Computer Interface Card Diagram	2-8
2-7	Option Priority in LSI-11 Backplanes	2-10
2-8	PDP-11 Computer Interface Card	2-11
2-9	PDP-11 Computer Interface Card Diagram	2-12
2-10	Interrupt Priority Levels	2-13
2-11	Interrupt Priority Levels Diagram	2-14
2-12	PDP-8 Computer Interface Card	2-15
2-13	Controller/Formatter Module DIP-Switch	2-17
2-14	Installing Chassis Slides	2-18
3-1	Back Panel of DSD 440	3-1
3-2	Orientation of Diskette for Insertion	3-2



## FIGURES (continued)

<u>Figure No.</u>	<u>Description</u>	<u>Page</u>
4-1	Register Formats	4-15
4-2	Command and Status Register Format	4-28
4-3	PDP-11 System Error and Status Register Bit Layout	4-32
4-4	PDP-8 Mode 1 Command Register Format	4-45
4-5	PDP-8 Mode 2, 12 Bit Mode, Command Register Format	4-46
4-6	PDP-8 Mode 2, 8 Bit Mode, Command Register Format/Sequence	4-46
4-7	Track Address Register	4-49
4-8	Sector Address Register	4-49
4-9	PDP-8 Data Buffer Register	4-49
4-10	Error/Status Register in Mode 1	4-51
4-11	PDP-8 Error/Status Register in Mode 2	4-52
5-1	Controller Module Indicator Lights and DIP-Switch Orientation	5-3
5-2	Controller DIP-Switch Settings	5-4
5-3	Controller LED Decoding Chart	5-5
5-4	DIP Switch Setting When DSD 440 is Shipped	5-7
 <b>Appendices</b>		
F-1	Single Sided Diskette Properties	F-1
F-2	Write Protect Notch Location	F-2
F-3	Single Density Track Format	F 4
F-4	DEC Double Density Track Format	F-5



## CHAPTER 1

### GENERAL INFORMATION

#### INTRODUCTION

The DSD 440 is a flexible disk data storage system. Data is stored in fixed-length blocks on industry standard, soft-sectored, 8-inch diameter flexible disks. Each flexible disk, or diskette", can store up to 512 kilobytes (8 bits per byte) of data. The average access time to this data is 296 milliseconds and the data transfer rate is 62.5 kilobytes per second. Figure 1-1 is a picture of the complete data storage system. It consists of a rack-mountable controller/drive subsystem, a computer interface module, and an interconnecting cable. Table 1-1 lists the DSD 440 specifications and environmental requirements.

When used with the appropriate interface module, the DSD 440 is completely software, hardware and media compatible with LSI-11, PDP-11, and PDP-8 computers, including those with extended memory. The DSD 440 can be configured as a DEC RX02 for DEC double density or IBM 3740 single density recording, or as a DEC RX01 for compatibility with earlier versions of your operating system.

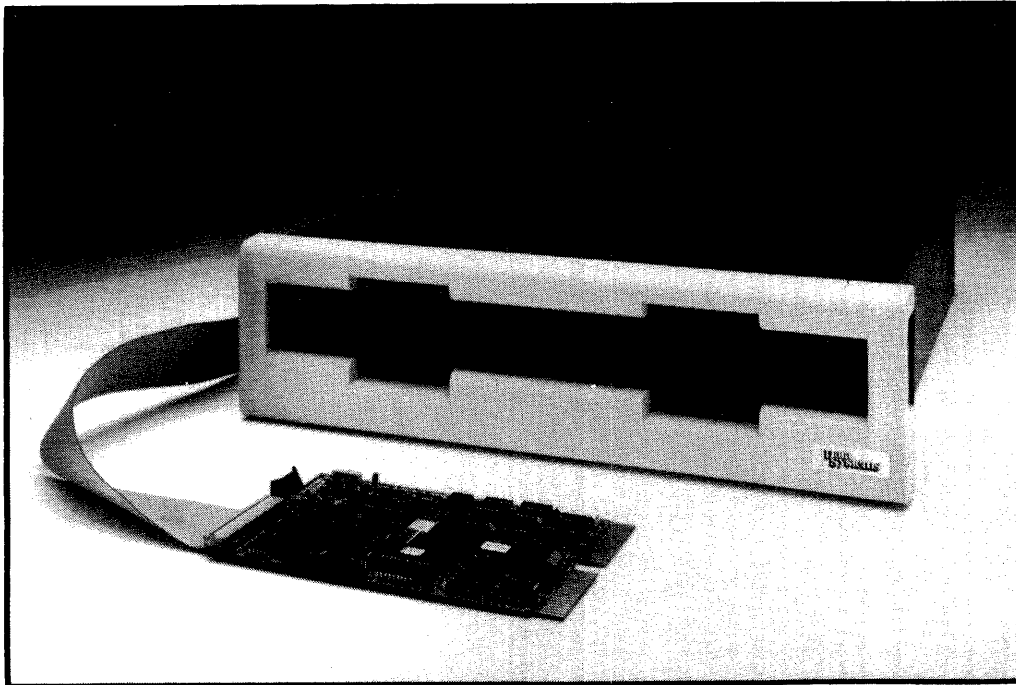


Figure 1-1. The DSD 440 System

Table 1-1. DSD 440 Specifications and Environmental Requirements

Capacity (Formatted)

Double Density

Bytes per surface	512,512
Bytes per sector	256
Sectors per track	26
Tracks per surface	77

Single Density

Bytes per surface	256,256
Bytes per sector	128
Sectors per track	26
Tracks per surface	77

Drives per Chassis	1 or 2
--------------------	--------

Recording Characteristics

Double density format	DEC RX02
Single density format	IBM 3740
Double density recording technique	DEC-Modified MFM
Single density recording technique	Double Frequency
Flux transition density maximum	3200 flux changes per inch
Track density	48 tracks per inch
Track-to-track spacing	.529 mm (.021")
Track width	.3048 mm (.012")

Speeds

Double Density

Diskette to controller bit rate	500 KHz
---------------------------------	---------

Controller to CPU memory transfer rate	24 Microseconds per word or 80 Kilowords per second
--	---

Sustained Throughputs

20 Kilobytes/second within a track (2-way interleave)  
17.6 Kilobytes/second across entire diskette (2-way interleave, 7 sector skew track-to-track)

Drive Data Transfer rate	62.5 Kilobytes/second 500 Kilobits/second
Diskette rotation	360 R.P.M. +/-2%
Head step rate	8 ms (Milliseconds) / track
Head load and settling time	35 ms
Step settling time	8 ms
Average access time	296 ms
Maximum access time	783 ms

## Interface Characteristics

### Interface Module Backplane Requirements

LSI-11 (Q-BUS)	1 half-quad slot
PDP-11 (UNIBUS)	1 quad SPC slot
PDP-8 (OMNIBUS)	1 quad slot

### Interface Module Power Consumption (+5 volts)

	Nominal	Maximum
DSD 440-4432 (LSI-11)	1.44 amp	2.00 amp
DSD 440-4430 (PDP-11)	1.30 amp	1.70 amp
DSD 440-2131 (PDP-8)	1.37 amp	2.35 amp

### Standard Device Addresses or Codes

DSD 440-4432 (LSI-11)	777170 - 777172
DSD 440-4430 (PDP-11)	777170 - 777172
DSD 440-2131 (PDP-8)	6750 - 6757

### Chassis Power Consumption

Controller module (at +5%)	3.40 amp nominal,	3.75 amp maximum
Single drive chassis (max)	1.50 watts idle,	204 watts busy
Dual drive chassis (max)	170 watts idle,	300 watts busy

Selectable input voltages	100 V AC or 120 V AC rms +/-10%
	220 V AC or 240 V AC rms +/-10%

Input frequencies	50 Hz +/-1 Hz
	60 Hz +/-1 Hz

### Fuse ratings

Single drive, 115 V AC	2.5 amp slow-blow
Dual drive, 115 V AC	3.0 amp slow-blow
Single drive, 220 V AC	1.25 amp slow-blow
Dual drive, 220 V AC	2.0 amp slow-blow

### Heat Dissipation (in BTU's per hour)

	Nominal	Maximum
All DSD 440 interface cards	24	40
Controller module	58	65
Single drive chassis		
(idle)	290	512
(busy)	503	791
Dual drive chassis		
(idle)	468	846
(busy)	681	1125

## Environment

U. L. Listing	EDP equipment, U.L. 478 standard
Operating temperatures	
Interface modules	0°C to 50°C (32°F to 122°F)
Controller module	0°C to 50°C (32°F to 122°F)
Chassis	0°C to 40°C (32°F to 104°F)
Diskettes	10°C to 51°C (50°F to 125°F)
Diskette maximum Thermal gradient	15°F per hour
Non-operating temperatures	
Interface modules	-40°C to 66°C (-40°F to 151°F)
Controller module	-40°C to 66°C (-40°F to 151°F)
Chassis	-40°C to 66°C (-40°F to 151°F)
Diskettes	-40°C to 52°C (-40°F to 125°F)
Humidity	
Interface modules, chassis and controller module	10% to 95% (non-condensating)
Diskettes	8% to 80% with a maximum wet bulb temperature of 29.4°C (85°F)
Sizes	
Chassis	5.25"h x 17.6"w x 21.0" d (31.3 cm x 44.7 cm x 53.3 cm)
Shipping carton	30.0"h x 17.6"w x 21.0"d (31.75 cm x 66.2 cm x 76.2 cm)
Controller module	17.1"h x 4.6"w x 1.0"d (43.43 cm x 11.68 cm x 2.54 cm)
Quad interface module	9.0"h x 10.5"w x 0.5"d (22.86 cm x 26.67 cm x 1.27 cm)
Dual height interface module	9.0"h x 5.2"w x 0.5"d (22.89 cm x 13.21 cm x 1.27 cm)
Weight	
Chassis	50 pounds (22.7 kg)
Shipping weight	74 pounds (33.6 kg)
Shock and Vibration	
Operating shock	1G for 10-20 milliseconds
Non-operating shock	15G for 10-20 milliseconds
Vibration	5 - 25 Hz @ .0014" 25 - 55 Hz @ .0007" 55 - 300 Hz @ .3G

## DSD 440 FEATURES AND BENEFITS

The DSD 440 is carefully designed, ruggedly built, and extensively tested to provide you with maximum performance and reliability. The following major features are incorporated into your DSD 440:

- **INTERFACES TO LSI-11, PDP-11 AND PDP-8 COMPUTERS**

With different interfaces, you can use your flexible disk system with different computers.

- **DOUBLE DENSITY RECORDING DOUBLES THE NUMBER OF BYTES PER SECTOR**

The data storage capacity and data transfer rate of your flexible disk system can be doubled in RX02 compatible mode.

- **BOTH SINGLE AND DOUBLE DENSITY MODES OF OPERATION**

You receive increased double density capability while retaining IBM single density and RX01-K media compatibility for interchange of data between systems.

- **DIRECT MEMORY ACCESS (DMA) FOR LSI-11 AND PDP-11 BASED SYSTEMS**

Faster data transfers are possible because blocks of data can be transferred directly between the DSD 440 and the computer's memory. This frees the computer to execute other programs while the data is being transferred at a rate of up to 160 kilobytes per second.

- **SUPPORT OF EXTENDED (18 BIT) CPU MEMORY ADDRESSING FOR THE LSI-11 AND PDP-11**

You can add memory addressing capacity to your LSI-11 or PDP-11 system, and your DSD 440 is able to address it directly.

- **HARDWARE BOOTSTRAP LOADER PROGRAM WITH SYSTEM AND DISK DIAGNOSTICS FOR LSI-11 AND PDP-11 COMPUTERS**

The bootstrap that resides on the interface module loads an RT-11 operating system program with the controller in either the RX01 or the RX02 mode. In addition, it confirms proper communications between the computer memory, the interface, and the disk controller.

- **BUILT-IN DRIVE AND CONTROLLER "HYPERDIAGNOSTICS"**

You can verify correct operation of the stand-alone drive and controller unit by selecting and initiating built-in maintenance routines. A bank of switches on the controller board selects special maintenance programs to be run on the controller microprocessor. Nine LED indicators on the

controller board designate the status of "HYPERDIAGNOSTICS" operation. These routines allow the DSD 440 to exercise and test itself without being connected to a computer.

- **WRITE PROTECTABLE DISKETTES**

Diskettes containing operating system masters, system source listings and test programs need never be exposed to the possibility of accidental overwriting.

- **DISKETTE FORMATTING CAPABILITY**

The DSD 440 allows you to write the format onto single-sided, soft sector diskettes in each of two possible formats: IBM 3740 single density or DEC double density. The physical sector sequence written on the diskette can be selected by the programmer so that hardware sector interleaving is possible. In addition, diskettes with magnetically damaged headers (formatting information) can be reformatted instead of discarded.

- **USES STANDARD SINGLE-SIDED DISKETTES**

Readily available IBM 3740 or RX01-K diskettes can be used for both single and double density recording.

- **OPTIMIZED CONTROLLER DATA TRANSFER ROUTINES**

The microcoded data transfer firmware in the DSD 440 controller increases system throughput when using the RT-11 foreground/background monitor.

- **FULL PHASE LOCK LOOP**

The data separator extracts valid data under widely varying conditions of AC line frequency, track alignment, signal strengths and DC supply voltages.

- **AUTOMATIC HEAD UNLOAD**

Diskette and recording head wear is reduced when there is no operation pending because the head is unloading from the diskette.

- **POWER LOSS DATA PROTECTION**

Power failure and restoration will not cause recording head transitions that overwrite your sensitive data.

- **POWERFUL LSI-11 AND PDP-11 COMPUTER-RESIDENT UTILITIES AND DIAGNOSTICS FOR INTERACTIVE TROUBLESHOOTING**

In addition to the standard DEC diagnostics, special programs including the FRD440 program provide many options for scanning diskettes, copying diskettes, formatting and direct utilization of special diagnostic diskettes.



- **DRIVE UNIT SELF-TESTING ON POWER-UP AND INITIALIZATION**

Turning on the DSD 440 automatically initiates a routine that checks the functions of the controller and drive hardware.

- **FRONT PANEL ACTIVITY INDICATORS**

The front panel LED indicators aid in fast error diagnosis and correction by indicating which drive is being accessed and by determining whether the head is loaded or not. The indicators flash to notify you when a drive error occurs.

- **FILTERS AC LINE POWER AND USES 115 or 230 VOLTS AT 50 OR 60 CYCLES PER SECOND**

You can order the DSD 440 with factory options for operation anywhere in the world.

- **LOW-PROFILE CHASSIS**

The compact, 5.25-inch profile and the clean packaging design of the DSD 440 make it attractive in all working environments including the desk top and office setting.

- **RUGGED, MODULAR ASSEMBLY**

With simple tools, you can quickly swap DSD 440 modules. There are no soldered connections between major components. The cabling is carefully secured and routed to avoid obstruction and minimize vibration stress. Keyed connectors eliminate connection guesswork.

These features make the DSD 440 a powerful, yet easy-to-use data storage unit for your DEC computer system.

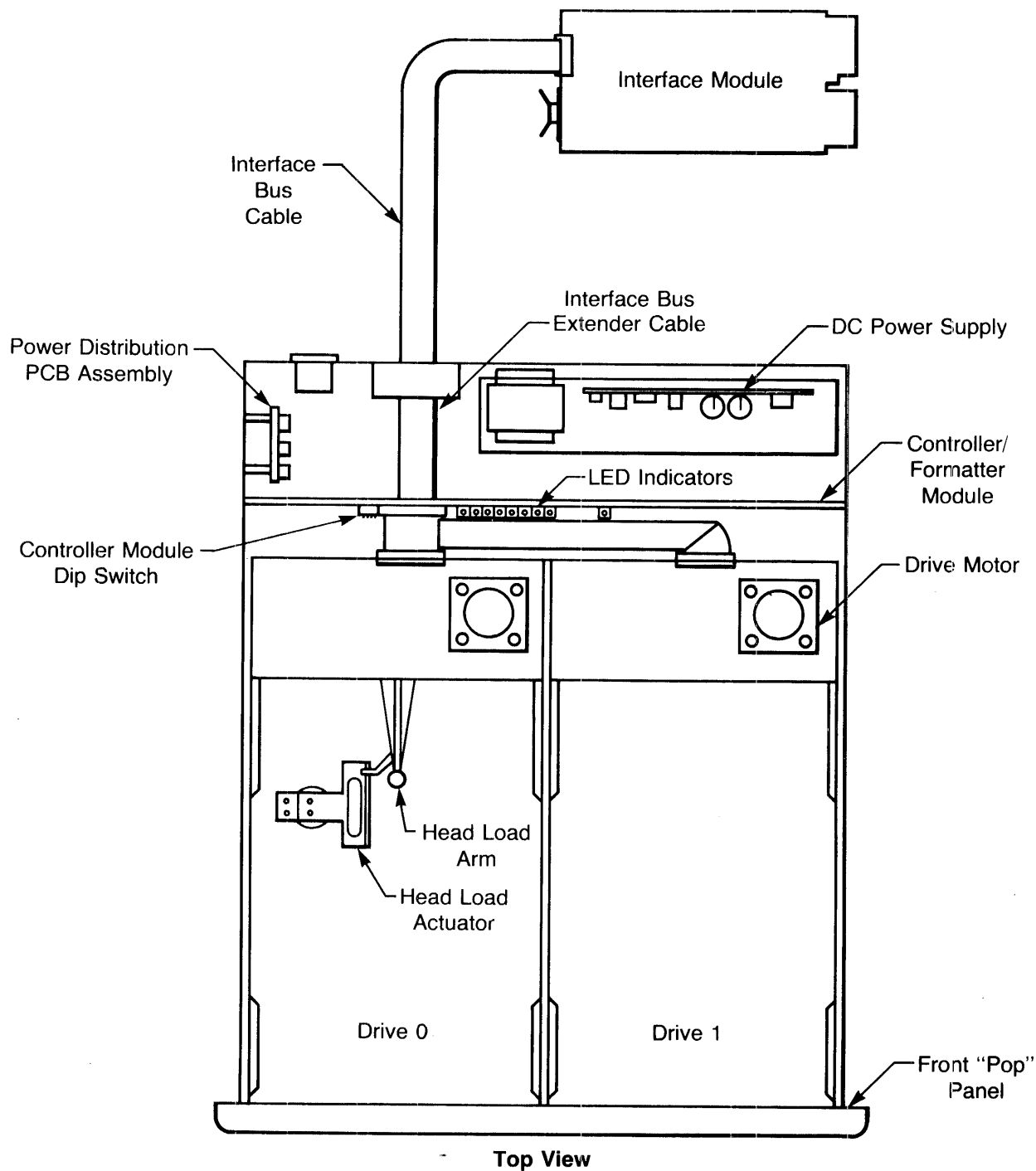
## **COMPONENTS**

The DSD 440 consists of a microprogrammed controller/formatter module, one or two disk drives, a computer interface card, an interconnecting cable and a power supply. With the exception of the interface card and cable, all components are enclosed in a 5.25-inch rack-mountable chassis. Figure 1-2 illustrates these components installed in the chassis.

### **CONTROLLER/FORMATTER MODULE**

The intelligence of the DSD 440 resides in the microcode of the controller/formatter module. It accepts commands sent across the interface cable and controls the operation of the disk drives.

This module contains a microprogrammed read/write controller and an 8-bit microprocessor. Connected to this board are the disk drives, the interface



**Figure 1-2. Components of the DSD 440 System**

bus connector, and a cable connected to the power distribution assembly. The controller/formatter module is shown in Figure 1-3.

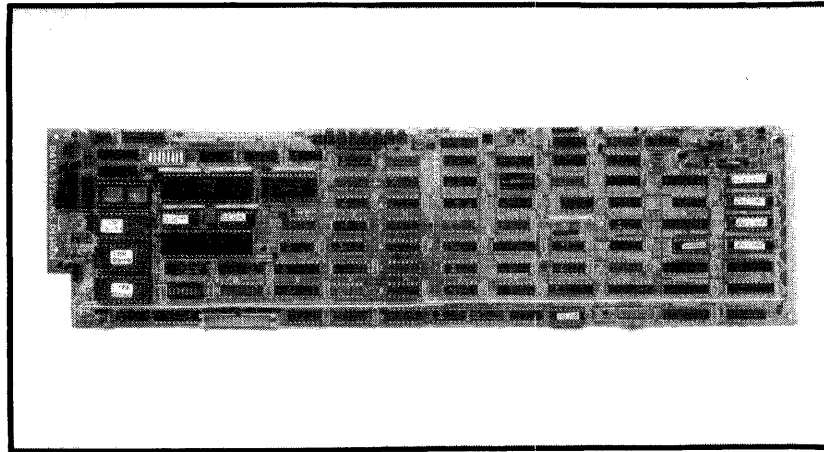


Figure 1-3. DSD 440 Controller/Formatter Module

Near the top of the controller/formatter module is a set of eight switches mounted in a dual in-line package (DIP). These switches are used to establish different system configurations and to specify the self-contained "HYPERDIAGNOSTICS" used during maintenance operations. Located near the switch is a row of eight LED indicator lights which help you interpret the status of "HYPERDIAGNOSTICS" operation. Appendix A contains the schematics for the controller/formatter module. Appendix B contains information about the standard jumper positions on this controller.

The controller/formatter is capable of formatting diskettes in DEC double density or IBM 3740 single density formats. This capability allows you to recover diskettes with altered formatting information. It also allows you to select a sequential interleave data pattern which may increase your system's throughput.

#### DISK DRIVES

The DSD 440 is equipped with one or two disk drives installed in the chassis. The drives write data or retrieve data from the flexible disks.

The drives used in the DSD 440 each consist of read/write and control electronics, a read/write head, a head positioning mechanism and a drive motor.

Each drive is fastened to the DSD 440 chassis by four screws accessible from the underside of the chassis. A 50-conductor flat-ribbon cable connects the controller module to the drives. Two additional cables provide AC and DC power to the connectors on each drive.

Appendix C contains a copy of the maintenance manual published by the drive manufacturer. Appendix D describes the standard jumper positions on each drive's printed circuit board.

### COMPUTER INTERFACE CARD

The computer interface card is a printed circuit board that accepts commands from the computer and passes them to the controller/formatter module.

Data Systems manufactures interface modules for the DEC LSI-11, PDP-11, and PDP-8 computers. These modules are shown in Figures 1-4, 1-5 and 1-6. Appendix A of this manual contains schematics of the computer interface cards. In addition, Data Systems Design can supply a complete specification of the signals and protocols on the interface cable. Customers wishing to interface the DSD 440 to other computers can design their own interface modules based on this specification.

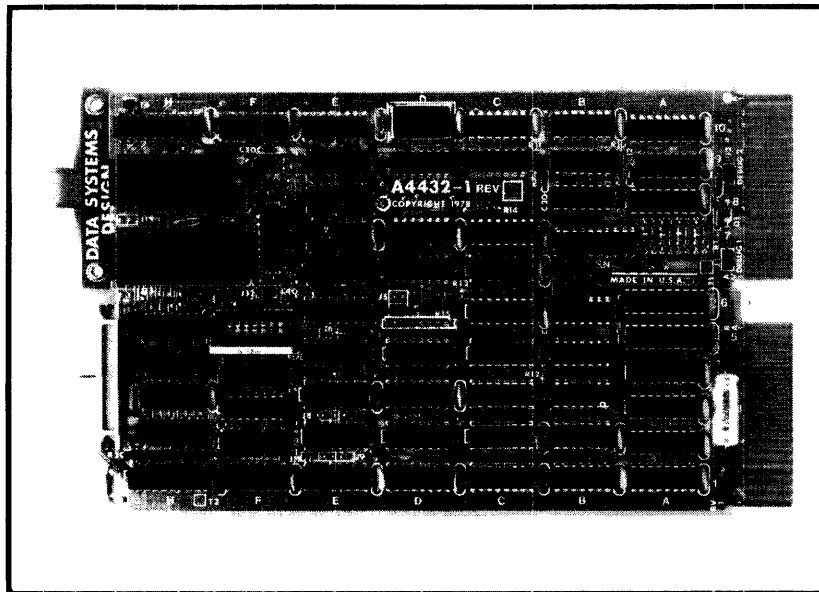


Figure 1-4. LSI-11 Computer Interface Card

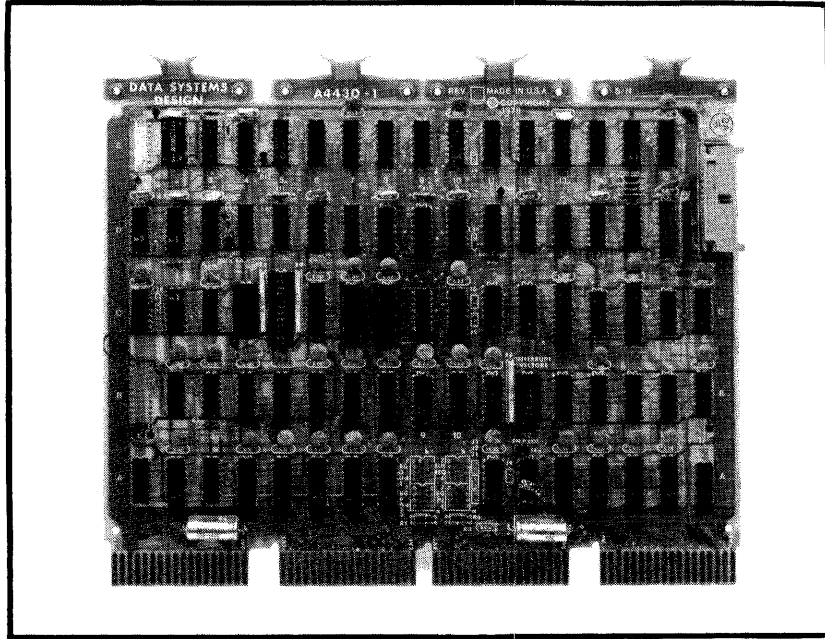


Figure 1-5. PDP-11 Computer Interface Card

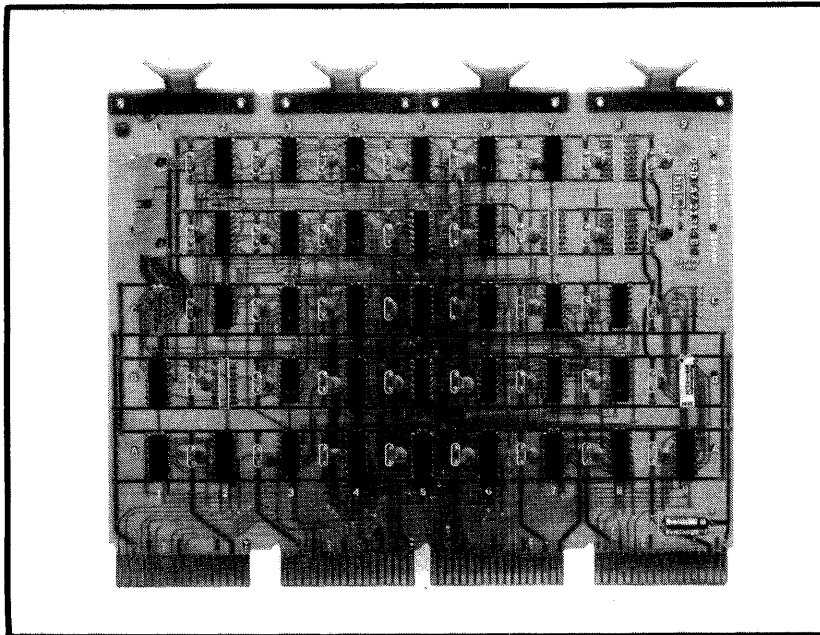


Figure 1-6. PDP-8 Computer Interface Card

## POWER SUPPLY

The DSD 440 power supply is an open-frame unit using linear regulators. Direct current output voltages include: +5 volts, +24 volts, and unregulated -12 volts. The power supply, shown in Figure 1-7, contains two trimmer potentiometers which can be used to adjust the +5 and +24 volt outputs. A schematic drawing, parts list, list of specifications, and trouble-shooting guide for the power supply are included as Appendix E of this manual.

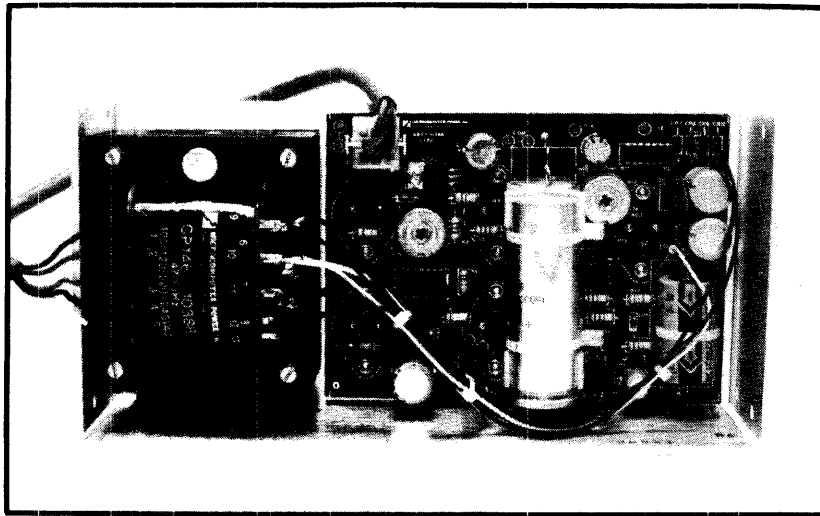


Figure 1-7. DSD 440 Power Supply

## INTERCONNECTING CABLE

The 26-conductor interconnecting cable serves as a signal path between the controller/formatter module and the computer interface card. The connectors on the ends of each cable are keyed on pin 26 to eliminate guesswork. Figure 1-8 is a picture of this cable.

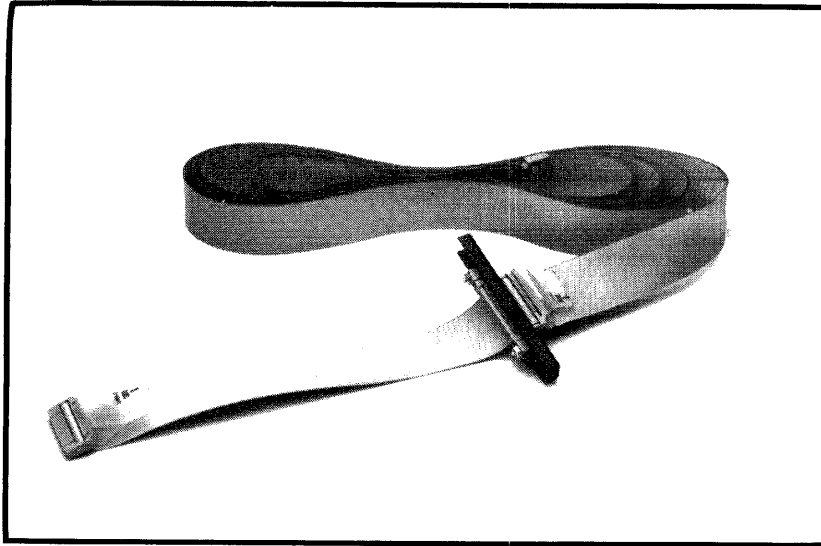


Figure 1-8. Interconnecting Cable





## CHAPTER 2

### INSTALLATION

#### UNPACKING AND INSPECTION

When your DSD 440 shipment arrives, inspect the shipping container immediately for evidence of mishandling during transit. If the container is damaged, request that the carrier's agent be present when the package is opened.

Compare the packing list attached to the shipping container against your purchase order to verify that the shipment is correct.

Unpack the shipping container and inspect each item for external damage such as broken controls and connectors, dented corners, bent panels, scratches and loose components.

If any damage is evident, notify DATA SYSTEMS DESIGN immediately.

Retain the shipping container and packing material for examination in the settlement of claims or for future use.

#### POWER REQUIREMENTS

The DSD 440 can operate on line voltages of either 120 or 240 AC volts. The line frequency must be within 1 Hz (cycles per second) of either 50 or 60 Hz.

The AC Power Connector assembly in the DSD 440 chassis contains a small printed circuit (PC) board. Figure 2-1 is a diagram of the connector assembly. You can make conversions between the two low line voltages (100 VAC or 115/120 VAC) by pulling out this board, changing its orientation and reinserting it. You can convert the DSD 440 between the two high line voltages (220 VAC or 230/240 VAC) using a similar procedure.

Although the board allows for voltage deviations of more than 10% of 120 VAC and 240 VAC, you should NEVER use this card to change between a 120 VAC and a 240 VAC system.

### CAUTION

The procedure required to convert between a low line voltage (100-120 VAC) and a high line voltage (220-240 VAC) is far more complicated than just changing the position of the printed circuit board. This conversion requires changing the fan, jumpers on the power distribution panel, the two AC spindle motors in the disk drives, the motor capacitors associated with the spindle

motors, and the fuse. **NEVER** change the position of the printed circuit board from low line voltages (100-120 volts) to high line voltages (200-240 volts) without changing the AC motors. These motors will be damaged if operated at the wrong voltage.

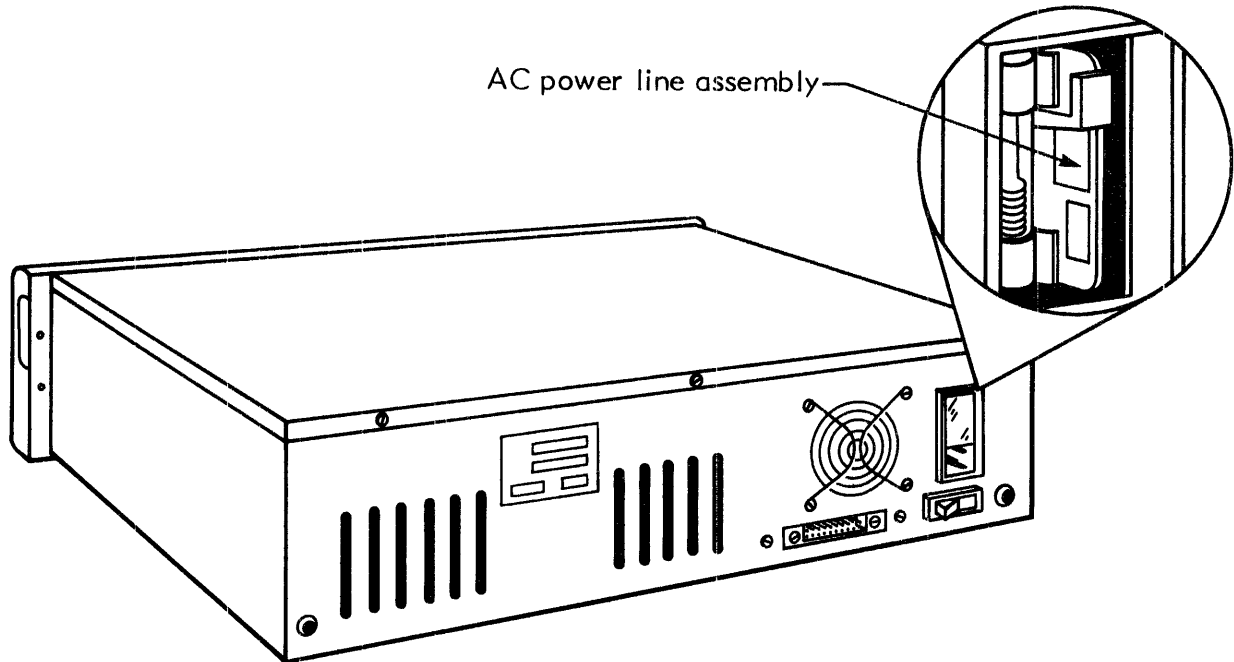


Figure 2-1. AC Power Connector Assembly

## OPERATIONAL ENVIRONMENT

All flexible disk systems manufactured by DATA SYSTEMS DESIGN perform efficiently in a normal computer room environment. Temperature, humidity, and cleanliness are three environmental considerations that can affect the reliability of diskette use.

### TEMPERATURE

The allowable operating temperature range for diskettes is shown in Table 2-1. The DSD 440 chassis should be installed where the ambient temperature range does not exceed these limits.

Table 2-1. Temperature Range for Diskettes

Operating Ambient Temperature Range  
10°C to 51°C (50°F to 125°F)

Storage Temperature Range  
-40°C to 52°C (-40°F to 125°F)

Maximum Thermal Gradient  
8° (15°F) Degrees/Hour

**HUMIDITY**

The DSD 440 requires humidity control for efficient operation. At a low humidity (dry air), static electricity can be generated when the read/write head contacts the diskette. When the electrical potential becomes high enough to cause a discharge, soft (non-repetitive) data errors may occur.

At a high humidity, the mylar diskettes absorb moisture from the air. This can move the centerline of a previously recorded track away from the centerline of the read/write head. The result will be a noticeable increase in the data error rate. The operating relative humidity range is 8 to 80% with a maximum wet bulb temperature of 29.4°C (85°F).

**CLEANLINESS**

Cleanliness is important wherever diskettes are to be stored, handled, and used. Store the diskettes in areas free of dust and corrosive chemicals. The storage area should also be free of strong magnetic fields which might damage the recorded data. When handling a diskette, never touch the exposed magnetic media.

If the DSD 440 is operated in an environment which has a high concentration of abrasive airborne particles, the useful life of the diskettes will be reduced and the data error rate increased.

For a further description of diskettes and their maintenance, refer to Appendix F.

**INSTALLING THE DSD 440 CHASSIS**

The DSD 440 chassis must be installed within ten feet (3 meters) of the interface module's location to accommodate the length of the interconnecting cable. If the computer system operator will be changing diskettes often, it may be convenient to install the chassis close to the console terminal.

The DSD 440 may be either mounted in a standard 19-inch rack or installed on a table top. The rack installation hardware consists of the items listed in Table 2-2.

The DSD 440 may also be ordered with optional chassis slides. Chassis slide installation is shown in Figure 2-14 on page 2-18 of this manual.

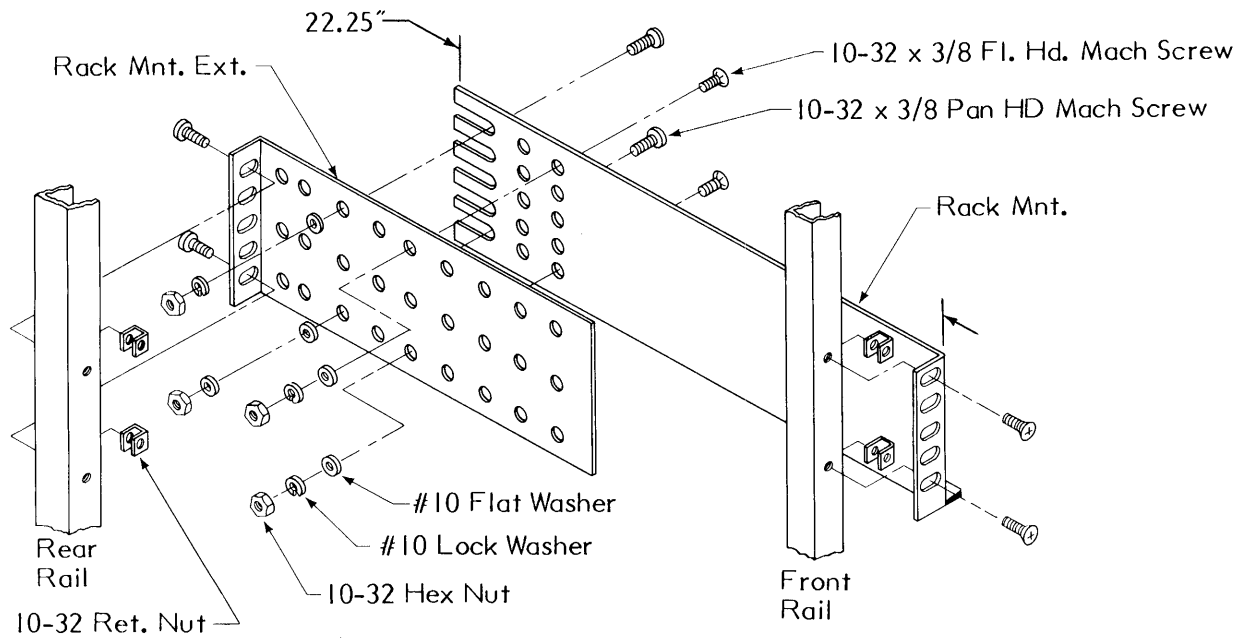
Table 2-2. Rack Installation Hardware

<u>QUANTITY</u>	<u>ITEM</u>
10	#10 x 32 Retainer Nuts
2	# 8 x 32 x 3/8 Allen Screws
2	#10 Lock Washers
2	#10 x 32 x 5/8 Captive Screws
14	#10 x 32 x 1/2 BH Machine Screws
16	#10 Star Washers
18	#10 Flat Washers
6	#10 x 32 Hex Nuts
2	Chassis Mounts

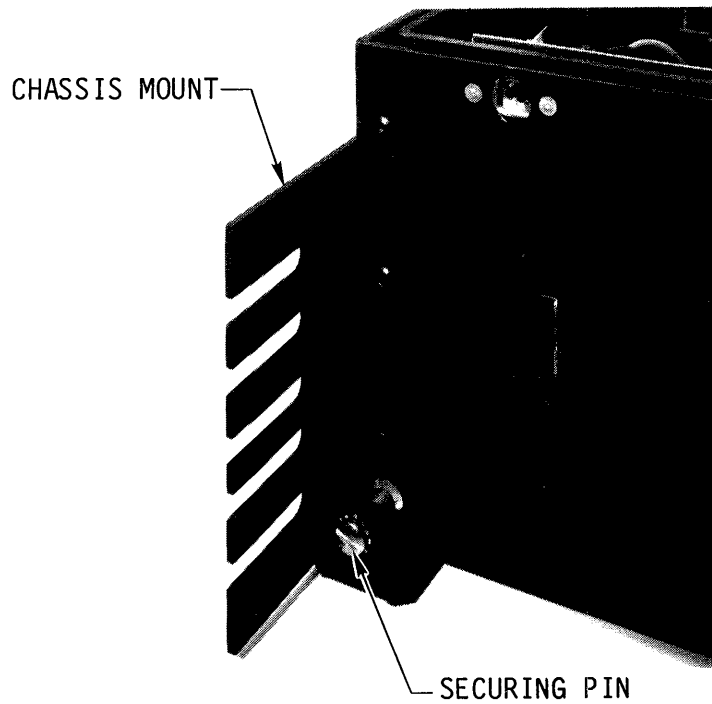
The DSD 440 chassis should be mounted in such a way that the air flow behind the fan is unrestricted. The temperature of the air entering the chassis should not exceed 40° C (104° F).

The following procedure should be used to mount the DSD 440 in a standard 19 inch instrumentation rack:

- 1) Attach the optional slim-line chassis mounts to your rack using the hardware supplied with the mounts. Note that the left and right rear extender brackets are not interchangeable. Figure 2-2 illustrates the correct relationship of the rack mounting components.
- 2) Check the DIP-Switch on the controller module inside the chassis to ensure that it matches your system configuration as described later in this chapter under "Changing the Operating Mode". The settings of this DIP-Switch are easier to modify before the chassis is secured in the rack.
- 3) Slide the DSD 440 chassis on the mounts until the two bullets at the rear of the chassis mounts engage the corresponding holes in the rear of the chassis as shown in Figure 2-3.
- 4) Remove the moded front "pop" panel from the chassis by pulling out the top of the panel.



**Figure 2-2. Installing Chassis Mounts**



**Figure 2-3. Securing the DSD 440 in a Rack**

- 5) Secure the chassis in the rack by bolting the front flange to the front rails of the rack as shown in Figure 2-4.

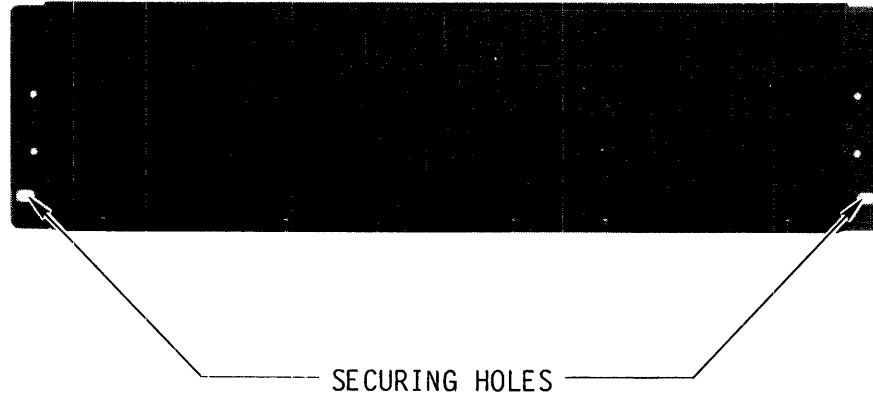


Figure 2-4. Front View of Chassis with Front Panel Removed to Show Securing Holes

You may now replace the "pop" panel by pushing it straight back onto the two "head locks".

#### INSTALLING THE DSD 440 INTERFACE MODULE AND CABLE

Ensure that all system and line power is off before proceeding with this section of the DSD 440 installation. There are separate procedures for LSI-11, PDP-11, and PDP-8 based systems.

##### LSI-11 BASED SYSTEMS

The DSD 440 interface module for LSI-11 based systems, including the PDP-11/03, is a dual-wide card marked "P/N 4432". This assembly is shown in Figure 2-5. DATA SYSTEMS DESIGN ships this interface module configured as follows:

REGISTER ADDRESS:	777170
BOOTSTRAP PROM:	ENABLED AT 173000
INTERRUPT VECTOR:	264
INTERRUPT PRIORITY:	BR4
OPERATING MODE:	MODE 2 (RX02 COMPATIBLE)

The module allows you to select one of four device register addresses, one of four bootstrap PROM (Programmable Read Only Memory) starting addresses, and a 7-bit interrupt vector address. Table 2-3 lists standard and alternate addresses for the registers and starting boot PROMs. As shown in Figure 2-6, there is a separate jumper which, when installed, disables the bootstrap PROM.

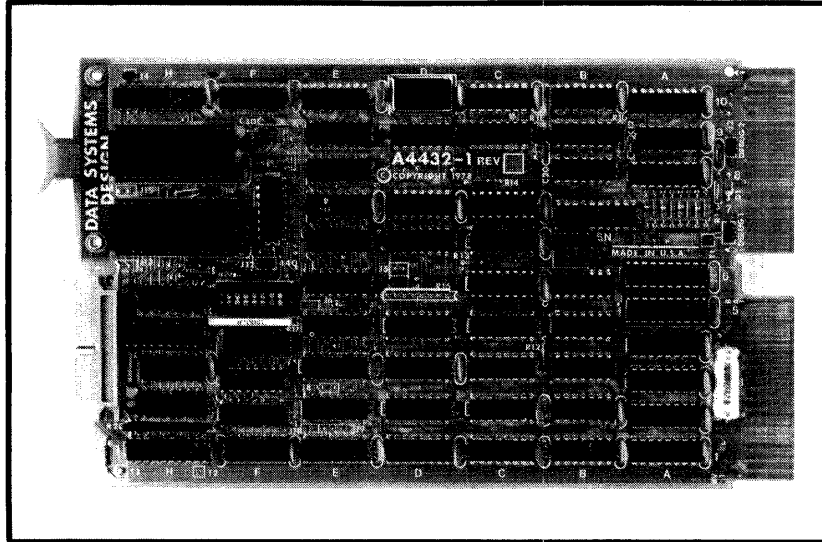


Figure 2-5. LSI-11 Computer Interface Card

Table 2-3. Register and Boot PROM Addresses

STARTING REGISTER ADDRESS	POSITION 1	POSITION 2
177170 (Standard)	CLOSED	CLOSED
177160	OPEN	CLOSED
177140	CLOSED	OPEN
177150	OPEN	OPEN
STARTING BOOT PROM ADDRESS	POSITION 3	POSITION 4
173000 (Standard)	CLOSED	CLOSED
171000	OPEN	CLOSED
175000 *	CLOSED	OPEN
166000 **	OPEN	OPEN

\* Use only this address of the system if configured for 31K of memory (0-174000).

\*\* Do not use this address for the bootstrap in systems with over 28K of memory.

DBST  
 Shorted = Disabled boot  
 Open = Enable boot

Interface  
 Cable Connector

Interrupt Vector  
 Jumpers IV8-IV2  
 Shorted = 0  
 Open = 1

Normal Vector  
 is 264<sub>8</sub>:

Octal  
 Numeral

IV8	Clsd	0	2
IV7	Opn	1	
IV6	Clsd	0	6
IV5	Opn	1	
IV4	Opn	1	
IV3	Clsd	0	4
IV2	Opn	1	
IV1	N/A*	0	
IV0	N/A*	0	

\*Can only be set  
 at 0 or 4.

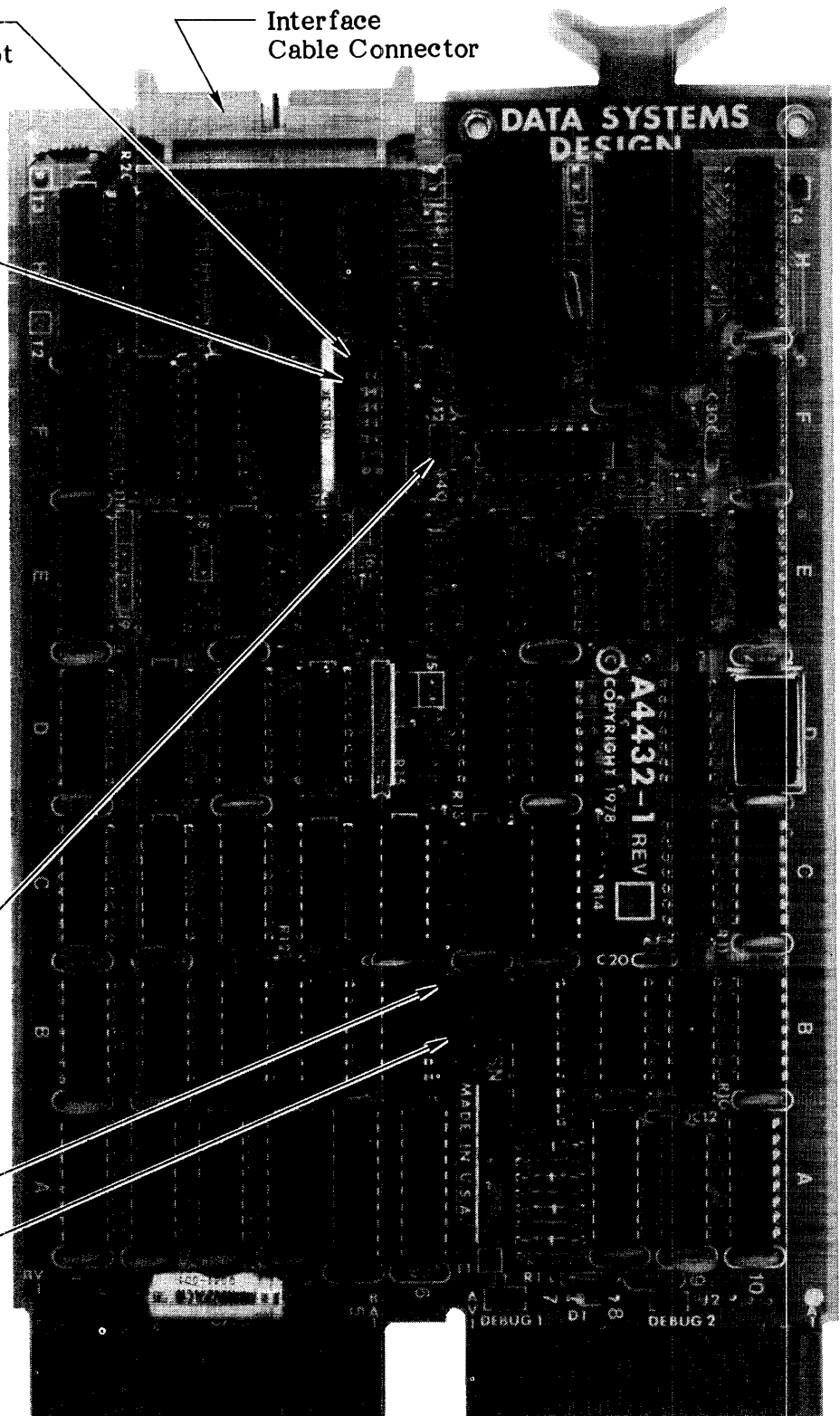
J12

Shorted = RX01  
 Open = RX-02

Starting  
 device and  
 boot select  
 jumpers

Position 1

Position 4



**Figure 2-6. LSI-11 Computer Interface Card Diagram**



## NOTE

When the interrupt vector jumpers are in place, the associated bit of the vector address is a "0". Thus, if all seven vector jumpers were to be installed, the vector address would be 000. Check your module against Table 2-4 and Figure 2-6 to ensure that it has been configured to match your system requirements before installation. Most system software assumes a device address of 177170 and an interrupt vector of 264. If you change either of these numbers, corresponding changes will be required in the software. Also, be sure to read the explanation of the bootstrap and diagnostic programs if non-standard addresses are used.

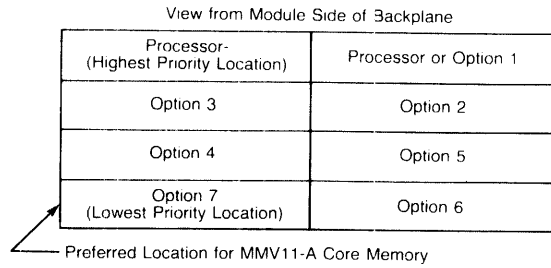
The following procedure describes how to install the LSI-11 interface module:

- 1) VERIFY LINE POWER IS OFF.
- 2) Plug one end of the interface cable into the interface module so that pin 1 (the striped side) is closest to the edge of the board. Note that the position of the clipped pin on the module connector matches the position of the plugged hole on the cable connector.
- 3) Plug the opposite end of the interface cable into the keyed connector mounted on the rear panel of the chassis. Note that the position of the clipped pin on the module connector matches the position of the plugged hole on the cable connector.

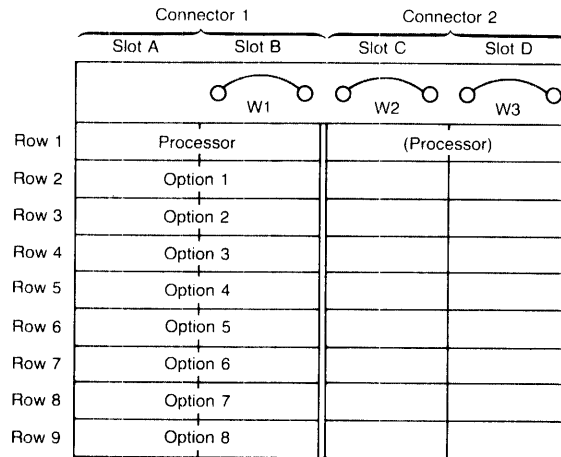
Now you are ready to plug the module into the lowest numbered available Q-Bus slot.

## NOTE

There must be no open Q-bus slots between the processor and the DSD 4432 interface module. Since this module uses both interrupts and DMA (Direct Memory Access), a break in either of the grant propagation chains will prevent the interface module from obtaining control of the Q-Bus. Figure 2-7 shows how Q-Bus slots are numbered on the standard backplanes available from DEC.

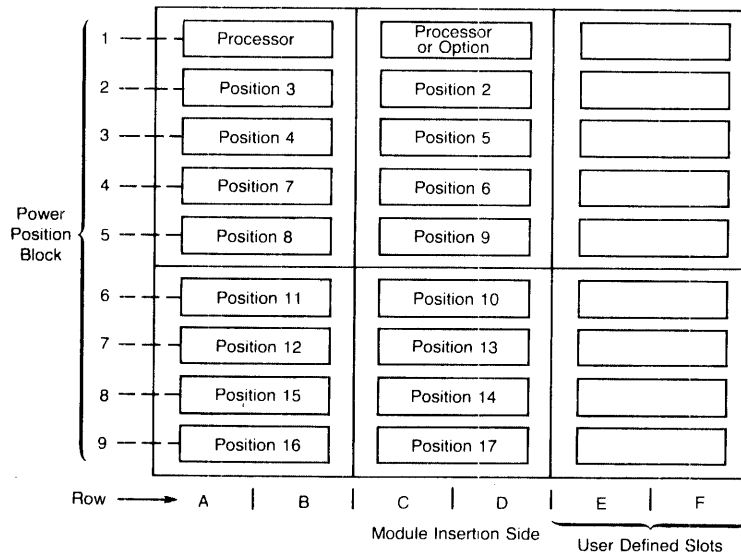


### DEC Backplane H9270



View is from Module Side of Connectors.

### DEC Backplane H9273-A



### DEC Backplane DDV11-8

**Figure 2-7. Option Priority in LSI-11 Backplanes**

## PDP-11 BASED SYSTEMS

The DSD 440 interface module for PDP-11 based systems, not including the PDP-11/03, is a quad card. This assembly is shown in Figure 2-8. DATA SYSTEMS DESIGN ships the interface module configured as follows:

REGISTER ADDRESS:	777170
BOOTSTRAP PROM:	ENABLED AT 77100
INTERRUPT VECTOR:	264
INTERRUPT PRIORITY:	BR5
OPERATING MODE:	MODE 2 (RX02 COMPATIBLE)

This interface module is marked "4430".

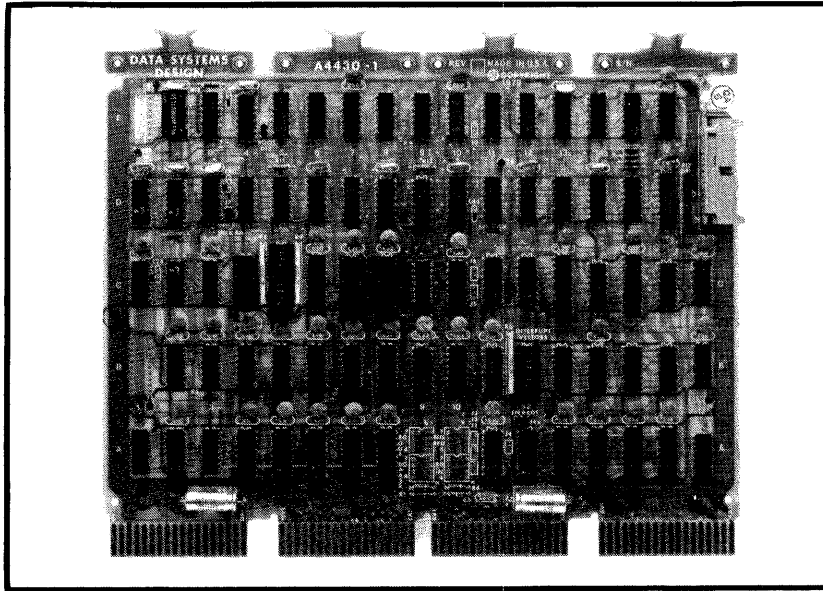
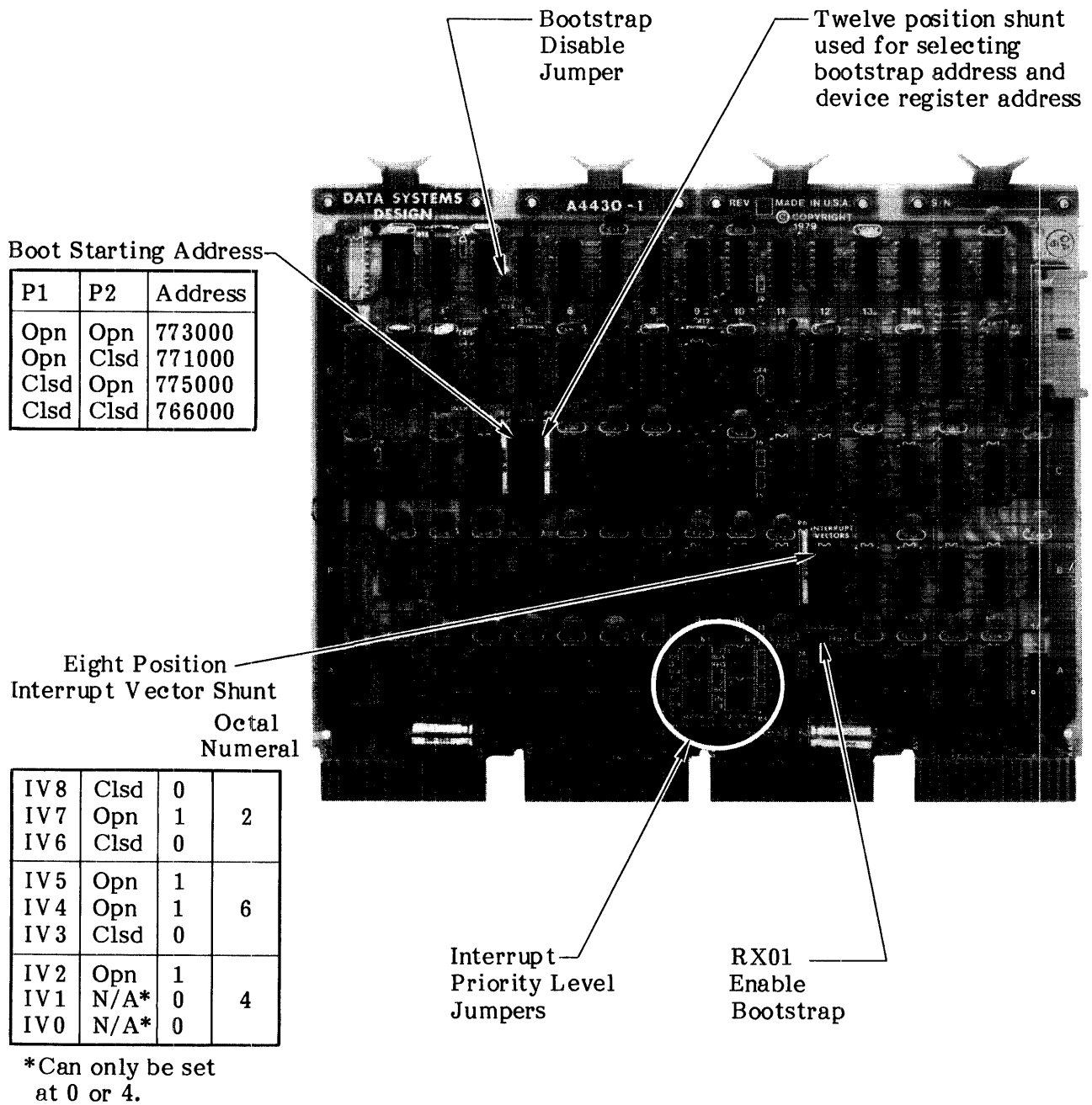


Figure 2-8. PDP-11 Computer Interface Card

The twelve position shunt located at coordinates C-5 on the 4430 interface module is used to configure device register addresses and the bootstrap program starting address. Figure 2-9 is an illustration of the PDP-11 interface module. It shows how the twelve shunt positions are numbered.

Shunt positions 1 and 2 are used to configure the bootstrap program starting address as follows:

STARTING BOOT PROM ADDRESS	POSITION 1	POSITION 2
773000	OPEN	OPEN
771000 (Standard)	OPEN	CLOSED
775000	CLOSED	OPEN
766000	CLOSED	CLOSED



**Figure 2-9. PDP-11 Computer Interface Card Diagram**

The bootstrap program contained on the interface module occupies 256 words of memory space, starting at one of the four selectable addresses shown above. If you do not want the bootstrap program to respond to any addresses, the bootstrap disable jumper should be installed as shown in Figure 2-9.

Shunt positions 3 through 12 correspond to address bits A3 through A12 respectively when configuring the device register address. A closed shunt position corresponds to a

binary 0 and an open shunt position corresponds to a binary 1. When the PDP-11 interface module is shipped, it is configured to respond to a base register address of 777170 (octal). This is done by having shunt positions 7 and 8 left closed, and positions 3, 4, 5, 6, 9, 10, 11 and 12 punched open.

The eight position shunt located at coordinates B-12 is used to configure the interrupt vector address. Figure 2-9 shows how the eight shunt positions are numbered. Position 1 is not used. Positions 2 through 8 correspond to interrupt vector address bits IV2 through IV8 respectively. A closed shunt position corresponds to a binary 0 and an open shunt position corresponds to a binary 1.

When this interface module is shipped, it is configured to have an interrupt vector address of 264 (octal). This is done by having shunt positions 3, 6 and 8 left closed, and positions 2, 4, 5 and 7 punched open.

If the interrupt priority level must be changed, cut and jumper the circuit board to resemble the diagram corresponding to the desired interrupt priority level as shown in Figures 2-10 and 2-11. If the priority levels will be changed often, cut the six permanent traces and install the four 8-pin IC sockets in the positions outlined on the board. Placing either four-position shunts or DIP-Switches in the sockets will allow for repeated jumper changes. The interrupt priority jumpers are located at coordinates A-9 and A-10 on the interface module circuit board. Interrupt priority level 4 is the lowest and level 7 is the highest.

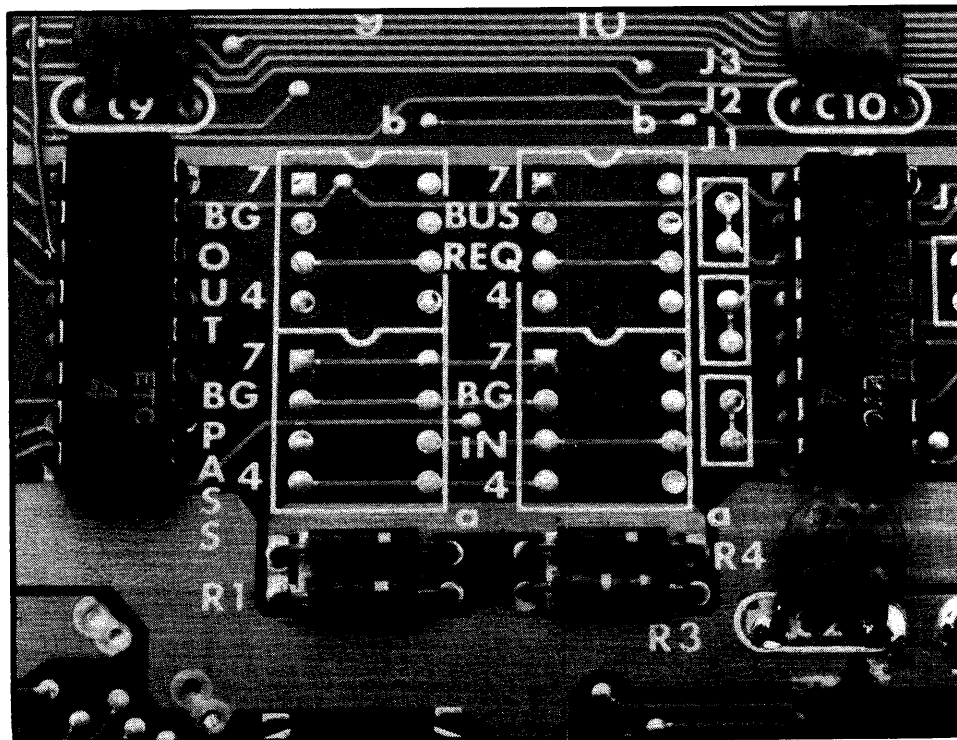


Figure 2-10. Interrupt Priority Level

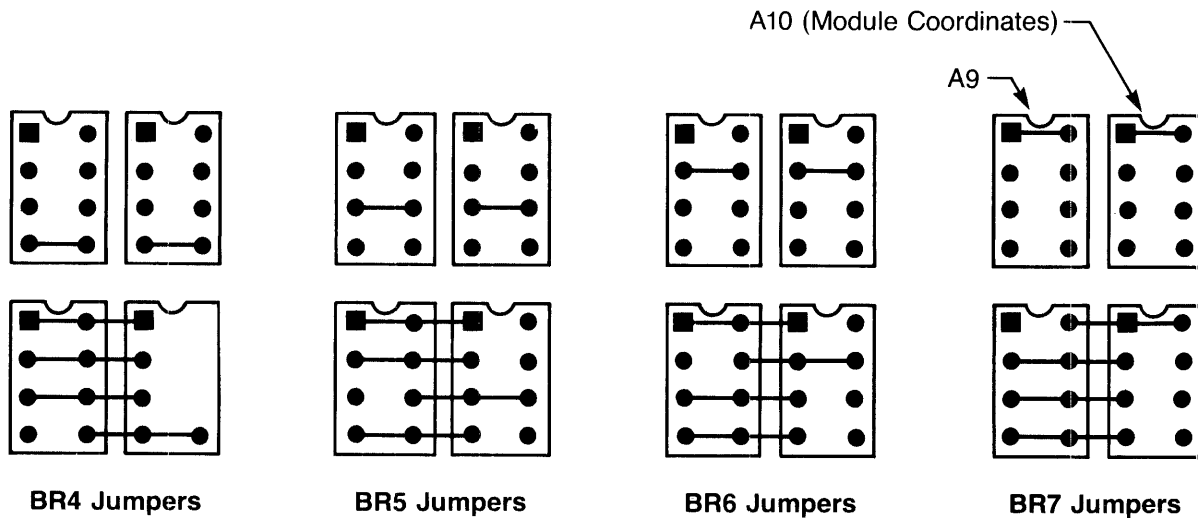


Figure 2-11. Interrupt Priority Level Diagram

If the system is to be operated in Mode 1 (RX01 compatible) then the EN RX01 jumper located near coordinates A-12 must be installed, and rocker switch #4 of the DIP-Switch set to open. This jumper is removed for RX02 compatible operation, and rocker switch #4 is closed.

The following procedure describes how to install the PDP-11 MODULE:

- 1) VERIFY LINE POWER IS OFF.
- 2) Check that the jumpers on the interface module are configured correctly.
- 3) Plug one end of the interface cable into the interface module so that pin 1 (the striped side) is closest to module handle.
- 4) Confirm that the position of the clipped pin on the module connector matches the position of the plugged hole on the cable connector.
- 5) Plug the module into a convenient SPC (Small Peripheral Controller) slot.

Be sure there is grant continuity between the processor and the interface module. If there are open SPC slots between the processor and the interface module, place a grant continuity card in slot D.

## NOTE

Since the 4430 interface module uses DMA (Direct Memory Access), you must be sure there is no backplane jumper or foil trace between backplane pins CA1 and CB1 of the selected SPC slot. These two pins normally connect NPG (Non-Processor Grant) IN to NPG OUT. Usually the pins are left connected since most Small Peripheral Controllers do not use DMA. If this jumper is not removed and a 4430 interface module is installed configured either for RX01 or RX02 compatible operation, the computer system will stop. Replace the jumper any time the 4430 module is removed. Otherwise, DMA devices on the UNIBUS will never receive NPG and the UNIBUS will stop each time a DMA cycle is attempted by one of these devices.

## PDP-8 BASED SYSTEMS

The DSD 440 interface module for PDP-8 based systems is a quad size card marked "DSD 2131". Figure 2-12 is a picture of this assembly. Data Systems Design ships this module jumpered to respond to device code 75 (octal). All device code jumpers except jumper 7 should be in place if this is the device code required. There are no additional jumpers or adjustments on the PDP-8 interface module.

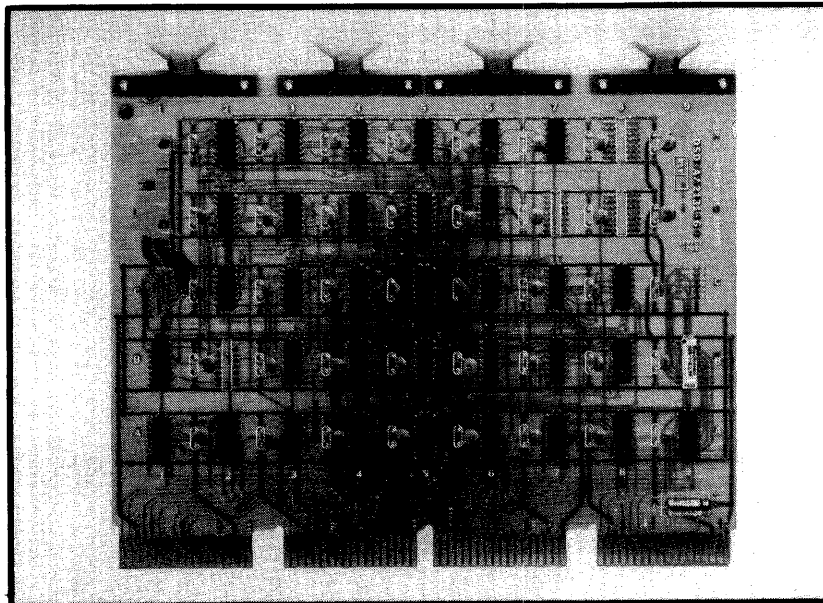


Figure 2-12. PDP-8 Computer Interface Card

The following procedure describes how to install the PDP-8 interface module:

- 1) VERIFY LINE POWER IS OFF.
- 2) Plug one end of the cable into the interface module. The position of the clipped pin on the module connector should match the position of the plugged hole on the cable connector.
- 3) Plug the opposite end of the interface cable into the connector mounted on the rear panel of the chassis. There is only one correct way to insert the cable.
- 4) Plug the module into an available OMNIBUS slot.

Since the PDP-8 interface module does not contain a bootstrap, a toggle-in bootstrap program is included in Appendix H of this manual. PDP-8A users may either toggle-in the bootstrap program or may install the two PROMs shipped with the PDP-8 interface card into their DEC M8317 bootstrap board. The replacement procedure for these PROMs is also contained in Appendix H.

#### CONFIGURING THE DSD 440 FOR RX01 OR RX02 MODE OPERATION

The DSD 440 can be configured to operate as a DEC RX02 for DEC double density or IBM 3740 single density recording, or as a DEC RX01 for compatibility with earlier versions of your DEC system software.

DATA SYSTEMS DESIGN ships the DSD 440 configured to operate as a DEC RX02 (Mode 2). If your DEC operating system diskette is configured for RX01 compatible operation, install the "EN RX01" jumper on your LSI-11 or PDP-11 interface module. This configures the DSD 440 for RX01 compatible operation. Figures 2-6 and 2-9 show the location of the jumper installation. Rocker switch #4 of the DIP-Switch located on the controller/formatter module must also be turned to the open position. This switch is illustrated in Figure 2-13. Then return the DSD 440 to Mode 2 operation by removing the "EN RX01" jumper on the LSI-11 or PDP-11 module, and closing rocker switch #4.

To create a double density system diskette for double density operation, follow the procedure in Chapter 4 titled "Generating an RX02 Compatible System Diskette".

No changes are required on the PDP-8 interface module when changing between RX01 and RX02 compatible operations.

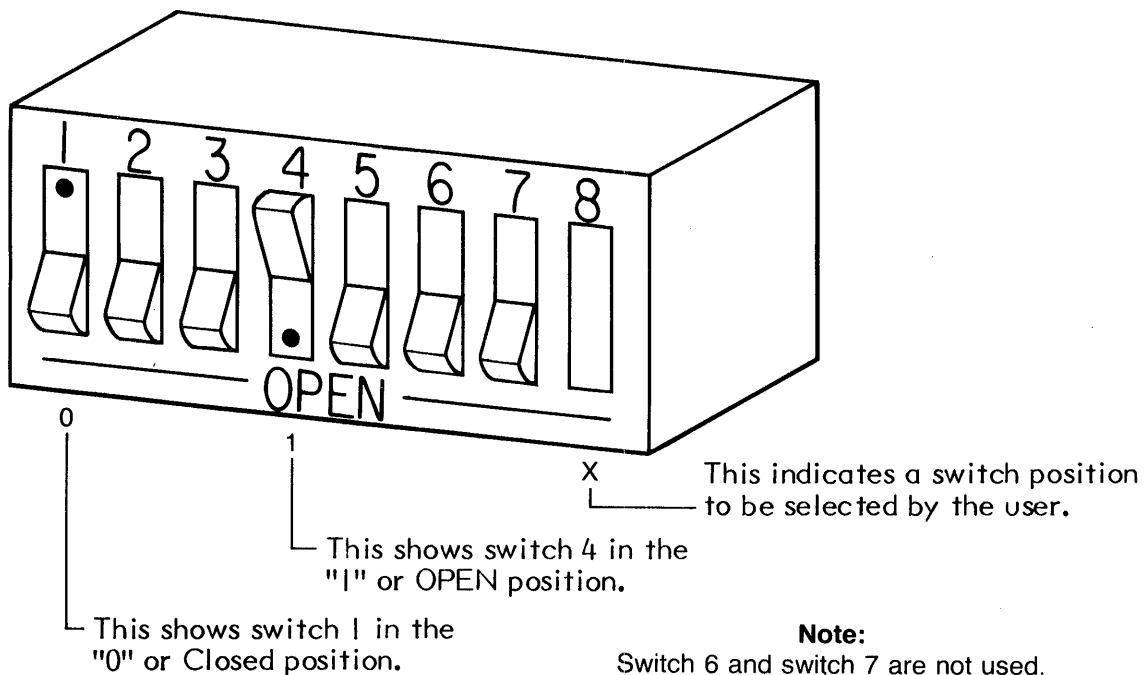


## DSD 440 CHASSIS CONNECTIONS

The following procedure describes the installation of the AC power cord and the interface bus cable:

- 1) VERIFY LINE POWER IS OFF.
- 2) Plug the female end of the power cord into the connector on the back of the chassis.
- 3) Plug the other end into the AC power receptable.
- 4) Route the free end of the interface bus cable over to the rear of the chassis.
- 5) Plug it into the 26 pin connector so that the striped side of the cable is toward the middle of the chassis.

Be sure the position of the clipped connector pin in the chassis rear matches the position of the plugged hole in the cable connector.



**Figure 2-13. Controller/Formatter Module DIP-Switch**

**ASSEMBLY INSTRUCTIONS**

Unpack your chassis slide kit and identify the right and left chassis slide by the stamped part no: Left is P/N XXXXXX-01, Right is P/N XXXXXX-02. (See detail-A.)

After identifying the right and left chassis slides (see chassis mounting), remove the inner slides by fully extending the slides and then releasing the safety stop. Assemble the inner slides to chassis using the fasteners shown.

To position the chassis slides, use the recommended dimensions (see detail-B). The positioning is contingent upon mounting your new system underneath or above the existing system. Align the flange of the chassis slide with the two nearest mounting holes of the rack.

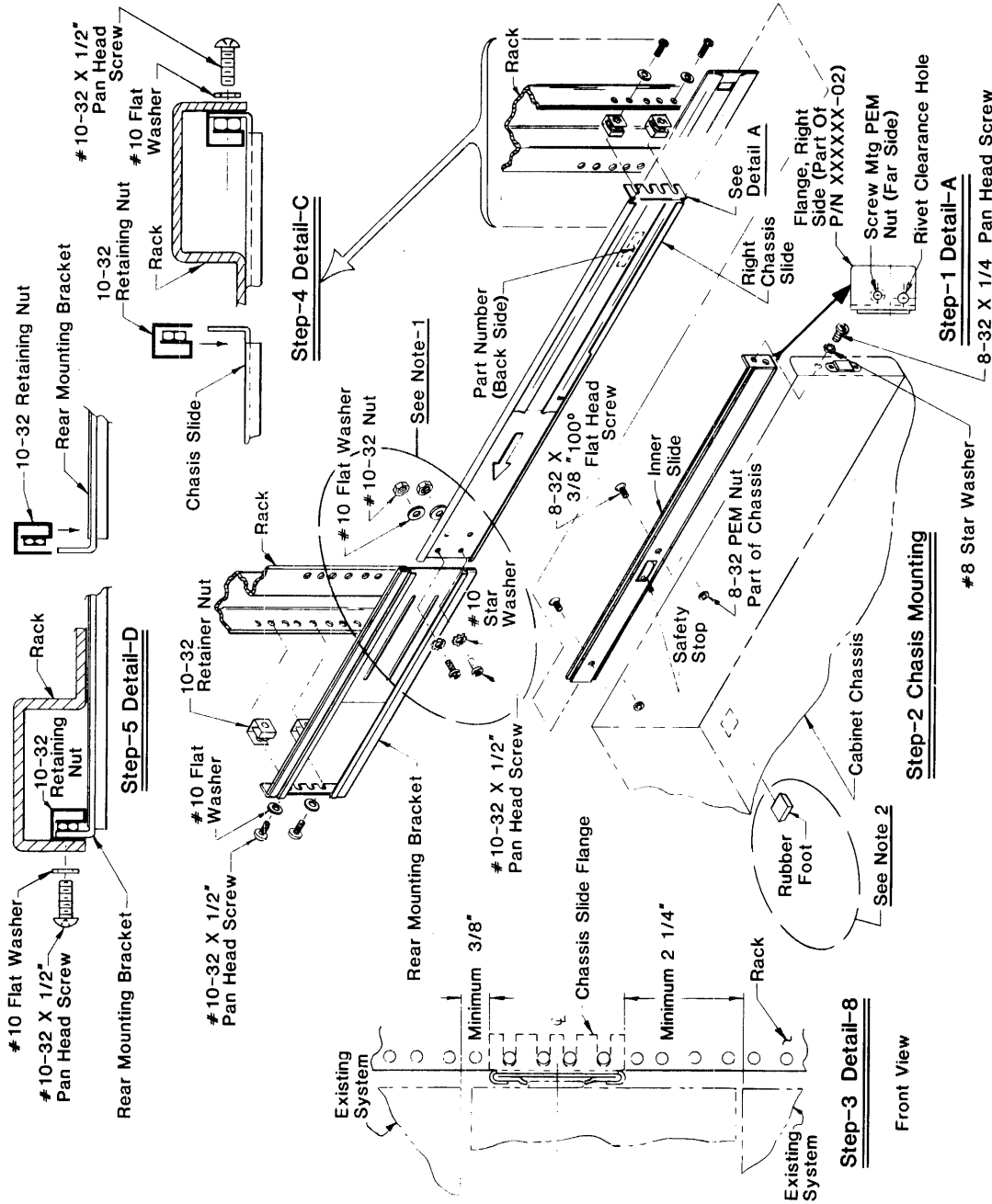
After determining which two holes/slots will be used, slide the retaining nuts into the appropriate slots on the mounting flange of the chassis slides (see detail-C). Fasten the chassis slides to the rack using the fasteners shown.

Slide the rear mounting bracket over the chassis slide until the flange meets the back of the rack. Align the bracket with the two nearest mounting holes on the slide and rear bracket level.

NOTE-1: For the extra long racks, additional hardware has been supplied for stiffening the assy.

2. Remove rubber feet from system before installing into rack.

NOTE: The rear has the same slot spacing relative to the center of the chassis slide. Slide the retainer nuts into the appropriate mounting slots, re-align the bracket to the holes and fasten with the hardware shown. (See detail-D)



STEP-1.

2.

3.

4.

5.

NOTE-1.

2.

NOTE:

**Figure 2-14. Installing Chassis Slides**

## CHAPTER 3

### OPERATION

This section explains how to turn on and operate your DSD 440 after correctly installing it. Chapter 2 details the installation procedure

#### APPLYING AC POWER

With the correct AC (alternating current) power connected, simply turn the line switch on the DSD 440 chassis to the "ON" position. The location of the line switch is shown in Figure 3-1.

The DSD 440 chassis and the interface module can be powered up in either order safely. There is no danger of writing on diskettes loaded in the drives during power up or power down cycles.

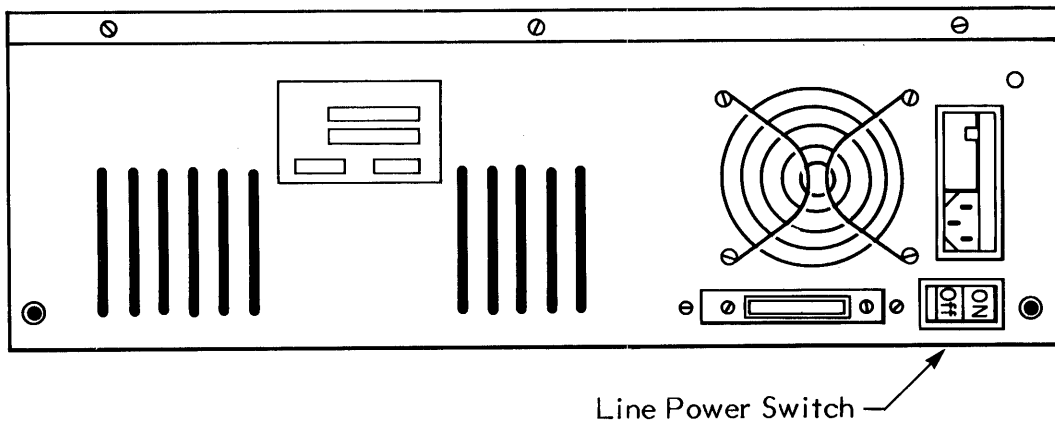


Figure 3-1. Back Panel of DSD 440

## INSERTING A DISKETTE

Ensure that the diskette is soft sectored and single sided in an eight inch square jacket. If you are not sure if your diskette meets this criteria refer to Appendix F for a more detailed description of diskettes. If you have the correct type of diskette, then open the DSD 440 drive door and insert it into the drive with the orientation shown in Figure 3-2. Close the drive door. You are now ready for an initialization response check of your computer and disk system.

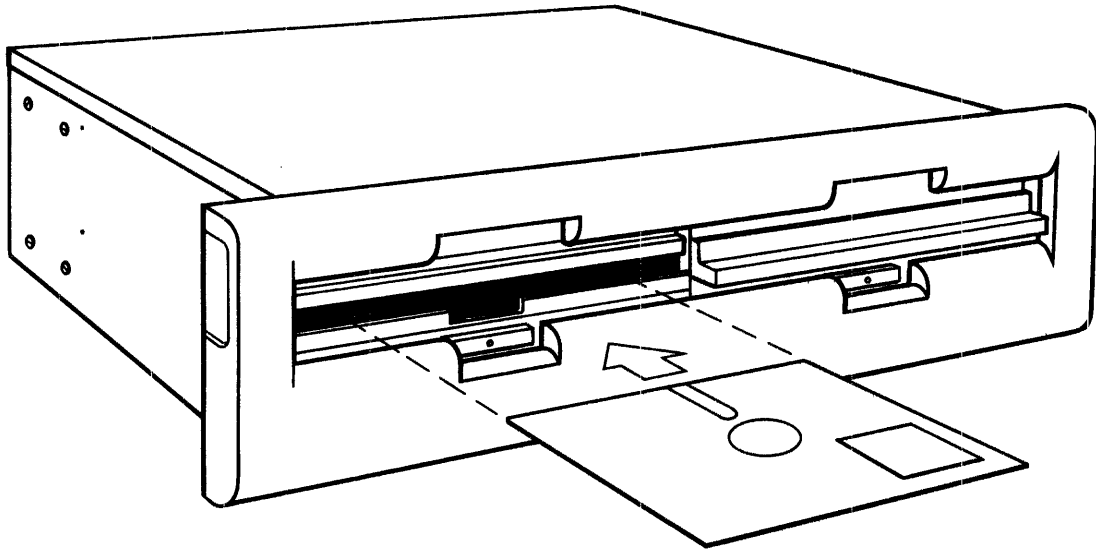


Figure 3-2. Orientation of Diskette for Insertion

## INITIALIZATION RESPONSE CHECK

When the DSD 440 is connected correctly to the host computer and if its drive doors are closed, it performs an initialization response (INIT) upon power-up. An initialization response can also be forced by some of the following operator console actions:

### LSI-11 BASED SYSTEMS

- 1) Flip the INIT switch (if there is one) on the host computer.
- 2) using ODT (Octal Debugging Tool), enter the "G" command at any arbitrary starting address.
- 3) Using ODT, write the number 40000 into the DSD 440 RXCS register, normally at address 777170.

### PDP-11 BASED SYSTEMS

- 1) Generate a UNIBUS INIT by depressing the START switch or button on the processor.
- 2) Using the console, write the number 40000 into the DSD 440 RXCS register, normally at address 777170.

### PDP-8 BASED SYSTEMS

- 1) Depress the system clear switch.
- 2) Load and execute the "clear all flags" I/O instruction.

Each time you generate an INIT, you should hear the drives as the controller moves the head to track 0. The activity lights on the front of the drives should come on briefly. If a diskette is loaded into drive 0 (normally the left hand drive) you should also hear the head load. The drive 0 activity light remains on slightly longer as the controller reads track 1/sector 1 of the diskette into the sector buffer.

If you did not observe the results described above, ensure that you have:

- 1) Applied power to both the computer mainframe and the DSD 440 chassis.
- 2) Connected both ends of the DSD 440 interface bus cable in the proper orientation. (The red drive select lights remain on if the cable is reversed.)
- 3) Generated a system or device initialize signal by one of the methods described above, and the signal is reaching the DSD 440 interface.
- 4) Closed the drive doors.
- 5) Set the controller module DIP-Switch as described earlier in this chapter.

If you are still unable to force an INIT, refer to Chapter 5.

## LSI-11 AND PDP 11 BOOTSTRAPPING (INITIAL PROGRAM LOADER)

Before attempting to bootstrap your DEC operating system diskette, ensure that the DSD 440 operating mode (RX01 or RX02 compatible) matches the device configuration of the system diskette. To convert your single density operating system diskette to double density operation, follow the procedure in Chapter 4 titled "Generating an RX02 Compatible System Diskette".

The DSD 440 features a built-in hardware bootstrap program on the LSI-11 and PDP-11 interface modules. When this 512-byte (256 16-bit words) program is executed by the computer, the operating system is loaded into memory from either a single or double density diskette automatically. The bootstrap also performs diagnostics which confirm operation of the interface, controller and processor memory. These diagnostics include:

- 1) A fill and empty buffer test which verifies the sector buffer and DMA transfer capability. It loads a data pattern, then reads it back and compares the results.
- 2) A command and status register bit-latch test that confirms correct operation using the DSD 440 interface register.
- 3) A computer memory test that checks all available memory for both data and address line errors.

If a malfunction is detected during execution of any of these tests, the computer either HALTs or continuously executes a program loop at an address specific to the problem at hand. If this occurs, the drives are quiet and nothing appears on the console terminal. In this case, halt the processor manually by pressing down the ENABLE HALT toggle switch on the front computer panel. This determines the address at which the program is looping. Once this address is known, refer to the bootstrap program listing in Appendix G to find out which test failed. LSI-11 users in this situation should remember that the ODT console monitor types the address of two locations past the HALT instruction.

After successful completion of the system diagnostics, the bootstrap program reads sector 1, track 1 of drive 0 into the controller sector buffer. Should this operation cause a density error, the density bit is set and the command is issued again. If any other error results, the processor halts leaving the drive number in Register O (RO), the memory address in Register 4 (R4), and the definitive error code in Register 6 (R6).

If the READ SECTOR operation is successful, the bootstrap program determines the operational mode of the DSD 440. If the system is configured for RX02 Mode, a DMA empty buffer cycle occurs. A programmed I/O cycle occurs if the system is in RX01 Mode.

At this point, the first word of data transferred to memory (at address 000000) is examined. If that word is NOP (No Operation) instruction (000240 octal), the bootstrap program concludes that the diskette is bootable. In this case the program counter is cleared and the secondary bootstrap program proceeds to load the operating system.

If the bootstrap program does not find an NOP instruction in address 0, it continues to try booting the diskette in drive 0 until any hard error is detected. When a hard error is induced (such as by popping the diskette out of the drive), the bootstrap program halts. If you have inserted the wrong diskette into drive 0, you should place a bootable diskette into drive 0 and start the bootstrap again. If you want to bootstrap the operating system using drive 1 as the system device, place a bootable diskette into drive 1 and generate a CPU "continue".

Normally, the bootstrap program is executed by loading the program counter with the diskette bootstrap program base address. This address is determined by the position of switches or jumpers on the interface modules. After loading the starting address, start the computer.

## PDP-8 SYSTEM BOOTSTRAPPING

Before attempting to bootstrap your DEC operating system, ensure that the DSD 440 operating mode (RX01 or RX02 compatible) matches the device configuration of the system diskette.

### NOTE

The DSD 440 operating mode must match the device configuration of the system diskette in order for successful bootstrapping to occur.

Your PDP-8 system may be bootstrapped by either of the following two methods:

- 1) The bootstrap program listed in Table H-2 (APPENDIX H) may be toggled in manually through the front panel. This bootstrap works in both RX01 and RX02 modes.
- 2) PDP-8/A users may replace the boot PROMS on their DEC 8317 options board using the procedure given in APPENDIX H. The PROMS provided will boot in both RX01 and RX02 modes with no switch changes required.

Both of the above bootstrap methods start execution of the bootstrap at location 33 (octal).

## BOOTSTRAPPING LSI-11 AND PDP-11 SYSTEMS WITH NON-STANDARD DEVICE ADDRESSES

Most DSD 440 systems are configured so the command and status register responds to address 177170. This address is typically regarded as the "standard" device address for the first storage peripheral installed on LSI-11 or PDP-11 based computer systems. Under certain circumstances, you may want to configure your DSD 440 system to respond to a non-standard device address. If this is done, the bootstrap procedure is slightly modified.

The following are descriptions of several types of bootstrap starting procedures:

- 1) If the shunts on the interface module are set up so that the bootstrap program base address is 173000 (773300) and the RXCS = 177170 (777170) (standard address), bootstrap by starting the computer at the bootstrap program base address.
- 2) If the interface module is modified so that the RXCS = 177150 (777150), the system is bootstrapped by starting the computer at the bootstrap program base address plus 20 (octal). In this case the base address is 173020.
- 3) If the interface module is modified so that the device address is any legal address other than 177150 or 177170, follow this procedure:
  - a) Write the device address (e.g., 177160) into memory address 000000.
  - b) Write the number 000340 into CPU register 0 (R0).
  - c) Write the number 000002 into CPU Register 1 (R1).

You can now start the computer at the bootstrap program base address plus 40 (octal). In this case the base address is 173040.

If the DSD 440 is configured for a non-standard bootstrap starting address, the system device handler on the operating system must be altered.

#### INITIALIZING LSI-11 and PDP-11 SYSTEMS WITH READ ONLY MEMORY (ROM) INSTALLED AS MAIN MEMORY

The DSD 440 bootstrap program reports a memory error if a block of ROM (read-only memory) is installed within the first contiguous block of read/write memory. This is because the bootstrap program sizes the read/write memory by reading sequential addresses from 0 until the first TRAP to 4. The bootstrap program also tests the read/write memory

The following is a description of how to bootstrap a DSD 440 if the standard bootstrap procedure results in memory errors.

- 1) Set computer for single-step/HALT operation by pressing down RUN/HALT switch on front panel.
- 2) Start bootstrap program as usual and single-step program by typing "P" or pressing CONTINUE after each instruction until you reach 74 address locations beyond the boot base address (i.e., 173074 if Boot Base Address = 173000).
- 3) Set computer register 2 (R2) or 777702 on the PDP-11 to the highest memory address you would like tested. On the LSI-11 type "R2/", after which the computer indicates the contents of R2. Type in the new contents followed by <CR>.



- 4) Set computer register 7 (R7) or 777707 on the PDP-11 to 106 address locations beyond the boot base address (i.e., 173106 if Boot Address = 173000).
- 5) Move the RUN/HALT switch to RUN and continue by pressing up on the RUN/HALT switch and by typing "P" on the system console.

The above procedure allows you to specify the upper address limit for the read/write memory test.

#### BOOTSTRAPPING WITHOUT SYSTEM TEST FUNCTIONS

The following procedure describes how to skip all of the system test functions included in the DSD 440 bootstrap program and to directly initialize the operating system:

- 1) The LINE-TIME CLOCK switch must be off.
- 2) Deposit the device address of the device to be bootstrapped in location 000000 (typically 177170).

You may now start the CPU at 524 address locations beyond the boot base address (i.e., 173524 if Boot Base Address = 173000).

#### ACCEPTANCE TESTING

When the DSD 440 is first installed, all of the recommended acceptance tests described in Chapter 5 should be performed as well as the initialization response check. These tests include the acceptance mode tests of the following diagnostics:

- FRD440 for LSI-11 and PDP-11 based systems, and
- VEP-210 for PDP-8 based systems

in addition to the general systems exerciser portion of the "HYPERDIAGNOSTIC" self-test routines.



## CHAPTER 4

### SOFTWARE

This chapter discusses the incorporation of single and double density data storage into your DEC computer system. First, the use of single density diskettes in a system configured for double density is described. Second, three formatting procedures are explained for double density diskettes. Third, the procedures for generating double density RT-11 and OS/8 system diskettes are detailed. Finally, the programming interface to the operating system is described, including register definitions.

#### SINGLE DENSITY DISKETTES IN A DOUBLE DENSITY RX02-COMPATIBLE SYSTEM

The DSD 440 allows previously recorded single density file diskettes to be read and written by a double density operating system. The double density software device handler also reads and writes previously recorded single density diskettes. The expected density of a diskette is specified with the initial command sent by the device handler to the DSD 440. If the density of the diskette does not match the density specified in the command, the DSD 440 will report a density error to the device handler. In response, the handler will retransmit the command specifying the correct density. The entire operation is automatic, thus freeing you from manually checking each diskette to determine its density.

#### FORMATTING DOUBLE DENSITY DISKETTES

##### CAUTION

Changing the density of a diskette results in lost data.  
Be sure to change the density only on blank diskettes or  
diskettes containing unwanted data.

Before you can generate a double density system or file storage diskette, you must generate diskettes with double density data address marks. These data address marks are used by the controller to distinguish single density diskettes from double density diskettes.

There are three methods to accomplish this. If you have no preformatted IBM 3740 single density diskettes then you can use the FRD440 program as described in Chapter 5. Use the FORMAT (or XFORMAT) and SET MEDIA commands of this program to generate single or double density formatted diskettes.

You can also use the "HYPERDIAGNOSTIC" drive utility routines to generate formatted diskettes from blanks. Use the "Write-Single-Density-IBM-Format" command and "Set Media Density" command. These drive utility routines are described in more detail in Chapter 5.

The third method requires a preformatted IBM 3740 single density diskette. A standard IBM 3740 single density diskette can be turned into a DEC double density diskette if you have the RT-11 utility program called FORMAT.SAV. If this file is not on your system diskette, locate it on one of the diskettes included in your RT-11-VO3B (or later) distribution kit. Run the program by typing the command:

.R FORMAT                   <CR>or carriage return

If the program is not on your system diskette, but it is on the diskette loaded in Drive 1, type the command:

```
.RUN DY1: FORMAT <CR>
```

When this utility program is loaded in memory and is ready to accept a command, it types an asterisk (\*). At this point, remove any diskette in Drive 1 and replace it with the blank diskette which you want to convert to a double density format. The write protect notch should be covered. To run the program, type the following command to the asterisk prompt:

```
*DY1: /Y <CR>
```

When the asterisk prompt returns, you can insert another diskette in Drive 1 to generate additional double density diskettes. If you want to return to the monitor, simply type "CTRL C".

Before you can transfer files to the new double density diskettes, you must initialize the directory with the following command:

```
INIT DY1:/NOQ <CR>
```

You are now ready to transfer files to your new double density diskettes.

#### GENERATING AN RX02 COMPATIBLE RT-11 SYSTEM DISKETTE

To use the DSD 440 system while it is configured in Mode 2, either the DYMNSJ.SYS or the DYMNFB.SYS monitor program must be installed on the system diskette. These programs are already installed in RT-11-V03B or RT-11-V04 (the GJ013-CX and GJ013-AX RX02 distributions of the operating system). If the RT-11 version you have was intended to be bootstrap loaded from an RX01 flexible disk, RK05 hard disk, or some device other than the RX02, the procedure below explains how to generate an RX02 compatible diskette.

If you have a version of RT-11 intended for the single density DSD 110, DSD 210 or DEC RX01 systems, you should reconfigure the DSD 440 to the RX01 compatible mode. The procedure for reconfiguring the DSD 440 is in Chapter 2.

The RT-11 operating system diskette and the operating mode of the DSD 440 must be compatible because different system monitors are used for single and double density operation. The DX monitor is used by RT-11 for RX01 mode. The DY monitor is used for RX02 mode.

The first step is to bootstrap load your single density version of RT-11. This should be done on drive 0 of the DSD 440. Once the operating system is loaded, locate a file named DYMNSJ.SYS if you normally use a single job monitor, or a file name DYMNFB.SYS if you use a foreground/background monitor. This file might be on your present system diskette, or it might be on another diskette included in your RT-11 distribution kit.

Find an otherwise unused diskette that is formatted for IBM 3740 single density operation. Ensure that it is write-enabled as described in Appendix F. Insert this diskette in Drive 1 and initialize the directory of that diskette by typing the following command to the RT-11 monitor:

.INIT DX1:/NOQ                   <CR> or Carriage Return

Copy all the files on the Drive 0 diskette (system diskette) to the Drive 1 diskette that will be needed on your new RX02 compatible system diskette. A command which will accomplish this and interrogate you about each file is shown below:

.COPY/SYS/QU DX0:\*. \* DX1:           <CR>

As each file name is typed by the computer, type a "Y" if that particular file should be copied to the diskette in Drive 1.

The RX02 compatible monitor file (DYMNSJ.SYS OR DYMNFB.SYS) should be included if it was found on your Drive 0 (system) diskette. Be sure that the file DXMNSJ.SYS or DXMNFB.SYS is included in the files that you copy. If these files were NOT located on the Drive 0 (system) diskette, then leave sufficient space on the Drive 1 diskette to accommodate one or both of these files. The DYMNSJ.SYS single job monitor file requires at least 63 blocks. The DYMNFB.SYS foreground/background monitor file requires at least 74 blocks.

Copy the DX bootstrap onto the Drive 1 diskette using the following command:

.COPY/BOOT DX1:DXMNSJ.SYS DX1: A <CR>    (Single Job Monitor)  
.COPY/BOOT DX1:DYMNFB.SYS DX1:A <CR>    (F/B Monitor)

Remove the distribution diskette presently in Drive 0 and set it aside. Move the diskette currently in Drive 1 over to Drive 0 and re-boot the system.

If the DYMNSJ.SYS and DYMNFB.SYS files were NOT located on the original Drive 0 system diskette, copy one or both of these new files onto the new Drive 0 diskette. This is done by placing the diskette which contained these files in Drive 1 and by using one or both of the following commands:

.COPY/SYS DX1:DYMNSJ.SYS DX0:DYMNSJ.SYS    <CR>

or

.COPY/SYS DX1:DYMNFB.SYS DX0:DYMNFB.SYS    <CR>

The last step is to install the bootstrap on the diskette you have placed in Drive 0 so that when you bootstrap, the RX02 compatible monitor (DY-monitor) is loaded instead of the RX01 compatible monitor (DX-monitor). To do this, type the following commands:

.COPY/BOOT DX0:DYMNSJ.SYS DX0:A            <CR>  
          (for the single job monitor)

or

.COPY/BOOT DX0:DYMNFB.SYS DX0:A            <CR>  
          (for the foreground/background monitor)

You now have a single density RX02 bootable diskette which will no longer boot on RX01 compatible hardware.

If you changed your DSD 440 to be RX01 compatible so you could perform the steps just described, change it back to RX02 (Mode 2) compatibility.

Bootstrap the new single density system diskette on the RX02 compatible DSD 440 and delete the RX01 compatible monitor (DX-monitor). This can be done by typing the following command:

```
.DELETE/SYS/NOQ DXMNSJ.SYS      <CR>  
    (for single job monitor)
```

or

```
.DELETE/SYS/NOQ DXMNFB.SYS      <CR>  
    (for foreground/background monitor)
```

The diskette can then be squished and re-booted by typing the following command:

```
.SQ/NOQ DY0:                    <CR>
```

Now you can generate a double density version of this single density (but RX02-compatible) diskette.

The first step in generating a double density system diskette, is to ensure you have a DEC double density diskette available. If you do not have such a diskette, refer to the previous section in this chapter "Formatting Double Density Diskettes".

Insert the double density formatted diskette generated initially into Drive 1. Type the following commands:

```
.COPY/SYS DY0:*. * DY1:         <CR>
```

and

```
COPY/BOOT DY1:DYMNSJ.SYS DY1:A  <CR>  
    (for single job monitor)
```

or

```
COPY/BOOT DY1:DYMNFB.SYS DY1:A  <CR>  
    (for foreground/background monitor)
```

When these commands are accomplished, you have completed the generation of a double density RX02 compatible system diskette. Place your new double density system in Drive 0 and reboot the RT-11 system. You should notice added space which is indicated by the number of free blocks on your system diskettes.

## SETTING MEDIA DENSITY ON THE PDP-8

Diskettes are usually supplied in single density format. They may be set to DEC double density with the SET MEDIA DENSITY command. There are two programs available which perform this function. DEC supplies a program called RXCOPY which is included with the OS/8 extensions. This program supports both single and double density set media density functions. Data Systems Design provides a program on the diagnostic

diskette called DSDMFT.SV which allows formatting either drive to either single or double density. One of these programs must be run before a diskette can be used in double density.

The following procedure explains how to run "DSDFMT.SV. The user types the quoted text and the system responds with the unquoted text.

```
."R DSDFMT "<CR>"
DSD FLOPPY DISK MEDIA DENSITY FORMATTER V1.1
  CTRL C RETURNS TO OS/8
  CTRL R RESTARTS DSDFMT
ENTER DRIVE (0 or 1): "1"
DRIVE 1 SELECTED <CURRENT>DENSITY
```

#### NOTE

At this time the controller must be configured for RX02 Mode. Do this by switching switch 4 on the controller board from open (RX01) to closed (RX02). DSDMFT will give the error message CONTROLLER NOT IN RX02 MODE and restart until this is done.

```
ENTER DENSITY (S OR D): "D"
DOUBLE DENSITY SELECTED
```

Control C may be typed at any time to return to OS/8.

#### GENERATING A DOUBLE DENSITY RX02 MODE OS/8 SYSTEM DISKETTE

This chapter describes the procedure for creating an OS/8 system diskette with full RX02 mode support. During this procedure the operating mode of the DSD 440 will be changed by toggling switch number 4 located on the controller board either to the closed position for RX02 mode or open position for RX01 mode operation (refer to Chapter 2, "Configuring the DSD 440 for RX01 Mode or RX02 Mode Operation"). Initially, the 440 will have to be booted using one of the bootstrap procedures explained in "Appendix H". After the initial boot, the 440 may be booted using the RXBOOT.SV program provided on the diagnostic diskette.

#### NOTE

1. If your system has less than 12K of memory, you will not be able to create a double density system diskette. However, you may still generate a single density RX02 bootable system diskette which will allow double or single density non-system diskette operation. Your processor's memory may be increased by adding core or relatively inexpensive CMOS memory.

2. You must have OS/8 version 3D or later to operate the 440 in RX02 mode. Once you have booted OS/8, if you cannot set today's date, you must obtain the version 3D updates from DEC\*.

The following outline details the steps involved in creating RX02 system diskettes. All "\$", "\*", and "." which appear at the beginning of the line are produced by the computer. All the text enclosed in "Quotes" must be typed in.

1. Make a copy of your master OS/8 diskette.
  - a. Put the DSD 440 in RX01 mode (refer to Chapter 2).
  - b. Insert the diagnostic diskette in drive 0.
  - c. Boot the diagnostic diskette (refer to Appendix H). If the processor halts while booting, it is because the diagnostic diskette is write protected. Just press "RUN" or "CONT" on the front panel. If you do not have a front panel, write enable the diagnostic diskette (refer to Appendix F).
  - d. When the monitor prompt appears on the terminal, type "R RXCOPY" (CR)  
The program will ask: COPY UNIT 0 to UNIT 1, YES OR NO?
  - e. Remove the diagnostic diskette from drive 0 and insert your master OS/8 diskette in drive 0.
  - f. Insert a blank single density, write-enabled diskette in drive 1,  
Type: "Y" (CR)  
If the system hangs, you have installed a Double Density diskette. Start over at step 1a. When the copy is complete, the program prints, DONE.
  - g. If you have installed the boot prompts, proceed to step 1h. If you boot from the front panel, here is a DSD time saver. Insert the diagnostic diskette in drive 0. Press the HALT switch and the INIT switch. Load address "7605" on the front panel and press CONT or RUN.  
This will restore the monitor.  
Type: "R RXBOOT" (CR)  
The processor will halt at location 72. Insert the copy of OS/8 just made in drive 1 into drive 0. This will be the OS/8 diskette referred to during the remainder of this chapter. Now press RUN or CONT. GO to step 1i.
  - h. Remove the copy in drive 1 and insert it in drive 0. This copy will be the OS/8 diskette referred to during the remainder of this chapter. Boot the OS/8 diskette.
  - i. Check the number of free blocks available.  
Type: ."R DIRECT" (CR)  
      \*"/E=2" (CR)  
      \*(CNTRL C)  
This will display all the empty file space. You will need 45 continuous free blocks. If you have them proceed to step 2. If you have 45 blocks, but they are not continuous, proceed to step 1k.

\* Trade Mark Digital Equipment Corporation



j. Delete enough files to make 45 free blocks.  
 Type: . "R PIP" (CR)  
       \*"SYS:FILENAME.EXT/D <" (CR)  
       \*(CNTRL C)

k. Squeeze files.  
 Type: . "R PIP" (CR)  
       \*"SYS:<SYS:/S" (CR)  
       ARE YOU SURE? "Y"  
       YES  
       \*(CNTRL C)

Go back to step 1i.

2. Installing the new device names and handlers.

a. Insert the diagnostic diskette in drive 1. PDP-8 users who have installed the boot prompts as described in Appendix H may proceed to step 2b. All others,  
 Type: . "R PIP" (CR)  
       \*"SYS:RXBOOT.SV<RXA1:RXBOOT.SV/I" (CR)  
       \*(CNTRL C)

b. Build the new system.

12K OR MORE

8K

Type: . "RUN SYS:BUILD" (CR)	SAME
\$"DELETE SYS" (CR)	SAME
\$"UNLOAD RX8E" (CR)	SAME
\$"UNLOAD RX01" (CR)	SAME
\$"LOAD RXA1:RXS44S" (CR) \$"LOAD RXA1:XRX01S" (CR)	
\$"LOAD RXA1:RXN440" (CR)	SAME
\$"INSERT RX1S SYS" (CR) \$"INSERT S210 SYS" (CR)	
\$"INSERT RX02 DY0" (CR)	SAME
\$"INSERT RX02 DY1" (CR)	SAME
\$"INSERT RX02 DX0" (CR)	SAME
\$"INSERT RX02 DX1" (CR)	SAME
\$"DSK=SYS" (CR)	SAME

c. Type: \$"PR" (CR)  
 This lists all installed devices and puts an asterisk before all devices inserted as active.

d. Save the build version.  
 Type: \$(CNTRL C)  
 (12K) "SAVE SYS:BLDX1S" (CR)  
 (8K) "SAVE SYS BLDX08" (CR)

e. Install this set of active build handlers into OS/8 by typing:  
 (12K) "RUN SYS:BLDX1S" (CR)     \$"BOOT" (CR)  
 (8K) "RUN SYS:BLDX08" (CR)     \$"BOOT" (CR)  
 The system will ask, WRITE ZERO DIRECTORY:  
 Type: "N" (CR)

- f. Patch the device length table. Note this may already have been done in PIP V14A or later.  
 Type: `."GET SYS:PIP.SV" (CR)`  
       `."OD" (CR)`  
       `"13632/"0000 "6044" (CR)`  
       `(CNTRL C)`  
       `."SAVE SYS:PIP.SV" (CR)`

You now have an RX02 bootable OS/8 diskette in drive 0.

- g. Put the DSD 440 in RX02 mode. If boot prompts are installed reboot and proceed to step 3, otherwise,  
 Type: `."R RXBOOT" (CR)`  
 When processor halts at location 72, press RUN or CONT. You are now booted in RX02 compatible mode.

### 3. Creating a double density diskette.

- a. Insert the diagnostic diskette in drive 1.  
 Type: `."RU DY1:DSDFMT" (CR)`  
 The program will respond by reporting its version number and active control functions. Next the program will query by typing:  
       ENTER DRIVE # (0 OR 1):

- b. Remove diagnostic diskette. Insert a blank write-enabled diskette in drive 1.  
 Type: `"1"`  
 The program will report the current density of the diskette in drive 1 and ask you to select a density.

WARNING: This following entry will destroy all data on the diskette being formatted!  
 Type: `"D"`

- c. When the operation is complete, the program will start over. When you have set the density on as many diskettes as you desire,  
 Type: `(CNTRL C)`  
 This will return you to OS/8.

If you have only 8K of memory your conversion is complete. You can now operate in RX02 mode with your new OS/8 single density system diskette and operate double or single density non-system diskettes. See the note on page 4-9.

Users with 12K or more of memory can convert their single density OS/8 system diskette to double density by proceeding with the following steps.

### 4. Transferring files from the single density OS/8 diskette to a double density diskette.

- a. Insert a double density diskette created in step 3 into drive 1.
- b. Type: `."R PIP" (CR)`  
       `*"DY1:<SYS:/Y/Z" (CR)`  
 Copy the system area and initialize the directory.

- c. **\*"DY1: SYS:/S/O" (CR)** Squeeze the system disk onto the double density disk. You now have a double density, RX02 mode, bootable, OS/8 diskette in drive 1. The RXS44S system handler installed in step 2 will work in both RX01 mode and RX02 mode; however, when accessing the diskette it assumes the diskettes are single density. If the diskette is double density, an error is generated and an automatic retry in double density is attempted. This process results in slightly longer file access time on double density diskettes. If the 440 is to be operated in RX02 mode only with a double density system diskette, it is recommended that the RXS44D handler now be installed. This handler works in RX02 mode only and differs from the RXS44S handler in that it assumes double density first, then on error attempts single density. The following step describes how to install it.

5. Installing the RXS44D handler.

- a. Put the double density OS/8 diskette created in step 4 in drive 0 and do a complete boot.
- b. Put the DSD diagnostic disk into drive 1. Insert the double density handler RXS44D.BN into the system as follows:

```
Type: ."RUN SYS:BLDX1S" (CR)
      $"UNLOAD RX1S" (CR)
      $"LOAD DY1:RXS44D.BN" (CR)
      $"INSERT RX2S SYS" (CR)
      $"DSK=SYS" (CR)
      $"PR" (CR)
      $"BOOT" (CR)
```

NOTE

If PIP version 12B or earlier is being used, the user must explicitly denote single or double density disks by DX0, DX1, or DY0,DY1 respectively while squeezing or zeroing. Otherwise erroneous device sizes will be installed in the directory. A squeeze with the correct device name will restore everything properly.

PROGRAMMING INTERFACE

The system interface with the DSD 440 varies according to the host computer type and the operational mode for which the system is configured. The characteristics of the DSD 440 operation are embedded in the controller. The controller utilizes five separate protocols to communicate with the interface module and host computer program.

The remainder of this chapter is organized by computer family and operational mode. MODES 1 (RX01 compatible) and 2 (RX02 compatible) of the LSI-11 and PDP-11 computers are discussed separately. Since the PDP-8 is not capable of DMA, MODE 1 operation and MODE 2 operation are not substantially different. For this reason, both operational modes of the PDP-8 interface are discussed together.

## NOTE

All address locations and numerical machine values are represented in octal format.

### DEC 11 FAMILY

The system interface with the DSD 440 is identical for the LSI-11 and PDP-11.

Data is transferred to and from the diskette in fixed length blocks called sectors. A sector contains 64 sixteen bit words when the system is being used in single density mode and 128 of these words in double density mode.

The programmer can direct the DSD 440 controller to perform several operations or tasks. Each of these tasks is used to facilitate the storage and retrieval of information on a diskette.

As an example, two operations are needed to move a sector of data from main memory to a particular sector on a diskette. The first operation is called FILL BUFFER. This is used to move the data from computer memory to a RAM buffer which is an internal part of the disk controller. The second operation is called WRITE SECTOR. This positions the read/write head of a flexible disk drive over the specified portion of the diskette, and writes the data stored in the controller's sector buffer on the diskette.

The programmer communicates his task requirements to the DSD 440 controller through two physical registers which are addressed as though they were in computer memory. The CONTROL and STATUS REGISTER is normally located at address 777170 octal. The DATA BUFFER REGISTER is normally located at address 777172 octal.

There are a total of seven "logical registers" that are mentioned throughout this chapter. These registers represent such information as data, controller status, track address and sector address. The programmer always reads and writes logical registers through the data buffer register, which is a physical register.

A task is initiated by writing a specific bit pattern to the control and status register. Each task is associated with a specific "protocol". A protocol is a set of rules which determine the parameters or data the computer should be passing through the data buffer register during the execution of a task.

For example, operations which move the read/write head in the disk drive require a track and sector address. The protocol for these functions is as follows:

- 1) The command is written to the control and status register.
- 2) The sector address is written to the data buffer register when the controller requests it.

- 3) The track address is written to the data buffer register when the controller requests it.

The DSD 440 operational modes influence the protocol associated with the various tasks. The main differences in these modes center on data transfer and storage characteristics. In Mode 1 programmed I/O is used exclusively for the transfer of both data and parameters between the computer and controller. In Mode 2, programmed I/O is used to transfer parameters, but DMA is used to transfer data between the controller and main memory.

In Mode 1 data is recorded on a diskette in single density only. In Mode 2, data is recorded on the diskette in either single or double density.

### MODE 1 (RX01 COMPATIBLE) OPERATION

The system assumes MODE 1 operation when the "RX01" switch (located on the controller module) is placed on the "1" position and by installing the "ENRX01" jumper on the PDP-11 or LSI-11 interface boards. Any program that runs successfully with the DSD 210, DSD 110, or the DEC RX-11 (or RXV-11) runs equally well on a DSD 440 system configured for operation in MODE 1.

#### Peripheral Device Registers

Programs communicate with the DSD 440 through two peripheral device registers. They are as follows:

<u>NAME</u>	<u>DESCRIPTION</u>	<u>OCTAL LOCATION</u>
RXCS	Command and status register	177170
RXDB	Data buffer register	177172

Peripheral device registers reside in the top 4K words of the DEC-11 family computers' memory address space. They are addressed as memory and any instruction that can operate on a memory location can operate on a peripheral device register in the same way. For information explaining how to assign non-standard bus addresses to these registers, see the section in Chapter 2 that describes installation of the interface module and cable.

#### Command and Status Register (RXCS)

Writing bit patterns to this physical register controls the DSD 440. The format for this register is shown in Table 4-1. The RXCS register also provides important status information and error indications when read by the program.

**Table 4-1. Mode 1 Command and Status Register**

Format for RXCS Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ER	IN							TR	IE	DN	UN1	FN	FN	FN	EX

BIT	MEANING
15	ER - Error detected, cleared by INITIALIZE or new command. Read Only
14	IN - INITIALIZE the DSD 440. The DONE flag will be negated, the controller will self-test, drive 1 will seek to track 0, drive 0 will seek to track 0. A READ SECTOR operation on drive 0, track 1, sector 1 will occur if a diskette is in place; the ERROR AND STATUS REGISTER will be set to 0, the INITIALIZE DONE bit will be set in the ERROR AND STATUS REGISTER, and if drive 0 is ready, then the DRIVE READY bit will be set in the ERROR AND STATUS REGISTER. The INITIALIZE bit takes precedence over all other bits in the RXCS register.
13-8	UNUSED
7	TR - TRANSFER REQUEST indicates to the program that the DATA BUFFER REGISTER has been emptied and needs loading or is loaded and needs emptying. Read only.
6	IE - INTERRUPT ENABLE permits an interrupt to occur when the DONE flag is set. It is a read/write bit.
5	DN - DONE flag indicates the completion of an operation. The DONE flag is a read only bit.
5-4	UN2 UN1 - Diskette drive unit select bits. The binary encoding of these bits selects drive 0-3. Drive selection only occurs if a drive related function is executed. A point of incompatibility exists when a triple or quad drive system is configured. DEC bootstraps assume that bit 5 is a "read only" bit, so they write into it with impunity. As a result, drive 2 is selected by mistake during bootstrapping. In systems configured for single or dual drive operation, bit 5 can be written into with impunity.
3-1	FN - FUNCTION SELECT  0 = FILL SECTOR BUFFER from memory 1 = EMPTY SECTOR BUFFER into memory 2 = WRITE SECTOR BUFFER TO disk 3 = READ SECTOR from disk to SECTOR BUFFER 4 = Not used 5 = READ STATUS (RXDB - RXES) 6 = Write sector with detected data address mark 7 = READ ERROR REGISTER (RXDB - RXER)  Function select bits are write only.
0	EX - Execute, when set, causes the function coded in RXCS bits 3-1 to be executed.

## Data Register (RXDB)

The RXDB, is physically a shift register that provides the communication link between the host processor and the DSD 440 system. The logical information passed through this physical register is based upon a predetermined protocol which is defined in the Mode 1 protocols section of in this chapter.

If the DSD 440 is not in the process of executing a command, the RXDB can be written without risk. However, during the execution of an instruction, the RXDB register will only provide or accept information (according to the RXDB protocol) when the TRANSFER REQUEST flag is set.

### NOTE

Data may be lost if the correct protocol is not followed.  
Only RXDB bits 0-7 are accepted by the controller. Bits 8 through 15 are ignored.

The following descriptions explain the various logical register formats of the physical Data Buffer Register or RXDB.

### Data Buffer Register

The data buffer register is a physical shift register used to transfer data to and from the controller data buffer. All information is transferred as a byte through bits 0-7 of the RXDB.

### Disk Track Address

At the proper time during commands requiring a track number (e.g., write sector, read sector), the track number is written to the physical RXDB register as if it were a logical register. This is the TRACK ADDRESS REGISTER (RXTA = 777172). Track numbers from 0-76 (decimal) are valid.

### Disk Sector Address

At the proper time during commands requiring a sector address (e.g., write sector, read sector) the sector address is written to the physical RXDB register as if it were a logical register. This is the SECTOR ADDRESS REGISTER (RXSA = 777172). Sector addresses from 1-26 (decimal) are valid. The controller microprocessor masks bits 6 and 7 of the RXSA to zeroes.

### SYSTEM ERROR AND STATUS REGISTER (RXES)

The RXES is a logical register that is implemented using the physical RXDB shift register. It provides status and error information about the drive that has been selected in bit 4 of the physical RXCS register. At the completion of a command, the controller places the RXES register into the data buffer register (RXDB = 777172) so that the host processor can check the status of the most recent operation.

**BIT 7 - DRV RDY - Drive Ready**

This bit, when set, indicates that the selected drive has a diskette installed correctly. The Drive Ready bit is only valid immediately following the Read Status function. The bit is valid for drive 0 immediately following an initialize.

**BIT 6 - DD - Deleted Data**

This bit indicates that a deleted data address mark was found during the last Read Sector operation or that the last command was Write Deleted Data Sector.

**BIT 5 - DRV DEN - Drive Density**

This bit indicates the density of the diskette installed in the selected drive. When asserted, double density is indicated. This bit is updated during read or write sector operations.

**BIT 4 - DEN ERR - Density Error**

This bit indicates that during a read or write sector operation, the controller found that the density of the selected diskette did not match the density given in the command. The operation is terminated and the error and done bits are set.

**BIT 3 - WP - Write Protect. (RX01 Mode only)**

This bit is set whenever a write is attempted on a write-protected diskette. This RXDB bit along with the ERROR and DONE bits of the RXCS is set when the controller/drive subsystem loses power, or the IBUS cable is disconnected.

**BIT 2 - ID - Initialize Done**

This bit indicates that the controller/drive subsystem has just completed an initialization sequence. This sequence may have been started by a power failure, bus INIT, or programmed INIT.

**BIT 1 - PAR - Parity Error**

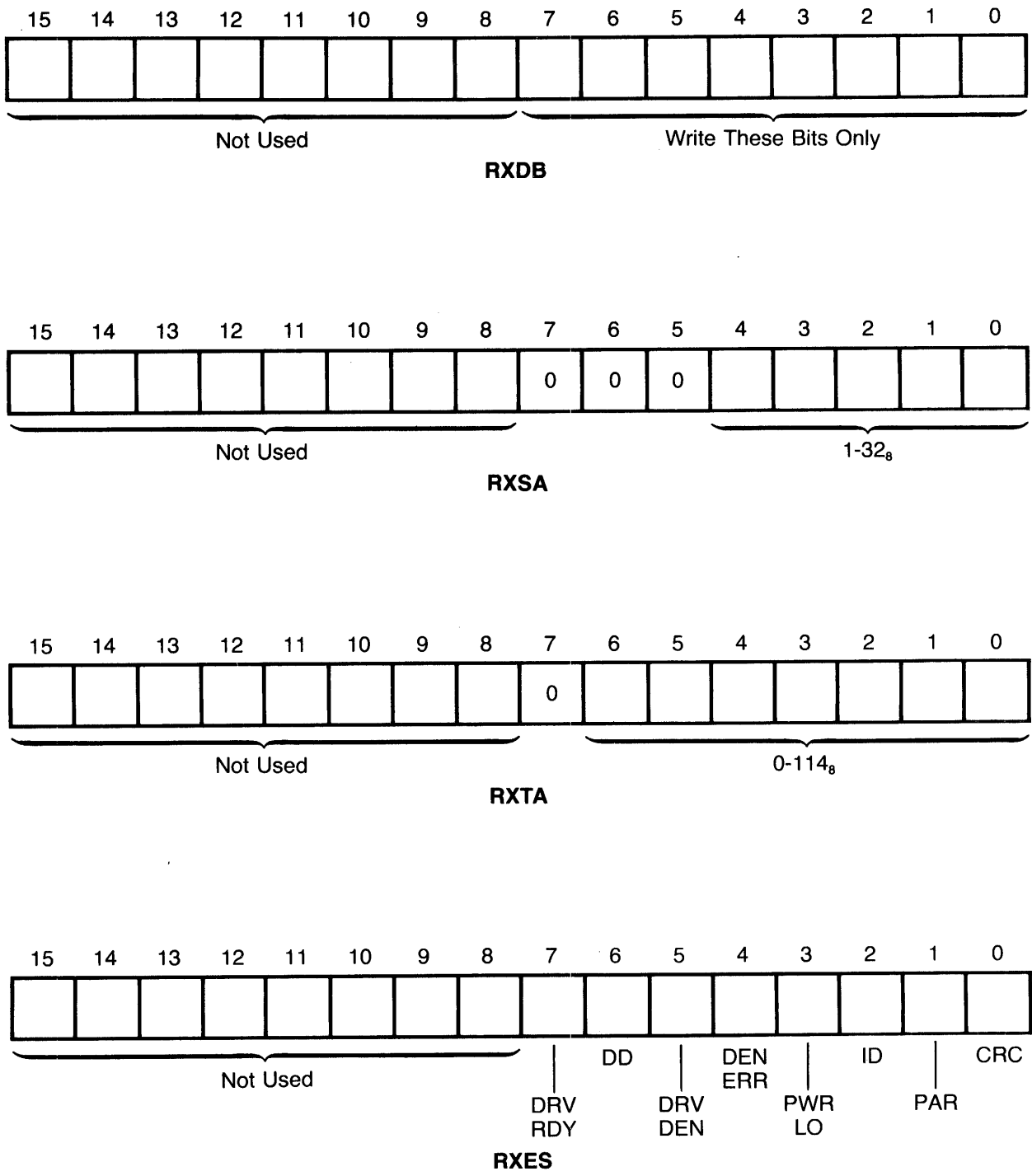
This bit indicates that a parity error was detected when a command or parameter was being shifted from the interface to the controller/drive subsystem. The operation is terminated; the ERROR and DONE bits are set.

**BIT 0 - CRC - CRC Error**

This bit indicates that a CRC (Cyclic Redundancy Check) Error was detected during the last Read Sector operation. The operation is terminated; the ERROR and DONE bits are set.

The bit layout of this register is shown in Figure 4-1.





**Figure 4-1. Register Formats**

## Mode 1 Protocols

Protocols are required in the DSD 440 because the computer interface module and the DSD 440 controller communicate mostly through a single physical I/O register (RXDB). Because of this constraint, the controller must identify parameters being passed to it by the order in which they are transmitted through the register link.

The following sections describe the proper protocol for each of the possible commands that can be sent to the controller. Failure to adhere to the correct protocol will result in lost or incorrect data.

### **FILL SECTOR BUFFER (000)**

The FILL SECTOR BUFFER command is used to fill a storage buffer inside the DSD 440 with 128 eight bit bytes of data from the host processor. Other functions can later be used to either write that data to the diskette, or transfer it back to the processor.

When the FILL SECTOR BUFFER command is given, the DSD 440 responds by clearing the DONE flag, RXCS bit 5. The controller then requests the first byte of data by setting the TRANSFER REQUEST flag, RXCS bit 7. At this time, one byte of data should be written into the lower eight bits of the RXDB register by the host processor. When the processor writes a byte into the RXDB register, the TRANSFER REQUEST flag is cleared.

When the TRANSFER REQUEST flag is again set by the controller, another byte of data is transferred to the RXDB register. This process is repeated until a total of 128 bytes have been transferred. When the controller has the 128 bytes needed to fill the buffer, TRANSFER REQUEST is left clear, and the DONE flag, RXCS bit 5 will be set. If the INTERRUPT ENABLE bit (RXCS bit 6), is set, an interrupt request will occur when the DONE flag is set.

### **NOTES**

- 1) Data will not be accepted unless the TRANSFER REQUEST flag is set.
- 2) If the ERROR flag, RXCS bit 15, is set, the specific error must be obtained from the RXER (see READ ERROR REGISTER section).
- 3) The controller will ignore all data sent after byte 128.
- 4) Since the FILL BUFFER command is not associated with any one drive, RXCS bit 4 does not affect this function.

Interrupts are generated by the logical "AND" of DONE and INTERRUPT ENABLE. If the DONE bit is set the first time you set the interrupt enable bit you will get a spurious interrupt.

#### EMPTY SECTOR BUFFER (001)

The EMPTY SECTOR BUFFER function is used to transfer the contents of the sector buffer to the computer. The sector buffer is loaded from a previous FILL SECTOR BUFFER or READ SECTOR command.

When the EMPTY BUFFER command is given, the controller responds by clearing the DONE flag, RXCS bit 5. The controller then sets the TRANSFER REQUEST flag (RXCS bit 7), to indicate that a byte of data is available for reading. The data byte appears in the lower 8 bits of the RXDB data register.

When the host computer reads the byte, the TRANSFER REQUEST flag is cleared. The TRANSFER REQUEST flag is again set when the controller has placed another byte of data in the RXDB register. This process is continued until all 128 bytes have been transferred to the host computer. After the 128 bytes of data have been transferred, the TRANSFER REQUEST flag will remain cleared and the DONE flag will be set. An interrupt request will be generated if the INTERRUPT ENABLE bit was set when DONE became true.

#### NOTES

- 1) Data will not be accepted unless the TRANSFER REQUEST flag is set.
- 2) If the ERROR flag, RXCS bit 15, is set, the specific error must be obtained from the RXER (see READ ERROR REGISTER section).
- 3) The controller will ignore all data sent after byte 128.
- 4) Since the EMPTY BUFFER command is not associated with any one drive, RXCS bit 4 does not affect this function.
- 5) The EMPTY BUFFER function does not modify the contents of the sector buffer.

Interrupts are generated by the logical "AND" of DONE and INTERRUPT ENABLE. If the DONE bit is set the first time you set the interrupt enable bit you will get a spurious interrupt.

#### WRITE SECTOR (010)

The WRITE SECTOR function is used to transfer the contents of the sector buffer to a specified track and sector of the diskette. When the WRITE SECTOR command is given, the controller clears the logical RXES register and the DONE flag.

Next, the controller sets the TRANSFER REQUEST flag, RXCS register bit 7, to request a sector address. The program responds by writing the desired sector address (RXSA) into the data buffer register (RXDB = 777172). This clears the TRANSFER REQUEST flag. As soon as the controller shifts the sector address over the interface cable, it asserts TRANSFER REQUEST again. This time the program responds by writing the desired track address (RXTA) into the data buffer register. This clears the TRANSFER REQUEST flag.

After the track address is received, the controller commands the selected drive to seek to the right track and locate the right sector. TRANSFER REQUEST stays unasserted for the remainder of the function.

If the correct track and sector are found, the controller writes the 128 bytes of data from the sector buffer, plus two bytes of Cyclic Redundancy Check (CRC) onto the diskette. When this is finished, the controller completes the function by writing the RXES data to the data buffer register and setting the DONE flag. As in all functions, an interrupt request is generated if the interrupt enable bit (RXCS bit 6), was set when DONE became true.

If the controller is unable to locate the specified diskette track, the RXER is set to a 150. If the specified sector cannot be found within two diskette revolutions, the RXER is set to a 70. Both of these error conditions cause the function to be terminated. The ERROR flag, RXCS bit 15, and the DONE flag, RXCS bit 5 are asserted. As with the error-free termination, an interrupt request will be generated if the interrupt enable bit was set when the DONE flag became true.

#### NOTES

- 1) The contents of the sector buffer are not modified by the WRITE SECTOR function.
- 2) The contents of the sector buffer ARE modified as a result of a power failure and an initialize command. Be sure that valid data is written back into the sector buffer following either of these conditions. This is especially true before executing the WRITE SECTOR command.
- 3) If the sector number written into the RXSA is 152 (octal) the WRITE SECTOR function becomes a WRITE FORMAT TRACK function.

#### READ SECTOR (011)

The READ SECTOR function is used to locate a specified track and sector of a diskette and then transfer the contents of the data field into the controller's sector buffer.

When the READ SECTOR command is given, the controller clears the RXES register and the DONE flag. Next, the controller sets the TRANSFER REQUEST flag, RXCS bit 7, to request a sector address. The program responds by writing the desired sector address (RXSA) into the data buffer register (RXDB = 777172). This clears the TRANSFER REQUEST flag. As soon as the controller shifts the sector address over the interface cable,

it asserts TRANSFER REQUEST a second time. The program should respond by writing the desired track address (RXTA) into the data buffer register which clears the TRANSFER REQUEST flag.

After receiving the track address, the controller causes the selected drive to seek to the desired track and locate the desired sector. TRANSFER REQUEST is left reset for the remainder of this function. If the correct track and sector are located, the controller looks for a Data Address Mark (DAM) or a deleted Data Address Mark (DDAM). When a valid mark is found, this marks the beginning of the 128 byte data field on the diskette.

At that point, the following 128 bytes are read from the diskette and stored in the controller data buffer. The two CRC bytes are read immediately after the data field. An error-free read is indicated if the address mark, 128 bytes of data, and two bytes of CRC produce a zero residue when passed sequentially through the CRC checker hardware circuits. As soon as the data is available in the buffer, the controller terminates the function by writing the RXES to the data buffer register and setting the DONE flag. An interrupt request will be generated if the interrupt enable bit, RXCS bit 6, is set when DONE was asserted.

If the deleted data address mark is detected, the controller sets the deleted data flag. This flag appears in the ERROR/STATUS register (RXES bit 6). If a CRC error is detected, the controller sets RXES bit 0 and the ERROR flag (RXCS bit 15) as an indication. Seek errors and missing sector errors are reported just as in the WRITE SECTOR function.

#### READ STATUS (101)

The READ STATUS command is used to determine the current status of the drive selected by RXCS bit 4. The status information passed back indicates if the drive is ready.

When the command is issued, the DONE flag is cleared. The controller checks to see that the selected drive's door is closed, a diskette is inserted, and that the diskette is up to speed.

The speed is determined by measuring the amount of time between successive index pulses. Since this measurement takes an average of 250 milliseconds, excessive use of the READ STATUS function will cause reduced throughput.

If the drive is ready, the controller sets bit 7 (DRV RDY) of the RXES. The controller terminates this function by shifting the RXES over to the RXDB and setting the DONE flag. An interrupt request will be generated if the interrupt enable bit (RXCS bit 6), was set when DONE became true.

#### WRITE DELETED DATA SECTOR (110)

This function performs the same task as WRITE SECTOR. The difference is that this command writes a deleted data address mark just before the data field. The standard WRITE SECTOR function writes a regular data address mark. When a sector which was written with a deleted data address mark is read, bit 6 of the RXES is set to reflect this.

## READ ERROR REGISTER (111)

When a command terminates because of an error condition, RXCS ERROR (bit 15 of the command and status register), is set. Under these conditions, a code is available in the RXER which can be used to identify the specific error. The READ ERROR REGISTER command is used to access that code. Table 4-2 shows the RXER code meanings.

When the READ ERROR REGISTER command is initiated, the DONE flag is cleared. The controller moves the logical RXER register into the physical data buffer register (RXDB), and signals completion of the transfer by asserting the DONE flag.

### NOTE

This is the only command that DOES NOT terminate with RXES placed in the RXDB. The information contained in the RXER should be read immediately after the ERROR flag (RXCS bit 15) is set. Subsequent commands or an INITIALIZE operation clear the RXER.

### Power Fail or Initialize Command

When a power failure occurs or DC power to the DSD 440 is interrupted, the controller gradually drains the filter capacitors and stops executing microcode. The DONE and ERROR bits are set in the command and status register, (RXCS). Bit 3 of the RXDB (PWR LO) is set.

When power is returned, the DSD 440 controller initiates the following sequence of events:

- 1) DONE is cleared.
- 2) Controller executes the hardware self-tests.
- 3) All drives homed to track 00.
- 4) RXES is cleared of all active error bits.
- 5) The controller reads sector 1, track 1 of unit 0 into buffer if a diskette is loaded in the drive.
- 6) Bit 2 of RXES (INITIALIZE DONE) is set.
- 7) Bit 7 (DRIVE READY) of RXES is updated according to the status of drive 0.
- 8) RXCS bit 5 (DONE) is set.

At this point, the DSD 440 is ready for operation.

Table 4-2. Error Register Codes (RXER)

<u>OCTAL CODE</u>	<u>DESCRIPTION</u>
000	No error
010	No drive 0 or drive 0 failed to find track 0 on INIT
020	No drive 1 when DIP switch indicates there should be a drive 1, or drive 1 failed to find track 0 on INIT
030	Track 0 found while stepping in on INITIALIZE
040	Track address passed to controller was invalid (>76)
050	Track 0 found before desired track while stepping
070	Requested sector not found in two revolutions
100	Write protect violation
110	No read data signal present
120	No preamble found
130	Preamble found, but no DAM or IDAM within window
140	CRC error on what appeared to be a header
150	Address in good header did not match desired track
160	Too many tries for an ID address mark
170	Data address mark not found in allotted time
200	CRC error on data field; RXES bit 0 also set
210	Parity error on interface cable; RXES bit 1 also set
220	Read/Write controller failed maintenance mode test
240	Density error
260	Indeterminate density
270	Read/Write controller write-format failure
320	Read/Write controller write circuit failure
330	Read/Write controller timed out on reset
340	Master controller out of sync with RD/WRT controller
360	Drive not ready during format command
370	AC power low caused abort of write activity

For additional information on the ERROR REGISTER codes shown in Table 4-2, refer to Appendix I.

## Diskette Formatting

The DSD 440 system can be used to write-format diskettes. This involves rewriting all of the header and data fields on a specified track. The entire track is always written.

**CAUTION:** Formatting will cause loss of all data on the formatted tracks.

The protocol to write-format a track is as follows:

- 1) The user program issues the WRITE SECTOR function code (010) to the controller.
- 2) When the controller requests a sector address by setting the TRANSFER REQUEST flag, the number 152 is written into the data buffer (RXDB).
- 3) When the controller sees this number, it branches to a special section of microcode which handles track formatting.
- 4) The controller sets the TRANSFER REQUEST flag and the program specifies the track to be formatted by writing a valid track address (RXTA) into the data buffer register.
- 5) The controller raises the TRANSFER REQUEST flag 26 more times. Each time the user program sees the TRANSFER REQUEST flag, another valid and unique sector address is written to the data buffer register.

### NOTE

The controller does NOT check these sector addresses for uniqueness or for being in the range 1-26. Unlike the READ and WRITE SECTOR functions, bits 6 and 7 of the sector addresses passed over to the controller are NOT masked to zeroes.

- 6) After receiving the 26th sector address, the controller seeks the heads to the specified track and awaits the index pulse.
- 7) Starting at the index mark, the controller writes the entire track. The sector addresses written in the sector headers are in the order that they were passed to the controller. This enables a wide variety of hard sector interleaving techniques to be implemented easily.

When a track has been fully written, the DONE flag is set and an interrupt is generated if the interrupt enable bit was set.

## Typical Sequences of Operations

The programming examples shown in Tables 4-3 and 4-4 illustrate how to write routines which successfully manipulate the DSD 440 System.



## READ STATUS

Status information is usually needed to determine the status of a drive or the cause of an error. To determine drive related status (DRIVE READY), the READ STATUS command should be used. When the ERROR flag (RXCS bit 15), is set following a function, the RXES should be read first. Remember that the logical RXES register is left in the physical RXDB register following all functions EXCEPT the READ ERROR REGISTER function.

As shown in Figure 4-1, the RXES has error bits for CRC ERROR, PARITY ERROR, POWER LOW, and DENSITY ERROR. If no error bits are set in the RXES, the definitive error code can be obtained using the READ ERROR REGISTER command. The code interpretations are shown in Table 4-2.

## COMMON PROGRAMMING MISTAKES

This section illustrates common programming mistakes that can cause data loss and/or error indications.

- 1) An illegal track or sector address is sent to the controller.
  - a) Valid sectors are 1-26. (decimal)  
(There is no sector 0)
  - b) Valid tracks are 0-76. (decimal)
- 2) The READ STATUS command requires up to two revolutions of the disk to complete. To avoid excessive delays, use this command only when necessary.
- 3) After reading or writing, the INITIALIZE DONE bit (RXES bit 2) may be checked for indication of power failure. A short power outage causes DONE to set without any error indication even though invalid data may have been read or written.
- 4) The drive select bit, RXCS bit 4 is not scanned by the controller during FILL BUFFER and EMPTY BUFFER functions.
- 5) A two-way-sector interleave should be used for maximal throughput when using a dma interface. A three way interleave is desirable when using programmed transfer.

A FILL BUFFER command usually precedes a WRITE SECTOR command. Similarly, a READ SECTOR command precedes an EMPTY BUFFER command.

## Interrupts

An interrupt is requested by the interface module whenever the INTERRUPT ENABLE and DONE bits of the physical command and status register, RXCS, both become set. Only a single interrupt can occur per request. The standard interrupt vector address location is 264.

Table 4-3. Fill / Empty RX01 Sector Buffer Example

```

177170          RXCS=177170 ;CONTROL AND STATUS REGISTER
177172          RXDB=177172 ;DATA BUFFER REGISTER
;FILL RX01 BUFFER EXAMPLE (128 BYTES ALWAYS)
000000 012737 RX1FIL: MOV #1, @#RXCS ;LOAD FILL BUFFER COMMAND
000001
177170
000006 012704          MOV #BUFADR, R4 ;UNIT SELECT IS IGNORED
000162          ;SET BUFFER ADDRESS INTO REGISTER
000012 012700          MOV #200, R0 ;SET BYTE COUNT INTO REGISTER
000200
000016 105737 2$      TSTB @#RXCS ;WAIT FOR TRANSFER REQUEST
177170
000022 100405          BMI 3$ ;MUST BE UP BEFORE DOING DATA XFER
000024 032737 BIT #40, @#RXCS ;CHECK FOR PREMATURE DONE (POWER GLITCH OR)
000400
177170
000032 001771          BEQ 2$ ;PARITY ERROR
000034 000410          BR FILERR
000036 112437 3$      MOVB (R4)+, @#RXDB ;LOAD A DATA BYTE (NOT WORD)
177172          ;WORD TRANSFER WORKS BUT ONLY LOW BYTE READ
000042 005300          DEC R0
000044 003364          BGT 2$ ;TRANSFER 128 BYTES
000046 032737 4$      BIT #40, @#RXCS ;DONE COMES UP AFTER LAST BYTE
000040
177170
000054 001374          BNE 4$ ;BUT WAIT BEFORE RETURNING
000056 005737 FILERR:TST @#RXCS ;ONLY POSSIBLE ERRORS ARE
177170          ;PARITY AND POWER DOWN DURING XFER
000062 000207          RTS PC ;NON EXISTANT MEMORY ERRORS WILL CAUSE TRAP
;SET BMI FLAG IF ERROR
000064          :EMPTY RX01 BUFFER EXAMPLE (128 BYTES ALWAYS)
012737 RX1EMP: MOV #3, @#RXCS ;LOAD EMPTY BUFFER COMMAND
000003
177170
000072 012704          MOV #BUFADR, R4 ;UNIT SELECT IS IGNORED
000162          ;PUT BUFFER ADDRESS INTO REGISTER
000076 012700          MOV #200, R0 ;128 BYTES MUST BE EMPTIED
000200
000102 105737 2$      TSTB @#RXCS ;BUT SOME CAN BE THROWN AWAY
177170          ;WAIT FOR TRANSFER REQUEST

```

Table 4-3. Fill / Empty RX01 Sector Buffer Example  
(Continued)

```

000106 100402          BMI 3$          ;SET - DO DATA XFER
000110 001774          BEQ 2$          ;NO OTHER FLAGS IN LOW BYTE BUT DONE FLAG
000112 000420          BR  EMPERR      ;IF DONE PREMATURELY PROBABLY POWER GLITCH
000114 113724 3$      MOVB  @#RXDB, (R4)+  READ A BYTE - ONLY LOW BYTE SIGNFICIAINT
177172
000120 005300          DEC  RO          ;FINISHED EMPTYING?
000122 003367          BGT  2$          ;NO - DO ANOTHER BYTE
000124 132737 4$      BITB  #240, #@RXCS  :THROW AWAY ANY EXTRA BYTES IF BYTE CNT NO
000240
177170
000132 001774          BEQ  4$          WAIT FOR DONE OR TRREQ
000134 100003          BPL  10$         ;DONE CAME UP
000136 113700          MOVB @#RXDB, RO  ;JUST LOAD RO, THEN TRY AGAIN
177172
000142 000770          BR   4$
000144 032737 10$     BIT   #40, @#RXCS
000040
177170
000152 001764          BEQ  4$          ;WAIT FOR DONE
000154 005737  EMPERR TST  @#RXCS  ;IF ERROR SET N FLAG FOR BMI ON RETURN
177170
000160 000207          RTS   PC
000162          BUFADR BLKB  128
000001          END

```

Table 4-4. Read / Write / Write Deleted Data RX01 Sector Example

```

177170          RXCS=177170          ;CONTROL AND STATUS REGISTER
177172          RXDB=177172          ;DATA BUFFER REGISTER
;
;READ RX01 SECTOR
01000 012700  READ: MOV #7, R0          ;BUILD READ SECTOR CMD IN R0
000007
01004 000405          BR SYNTAX
;
;WRITE RX01 SECTOR ROUTINE
01006 012700  WRITE: MOV #5, R0        ;BUILD WRITE SECTOR CMD IN R0
000005
01012 000402          BR SYNTAX
;
;WRITE DELETED DATA RX01 SECTOR
01014 012700  WRTDD: MOV #15, R0      ;BUILD WRITE DELETED DATA
000015
;
01020 005767  SYNTAX: TST UNIT          ;SECTOR COMMAND IN R0
000114          ;UNIT 0 OR UNIT 1?'
01024 001402          BEQ 1$
01026 052700          BIS #20,R0        ;SET UNIT 1
000020
01032 010037  1$:  MOV R0,@#RXCS        ;ISSUE COMMAND TO CONTROLLER
177170
01036 105737  2$:  TSTB @#RXCS          ;WAIT FOR TRANSFER REQUEST
177170
01042 100375          BPL 2$
01044 016737          MOV SECTOR,@#RXDB ;PASS SECTOR TO CONTROLLER
000064
177172
01052 105737  3$:  TSTB @#RXCS          ;WAIT FOR TRANSFER REQUEST
177170
01056 100375          BPL 3$
01060 016737          MOV TRACK,@#RXDB   ;PASS TRACK TO CONTROLLER
000052
177172
01066 032737  4$:  BIT #100040,@#RXCS  ;TEST FOR DONE AND ERROR
100040
177170
01074 001774          BEQ 4$
01076 100401          BMI ERFIN          ;ERROR BIT SET?
01100 000207          RTS PC              ;RETURN TO CALLING ROUTINE
;
01102 013767  ERFIN:MOV @#RXDB,GENSTT   ;SAVE GENERAL STATUS
177172
000032
01110 012737          MOV #17,@#RXCS    ;REQUEST DEFINITIVE STATUS
000017
177170

```

Table 4-4. Read / Write / Write Deleted Data RX01 Sector Example  
(continued)

```

01116 105737 5$: TSTB @#RXCS ;LOOK FOR DONE FLAG
      177170
01122 001775 ; BEQ 5$
01124 013700 MOV @#RXDB, RO ;LEAVE DEFINITIVE STATUS IN RO
      177172
01130 000261 SEC ;CARRY FLAG SET INDICATE ERROR
01132 000207 RTS PC ;RETURN TO CALLING ROUTINE
;
01134 000001 SECTOR: .WORD 1 ;DESIRED SECTOR ADDRESS
01136 000001 TRACK: .WORD 1 ;DESIRED TRACK ADDRESS
01140 000000 UNIT: .WORD 0 ;UNIT - 0=DRIVE 0, 1=DRIVE 1
01142 000000 GENSTT: .WORD 0 ;GENERAL STATUS VARIABLE

```

### MODE 2 (RX02 COMPATIBLE) OPERATION

The system assumes MODE 2 operation when rocker #4 on the controller module DIP-Switch is placed in the closed position. The system operates only according to MODE 2 protocol when connected to an interface module capable of DMA. Any program that runs with the DEC RX211 (or RXV211) runs successfully on a DSD 440 system configured for operation in MODE 2.

### Peripheral Device Registers

Programs communicate with the DSD 440 through two peripheral device registers:

<u>NAME</u>	<u>DESCRIPTION</u>	<u>OCTAL LOCATION</u>
RX2CS	Command and status register	177170
RX2DB	Data buffer register	177172

Peripheral device registers reside in the top 4K words of DEC-11 family computers' memory address space. They are addressed as memory and any instruction that operates on a memory location can operate on a peripheral device register in the same way.

### Command and Status Register (RX2CS)

Writing the bits of this physical register controls the DSD 440. The format for this register is shown in Figure 4-2. The RX2CS register also provides important status information and error indications when read by the user program.

BIT 15 - ER - Error detected, cleared by INITIALIZE or the issuance of a new command. Read Only bit.

BIT 14 - IN - INITIALIZE the DSD 440

The DONE flag is negated, the controller resets some internal variables, and then executes the self-test microcode. The disk drives are homed to track 0.

If the controller is configured in "NORMAL" mode, the controller reads track 1, sector 1 of the diskette in drive 0. When the READ SECTOR function is attempted, the INITIALIZE DONE bit in the error/status register is set. If there was a readable diskette in drive 0, the DRIVE READY bit is also set. If the diskette is in double density, then the drive density bit is set. The DONE flag is set when the controller has completed the Initialization sequence. The INITIALIZE bit takes precedence over all other bits in this register. Bit 14 is a Write Only bit.

**BIT 13 - A17 - Extended address bit 17**

This write only bit is asserted on UNIBUS or Q-BUS address line 17 when the DSD 440 is transferring data via direct memory access. This bit is cleared by an INITIALIZE. A17 will toggle if A01-A16 are all ones and the bus address register is incremented by the logic.

**BIT 12 - A16 - Extended address bit 16**

This write only bit is asserted on UNIBUS or Q BUS address line 16 when the DSD 440 is transferring data via direct memory access. This bit is cleared by an INITIALIZE. A16 will toggle if A01-A15 are all ones and the bus address register is incremented by the logic.

**BIT 11 - RX02 System identification bit**

This read only bit provides an easy way for software to differentiate RX01 systems from RX02 systems.

**BITS 10, 9 - RESERVED FOR POSSIBLE FUTURE USE**

**BIT 8 - DEN - Density of function**

This read/write bit specifies the density of the function encoded in bits 1-3. High density is specified when this bit is set.

**NOTE**

Even though the FILL BUFFER and EMPTY BUFFER functions do not involve magnetic media, a valid density bit is required so that the controller can evaluate the validity of the word count parameter.

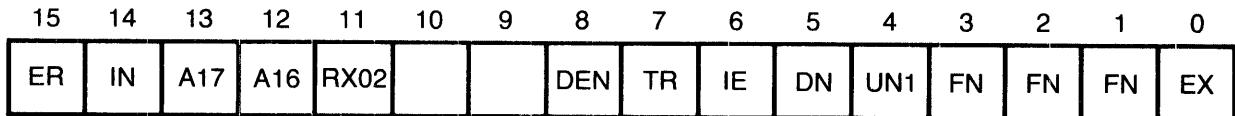


Figure 4-2. Command and Status Register Format

BIT 7 - TR - TRANSFER REQUEST flag

This read only bit indicates to the program that the DATA BUFFER REGISTER is empty and needs loading, or is loaded and needs emptying.

BIT 6 - IE - INTERRUPT ENABLE bit

This read/write bit, when set, allows an interrupt to be generated whenever the DONE flag is set.

BIT 5 - DN - DONE flag indicates the completion of an operation

This read only bit works in conjunction with the interrupt enable bit to generate interrupts.

BIT 4 - UN1 - Drive unit select bit

The binary encoding of this read/write bit selects drive 0-1. Drive selection only occurs if a drive related function is executed.

BITS 3-1 - FN - FUNCTION SELECT

The binary encoding of these write only bits selects the function to be performed by the DSD 440 system.

<u>BINARY</u>	<u>OCTAL</u>
000	0 = Fill buffer
001	1 = Empty buffer
010	2 = Write sector
011	3 = Read sector
100	4 = Set media density
101	5 = Read status
110	6 = Write deleted data sector
111	7 = Read error code

BIT 0 - EX - Execute the function encoded in bits 3-1 of this register. This is a write only bit.

Data Register (RX2DB)

The RX2DB is physically a shift register that provides the communication link between the host processor and the DSD 440 system. The logical register information passed through is based upon a predetermined protocol.

If the DSD 440 is not in the process of executing a command, the RX2DB is written without risk of adverse effects. However, during the execution of an instruction, the RX2DB register provides or accepts information (according to the RX2DB protocol) when the TRANSFER REQUEST flag is set.

#### CAUTION

Data may be lost if the correct protocol is not followed.

The following descriptions explain the various logical register formats of the physical Data Register (or RX2DB).

#### Data Buffer Register

The data buffer register is a physical shift register that transfers data to and from the controller data buffer. All information is transferred as a byte through bits 0-7 of the RXDB.

#### Disk Track Address (RX2TA)

At the proper time during commands requiring a track number (e.g., write sector, read sector), the track number is written to the physical RX2DB register as if it were a logical register. This is the TRACK ADDRESS REGISTER (RX2TA = 777172). Track numbers from 0-76 (decimal) are valid.

#### Disk Sector Address (RX2SA)

At the proper time during commands requiring a sector address (e.g. write sector, read sector), the sector address is written to the physical RX2DB register as if it were a logical register. This is the SECTOR ADDRESS REGISTER (RX2SA = 777172). Sector addresses from 1-26 (decimal) are valid. Bits 7 and 6 of the RX2SA are masked to zeroes.

#### Word Count Register

The WORD COUNT REGISTER specifies the number of words to be transferred between the controller sector buffer and main memory via direct memory access. For a double density sector, the maximum word count is 128 decimal words (256 bytes). The maximum word count in single density is 64 decimal words (128 bytes). The programmer loads the actual count into this register, NOT the 2's complement of the count.

#### Bus Address Register

This register specifies the bus address to which data is to be transferred during any DMA operation. It is a 16 bit counter on the DSD 440 interface module for the PDP-11 and LSI-11. It increments by two following each data transfer.

The bus address register cannot be read. It should always be loaded with the starting address of a data buffer in memory at the appropriate time during the FILL BUFFER, EMPTY BUFFER, or READ EXTENDED STATUS functions. If you try to load bit 0 with a 1 it will be ignored.

#### System Error and Status Register (RX2ES)

The RX2ES register is another logical register that is implemented using the physical RX2DB shift register. It provides status and error information about the drive that is selected by bit 4 of the physical RX2CS register. At the completion of a command, the controller places the RX2ES register into the data



buffer register (RX2DB = 777172) so that the host processor can check the most recent operation. When the controller completes a function which did not actually select a drive (e.g., FILL BUFFER, EMPTY BUFFER), the RX2ES "UNIT SEL" bit and "DRV DEN" bit remains unmodified. All the other RX2ES bits are cleared at the initiation of each new function. See Figure 4-3 for the bit layout of this register.

**BIT 11 - NXM - Non-existent Memory Error**

This bit is set if during a DMA cycle the interface does not receive a bus reply when it tries to write/read a word to or from memory. Usually this means the address in the RX2BA or the extended address bits in the RX2CS are invalid. The operation is terminated; the error and done bits are set. To recover from this error condition, generate either a bus INIT or a programmed INIT.

**BIT 10 - WC OVFL - Word Count Overflow**

This bit is set if the word count specified during a fill or empty buffer command is too large for the sector size indicated by the density bit. The operation is terminated; the Error and Done bits are set.

**BIT 8 - UNIT SEL - Unit Select**

This bit indicates which drive was selected during the last read or write operation. It is set to indicate drive 1; cleared to indicate drive 0.

**BIT 7 - DRV RDY Drive Ready**

This bit, when set, indicates that the selected drive has a diskette correctly installed and up to speed. The Drive Ready bit is valid immediately following the Read Status function. The bit is also valid for drive 0 immediately following an initialize.

**BIT 6 - DD - Deleted Data**

This bit indicates that a deleted data address mark was found during the last Read Sector operation or that the last command was Write Deleted Data Sector.

**BIT 5 - DRV DEN - Drive Density**

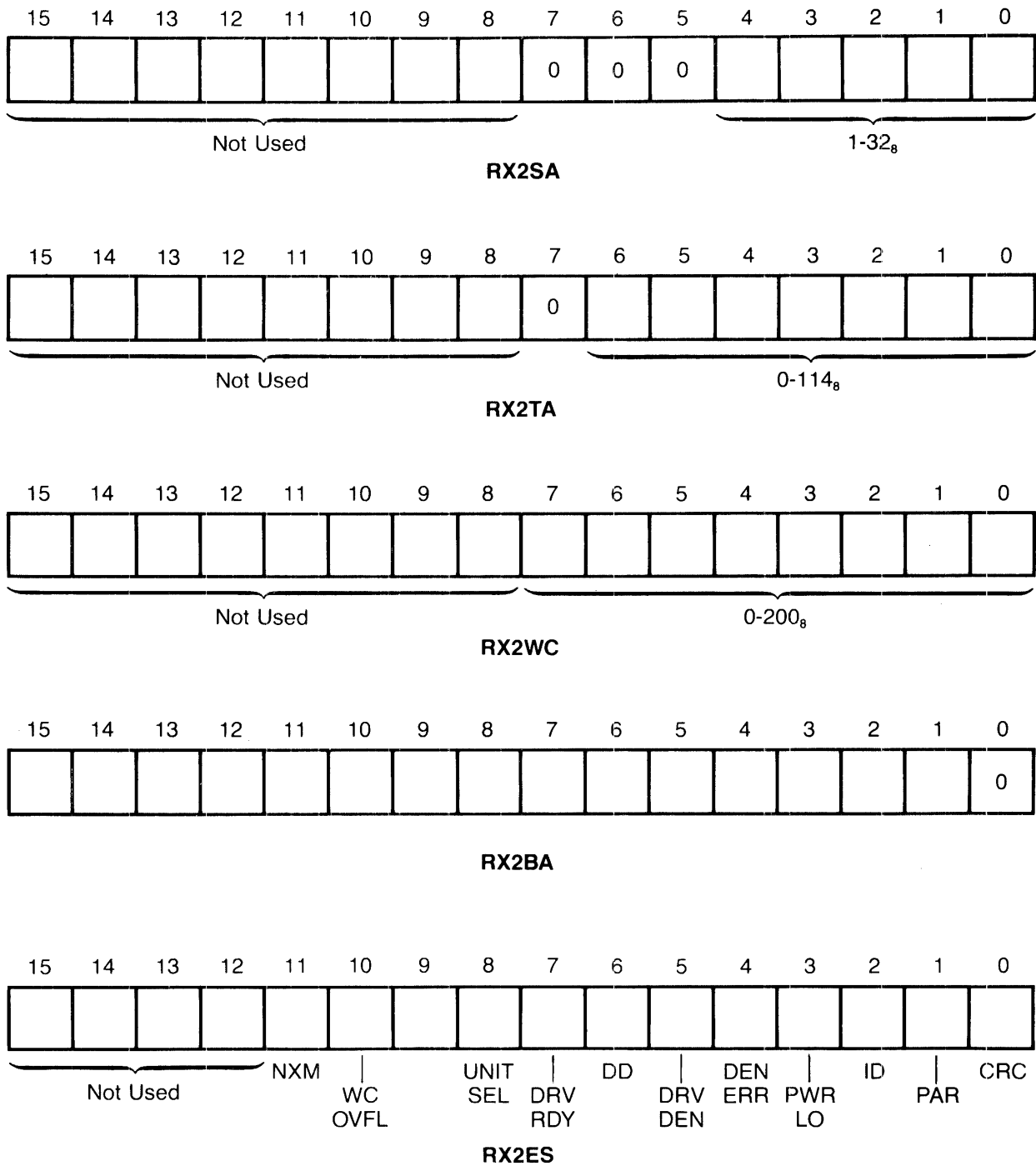
This bit indicates the density of the diskette installed in the drive indicated by bit 8. It is updated during read or write sector operations.

**BIT 4 - DEN ERR Density Error**

This bit indicates that during a READ SECTOR, WRITE SECTOR, WRITE DELETED DATA SECTOR, or READ STATUS operation the diskette density did not agree with the density bit of the RX2CS. The operation is terminated; the ERROR and DONE bits are set.

**BIT 3 - PWR LO Power Low**

This bit indicates a power failure in the controller/drive subsystem. It will also be set if the interface cable becomes disconnected. Any operation is terminated; the ERROR and DONE bits are set.



**Figure 4-3. PDP-11 System Error and Status Register Bit Layout**

#### BIT 2 - ID - Initialize Done

This bit indicates that the controller/drive subsystem has just completed an initialization sequence. This sequence may have been started by a power failure, bus INIT, or programmed INIT.

#### BIT 1 - PAR - Parity Error

This bit indicates that a parity error was detected when a command register was being shifted from the interface to the controller/drive subsystem. The operation is terminated; the ERROR and DONE bits are set.

#### BIT 0 - CRC - Cyclic Redundancy Check Error

This bit indicates that cyclic redundancy error was detected during the last Read Sector operation. The operation is terminated; the ERROR and DONE bits are set.

### Mode 2 Protocols

Protocols are required in the DSD 440 because the computer interface module and the intelligent portion of the DSD 440 are connected by a single serial data link. Therefore, the controller must identify parameters based on the order in which they are transmitted across the data link.

The following sections describe the protocol for each command that can be sent to the controller. Failure to adhere to the correct protocol results in lost or incorrect data.

#### FILL SECTOR BUFFER (000)

The FILL SECTOR BUFFER command is used to fill a storage buffer inside the DSD 440 with 128 or 256 eight-bit bytes of data from computer memory. Other functions can be used later to write that data to the diskette, or transfer it back to memory.

When the FILL SECTOR BUFFER command is given, the DSD 440 responds by clearing the DONE flag, RX2CS bit 5. The controller then requests a word count by setting the TRANSFER REQUEST flag. The program should respond by writing a valid RX2WC into the RX2DB. When TRANSFER REQUEST is again asserted by the controller, the program should respond by writing a valid starting memory address (RX2BA) into the RX2DB.

As soon as the RX2BA is loaded, TRANSFER REQUEST is cleared and remains cleared for the duration of this function. The data bytes are now transferred directly from memory to the controller sector buffer. When the word count is decremented to zero and the controller has zero-filled the remainder of the sector buffer (if necessary), DONE is asserted. An interrupt request is generated if the interrupt enable bit, RX2CS bit 6, was set when DONE became true. The RX2ES register will be found in the RX2DB at the completion of the function.

## NOTES

- 1) Bit 4 of the RX2CS does not affect this function since no disk drives need be selected.
- 2) The DENSITY bit, RX2CS bit 8, must be correctly set since this bit is used by the controller in evaluating the validity of the word count.

### EMPTY SECTOR BUFFER (001)

The EMPTY SECTOR BUFFER function is used to transfer the contents of the sector buffer to main memory. The sector buffer is loaded from a previous FILL SECTOR BUFFER or READ SECTOR command.

When the EMPTY SECTOR BUFFER command is given, the controller responds by clearing the DONE flag (RX2CS bit 5). The controller then sets the TRANSFER REQUEST flag (RX2CS bit 7) to request the word count register. The program should respond by loading a valid word count into the data buffer register.

When TRANSFER REQUEST is asserted again, the program responds by loading the starting memory address into the data buffer register. When this is done, the controller clears the TRANSFER REQUEST flag and it remains clear for the rest of the operation.

The data in the sector buffer is transferred to memory one word at a time until the word count is decremented to zero. When the data has been transferred, the controller places the RX2ES into the data buffer register and sets the DONE flag. If the interrupt enable bit is set, an interrupt request is initiated when DONE becomes true.

The notes above that apply to the FILL BUFFER command apply equally to the EMPTY SECTOR BUFFER command. In addition, note that the EMPTY BUFFER function does not modify the contents of the sector buffer.

### WRITE SECTOR (010)

The WRITE SECTOR function is used to transfer the contents of the sector buffer to a specified track and sector of the diskette.

When the WRITE SECTOR command is given, the controller clears the logical RX2ES register and the DONE flag. Next, the controller sets the TRANSFER REQUEST flag, RX2CS register bit 7, to request a sector address. The program responds by writing the desired sector address (RX2SA) into the data buffer register. This clears the TRANSFER REQUEST flag. As soon as the controller shifts the sector address over the interface cable, it asserts TRANSFER REQUEST again. This time the program responds by writing the desired track address (RX2TA) into the data buffer register. This clears the TRANSFER REQUEST flag.

After the track address is received, the controller causes the selected drive to seek the desired track. TRANSFER REQUEST is left reset for the remainder of the function. The heads of the selected drive are positioned over the specified track and are loaded against the media. If the controller does not know the density and format of the media, it determines the density and format by reading a random sector.

If media density does not agree with the command density (RX2CS bit 8), the operation is terminated. Bit 4 of the RX2ES register indicates a density error. If the densities agree, the controller checks the track address and looks for the specified sector address. If the correct track and sector are found, the controller writes either 128 bytes of single density data or 256 bytes of double density data from the sector buffer to the diskette. Two CRC bytes are written immediately after the data.

If the controller is unable to locate the specified diskette track, the RX2ER is set to a 150. If the specified sector cannot be found within two diskette revolutions, the RX2ER will be set to a 70. These error conditions, and the density error, cause the function to be terminated. The ERROR flag, RX2CS bit 15, and the DONE flag, RX2CS bit 5 are asserted when the function completes in this way. As with the error-free termination, an interrupt request is generated if the interrupt enable bit was set when the DONE flag became true.

#### NOTES

- 1) The contents of the sector buffer are not modified by the WRITE SECTOR function.
- 2) The contents of the sector buffer are modified as a result of a power failure or the initialize command. Programmers must be sure that valid data is written back into the sector buffer following either of these conditions. This is especially true before executing the WRITE SECTOR command.
- 3) If a sector number of 152 or 153 is written to the RX2SA, the WRITE SECTOR function turns into a WRITE FORMAT TRACK function.

#### READ SECTOR (011)

The READ SECTOR function is used to locate a specified track and sector of a diskette and then transfer the contents of the data field into the controller's sector buffer.

When the READ SECTOR command is given, the controller clears the RX2ES register and the DONE flag. Next, the controller sets the TRANSFER REQUEST flag (RX2CS bit 7) to request a sector address. The program should respond by writing the desired sector address (RX2SA) into the data buffer register (RX2DB = 777172). This clears the TRANSFER REQUEST flag. As soon as the controller shifts the sector address over the interface cable, it asserts TRANSFER REQUEST a second time. The program responds by writing the desired track address (RX2TA) into the data buffer register which clears the TRANSFER REQUEST flag.

After receiving the track address, the controller causes the selected drive to seek the desired track. TRANSFER REQUEST is left reset for the remainder of this function. The controller loads the heads against the media and determines the density of the media if it is not already known. If the diskette density does not agree with the command density (RX2CS bit 8), an error is reported and the function is terminated.

If the densities agree, the controller looks for the specified sector. When the right sector is located, the controller looks for the appropriate data, or deleted data address mark.

When the mark is found, the controller transfers the following 128 (or 256) bytes into the sector buffer. The two CRC bytes are read immediately after the data field. An error-free read is indicated if the address mark, data bytes, and two bytes of CRC check bytes produce a zero residue when passed sequentially through the CRC checker hardware circuits. As soon as the data is available in the buffer, the controller terminates the function by writing the RX2ES to the data buffer register and setting the DONE flag. An interrupt request is generated if the interrupt enable bit was set when DONE became true.

If the deleted data address mark was detected, the controller will set the deleted data flag. This flag appears in the ERROR/STATUS register (RX2ES bit 6). If a CRC error is detected, the controller will set RX2ES bit 0 and the ERROR flag (RX2CS bit 15). Seek errors and missing sector errors are reported just as in the WRITE SECTOR function.

#### SET MEDIA DENSITY (100)

This command is used to initialize an entire diskette to some specified density. When the SET MEDIA DENSITY command is executed, the controller attempts to write zeroes in every data field on the diskette. Bit 8 of the RX2CS determines the recording density and the type of Data Address Mark to be written in each data field. No sector headers are written when the SET MEDIA DENSITY command is executed.

#### Function Protocol

When the command is received, the controller clears the DONE flag and the logical RX2ES register. Next, the controller sets the TRANSFER REQUEST flag. The program responds by writing a "key byte" into the physical RX2DB. If the key byte is an ASCII "I" 411 or 111 octal, the SET MEDIA DENSITY function is executed. If the byte written into the RX2DB is not an "I", the DONE and ERROR flags are set and the operation terminates. The error register is loaded with a 250 to indicate an invalid key. The purpose of the key is to make it difficult to erase all of the data on a diskette.

As soon as the safety character "I" is received, the controller moves the heads to track 0. When sector 1 is found, the controller starts writing. If bit 8 of the RX2CS was a 0, a single density Data Address Mark and 128 FM zeroes are written. If bit 8 of the RX2CS was a 1, a double density Data Address Mark and 256 DEC MFM zeros are written. After writing all 25 sectors on track 0, the controller seeks to track 1, 2, ... writing all 26 sectors on each track. This continues until either every sector has been written through track 76: sector 26, or a bad header is found. The ERROR and DONE flags are set if the operation terminates due to a bad header.

The SET MEDIA DENSITY function takes about 26 seconds, depending on the sector interleave. It should never be interrupted before it is done. If the function does not terminate normally, an illegal diskette which has Data Address Marks of both densities may have been created. If this happens, the diskette should be completely rewritten. If the

SET MEDIA DENSITY function is not complete because of an unreadable header, the TRACK FORMAT procedure can be used to rewrite the incorrect header information.

#### READ STATUS (101)

The READ STATUS command is used to determine the current status of the drive selected by RX2CS bit 4. The status information passed back is: 1) drive readiness, and 2) the density of the diskette currently in the drive.

When the command is issued, the DONE flag is cleared. The controller checks the selected drive's door is closed, a diskette is inserted, and that the diskette is up to speed. Diskette speed is determined by measuring the amount of time between successive index pulses. Since this measurement takes an average of 250 milliseconds, excessive use of the READ STATUS function will cause reduced throughput. If the drive is ready, the controller sets bit 7 (DRIVE READY) fo the RX2ES. Next, the controller loads the heads and reads the first sector it finds.

If a double density address mark is detected, bit 5 (DRV DEN) of the RX2ES is set. If a single density mark is found, bit 5 is cleared. The controller terminates the function by shifting the RX2ES over to the RX2DB and setting the DONE flag. An interrupt request is generated if the interrupt enable bit, RX2CS bit 6, was set when DONE became true.

#### WRITE DELETED DATA SECTOR (110)

This function performs the same task as WRITE SECTOR except it writes a deleted data address mark just before the data field. The standard WRITE SECTOR function writes a regular data address mark. When a sector written with a deleted data address mark is read, bit 6 of the logical RX2ES register is set. The density bit associated with this function (RX2CS bit 8) determines whether a single or double density deleted data address mark is written.

#### READ EXTENDED STATUS (111)

The READ EXTENDED STATUS command is used to retrieve a number of internal controller registers, including the error register. These registers are transferred to memory using direct memory access. As soon as the command is loaded into the RX2CS, the DONE flag goes false. The controller then asserts the TRANSFER REQUEST flag.

The program then loads a starting memory address into the RX2DB. The controller transfers 4 words directly to memory. When the words are in memory, the controller asserts DONE. This generates an interrupt request if interrupt enable had been previously set.

The words transferred to memory are as follows:

Word 1 - Lo byte	Definitive error code (see Table 4-2)
Word 1 - Hi byte	Word count register
Word 2 - Lo byte	Current track address of drive 0
Word 2 - Hi byte	Current track address of drive 1

Word 3 - Lo byte	Target track of current disk access
Word 3 - Hi byte	Target sector of current disk access
Word 4 - Bit 0	Density of read error register command
Word 4 - Bit 4	Drive density of drive 0
Word 4 - Bit 5	Head load bit
Word 4 - Bit 6	Drive density of drive 1
Word 4 - Bit 7	Unit select bit
Word 4 - Hi byte	Track address of selected drive

### Power Fail

When a power failure occurs or DC power to the DSD 440 is interrupted, the controller gradually drains the filter capacitors and stops executing microcode. The program knows the controller/drive subsystem has lost power when the DONE and ERROR bits are set in the RX2CS, and the PWR LOW bit is set in the RX2DB.

When power is restored, and the controller DIP-Switch is configured for "NORMAL" mode, the DSD 440 controller initiates the following sequence:

- 1) DONE is cleared.
- 2) Controller executes the hardware self-tests.
- 3) All drives positioned to track 00.
- 4) RX2ES is cleared of all active error bits.
- 5) The controller reads sector 1, track 1 of unit 0 into buffer.
- 6) Bit 2 of RX2ES (INITIALIZE DONE) is set.
- 7) Bit 7 (DRIVE READY) and 5 (DRIVE DENSITY) of RX2ES are updated according to the status of drive 0.

At the end of this sequence, RX2CS bit 5 (DONE) is set.

### Diskette Format Programming

When configured for Mode 2 operation, the DSD 440 can write-format diskettes in two unique formats.

#### NOTE

The DEC RX02 does not support the command protocol described below because the DSD 440's commands are larger than the set of commands for the RX02.

Each time the write-format command protocol is executed, one entire track is rewritten. The protocol starts when the user program sends the WRITE SECTOR function code (010) to the controller. The state of the density bit, RX2CS bit 8, is unimportant during the execution of this particular command. After receiving the command,



the controller clears the RX2ES register and the DONE flag. Next, the controller sets the TRANSFER REQUEST flag, RX2CS bit 7, to request a sector address.

If the track format desired is IBM 3740 single density, the user program writes the number 152 into the data buffer register. If the track format desired is DEC double density, the user program writes the number 153 into the data buffer register. When the controller reads these "unusual" sector addresses, it jumps out of the WRITE SECTOR microcode and into special microcode designed to format tracks.

The protocol continues as follows: The controller sets the TRANSFER REQUEST flag to request a track address. The user program responds by writing a valid track address into the data buffer register. Next, the controller enters a program loop where 26 sector addresses are requested. Each time the user program sees the TRANSFER REQUEST flag, another valid and unique sector address is written into the data buffer register. The controller does NOT verify either the uniqueness or the validity (in the range 1-26) of the sector addresses being passed.

After the 26th sector address is received, the TRANSFER REQUEST flag remains false. The controller positions the heads to the specified track and awaits an index pulse. Starting at the index mark, the controller writes the entire track according to the specified format. The sector addresses passed to the controller are written in the sector headers in the order they passed to the controller. This enables hard sector interleaving techniques that improve the throughput of the disk system to be implemented.

#### Typical Sequences of Operations

The programming examples shown in Tables 4-5 and 4-6 illustrate how to write routines which will manipulate the DSD 440 Flexible Disk System in Mode 2 operation (RX02 compatible).

#### READ STATUS

Status information helps determine the status of a drive or the cause of an error. To determine drive related status (DRIVE READY, DRIVE DENSITY) the READ STATUS command should be used. When the ERROR flag (RX2CS bit 15) is set following a function, the RX2ES should be read first. Remember that the RX2ES is left in the RX2DB following all functions. The RX2ES has error bits for: CRC ERROR, PARITY ERROR, POWER LOW and DENSITY ERROR.

If no error bits are set in the RX2ES, the definitive error code can be obtained using the READ EXTENDED STATUS command. Code interpretations are shown in Table 4-7. Refer to Appendix I for more detailed error-code interpretations.

#### COMMON PROGRAMMING MISTAKES

This describes common programming mistakes that can cause data loss and/or error indications.

- 1) Illegal track or sector address sent to the controller.
  - a) Valid sectors are 1-26 (decimal)  
(There is no sector 0)
  - b) Valid tracks are 0-76 (decimal)

Table 4-5. Fill / Empty RX01 Sector Buffer Example

```

;PROGRAMMING EXAMPLE
;FILL/EMPTY RX02 SECTOR BUFFER
;
177170          RXCS=177170          ;CONTROL AND STATUS REGISTER
177172          RXDB=177172          ;DATA BUFFER REGISTER
;
;EMPTY RX02 SECTOR BUFFER
00000 012700    EMPBUF  MOV #3, RO      ;BUILD EMPBUF COMMAND IN RO
00003 000003
00004 000402          BR  FNCENT
;
;FILL RX02 SECTOR BUFFER ROUTINE
00006 012700    FILBUF: MOV #1, RO      ;BUILD EMPBUF COMMAND IN RO
00001 000001
00012 005737    FNCENT  TST  UNIT      ;UNIT 0 OR UNIT 1?
00015 000154'
00016 001402          BEQ  1$
00020 052700          BIS  #20, RO      ;SET UNIT
00020 000020
00024 005737    1$:    TST  DEN        ;HIGH OR LOW DENSITY?
00024 000146'
00030 001402          BEQ  2$
00032 052700          BIS  #400,RO     ;SPECIFY HIGH DENSITY
00032 000400
00036 010037    2$:    MOV  RO,@#RXCS   ;ISSUE COMMAND TO CONTROLLER
00036 177170
00042 105737    3$:    TSTB @#RXCS     ;WAIT FOR TRANSFER REQUEST
00042 177170
00046 100375          BPL  3$
00050 013737    MOV  WRDCNT,@#RXDB     ;PASS WORDCOUNT TO CONTROLLER
00050 000150'
00050 177172
00056 105737    4$:    TSTB @#RXCS     ;WAIT FOR TRANSFER REQUEST
00056 177170
00062 100375          BPL  4$
00064 013737    MOV  BUFADR,@#RXDB    ;PASS BUS ADDRESS TO CONTROLLER
00064 000152'
00064 177172
00072 032737    5$:    BIT  #100040,@#RXCS ;TEST FOR DONE AND ERROR
00072 000040
00072 177170
00100 001774          BEQ  5$
00102 100401          BMI  ERFIN      ;ERROR BIT SET?
00104 000207          RTS  PC
;
00106 012737    ERFIN: MOV  @17,@#RXCS  ;GET DEFINITIVE STATUS
00106 000017
00106 177170
01114 105737    6$:    TSTB @#RXCS     ;WAIT FOR TRANSFER REQUEST
01114 177170
00120 100375          BPL  6$

```

Table 4-5. Fill / Empty RX01 Sector Buffer Example  
(continued)

```

00122 012737          MOV #ERBUF,@#RXDB          ;SEND ERROR BUFFER ADDRESS
      000156'
      177172
00130 032737 7$      BIT #40,@#RXCS          ;WAIT FOR DONE BIT
      000040
      177170
00136 001774          BEQ 7$
00140 113700          MOVB @#ERBUF, R0          ;LEAVE ERROR REGISTER IN R0
      000156'
00144 000000          HALT

;
00146 000000 DEN:      .WORD 0                ;DENSITY - 0=SINGLE 1=DOUBLE
00150 000100 WRDCNT:  .WORD 100              ; - FULL SD BUFFER = 100
                                           ; - FULL DD BUFFER = 200
00152 002000 BUFADR:  .WORD 2000             ;BUFFER ADDRESS VARIABLE
00154 000000 UNIT:    .WORD 0                ;0=DRIVE 0, 1=DRIVE 1
00156 000000 ERBUF:   .WORD 0                ;ERROR BUFFER

```

Table 4-6. Read / Write RX02 Sector Example

```

;PROGRAMMING EXAMPLE
;READ/WRITE RX02 SECTOR
;
177170          RXCS=177170          ;CONTROL AND STATUS REGISTER
177172          RXDB=177172          ;DATA BUFFER REGISTER
;
;READ RX02 SECTOR
00000 012700  READ:  MOV #7, R0          ;BUILD READ SECTOR CMD IN R0
00007 000007
00004 000402          BR SYNTAX
;
;WRITE RX02 SECTOR ROUTINE
00006 012700  WRITE: MOV #5, R0          ;BUILD WRITE SECTOR CMD IN R0
00005 000005
00012 005737  SYNTAX: TST UNIT          ;UNIT 0 OR UNIT 1?
000154' 000154'
00016 001402          BEQ 1$
00020 052700          BIS #20,R0          ;SET UNIT
00020 000020
00024 005737  1$      TST DEN          ;HIGH OR LOW DENSITY?
00024 000146
00030 001402          BEQ 2$
00032 052700          BIS #400,R0        ;SPECIFY HIGH DENSITY
00032 000400
00036 010037  2$:     MOV R0,@#RXCS      ;ISSUE COMMAND TO CONTROLLER
00036 177170
00042 105737  3$:     TSTB @#RXCS        ;WAIT FOR TRANSFER REQUEST
00042 177170
00046 100375          BPL 3$
00050 013737          MOV SECTOR,@#RXDB  ;PASS SECTOR TO CONTROLLER
00050 000150'
00050 177172
00056 105737  4$:     TSTB @#RXCS        ;WAIT FOR TRANSFER REQUEST
00056 177170
00062 100375          BPL 4$
00064 013737          MOV TRACK,@#RXDB   ;PASS TRACK TO NUMBER TO
00064 000152          ;CONTROLLER
00064 177172
00072 032737  5$:     BIT #100040,@#RXCS ;TEST FOR DONE AND ERROR
00072 100040
00072 177170
00100 001774          BEQ 5$
00102 100401          BMI ERFIN          ;ERROR BIT SET?
00104 000207          RTS PC
;
00106 012737  ERFIN:  MOV #17,@#RXCS      ;GET DEFINITIVE STATUS
00106 000017
00106 177170
00114 105737  6$:     TSTB @#RXCS        ;WAIT FOR TRANSFER REQUEST
00114 177170
00120 100375          BPL 6$

```

Table 4-6. Read / Write / RX02 Sector Example  
(continued)

```

00122 012737      MOV #ERBUF,@#RXDB      ;SEND ERROR BUFFER ADDRESS
      000156 '
      177172
00130 032737 7$   BIT #40, @#RXCS      ;WAIT FOR DONE BIT
      000040
      177170
00136 001774      BEQ 7$
00140 113700      MOVB @#ERBUF, RO      ;LEAVE ERROR REGISTER IN RO
      000156
00144 000000      HALT
;
00146 000000 DEN:      .WORD 0      ;DENSITY 0=SINGLE 1=DOUBLE
00150 000001 SECTOR:  .WORD 1      ;DESIRED SECTOR ADDRESS
00152 000001 TRACK:   .WORD 1      ;DESIRED TRACK ADDRESS
00154 000000 UNIT:    .WORD 0      ;UNIT - 0=DRIVE 0 1=DRIVE 1
00156 000000 ERBUF:   .WORD 0      ;ERROR BUFFER

```

- 2) The READ STATUS command requires up to two revolutions of the disk to complete. To avoid excessive delays, use this command only when necessary.
- 3) After reading or writing, the INITIALIZE DONE bit, RX2ES bit 2, may be checked for an indication of power failure. A short power outage will cause DONE to set without any error indication.
- 4) The drive select bit, RX2CS bit 4, is not decoded by the controller during FILL BUFFER and EMPTY BUFFER functions.
- 5) It is recommended that a two-sector interleave (Sectors 1, 3, 5 etc.) be used for optimal data transfer rate.

Typically, a FILL BUFFER command precedes a WRITE SECTOR command. Similarly, a READ SECTOR command precedes an EMPTY BUFFER command.

### Interrupts

An interrupt is requested by the interface module whenever the INTERRUPT ENABLE and DONE bits of the RX2CS both become set. The standard interrupt vector address is location 264.

### DEC PDP-8 PROGRAMMING INTERFACE

When connected to a PDP-8, the DSD 440 disk system is capable of writing either 8 bit or 12 bit data words on the diskette. Using 8 bits per word, 128 bytes can be written per sector with no waste. With 12 bits per word, 64 twelve bit data words are written per sector and the controller fills the unused portion of the sector with zeroes. In double density operation, 8 and 12 bit words can still be selected. The number of words that fit into each sector is doubled.

Table 4-7. Definitive Error Register Codes for the RXER

(FUNCTION 111)

<u>OCTAL CODE</u>	<u>DESCRIPTION</u>
000	No error
010	No drive 0 or drive 0 failed to find track 0 on INIT
020	No drive 1 when DIP switch indicates there should be a drive 1, or drive 1 failed to find track 0 on INIT
030	Track 0 found while stepping in on initialize
040	Track address passed to controller was invalid (>76)
050	Track 0 found before desired track while stepping
070	Requested sector not found in two revolutions
075	Too many bad headers on "IBM 2D" diskette
100	Write protect violation
110	No read data signal present
120	No preamble found
130	Preamble found, but no address mark within window
140	CRC error on what appeared to be a header
150	Address in good header did not match desired track
160	Too many tries for an ID address mark
170	Data address mark not found in allotted time
175	DEC double density address mark on non-DEC diskette
200	CRC error on data field; RXES bit 0 also set
210	Parity error on interface cable; RXES bit 1 also set
220	Read/write controller failed maintenance mode test
230	Invalid word count specified
240	Density error; DEC format
250	Wrong key for set media density or format command
260	Indeterminate density, or no diskette present
270	Read/write controller write-format failure
320	Read/write controller detected write circuit failure
330	Read/write controller timed out on reset
340	Master controller out of SYNC with RD/WRT controller
350	Non-existent memory error during DMA
360	Drive not ready during format command
370	AC power low caused abort of write activity

## INSTRUCTION SET

The eight input/output transfer (IOT) instructions used to program the DSD 440 are shown in Table 4-8. These same IOT instructions are used in both Mode 1 (RX01) and Mode 2 (RX02) operation.

Table 4-8. Input/Output Transfer Instructions

<u>INSTRUCTION</u>	<u>MNEMONIC</u>	<u>DESCRIPTION</u>
67 x 0		No operation
67 x 1	LCD	Load command, clear accumulator
67 x 2	XDR	Transfer data register
67 x 3	STR	Skip on transfer request, clear flag
67 x 4	SER	Skip on error flag, clear flag
67 x 5	SDN	Skip on done flag, clear flag
67 x 6	INTR	Enable or disable disk interrupts
67 x 7	INIT	Initialize controller and interface

(The digits represented by "x" can be any octal digit.)

### The LOAD COMMAND (LCD) Instruction (67 x 1)

This instruction transfers the contents of the accumulator to the interface register and then clears the accumulator. The command word interpretation for Mode 1 is shown in Figure 4-4. Figure 4-5 shows the interpretation when the system is configured in Mode 2 for 12 bit mode (bit 5 = 0).

The format shown in Figure 4-12 is used in Mode 2 when 8 bit transfers are selected (bit 5 = 1). In this case, the first 8 bit command segment is sent to the controller using the LCD instruction. The program must then execute the STR instruction before sending the upper four bits of the command register to the controller with the XDR instruction.

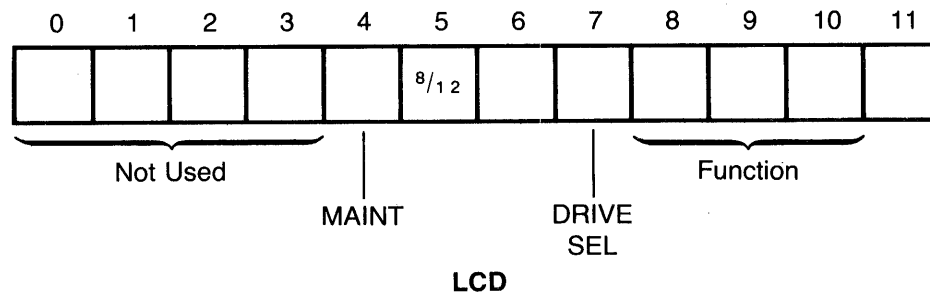
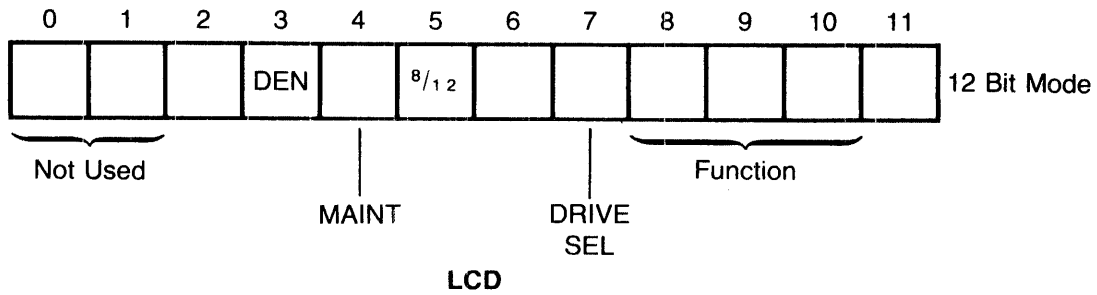
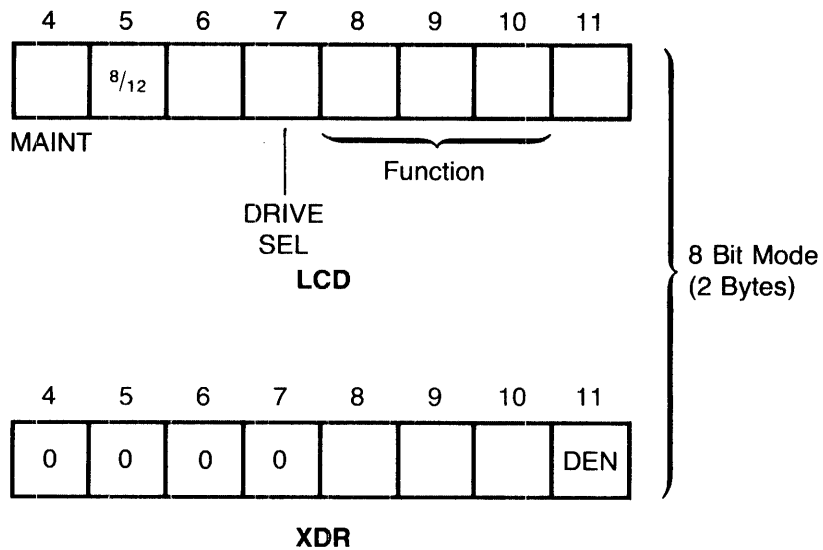


Figure 4-4. PDP-8 Mode 1 Command Register Format



**Figure 4-5. PDP-8 Mode 2, 12 Bit Mode, Command Register Format**



**Figure 4-6. PDP-8 Mode 2, 8 Bit Mode, Command Register Format/Sequence**

**The TRANSFER DATA REGISTER (XDR) Instruction (67 x 2)**

This instruction is used to transfer a word between the accumulator and the interface register. The DSD 440 controls the direction of transfer based on the context of the function in progress. The length of the word transferred is governed by the mode last selected (8-bit or 12-bit).

If the Done flag is false, executing this instruction can indicate one of two conditions to the DSD 440:

- 1) The PDP-8 has accepted the last word supplied by the controller.
- 2) The PDP-8 has provided the last word requested by the controller.



Data transfers from the accumulator always leave the accumulator unchanged. When the controller is done with the current function (as indicated by the Done flag), the XDR instruction can be used to transfer the error-status register from the interface register to the accumulator. Data transfers to the accumulator are 12-bit JAM transfers using twelve-bit word, or eight-bit "OR"ed transfers with eight bit words.

#### The SKIP ON TRANSFER REQUEST (STR) Instruction (67 x 3)

This instruction causes the next instruction to be skipped if the DSD 440 has set the Transfer Request flag. The instruction also clears the flag. The program should test Transfer Request just before using the XDR instruction to transfer data or parameters to or from the controller.

#### The SKIP ON ERROR (SER) Instruction (67 x 4)

This instruction causes the next instruction to be skipped if the Error flag has been set by the controller. The Error flag is then cleared. The controller always sets the Done flag together with the Error flag (but not vice versa). A typical program executes a Skip On Done (SDN) instruction followed by a Skip On Error (SER) instruction.

#### The SKIP ON DONE (SDN) Instruction (67 x 5)

This instruction causes the next instruction to be skipped if the Done flag has been set by the controller. The Done flag is then cleared by the instruction. If interrupts are enabled, the Done flag generates an interrupt when it is first asserted by the controller.

#### The INTERRUPT ENABLE/DISABLE (INTR) Instruction (67 x 6)

If accumulator bit 11 is set when this instruction is executed, the interrupt enable flip flop is set. If bit 11 is cleared, then the interrupt enable flip flop is reset. Interrupts are normally generated when the interrupt enable flip flop is set and the Done flag has just become true. Interrupts are disabled by a CLEAR ALL FLAGS instruction, a bus initialize, or a programmed initialize.

#### The INITIALIZE (INIT) Instruction (67 x 7)

This instruction initializes the DSD 440 system by repositioning the read/write heads of the disk drives to track 0. The controller microprocessor runs all the hardware self-test routines.

If a diskette is properly installed in drive 0, the data contained in track 1: sector 1 is read into the sector buffer. The error/status register is cleared and then the Initialize Done bit (the logical RXES register bit 9) is set. This initialize sequence can take as long as 1. The INIT instruction and the CLEAR ALL FLAGS instruction are both capable of starting this sequence of events.

## REGISTER DESCRIPTIONS

The interface register is the only physical register in the DSD 440 interface hardware for the PDP-8. It can represent any one of six logical controller registers depending on the protocol of the function in progress. Following is a description of each possible register format.

### Command Register

The command register has the format shown in Figure 4-4 when the controller is configured in Mode 1. Since the high order four bits are never used in Mode 1, the entire command can be communicated to the controller with a single LCD (load command, clear accumulator) instruction regardless of the 8/12 bit mode bit. Bit 4 is ignored in the DSD 440.

Bit 5 is used to select the word length. When cleared, bit 5 sets the word length to a full twelve bits. When set, bit 5 sets the word length to eight bits.

Bit 7 specifies the drive to be used for the current operation. Drive 0 is used if bit 7 is clear and drive 1 is used if bit 7 is set. Operations that do not involve drives (e.g., FILL BUFFER, EMPTY BUFFER) are not affected by the drive select bit. In Mode 2, the density bit specifies a double density operation when set, and a single density operation when cleared. An error is reported if the density of the diskette in the selected drive does not match density specified by the density bit.

Bits 8, 9, and 10 are used to encode the desired function. Table 4-9 lists the codes:

Table 4-9. Command Register Function Codes

<u>CODE BITS</u>	<u>MODE 1 FUNCTION</u>	<u>MODE 2 FUNCTION</u>
0 0 0	Fill buffer	Fill buffer
0 0 1	Empty buffer	Empty buffer
0 1 0	Write sector	Write sector
0 1 1	Read sector	Read sector
1 0 0 *	NOOP (clear INIT done) *	Set media density
1 0 1	Read status	Read status
1 1 0	Write deleted data sec.	Write deleted data sec.
1 1 1	Read error register	Read error register

### Track Address Register

When a particular function requires a track address, this register is written to the controller interface register with an XDR (transfer data register) instruction. Functions which require track addresses include: WRITE SECTOR, READ SECTOR, WRITE-FORMAT TRACK, and WRITE DELETED DATA SECTOR. Valid numbers for this register are 0-76 decimal. The track address register works the same in both Mode 1 and Mode 2 system operation. Figure 4-7 is a diagram of this register.

### Sector Address Register

This register is written to the controller interface register with an XDR (transfer data register) instruction when a particular function requires a sector address. Functions which require sector addresses include: WRITE SECTOR, READ SECTOR, and WRITE DELETED DATA SECTOR. Valid numbers for this register are 1-26 decimal. Figure 4-8 is a diagram of this register. This register works the same in both Mode 1 and Mode 2 system operation.

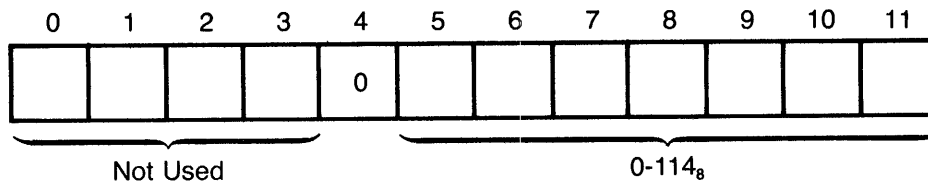


Figure 4-7. Track Address Register

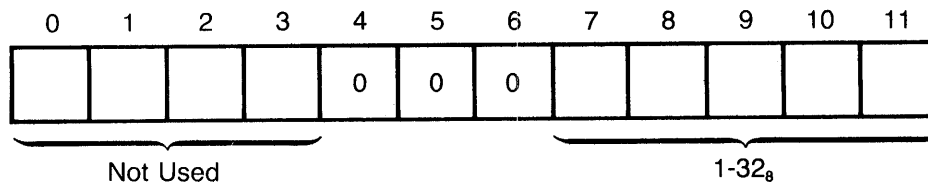


Figure 4-8. Sector Address Register

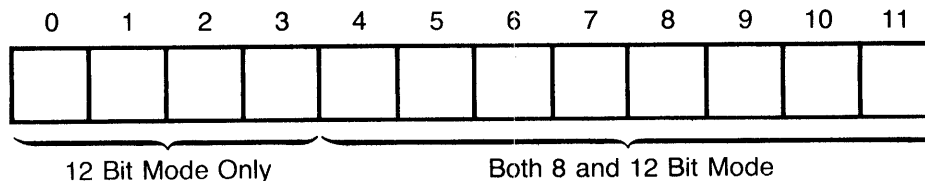


Figure 4-9. PDP-8 Data Buffer Register

## Data Buffer Register

All data transferred to and from the diskette must pass through this register. The state of bit 5 in the command register determines which bits of this register are significant. If bit 5 was a zero, then the machine is configured for 12 bit operation. All 12 bits of the register contain valid data. If bit 5 is a one, then the machine is in 8 bit operation and only bits 4-11 contain valid data. This applies equally to both Mode 1 and Mode 2 system operation. Figure 4-9 is a diagram of this register.

## ERROR AND STATUS REGISTER IN MODE 1 OPERATION

Figure 4-10 shows the bit assignments of the error and status register when the controller is configured for Mode 1 operation. This register is available in the interface register upon completion of any function except READ ERROR REGISTER. The READ STATUS function is used to access this register when a valid DRIVE READY bit is important. The XDR instruction transfers the error/status register from the interface register to the accumulator. The error/status register bits are assigned the following meanings:

BIT 0-3 - Not Used

BIT 4 - DRV RDY - Drive Ready

This bit, when set, indicates that the selected drive has a diskette correctly installed and up to speed. The Drive Ready bit is only valid immediately following the Ready Status function. The bit is also valid for drive 0 immediately following an initialize.

BIT 5 - DD - Deleted Data

This bit indicates a deleted data address mark was found during the last Read Sector operation or that the last command was Write Deleted Data Sector.

BIT 6-8 - Not used in Mode 1

BIT 9 - ID - Initialize Done

This bit indicates that the controller/drive subsystem has just completed an initialization sequence. This sequence may have been started by a power failure, bus INIT, or programmed INIT.

BIT 10 - PAR - Parity Error

This bit is set to indicate that a parity error was detected when a command register was being shifted from the interface to the controller/drive subsystem. The operation is terminated; the Error and Done flags are set.

BIT 11 - CRC - Cyclic Redundancy Check Error

This bit is set to indicate that a cyclic redundancy check error was detected during the last Read Sector operation. The operation is terminated; the Error and Done flags are set.

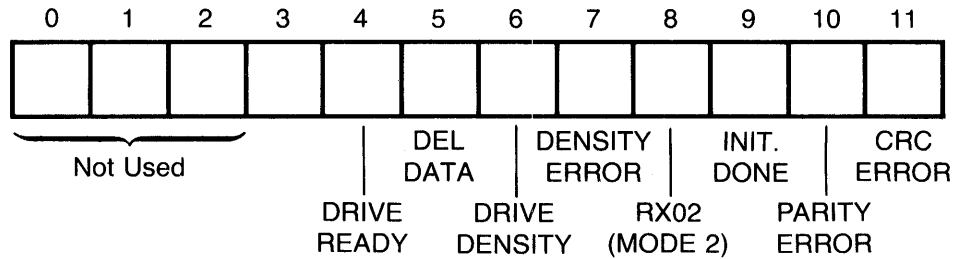


Figure 4-10. Error/Status Register in Mode 1

ERROR AND STATUS REGISTER IN MODE 2 OPERATION

Figure 4-11 shows the bit assignments of the error and status register when the controller is configured for Mode 2 operation. This register is available in the interface register upon completion of any function except READ ERROR REGISTER. The READ STATUS function is used to access this register when a valid DRIVE READY bit is important. The XDR (transfer data register) instruction is used to transfer the error/status register from the interface register to the accumulator. The error/status register bits are assigned the following meanings:

BIT 0-3 - Not used

BIT 4 - DRV RDY - Drive Ready

This bit, when set, indicates that the selected drive has a diskette correctly installed and up to speed. The Drive Ready bit is only valid immediately following the Read Status function. The bit is also valid for drive 0 immediately following an initialize.

BIT 5 - DD - Deleted Data

This bit indicates that a deleted data address mark was found during the last Read Sector operation or that the last command was Write Deleted Data Sector.

BIT 6 - DRV DEN - Drive Density

This bit indicates the density of the diskette in the selected drive. A binary 1 means double density and a binary 0 indicates single density.

BIT 7 - DEN ERR Density Error

This bit indicates that during a read or write sector operation the diskette density did not agree with the density bit in the command register. The operation is terminated and the Error and Done flags are set.

**BIT 8 - RX02 (Mode 2)**

This bit indicates that the interface module is connected to a controller/drive subsystem which is capable of both single and double density operation.

**BIT 9 - ID - Initialize Done**

This bit is set to indicate that the controller/drive subsystem has just completed an initialization sequence. This sequence may have been started by a power failure, bus INIT, or programmed INIT.

**BIT 10 - PAR - Parity Error**

This bit is set to indicate that a parity error was detected when a command register was being shifted from the interface to the controller/drive subsystem.

**BIT 11 - CRC Cyclic Redundancy Check Error**

This bit is set to indicate that a cyclic redundancy check error was detected during the last Read Sector operation. The operation is terminated; the Error and Done flags are set.

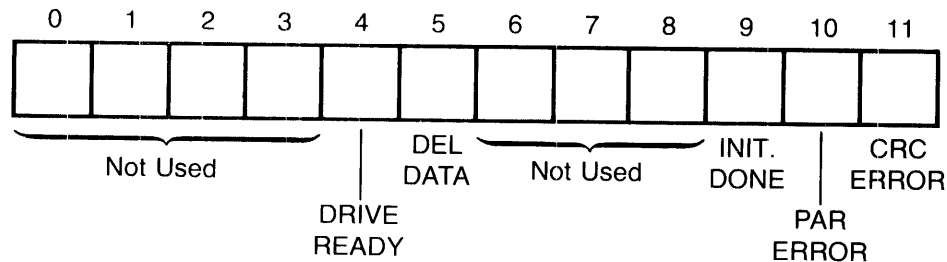


Figure 4-11. PDP-8 Error/Status Register in Mode 2

**FUNCTION CODE DESCRIPTIONS**

A diskette's function is initiated by writing one of the function codes as shown in Table 4-9 to the command register using the LCD instruction. The Done flag should always be tested and cleared with the SDN (Skip On Done) instruction before issuing the command instruction that verifies the controller is really in the Done state. The protocol associated with each of the functions shown in Table 4-9 is described in detail.

Fill Buffer (000)

This function is used to load the controller sector buffer with the data to be written into a sector or to be transferred later to the host computer. The amount and format of the data is determined by the operating mode of the controller (Mode 1 or Mode 2) and the word length bit of the command register.

After the command is issued to the controller with the LCD (Load Command) instruction, the controller asserts the Transfer Request flag once for each 8 or 12 bit word needed from the host computer to place in the sector buffer. The host computer tests and clears the Transfer Request flag with the STR (Skip On Transfer Request) instruction and then transfers a data word to the controller with the XDR (Transfer Data Register) instruction. When the controller determines that the appropriate number of words has been transferred in this manner, the Done flag is asserted and an interrupt occurs, assuming interrupts have been enabled. If an XDR instruction is executed after Done is set, it has the effect of loading the error/status register into the accumulator. Table 4-10 shows the relationship between the number of XDR instructions needed to fill the buffer, the operating mode, and the word length.

Table 4-10. Operating Mode, Word Length, XDR Instructions Relationship

<u>OPERATING MODE</u>	<u>DATE WORD LENGTH</u>	<u>NUMBER OF CYCLES</u>
1 (RX01 compatible)	8 bits	128
1 (RX01 compatible)	12 bits	64
2 (RX02 compatible)	8 bits	256 DD, 128 SD *
2 (RX02 compatible)	12 bits	128 DD, 64 SD

\* (SD = Single Density, DD = Double Density)

#### Empty Buffer (001)

This function is similar to the FILL BUFFER function except that words are moved from the interface register to the accumulator every time the XDR (Transfer Data Register) instruction is executed. Table 4-10 indicates the number of XDR instructions as a function of the word length and the operating mode. Execution of this function does NOT modify the contents of the controller sector buffer. When the controller asserts the Transfer Request flag, this indicates that a word is in the interface register.

The program must test the Transfer Request flag with the STR (Skip On Transfer Request) instruction before moving the word to the accumulator with the XDR instruction. When the buffer is emptied, the controller asserts the Done flag and places the error/status register into the interface register. An XDR instruction executed after Done is set moves the error/status register into the accumulator. An interrupt occurs when the controller sets the Done flag, assuming interrupts have been enabled.

#### Write Sector (010)

This function is used to transfer the data contained in the sector buffer to a specified unit, track and sector. After the WRITE SECTOR function is decoded by the controller, the error/status register is cleared and the Transfer Request flag is set. The program must test the Transfer Request flag with the STR (Skip On Transfer Request) instruction, which clears the flag.

The program must then transfer a valid sector address register to the controller using the XDR instruction. After the controller receives the sector address, it sets the Transfer Request flag a second time. The program must test the flag with the STR (Skip

On Transfer Request) instruction, which clears the flag. It then transfers a valid track address register to the controller using the XDR instruction.

The controller checks for density compatibility by comparing the diskette density with the density bit passed in the command register (if in Mode 2) or with low density (if in Mode 1). If the densities are incompatible, a density error is reported and the function is terminated. Assuming compatible densities, the controller attempts to locate the specified sector so that the data contained in the buffer, together with two CRC characters, can be written onto the diskette. The content of the data buffer is not modified regardless of whether it is written successfully on the diskette or not. When the function is completed, the error/status register is loaded into the interface register and the Done flag is set. An interrupt is generated when Done is set, assuming interrupts have been enabled.

### Read Sector (011)

The READ SECTOR function is used to locate a specified track and sector of a diskette and then transfer the contents of the data field into the controller's sector buffer. After the READ SECTOR function is decoded, the controller clears the error/status register. Next, the controller asserts the Transfer Request flag to request a sector address. The program should be in a loop, testing Transfer Request with the STR (Skip On Transfer Request) instruction. When the flag becomes true, the skip takes place and Transfer Request is cleared automatically.

At this point, the program loads the accumulator with the sector address register and executes the XDR (Transfer Data Register) instruction to transfer the AC to the controller. The controller asserts the Transfer Request flag as soon as it is ready for a track address. The program should test Transfer Request and should transfer a track address to the controller in the same manner used for the sector address.

The controller verifies the legality of the track address supplied by the program. If the address is illegal, the Error and done flags are set and the error/status register is moved into the interface register, thus terminating the function. If the track address is legal, the controller moves the head to the specified track and loads the head against the media. The controller checks for density compatibility by comparing the actual diskette density with the density bit passed in the command register (if in Mode 2), or with low density (if in Mode 1). If the densities are incompatible, a density error is reported and the function is terminated.

If the densities are compatible, the controller seeks for the specified sector. If a sector match is not found within two diskette revolutions, an octal 70 is placed in the error register and the done and error flags are set to terminate the function. Once the correct sector header is found, the controller starts looking for either a data address mark or a deleted data address mark of the appropriate density. The controller uses the mark to synchronize with the data. Then the appropriate amount of data is transferred from the diskette to the controller sector buffer. An error-free read is indicated if the address mark, data, and two CRC bytes produce a zero residue when sequentially passed through the CRC checker hardware circuits. If a CRC error is detected, the controller sets bit 11 of the Error/Status Register (RXES), loads an octal 200 into the error register (RXER), and then sets the Error and Done flags.



The function is always terminated when the error/status register is loaded into the interface register and the Done flag is set. Setting the Done flag produces an interrupt if interrupts have been enabled. If the data address mark found is a special mark called a "deleted data address mark", bit 5 of the error/status register will be set at the completion of this function.

### Set Media Density (100)

If the controller is configured in Mode 1 (RX01 compatible), a command code (1C) puts the error/status register in the interface register and sets the Done flag. If the controller is configured in Mode 2 (RX02 compatible), this command code can be used to initialize an entire diskette to a specified density.

#### Function Protocol

After receiving the entire command register, the controller checks that it is configured in Mode 2 (RX02 compatible). If not, the function is immediately terminated.

The Transfer Request flag is asserted to request a "key byte" from the user program. The key byte specifies the SET MEDIA DENSITY function if it is an ASCII "I" (0111) octal. If this particular key byte is not transferred to the controller at this time, the Error and Done flags are set and the function terminates. The error register is loaded with a (250) octal to indicate an invalid key error.

#### Set Media Density

As soon as the key byte "I" is received, the controller moves the head of the specified drive to track 0. When sector 1 is found, the controller starts writing. If the density bit is a binary 0, a single density Data Address Mark and 128 eight-bit FM (frequency modulated) zeroes are written followed by 2 CRC bytes.

If the density bit is a 1, a double density Data Address Mark and 256 eight-bit DEC MFM zeroes are written followed by 2 CRC bytes. After writing all 26 sectors on track 0, the controller sequentially seeks to track 1, 2, ... 76 writing all 26 sectors on each track. This process continues until every sector through track 76, sector 26 has been written or a bad header is found. The Error and Done flags are set if the operation terminates due to a bad header.

The SET MEDIA DENSITY function requires about 15 seconds to complete and should never be interrupted before it is completed. If this function does not terminate normally, an illegal diskette which has Data Address Marks of both densities may have been created. If this happens, the diskette should be rewritten. If the SET MEDIA DENSITY function does not finish normally because of an unreadable header, the WRITE-FORMAT option can be used to rewrite the bad header. The SET MEDIA DENSITY function only writes the data fields, not the headers.

### Read Status (101)

When the controller decodes the READ STATUS command, several bits in the error/status register are updated. The error/status register is then transferred to the interface register. Figures 4-10 and 4-11 show the format of the error/status register in Mode 1 and Mode 2 respectively. The INIT DONE status bit is always reset when this function is executed. The DRIVE READY bit is updated according to whether the selected drive has both power, and a diskette properly installed and up to speed.

Since the controller determines diskette rotational speed by measuring the amount of time between two successive index pulses, this function can require up to 250 milliseconds to execute. Because of this, excessive use of this function will result in substantially reduced throughput. The DELETED DATA bit, PARITY ERROR bit, and CRC ERROR bit are NOT modified by this function.

If the controller is configured for Mode 2 operation, other bits in the error/status register will be modified in addition to those already mentioned. The DRIVE DENSITY bit is updated to reflect the density of the diskette installed in the selected drive. The controller determines the density by loading the head, wherever it happens to be, and by trying to read the first sector that passes. If the density of the diskette is different from that indicated by the density bit in the command register, then the DENSITY ERROR bit of the error/status register will be set.

### Write Deleted Data Sector (110)

This function is identical to the WRITE SECTOR function except that a deleted data address mark is written prior to the data field instead of a normal data address mark. When the WRITE DELETED DATA SECTOR function is executed, the DELETED DATA bit is set in the error/status register. When a sector which was written using this function is read later, the DELETED DATA bit is set in the error/status register.

### Read Error Register (111)

This function is used to obtain explicit error information after the error flag has been detected using the SER (Skip On Error Flag) instruction. When the controller decodes this function code, the error code is transferred to the interface register and the Done flag is set.

The user program should detect Done with the SDN (Skip On Done Flag) instruction, and then transfer the error code from the interface register to the accumulator using the XDR (Transfer Data Register) instruction. This is the only function which does not terminate with the error/status register left in the interface register.

The interpretation of each error code is shown in Table 4-2. Some of these codes are only possible if the system is configured in Mode 2 (RX02 compatible).

### POWER RESTART

When a power failure occurs or when DC power to the DSD 440 controller is interrupted, the controller ceases its current function. All disk activity is terminated and the head is unloaded from the currently selected diskette.

When power is reapplied, the controller performs the following sequence:

- 1) Done flag is cleared.
- 2) Controller executes the hardware self-tests.
- 3) All drive heads are repositioned to track 00.
- 4) The error/status register is cleared of all active error bits.
- 5) The controller reads sector 1, track 1 of unit 0 into buffer. (If there is a diskette ready in drive 0.)
- 6) The INITIALIZE DONE bit is set in error/status register.
- 7) The DRIVE READY AND DRIVE DENSITY (if in Mode 2) bits of the error/status register are updated according to status of drive 0.

At the end of this sequence, the Done flag is set.

#### DISKETTE FORMATTING IN PROGRAMMABLE DENSITY

Each time the write-format command protocol is executed, an entire track is rewritten. The protocol starts when the user program sends the WRITE SECTOR function code (010) to the controller. (The state of the density bit, whether transmitted in bit 3 or bit 11, is unimportant.) After receiving the command, the controller clears the error/status register and sets the Transfer Request flag. The user program must test this flag with the STR (Skip On Transfer Request) instruction, which also clears the flag.

Instead of a valid sector address, the user program specifies a single density track format operation by transferring 152 octal to the controller with the XDR (Transfer Data Register) instruction. If the track format desired is DEC double density, then the number 153 octal is transferred.

When the controller sees these unusual sector addresses, it jumps out of the WRITE SECTOR microcode and into special microcode designed to format tracks. The protocol continues as follows:

- 1) The controller sets the transfer request flag to request a track address. The user program should respond by writing a valid track address into the data buffer register.
- 2) The controller enters a program loop where 26 sector addresses are requested. Each time the user program sees the TRANSFER REQUEST flag, another valid and unique sector address is written into the data buffer register.

Note that the controller does NOT verify either the uniqueness or the validity (in the range 1-26) of the sector addresses being passed at this time.

After the 26th sector address is received, the TRANSFER REQUEST flag will remain false. The controller will seek the heads to the specified track and await an index pulse. Starting at the index mark, the controller will write the entire track according to the specified format. The sector addresses that were passed to the controller will be written in the sector headers in the same order that they were passed to the controller. This enables hard sector interleaving techniques that improve disk system throughput.



## CHAPTER 5

### TESTING

The DSD 440 diagnostics simplify incoming inspection and speed fault isolation with easy-to-use maintenance capabilities including both extensive self-testing and complete interactive system level testing. These capabilities, combined with the reliability and modular construction of the DSD 440 system, help minimize the cost of long-term ownership. Additionally, the DSD 440 is able to run DEC diagnostics without modification.

This chapter covers both the self-testing capabilities of the DSD 440 and the diagnostic tests supplied by Data Systems Design which can be used with a computer and console. Recommendations are made for initial DSD 440 acceptance testing and for subsequent troubleshooting of suspected system malfunctions.

#### DSD 440 SELF-TESTS

There are two types of DSD 440 self-tests. First, there are those that execute automatically on power-up. Then, there are user selectable self-tests. Both types should be performed when the DSD 440 is first installed or whenever a computer system fault condition is suspected. These tests should be performed before any computer resident diagnostics are attempted.

#### AUTOMATIC SELF-TESTING

The microprocessor in the DSD 440 controller executes several system hardware tests following power-up or an initialization. These tests are executed even when the system is in a user selectable, stand alone, self-test mode. You cannot inhibit these tests from executing, and you cannot operate the system should one of them detect a malfunction.

Just before each of these controller test routines is executed, the microprocessor writes the error code associated with the failure of that particular test in LED indicators 5 through 8 on the controller board. In the event the test detects a malfunction, the controller microprocessor will halt leaving the error code displayed. The green RUN LED (labelled "LED 9") indicator and LED indicators 1, 3 and 4 will be off. LED indicator 2 will be on. LED indicators 5, 6, 7 and 8 contain the error code. The error code interpretations are in Table 5-1.

Should an error occur and the solution to the problem is not obvious, try cycling the main power several times. If the error persists, call the Data Systems Design Customer Service Department for assistance.

In addition to the controller self-tests, the DSD 440 has self-test routines built into the bootstrap program on the LSI-11 and PDP-11 interface cards. These routines and their error conditions are described in Chapter 3 in the section titled "System Bootstrapping".

## USER SLECTABLE SELF-TESTS

The DSD 440 controller may be operated in two modes. In NORMAL mode, the DSD 440 controller is connected to a host computer through an interface module. The user program and/or operating system software controls all of the functions performed by the data storage system.

In "HYPERDIAGNOSTIC" mode, the interface cable is disconnected from the DSD 440 chassis and the controller microprocessor executes routines which are selected by the eight position DIP-switch on the controller module. These routines are started and stopped by applying and removing AC power. To select a test routine, the main AC power switch located on the rear of the chassis must be in the OFF position. You can then select the individual "HYPERDIAGNOSTIC" routine by changing the settings of the DIP-Switch. To initialize a test, resume power.

These DIP-Switch selectable routines were named "HYPERDIAGNOSTICS" because they are a decisive improvement over standard self-test procedures. They perform the following types of diagnostic functions.

- Simplified acceptance tests requiring no special test equipment for operation.
- Drive-independent controller self-tests.
- Extensive drive utility routines and composite system exercisers.
- Simplified disk drive alignment and adjustment procedures.

In addition, nine LED indicators on the DSD 440 controller module designate the status of "HYPERDIAGNOSTIC" operation.

The major benefit of the "HYPERDIAGNOSTICS" is in their stand-alone ability to easily verify correct disk drive system and controller operation. With this feature, faults in subsystems may be quickly isolated when a total computer system malfunctions.

## TEST SELECTION INDICATOR LIGHTS AND DIP-SWITCH LOCATION

Figure 5-1 shows the relative location of the nine indicator LEDs and the DIP-Switch test selector on the controller module.

Note that two of the LEDs are green and the remaining seven are red. LED 1 is green, and is located nearest the DIP-Switch. LEDs 2 through 8 are all red, and are located next to LED 1. The meanings of LEDs 1 through 8 will vary according to whether the system is in NORMAL or "HYPERDIAGNOSTIC" mode, and whether the microprocessor is running or halted. LED 9, which also is green, is ON when the microprocessor is running and OFF when the microprocessor is halted. LED 9 referred to as the RUN LED.

### NOTE

If there is ever doubt as to whether a particular LED indicator is ON or OFF, view the indicator from directly on axis.

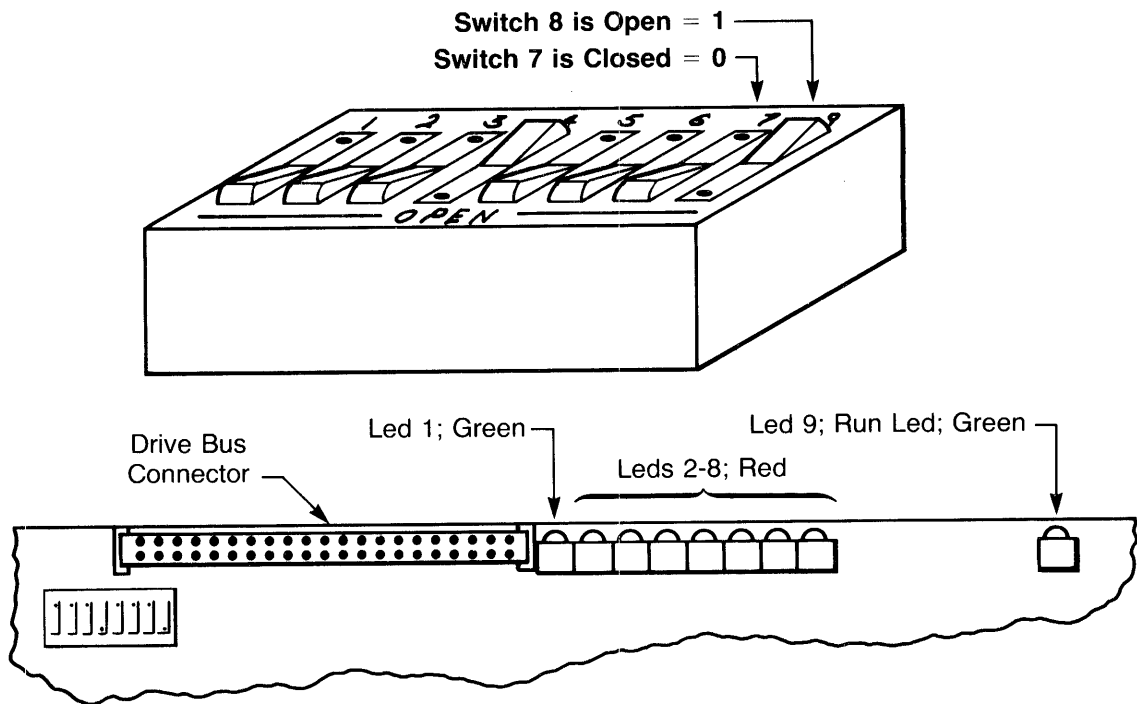


Figure 5-1. Controller Module Indicator Lights and DIP-Switch Orientation

The drive activity LEDs are mounted in the eject button on the front of each disk drive. These indicate when the head is loaded against the media, thus indicating the drive door should not be opened. When the system is operating in NORMAL mode, these LEDs may flash on and off about once every second to indicate an error condition. This flashing continues until an INIT occurs or two minutes have elapsed since the occurrence of the error.

The 8 position DIP-Switch is used to select options in Normal Mode operation and try to select tests to be run in "HYPERDIAGNOSTIC" Mode. The key in Figure 5-2 shows how the switches are numbered, and which physical position of a switch corresponds to a binary "1" and which position corresponds to a "0".

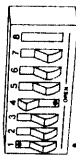
#### READING THE INDICATOR LIGHTS

The nine indicator light emitting diodes (LEDs) are on the controller module. They are used to encode different information at various times. The encoding algorithms explained in this chapter are summarized in Figure 5-3.

# DSD 440 HyperDiagnostics™

## CONTROLLER DIP-SWITCH SETTINGS

Key:



These Dip Switches are found on the controller board inside the DSD 440 chassis

This indicates a switch position to be selected

This shows switch 4 in the "1" or OPEN position.

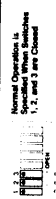
The shows switch 1 in the "0" or CLOSED position.

Note: Switch 6 and switch 7 are not used

Assistance is available from DSD Customer Service.\*  
CORPORATE HEADQUARTERS  
EASTERN REGION SALES  
10000 Highway 101  
San Jose, CA 95131  
(408) 946-5900  
(817) 789-7800

\*For service outside the United States contact the nearest DSD International Distributor.

### Normal Operation



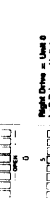
Normal Operation:  
Switches 1, 2, and 3 are Closed



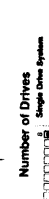
RX01/RX02 Mode  
EN RX01 jumper should be removed on interface



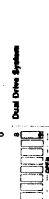
RX07: Compatible Operation  
EN RX07 jumper should be installed on interface



Mapping of Drive Unit Numbers  
Left Drive = Unit 0  
Right Drive = Unit 1

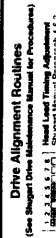


Number of Drives  
Single Drive System

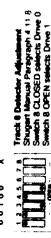


Dual Drive System

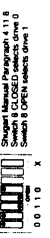
### HyperDiagnostics



Drive Alignment Routines  
(See Startup Drive Manual for Procedures)  
Need Local Trackhead for Procedures  
Startup Manual Paragraph 4.1.3  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



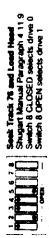
Track 8 Detector Adjustment  
Startup Manual Paragraph 4.1.9  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects Drive 1



Seek Track 81 and Load Head  
Startup Manual Paragraph 4.1.8  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



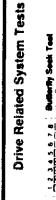
Seek Drive 81 and Load Head  
Startup Manual Paragraph 4.1.9  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



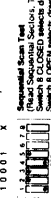
Seek Track 36 and Load Head  
Startup Manual Paragraph 4.1.10  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



Seek Track 76 and Load Head  
Startup Manual Paragraph 4.1.10  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



Battery Backup Test  
(No Indication)  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



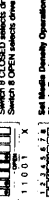
Sequential Scan Test  
(Read Stoppage Switches, T, a, b, c)  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



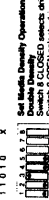
Battery Scan Test  
(Read Stoppage Switches, T, a, b, c)  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



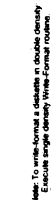
Inoperative Write/Read Test  
(Read Stoppage Switches, T, a, b, c)  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



Write-Format & Initialize & Single Density Using Recommended Switching  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



Set Media Density Operation  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1



Set Media Density Operation  
Switch 8 CLOSED selects drive 0  
Switch 9 OPEN selects drive 1

Note: To write-normal a density in double density format, switch 8 must be OPEN.  
2. Enable double density Set Media Density.  
3. Switch 9 Selects Drive  
0 Drive 0  
1 Drive 1

### Drive Independent System Tests



Test Read/Write Controller



Test Drive Encoder Loop and CRC Checksum (Optional Hardware)



Verify Printer Operations of the Interface and Drive Belt Cables



Continuously Execute Hardware Self-Test Microcode



Dip-Switch / LED: E-Test



General System Exercisers  
General Exerciser Routine Starting with three-Cycle Single Density  
Switch 8 OPEN selects Drive 0  
Switch 9 OPEN selects Drive 1



General Exerciser Routine  
Switch 8 CLOSED selects Drive 0  
Switch 9 OPEN selects Drive 1

Figure 5-2. Controller DIP-Switch Settings





## NORMAL MODE

In normal mode, the DSD 440 controller is connected to a host computer through an interface module. The user program and/or operating system software controls all of the functions performed by the data storage system.

### DIP-Switch Settings for Normal Operation

The eight rocker switches in the controller module DIP-Switch have the following meanings when the system is operated in NORMAL mode:

#### Switch 1, Switch 2, and Switch 3

These must all be 0 (top depressed) for normal mode. If any are non-zero then "HYPERDIAGNOSTIC" mode is selected.

#### Switch 4

Set switch 4 to 0 (top depressed ) for Mode 2 (RX02 mode). Set switch 4 to 1 (bottom depressed) for Mode 1 (RX01 mode).

#### Switch 5

When switch 5 is a 0, the left hand disk drive is drive 0 and the right hand disk drive is drive 1. When switch 5 is a 1, the left hand disk drive is drive 1 and the right hand disk drive is drive 0. This switch permits an easy re-mapping of the right hand disk drive to drive 0 in the event that the left hand drive fails.

#### Switch 6, Switch 7

These are not used in NORMAL mode.

#### Switch 8

When switch 8 is a 0, the DSD 440 controller recognizes one disk drive. When switch 8 is a 1, the controller recognizes two drives.

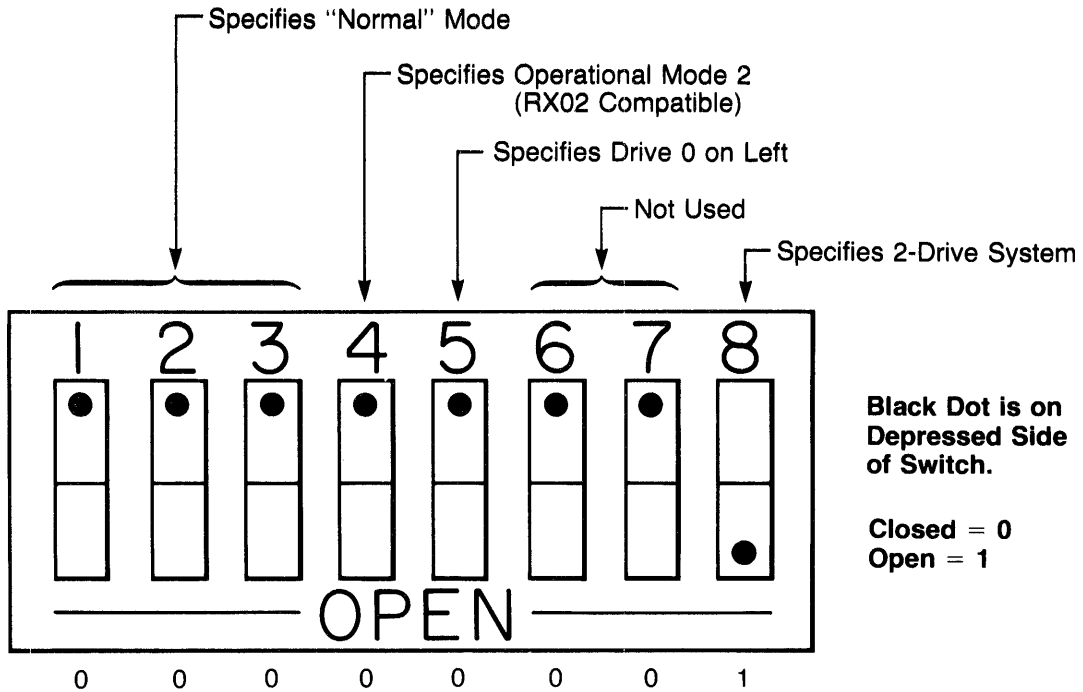
As shown in Figure 5-4, the standard DSD 440 system is shipped with all switches in the 0 position EXCEPT Switch 8. This means: NORMAL operation, Mode 2 (RX02 compatible), normal drive mapping, and two disk drives.

### Indicator Light (LED) Definitions During Normal Operation

This section defines the meanings of LED 1 through LED 8 when the DSD 440 system is connected to the host computer (NORMAL mode). The chassis cover must be moved to see these LED indicators.

LED 1, when on, indicates that the DSD 440 system is operating in NORMAL mode. LED 1 is green.

LED 2, when on, indicates that the DSD 440 controller microprocessor is waiting for the host computer to issue a new command, write a parameter to the data buffer register, or transfer a data byte from or to the data buffer register.



**Figure 5-4. DIP-Switch Setting When DSD 440 is Shipped**

LED 3, when on, indicates that the controller is writing on a diskette.

LED 4, when on, indicates that the controller is reading from a diskette.

LED 5 through LED 8 are used to display an "error class" code. When a LED is on, this corresponds to a binary 1. When a LED is off, it corresponds to a binary 0. The code bits read from left to right where LED 5 is the most significant bit and LED 8 is the least significant bit. Each error class code represents a grouping of one or more definitive error codes that are passed to the main computer on command. The errors are grouped into 16 classes so that all possible errors can be coded visually using only four LED indicators. These error class codes are listed in Table 5-1. The Error Register (ERREG) codes referenced in Table 5-1 are more fully described in Table 5-2.

**Table 5-1. Error Class Codes in NORMAL Mode (Both Green LEDs On)**

<u>BINARY CODE SEEN ON LED#S 5678</u>	<u>INTERPRETATION</u>
0000	No errors have occurred since power on or last INIT
0001	Operator Error-Write protect violation (ERREG = 100) -or drive not ready (ERREG = 360)

Table 5-1. Error Class Codes in NORMAL Mode (Both Green LEDs On)  
(Continued)

<u>BINARY CODE SEEN ON LED#S 5678</u>	<u>INTERPRETATION</u>
0010	Programming error-density/key (ERREG = 240 OR 250)
0011	Programming error-drive/track address (ERREG =040)
0100	Programming error-word count/NXM (ERREG = 230 OR 350)
0101	Indeterminate density (ERREG = 260)
0110	Seek error (ERREG = 150)
0111	Header CRC error (ERREG = 140)
1000	Data CRC error (ERREG = 200)
1001	Sector unrecoverable (ERREG = 070, 120, 130, 160, OR 170)
1010	No read data signal present (ERREG = 110)
1011	Read/Write contr. failure (ERREG = 220, 320, OR 330)
1100	Master controller failure (ERREG = 340)
1101	Drive failure (ERREG = 010, 020, 030, 050)
1110	Interface parity error (ERREG = 210)
1111	Write or write-format failure (ERREG = 270 or 370)

The error class code is displayed in the LEDs as soon as the error is detected. The code resets to zero if the power is switched off and then on or if an INIT is generated over the IBUS cable. The drive activity LEDs are also used to indicate the occurrence of errors. Whenever bit 15 of the control and status register indicates the occurrence of an error (other than density error), the controller microprocessor flashes the activity LED of the drive associated with the error about every second. This flashing stops when either a system initialize is forced by the host computer, or after approximately two minutes elapse.

#### "HYPERDIAGNOSTIC" MODE

The "HYPERDIAGNOSTICS" are used when you want to adjust, exercise, or test your controller or drives independently of the host computer system and associated software.

The DSD 440 chassis need only be connected to an AC power outlet to run the "HYPERDIAGNOSTICS". You select particular tests and particular disk drives using the DIP-Switch on the controller board. Test results are designated through a combination of the nine indicator LEDs on the controller board in the DSD 440 chassis and, when needed, on an oscilloscope. After the switch and LED conventions are explained, the details of each "HYPERDIAGNOSTIC" routine will be discussed.

Table 5-2. Error Register Codes

<u>CODE ERROR (OCTAL)</u>	<u>CLASS</u>	<u>MEANING</u>
010	D	Drive 0 failed to home on INIT
*020	D	Drive 1 failed to home on INIT
030	D	Encountered track 0 while stepping in on INIT
040	3	Invalid drive or track address specified
050	D	Track 0 encountered unexpectedly
070	5	Requested sector not found in two revolutions
100	1	Attempted to write on protected diskette
110	A	No read data signal present
120	9	Preamble not found
130	9	Preamble found but no ID address mark followed
*140	7	Header CRC error
150	6	Track or head address mismatch
160	9	Too many tries for an ID address mark
170	9	Preamble found but no data address mark followed
200	8	Data CRC error
210	E	Interface parity error
220	B	Read/Write controller self test failure
230	4	Invalid word count specified
240	2	Density error
250	2	Wrong key for set media density
260	5	Indeterminate density
270	F	Read/Write controller write-format failure
320	B	Read/Write controller write failure
330	B	Read/Write timed out on reset
340	C	Master controller out of sync with Read/Write controller
350	4	Non-existent memory encountered during DMA
360	1	Drive not ready during write-format command
370	F	AC low abort of write or write-format

\* These codes do not assert error in RXCS, all others do

NOTE

With the exception of the cable orientation test, all the "HYPERDIAGNOSTIC" tests should be executed with the interface bus (IBUS) cable disconnected from the DSD 440 chassis.

## DIP-Switch Settings for "HYPERDIAGNOSTIC" Operation

### Switch 1 through Switch 5

These are used to select the desired test. Any time switches 1, 2, and 3 are all zeroes, the microprocessor assumes "NORMAL" mode operation. Switch 1 is the most significant bit and switch 5 is the least significant bit.

### Switch 6, Switch 7

These switches are used in "HYPERDIAGNOSTIC" mode only during the DIP-SWITCH/LED test.

### Switch 8

When the switch is a 0, drive 0 is selected; when it is a 1, drive 1 is selected. Not all the tests involve a drive, so in some cases the position of switch 8 is irrelevant.

The general exerciser tests are capable of testing more than one drive and should be used for initial acceptance testing. These two tests (switch codes 11110 and 11111) interpret switch 8 as the number of drives to be exercised. If switch 8 is a 0, only drive 0 will be exercised. If switch 8 is a 1, both drive 0 and drive 1 will be exercised.

### NOTE

There is no drive mapping function available in "HYPERDIAGNOSTIC" mode. This function is only available when the system is operating in NORMAL mode.

The following is the detailed procedure for running a particular "HYPERDIAGNOSTIC":

- 1) Remove power from the controller/drive subsystem using the AC switch on the rear of the chassis. Do not tamper with the DIP-Switch settings while power is on EXCEPT when explicitly directed to do so for a particular test.
- 2) Set the eight switches to select the desired test, and in some cases, the desired drive. The DIP-Switch configurations are shown in Figure 5-2.
- 3) Use a pointed object (such as a ball point pen) to depress the rocker switches and set them to the desired setting.
- 4) Restore power.

A technique that can be used to confirm your switch settings is the DIP-SWITCH/LED "HYPERDIAGNOSTIC". The code for this test is (10000). Once the

microprocessor recognizes this test code, it reads the DIP-Switch and echoes the setting in the LEDs. Once this test is running, change the DIP-Switch to the desired questionable setting and verify the setting in the LEDs.

If the LEDs reflect the correct switch setting, the specific function indicated by the switches can be executed by simply powering the unit down, and then up again.

### Indicator Light (LED) Definitions During "HYPERDIAGNOSTIC" Operation

Except where noted otherwise, the LED indicators function as follows:

- LED 1 and LED 2 are off to indicate that the microprocessor is in "HYPERDIAGNOSTIC" mode and is not executing any of the hardware self-test routines.
- LED 3 is on to indicate the system is writing on a diskette. In general, you should not turn off the power while LED 3 is still on, as a FORMAT routine or SET MEDIA DENSITY may be in progress.
- LED 4 is on to indicate the system is reading from a diskette.

As with the hardware self-tests, the controller microprocessor halts whenever it detects an error. You can determine when the microprocessor is halted by observing the green "RUN" LED shown in Figure 5-1. An error code is displayed in LEDs 5 through 8 when the microprocessor detects an error and halts. Errors that involve a host computer interface, such as non-existent memory and parity errors, do not occur during execution of any of the "HYPERDIAGNOSTICS".

Table 5-3 shows the code interpretations. The activity LED of the drive selected when the failure was detected is left on.

Table 5-3. "HYPERDIAGNOSTIC" Error Code Interpretation

<u>BINARY CODE</u> <u>LED 5678</u>	<u>INTERPRETATION</u>
0000	No errors have occurred since power on or last INIT
0001	Operator error - write protect violation (ERREG = 100) - or drive not ready (ERREG = 360)
0010	This error code not currently assigned
0011	IBUS cable backwards or interface module without power
0100	Drive bus cable is installed backwards
0101	Indeterminate density (ERREG = 260)
0110	Seek error (ERREG = 150)
0111	This error code not currently assigned

Table 5-3. "HYPERDIAGNOSTIC" Error Code Interpretation (continued)

<u>BINARY CODE</u> <u>LEDs 5678</u>	<u>INTERPRETATION</u>
1000	*Data CRC error (ERREG = 200)
1001	Sector unrecoverable (ERREG = 070, 120, 130, 160, OR 170)
1010	Drive read signal lost (ERREG = 110)
1011	Read/Write controller failure (ERREG = 220, 320, OR 330)
1100	Master controller failure (ERREG = 340)
1101	Drive failure (ERREG = 010, 020, 030, 050)
1110	Data pattern read not the same as pattern written
1111	Write or write-format failure (ERREG = 270 or 370)

\* Most CRC errors are attributed to worn media. Switch diskettes and restart the "HYPERDIAGNOSTIC" by cycling the AC power switch.

#### TEST STRATEGY USING "HYPERDIAGNOSTIC"

A substantial portion of a DSD 440 system can be checked using only the "HYPERDIAGNOSTIC" routines included in the controller microcode. No host computer or interface module is required to perform the initial test procedure described below. This procedure is used to confirm that the controller/drive subsystem is fully operational, but any system malfunction related to an interface module or interface cable will not be detected. If your system is either unable to boot or unable to run the programs on the diagnostic diskette, the "HYPERDIAGNOSTICS" are an excellent way to determine if the problem lies in the interface module/cable or in the controller/drive subsystem. The following procedure describes how to check your DSD 440 with the "HYPERDIAGNOSTICS":

- 1) VERIFY THE POWER SWITCH IS OFF.
- 2) Remove the top cover by turning the three fasteners at the rear of the cover and sliding it straight off.
- 3) Connect the AC power cord. Make sure the interface cable is not connected to the DSD 440 chassis.
- 4) Place write-enabled, blank, formatted diskettes in both drive 0 and drive 1.
- 5) Using a ball point pen, or similar object, place the code 10000001 in the DIP-Switch. This particular code selects the DIP-SWITCH/LED "HYPERDIAGNOSTIC" test.



- 6) Power-up the unit by flipping the power switch ON. The RUN light or LED (Light Emitting Diode) should remain on. Lights 1 through 8 should echo the settings of switches 1 through 8 by being lighted for a binary "1" and off for a binary "0".
- 7) Place the code 11110001 (GENERAL EXERCISER TEST) in the DIP-Switch. Verify that lights 1 through 4 and 8 are on and 5 through 7 are off.
- 8) Start the general exerciser test by turning power off and then back on. The test will start on drive 1. After about 6-1/4 minutes, the test should switch over to drive 0.

The test will continue alternating between drives until power is removed or an error is detected. If an error occurs, the microprocessor will HALT, causing the RUN light to be extinguished. An error code will be displayed in LEDs 5 through 8. The controller/drive subsystem should be considered fully functional if the GENERAL EXERCISER TEST can be run for a half hour or longer without the occurrence of an error halt. Should an error occur, make sure you are using good quality diskettes. If you believe the unit is malfunctioning, contact the Data Systems Design Customer Service Department for assistance.

#### NOTE

The hardware self-tests and the "HYPERDIAGNOSTICS" report errors by writing an error code in LEDs 5 through 8 and then halting. The microprocessor executes the hardware self-tests before it gets to the "HYPERDIAGNOSTIC" routine encoded in the switches. As shown in Figure 5-3, LED 2 will be ON following an error detected by the hardware self-test routines. LED 2 is OFF following an error detected by most of the "HYPERDIAGNOSTIC" routines.

LEDs 5 through 8 have a different meaning when the microprocessor is executing "HYPERDIAGNOSTIC" routine but has not yet detected any error. If the RUN LED is still ON, LEDs 5 and 6 encode density, and LED 8 encodes selected drive. Coding is shown in Table 5-4.

Table 5-4. Interpretation of LEDs 5, 6 and 8 During Execution of "HYPERDIAGNOSTICS" While RUN LED is ON

LED 5	LED 6	DENSITY
Off	Off	Indeterminate density (drive probably not ready)
Off	On	IBM 3740 single density
On	Off	DEC double density
LED 8		SELECTED DRIVE
Off		0
On		1

## INDIVIDUAL "HYPERDIAGNOSTIC" TESTS

There are five types of "HYPERDIAGNOSTIC" tests. Each type is described below:

- General System Exercisers
- Drive Alignment Routines
- Drive Independent System Tests
- Drive Related System Tests
- Drive Utility Routines

GENERAL SYSTEM EXERCISERS. These tests are designed to exercise all parts of the DSD 440 controller/drive subsystem. There are two tests of this type:

Unlike the other "HYPERDIAGNOSTIC" tests, this and the following routine operate on multiple drives.

- GENERAL EXERCISER STARTING WITH WRITE-FORMAT SINGLE DENSITY (11110)

Switch 8 is used to specify the drive to be exercised. As an example, if switch 8 were set to a 1, the general exerciser would first exercise drive 1, then drive 0, then drive 1, etc. If switch 8 were set to a 0, only drive 0 would be exercised. It is important that all drives to be exercised are loaded with write-enabled diskettes. The sequence of operations is listed below:

- 1) Execute hardware self-tests (no drives involved).
- 2) Write-format selected drive according to IBM 3740 single density standard.
- 3) Do sequential read of all sectors on selected drive.
- 4) Do sequential write/read of all sectors on selected drive.
- 5) Do butterfly read of all sectors on selected drive.
- 6) Do a double density set media density on selected drive.
- 7) Do a sequential read of all sectors on selected drive.
- 8) Do sequential write/read of all sectors on selected drive.
- 9) Do a butterfly read of all sectors on selected drive.
- 10) Do a single density set media density on selected drive.
- 11) Determine next logical drive unit, and if that unit has not already been write-formatted once, go to step (2); other wise go to step (3).

This general exerciser test takes about six minutes per drive, including the write-format cycle that occurs during the first pass for each drive.

- GENERAL EXERCISER (11111)

This test is similar to the previous one (11110), except that single density write-format routine indicated in step (2) is not executed. This test requires about five and one half minutes per pass on each drive.

DRIVE ALIGNMENT ROUTINES. These routines are used with an alignment diskette available from the drive manufacturer. Details regarding this diskette and the actual alignment procedures are in Appendix D. These routines execute even if no diskette is installed and/or the drive door is left open. No ERROR-HALT will occur if the selected drive is not ready. The following is a description of the six test routines executed by the microprocessor:

- HEAD LOAD TIMING ADJUSTMENT ROUTINE (00100)

This routine starts by moving the selected drive's read/write head to track 00. Once there, the head is loaded and unloaded at approximately 5 times per second. The head is loaded for 100 milliseconds, and then unloaded for 100 milliseconds before the cycle is repeated. The routine is terminated by disconnecting AC power from the chassis.

Paragraph 4.6.3 in Appendix C specifies this type of test routine.

- TRACK 00 DETECTOR ASSEMBLY ADJUSTMENT ROUTINE (00101)

This routine starts by moving the selected drive's read/write head to track 00. The head is then alternately moved between track 01 and track 02 about once every 70 milliseconds. The head is loaded during this test. The routine is terminated by disconnecting AC power from the chassis.

Paragraph 4.11.8 in Appendix C specifies this type of procedure for track 00 flag adjustment.

- SEEK TRACK 01 AND LOAD HEAD (00110)

This routine starts by moving the selected drive's read/write head to track 00. Next, the head is positioned at track 01 and loaded against the media. The head remains loaded until power is removed.

Paragraph 4.11.8 in Appendix C specifies this type of routine for track 00 flag adjustment.

- SEEK TRACK 02 AND LOAD HEAD (00111)

This routine starts by moving the selected drive's read/write head to track 00. Next, the head is positioned at track 02 and loaded against the media. The head remains loaded until power is removed.

Paragraph 4.11.8 in Appendix C specifies this type of routine for track 00 flag adjustment.

- **SEEK TRACK 38 AND LOAD HEAD (01000)**

This routine starts by moving the selected drive's read/write head to track 00. Next, the head is positioned at track 38 and loaded against the media. The head remains loaded until power is removed.

This routine is used during the HEAD RADIAL ADJUSTMENT described in paragraph 4.11.3 of Appendix C. The alignment diskette is required to perform this routine.

- **SEEK TRACK 76 AND LOAD HEAD (01001)**

This routine starts by moving the selected drive's read/write head to track 00. Next, the head is positioned at track 76 and loaded against the media. The head remains loaded until power is removed.

This routine is used during the READ/WRITE HEAD AZIMUTH ALIGNMENT described in Appendix C. The SA-120 alignment diskette is required to perform this routine.

If the results of these test routines indicate that the drives require alignment, follow the instructions in Appendix C. To take the drives out of the DSD 440 system, remove the screws from the bottom of the chassis so that the drives can be turned on their side. Many of the adjustment screws and oscilloscope test points are located on the underside of the drives.

DRIVE INDEPENDENT SYSTEM TESTS. These tests are designed to exercise those parts of the DSD 440 controller which are not drive-dependent. There are five tests of this type:

- **TEST READ/WRITE CONTROLLER (01100)**

This routine causes the read/write controller hardware to be continuously cycled through its internal self-test microcode. Run this test if you believe the read/write controller hardware is not performing reliably. Should this test generate an error, the code is 1011. LED 2 will also be ON if this error is reported.

- **TEST PHASE LOCKED LOOP/CRC GENERATOR (01101)**

This routine checks the operation of the phase locked loop (PLL) circuitry by counting the number of PLL CLOCK cycles that occur during a 50 millisecond interval. This test determines if a READ problem is being caused by the phase locked loop circuitry. If it is, the error code is 1010.

**NOTE**

LED 2 will be on if this error is reported.

The second half of this test verifies that the CRC generator/checker and serial data path are functioning properly. If this test detects a malfunction, the error code is 1100. LED 2 will also be on.

- TEST CABLE ORIENTATION (01110)

This test is used to verify that both the interface bus cable (connecting the controller to the interface module) and the drive bus cable (connecting the controller to the drives) are installed correctly. The interface bus cable must be connected on both ends and the interface module must have power in order to run this test. The interface bus cable must be disconnected at one or both ends when running all "HYPERDIAGNOSTIC" tests EXCEPT this one. If there is an error, Table 5-4 indicates which cable is causing the problem. If there is no cable at all, this test will pass.

- HARDWARE SELF-TEST LOOP (01111)

The DSD 480 microprocessor executes the hardware self-test once after powering up. When this "HYPERDIAGNOSTIC" routine is selected, the microprocessor keeps executing the hardware self-test indefinitely until either power is removed or an error is detected. LEDs 5, 7, and 8 flash on and off when this routine is executing error-free.

- DIP-SWITCH/LED TEST ROUTINE (10000)

This routine is used to determine if the microprocessor can read all 8 switches in the DIP-Switch, and illuminate LED's 1 through 8. The routine reads the DIP-Switch and writes that byte to the LED bank. For example, if Switch 2 is in the 1 position, then LED 2 is on. If Switch 2 is in the 0 position, then LED 2 is off.

Since the microprocessor is executing this loop continuously, the state of a LED should appear to change immediately following a switch position change.

This is the only case in which the settings of the DIP-Switch should be changed with power on. Changing them at other times can cause erroneous operation.

These tests are terminated by removing power.

DRIVE RELATED SYSTEM TESTS. These tests are designed to exercise the drives on the assumption the DSD 480 controller is fully operational. There are five tests of this type:

- BUTTERFLY SEEK TEST (10001)

This routine starts by moving the selected drive's read/write head to track 00. The head positioner is then moved back and forth in a "butterfly" pattern. This pattern consists of the following series of tracks: 76, 01, 75, 02, 74, 03 . . . After one complete cycle, the microprocessor tries to move the head positioner to track 00. If the track 00 signal is not asserted, the error code is reported in the LED indicators and the microprocessor halts. If the track 00 signal is asserted, the test is repeated.

- SEQUENTIAL SCAN TEST (10010)

This routine starts by moving the selected drive's read/write head to track 00. The head is then loaded and the media density is determined and displayed in the LEDs. The controller then sequentially reads every sector of every track. If no errors occur, the test is repeated. If an error does occur, the microprocessor halts and the error code is displayed in LEDs 5 through 8. The meanings associated with the 16 possible error codes are shown in Table 5-4. The code displayed in LEDs 5 through 8 is not an error code UNLESS the green LED (LED 9) is off and the head positioner is not moving.

- BUTTERFLY SCAN TEST (10011)

This routine is similar to the sequential scan test, except for the sequence of tracks read. This test takes longer than the sequential scan test to read the same total number of sectors because of the added positioner step and head load time. This test detects problems associated with seeking and/or reading.

NOTE

The data on the diskette during the above two tests may be any validly formatted single or double density data. Only a CRC error or other hardware detected error will be reported, as the data itself is ignored.

- SEQUENTIAL WRITE/READ TEST (11000)

This routine starts by moving the selected drive's read/write head to track 00. Next, the density of the write-enabled diskette inserted in the drive is determined. The routine then sequentially writes pseudo-random data on every track and sector of the diskette, in the appropriate density. After writing, every track and sector on the diskette is sequentially read and compared to what was written. Any error encountered is identified in the LEDs when it occurs and the machine halts. The write cycle only happens once. The read cycle is repeated indefinitely until power is disconnected or an error is detected. LEDs 3 and 4 are used to determine if the routine completes the write cycle.

These tests should be executed whenever a drive is replaced or realigned.

DRIVE UTILITY ROUTINES. These tests are designed to exercise the formatting capability of the DSD 440. There are three tests of this type:

- WRITE-FORMAT DISKETTE IN IBM 3740 Single Density Format (11001)

This routine starts by moving the selected drive's read/write head to track 00. Next the entire write-enabled diskette is formatted according to the IBM 3740 single density standard. The sector addresses are written sequentially on each track. After all tracks have been written, control is transferred directly to the sequential scan test, which then keeps reading the diskette indefinitely until an error is detected or power is disconnected. LEDs 3 and 4 are used to determine when the write cycle is completed.

- SET MEDIA TO SINGLE DENSITY (11010)

This routine starts by moving the selected drive's read/write head to track 00. Next, every sector on the write-enabled diskette is written with a single density data address mark, 128 bytes of zeroes, and 2 CRC bytes. Unlike the previous write-format routine, this one does not modify the sector headers. Control is transferred to the sequential scan test as soon as all sectors are written. LEDs 3 and 4 are used to determine when the writing has stopped and the reading has begun.

- SET MEDIA TO DOUBLE DENSITY (11011)

This routine starts by moving the selected drive's read/write head to track 00. Next, every sector on the write enabled diskette is written with a double density data address mark, 256 bytes of DEC modified frequency modulation (MFM) zeroes and 2 Cyclic Redundancy Check (CRC) bytes. This routine does not modify the sector headers. Control is transferred to the sequential scan test as soon as all sectors are written. LEDs 3 and 4 are used to determine when the writing has stopped and the reading has begun.

These tests can also be used to reformat magnetically damaged diskettes without a computer.

## COMPUTER RESIDENT DIAGNOSTICS

There are two types of computer resident tests. Both are designed to be used with a computer, flexible disk storage system and a console terminal. The first type are the diagnostics supplied by Digital Equipment Corporation for the RX01 and RX02. To use these tests refer to the appropriate DEC manuals.

The second type of computer resident diagnostic is supplied by Data Systems Design on the diagnostic diskette shipped with your system. There are two tests of this type. Both are interactive and are accessed from the system console. FRD440 is designed for LSI-11 and PDP-11 based systems. This diagnostic also includes other useful system utilities such as diskette formatting routines. VEP 210 is designed for PDP-8 based systems. One or the other of these two diagnostic programs should be used as part of the initial acceptance testing of the DSD 440.

### FRD440 DIAGNOSTIC PROGRAM FOR LSI-11 AND PDP-11 SYSTEMS

All DSD 440 systems with an LSI-11 or PDP-11 interface board are shipped with a diskette containing an interactive diagnostic program called FRD440. This section explains the operation of this comprehensive set of tests and utility programs. Note that the FRD440 program has been replaced by the FLPEXR diagnostic program, which is covered in Appendix J. Recent purchasers should refer to this appendix for diagnostic programs.

### PROGRAM LOADING AND MONITOR PROTOCOL

FRD440 requires a standard console device, an LSI-11 or PDP-11 computer and at least 12K words of memory. Loading FRD440 can be accomplished by two methods. One method is to bootstrap the diagnostic diskette. This loads FRD440 into memory automatically.

The other method requires an RT-11 operating system. The FRD440 diagnostic diskette has an RT-11 compatible directory and file space. The files on the diagnostic diskette can be accessed using standard RT-11 procedures. For example, FRD440 can be run from an RT-11 system by typing:

```
.RU <DEV:>FRD440    <CR>
```

where <DEV:> might be DX0:, DX1:, DY0:, DY1: as appropriate.

Once the FRD440 diagnostic program has been loaded into memory, the diagnostic diskette should be removed from the drive so it is not erased. The information on the diagnostic diskette is recorded in single density format. This, and the fact that both bootstrap and diagnostic programs handle RX01 and RX02 protocols, make the FRD440 diagnostic diskette useful on any DEC compatible disk system.

After FRD440 is loaded into memory, a brief description is displayed on the terminal which includes the version number of the program and a memory map. This memory map indicates the ranges of the address space which responds with SSYNC (or BRPLY) when accessed by the host computer. A list of all the available commands may be obtained by typing an "H" (HELP).

Two high quality, write-enabled formatted diskettes of the same density should be installed in the DSD 440 drives before proceeding with any of the tests.

FRD440 types "<CRLF>#" when starting, and then the program attempts an INIT (initialize) instruction. When the INIT cycle is successful, the program types the prompt word: "MODE:". This prompt string allows the operator to input a command. Each of the possible commands is described below.

Legal responses to "MODE:" are listed below. Only the characters enclosed in parenthesis need to be typed. The parenthesis should NOT be typed:

```
(A)CCEPTANCE TEST  
(S)H)ORT  
(V)ERIFY  
(R)EE)NTER ACCEPTANCE TEST  
(H)ELP  
(M)A)P ADDR  
(M)P)X-MAP ADDR  
(F)ILL-EMPTY  
(S)EQW) /R  
(S)EQR)D  
(R)N)D R/W  
(R)D) RANDOM  
(S)C)AN  
(R)A)NDOM  
(S)K) RANGE  
(S)A) 125  
(S)T)ATUS  
(R)E)S) STATUS  
(S)V)-STATUS
```



(REC)OVER STATUS  
(DUMPC)  
(S)INGLE  
(T)AP  
(L)INIT  
(SET )UNIT  
(SET-)TRACK  
(SEC)TOR INCREMENT  
(SETW)COUNT  
(DE)NSITY LOCKUP  
(I)NTERRUPT STT  
(SETD)EVICE  
(ME)DIA-DEN-ALL  
(SETM)EDIA DENSITY  
(X)FORMAT REALLY  
(DU)PLICATE  
(C)OMPARE  
(DUMPO)  
(DUMPB)  
(DUMPA)

The program will fill in the remaining characters and then proceed to execute the function.

#### FRD440 PROGRAM FUNCTIONAL MODES OF OPERATION

This section describes each functional mode of interactive operation for the FRD440 diagnostic program. The purpose of each test and all communication protocol with the operator is also described.

- ACCEPTANCE TEST

The ACCEPTANCE TEST is generally used to verify that a disk system is functioning properly after it is installed. The cumulative error status is maintained and can be observed at any time by typing "<LF>". The ACCEPTANCE TEST consists of an ordered execution of the specialized tests described below. It runs indefinitely unless you stop the test by typing "CTRL R".

The sequence of tests is: SCAN, SEQUENTIAL WRITE, SEQUENTIAL READ, ENABLE INTERRUPTS, SEQUENTIAL WRITE, SEQUENTIAL READ, FILL-EMPTY (NO DRIVE ACTIVITY), RANDOM READ/WRITE, SEQUENTIAL READ. A single pass of SK RANGE is executed to test the drives. After each pass the data generation algorithm is changed. The acceptance test outputs short messages which indicate the active test. Error information is displayed on the console terminal as it is detected.

- SHORT

This interactive program changes the track range used by the ACCEPTANCE TEST so that only the first 9 tracks of each selected drive are tested. This SHORT ACCEPTANCE TEST is repeated until stopped.

- **VERIFY**

The VERIFY test does one pass of a SHORT ACCEPTANCE TEST, on the first 7 tracks and then resets the limit variables back to the normal default values. It then induces an automatic "CTRL P" to inhibit all but error printout and initiates the ACCEPTANCE TEST.

- **REENTER ACCEPT**

This function re-enters the ACCEPTANCE TEST after the seek test has been performed.

- **HELP**

The HELP command causes all the valid "MODE:" responses to be displayed on the console terminal. The "MODE:" prompt is typed when this function is complete.

- **MAP ADDRESS — MAP ADDR**

The MAP ADDRESS command causes a memory and device address map of your system to be displayed on the console terminal. This is the same map displayed when the FRD440 program is first loaded. In addition, the interrupt vector address associated with each disk interface is displayed. The "MODE:" prompt is typed when this function is complete.

- **MULTIPLEXED MAP ADDRESS — MPX-MAP ADDR**

The MULTIPLEXED MAP ADDRESS function causes the MAP ADDRESS command described above to be executed repeatedly until the operator types a "CTRL R". This is useful while testing the address decoding logic of memory or a peripheral interface.

- **FILL-EMPTY**

The FILL-EMPTY test checks the FILL BUFFER and EMPTY BUFFER controller commands. If the controller under test is configured in RX01 compatible mode, then the test involves only programmed I/O. If the controller is configured as an RX02, the controller does FILL/EMPTIES into three different buffers so as to verify proper operation of all possible address bits. FILL/EMPTIES are done in both densities covering all possible word counts. Since this test does not manipulate the drives, the DSD 440 will operate in relative silence. This test continues until you type a "CTRL R".

- **SEQUENTIAL WRITE/READ — SEQW/R**

The SEQUENTIAL WRITE/READ test writes pseudo-random data sequentially on all selected drives. The test then reads all the data and checks it. The message "WRITING" is typed on the console terminal when the test first starts writing. The message "READING" is typed when the test starts reading. This test continues until the operator types "CTRL R". It also performs a set media density operation if the diskette on a drive is not of the expected density.

## NOTE

The following three tests require a SEQUENTIAL WRITE pass be done first in order to initialize the pseudo-random data. Data compare errors are reported if this is not done.

- SEQUENTIAL READ — SEQRD

The SEQUENTIAL READ test reads the data on all selected drives sequentially and compares the data pattern against what was written. The program types "READING" at the beginning of each pass. This test continues until you type "CTRL R".

- RANDOM READ/WRITE — RND R/W

The RANDOM READ/WRITE test selects a random sector of a selected drive, then reads or writes it. It checks data when appropriate. This test continues until you type "CTRL R".

- READ RANDOM — RD RANDOM

The READ RANDOM test reads selected sectors randomly. Data is checked following each read performed. This test continues until you type "CTRL R".

- SCAN

The SCAN test reads all sectors on all selected drives sequentially and checks for CRC errors. It also determines media density. No direct data checking takes place in this test, only status is checked. After all units are scanned once, the "MODE:" prompt is displayed on the console.

- RANDOM

The RANDOM test reads randomly selected sectors on all selected drives. Only status is checked. This test continues until you type "CTRL R".

- SEEK RANGE — SK RANGE

The SEEK RANGE function is a versatile drive test that performs all possible seeks within the operator specified track and seek length boundaries. It specifies a read on the first sector that can be read on the destination track after compensating for step and head load times. Thus it is a worst case test of the drive stepper motor and had settling.

- SA 125 ALIGNMENT

The SA 125 test is for use with a special test diskette available from the drive manufacturer. Refer to Appendix C for more information. The test

determines the off-track margins and head alignment of a disk drive without using test equipment. This test verifies proper head alignment.

Another procedure and test diskette (SA 120) is used to perform a head alignment. Measurements are made by reading one of two specially written tracks on the SA 125 diskette. These tracks are formatted with the heads in alignment and with data written in increasing radial offsets from the track centerline. By determining which sectors are read correctly, the actual position of the head and the radial reliability margins can be observed. Each track has the offset pattern written twice. As the SA 125 test is executed, a graphic display indicating which sectors have been read correctly is written repeatedly to the console device.

Symbols in the display have the following meanings:

- "\*" Error reading both sectors at a particular offset.
- "L" Error reading lower numbered sector at a particular offset.
- "H" Error reading higher numbered sector at a particular offset.
- "." Both sectors read correctly.
- "<.>" Both zero offset sectors read correctly.
- "HEAD@" Specifies the calculated head position (where '-' is outwards).
- " RNG=" Total number of offsets across which the head can read correctly. Good range = 15 (octal).

**EXAMPLE:**

```
OUT = * * * . . . <.> . . . * * * - IN HEAD @ 0.0 RNG=7
```

This display indicates that the drive can read correctly those sectors with offsets of less than 4 mils. The range indicated is bad.

**EXAMPLE:**

```
OUT = . . . . . <.> . H * * * * - IN HEAD @ -5.0 RNG=10
```

This display indicates that the head is positioned too far outward.

To run this alignment verification test, type "SA" in response to the "MODE:" prompt typed by the FRD440 program. Next, the FRD440 program will ask "UNIT:". Insert a write protected SA 125 test diskette into the drive to be tested, and type the logical unit number of that drive followed by a space or a carriage return.

Next, FRD440 will type the message "TRACK: 111". To select test track 111 (octal), type a carriage return. To select the other test track, type "107" followed by a carriage return.

At this point, the SA 125 test will output maps of the radial offsets of the sectors read correctly from most outward offset (negative position) through most inward offset. The test can be terminated at any time by typing a CTRL R. Remove the SA 125 test diskette when it is not in use.

- STATUS

The STATUS function causes all the current status information including hardware errors, data errors, and pass counts to be displayed on the console terminal. Displaying status information does not reset the status counts. See the RESET STATUS function below. The "MODE:" prompt is typed when this function is complete.

- RESET STATUS - RES STATUS

The RESET STATUS function first displays all the available status counts. Next, the display will ask whether all of the status counts need resetting. You respond with a "Y", all of the error, pass, etc., counts will be reset to zero. The "MODE:" prompt is typed when this function is complete.

- SAVE STATUS — SV-STATUS

The SAVE STATUS command causes all the status counts associated with a particular drive to be written on track 0, sector 1 of the diskette in that drive. Only the SET MEDIA DENSITY commands over-write track 0, so the status data associated with each drive can be safely stored away. This function is used by the acceptance test so that it can survive a loss of main computer CPU memory without any loss of cumulative error data. The "MODE:" prompt is typed when this function is complete.

- RECOVER STATUS

The RECOVER STATUS routine performs the opposite function performed by the SAVE STATUS function. The status data stored away on track 0, sector 1 of the diskette in each drive is transferred back from the diskette to the status/counter variables in memory. The "MODE:" prompt is displayed when this function is complete.

- DISPLAY CIRCULAR OUTPUT BUFFER — DUMPC

The DUMPC function is used to display the circular output buffer associated with all console terminal output. This function is useful on systems where the console terminal is a CRT. Messages previously output can be re-examined on the console. The circular buffer can be cleared after it is displayed by this function.

- SINGLE

The SINGLE function causes an operator specified drive, track, and sector address to be read continuously. This test is useful in determining if a particular sector is prone to intermittent CRC errors. SINGLE

is also used to measure head/media wear because the head remains in contact with the media continuously. This test continues until you type "CTRL R".

- TAP

The TAP function reads repeatedly a sector you specify and unloads the head after each read operation. TAP measures head/media wear. This test continues until the operator types "CTRL R".

- LINIT

The LINIT function sends repeated INITIALIZE commands to the DSD 440. It verifies correct status to insure that each initialization instruction is executing correctly. This test continues until you type "CTRL R".

#### MODE SETTING COMMANDS

- SET UNIT

This function enables the operator to specify which drives are to be accessed by the various test functions. The default drives are units 0 and 1. The currently selected units are printed first. It prompts with "UNIT:", expecting a number between 0 and 3, inclusive. Unit numbers are accepted as long as they are valid. When a non-number is typed to a unit request, the units currently selected are prompted and the program returns to MODE.

- SET-TRACK

This function enables the operator to specify lower and upper track limits for all other test functions. The default lower track limit is track 0 and upper track limit is track 76. The "MODE:" prompt is issued after you enter the new limits.

- SET-INCREMENT

This function enables you to specify the sector increment value. The number is added to the present sector address to determine the next sector address in the functions that read multiple sectors on a single track. If this number were 1 and the diskette did not have an interleaved format, an entire revolution would be required to read each sector. On LSI-11 processors, the default increment value is 3. On PDP-11 processors the default increment value is 2. The "MODE:" prompt is issued after the new value has been entered.

- SET WORD COUNT — SETWCOUNT

The SET WORD COUNT function enables you to specify the maximum number of words transferred when the DSD 440 performs a DMA FILL/EMPTY BUFFER operation. Only the functions which do data checking use this word count variable. The FILEMP (fill empty buffer) test controls word count independently of this variable. Default word count stored in this variable is 64 words in single density and 128 words in double density.

- DENSITY LOCKUP

This function is used to inhibit switching between low and high density during the ACCEPTANCE and VERIFY test sequences. The DENSITY LOCKUP function asks you to determine whether alternating single and double density or only flag word density should be tried.

- SET INTERRUPT STATUS — INTERRUPT STT

The SET INTERRUPT STATUS function enables you to test the disk system with interrupts either enabled or disabled. If interrupts are enabled, the program ensures that an interrupt occurs whenever it is appropriate. The operator enters a 0 to disable interrupts and a 1 to enable interrupts. This function is also used in ACCEPTANCE and VERIFY to set "Interrupts Enabled" and "Interrupts Disabled".

- SET DEVICE

This function facilitates testing controllers that are not configured at the standard device I/O address and interrupt vector. It also enables the FRD440 test program to simultaneously exercise multiple controllers. The function protocol asks you for device address, interrupt vector, and flag word. If a space is typed, the program steps past that field, leaving it intact. To return to "MODE:", type a "CR" (carriage return) in response to "RXCS:". The flag word is organized as follows:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
				DMA	DBS		DDN	MPX	US3	US2	US1	US0	MPN	MPN	MPN

When set to a 1, the bit labeled:

DMA	indicates the device should be tested as an RX02.
DBS	indicates the device is double sided.
DDN	indicates double density operation is enabled.
MPX	indicated a multiplexed unit.
US3	indicates this device contains a drive unit 3.
US2	indicates this device contains a drive unit 2.
US1	indicates this device contains a drive unit 1.
US0	indicates this device contains a drive unit 0.
MPN	coded multiplexed system numbers (normally 0).

US0, US1, US2, US3 do an implicit "SET UNIT" function when set.  
 The normal flag variable for RX02 mode is 4400 (octal).  
 The normal flag variable for RX01 mode is 0000 (octal).

- SET AUTO

This function is an "automatic" SET DEVICE. An internal table containing the most common disk control and status register addresses is scanned. If a response is obtained from a CSR address, the table is updated to indicate the interrupt vector address associated with that controller. The flag word is also modified as appropriate for the particular system assuming drives 0 and 1. A DSD 440 system configured for Mode 2 operations has its flag word set to 4430.

## FORMAT INITIALIZATION COMMANDS

- SET MEDIA DENSITY

This function enables the operator to initialize a diskette to single density or double density format. The function prompts for function confirmation, unit, and desired density. To select single density, respond with an "S". Type "D" to select double density.

The SET MEDIA DENSITY command is used to implement this function, so no headers are rewritten; however, this causes any status that may have been saved on track 0, sector 1 to be erased. The "MODE:" prompt is issued when this function is complete. This function causes any status saved on track 0, sector 1 to be erased.

- SET ALL MEDIA DENSITIES - MEDIA-DEN-ALL

This function initializes all the scratch diskettes loaded in the DSD 440 to the same density. It also prompts you for the diskette density you require. To select single density, respond with an "S". Type "D" to select double density. The SET MEDIA DENSITY command is used to implement this function, so no headers are rewritten; however, this causes any status that may have been saved on track 0, sector 1 to be erased. This "MODE:" prompt is issued when this function is complete.

- REFORMAT -- XFORMAT

This function is used to rewrite diskette headers, as well as all the other data on a particular diskette. It also prompts you for confirmation, unit, and sequential or interleaved format.

Sequential track header format is 01, 02, 03, . . . 24, 25, 26.

The RT-11 maps block numbers 0, 1 . . . into the sector sequence 1, 3, 5, 7, 9, 11, 13, 15, . . . to achieve a "two-way interleave". This provides enough time to process each sector before the next sector comes around on the diskette. The interleave option in XFORMAT writes the following sequence of sector numbers on the diskette following the index pulse:

01	19	12	03	21	14	05	23	16	07	25	18	09
02	20	11	04	22	13	06	24	15	08	26	17	10

When the two way logical sector interleave generated by RT-11 is combined with the physical sector sequence written on the diskette by the XFORMAT interleaved function, a net three-way system interleave is achieved, thus there are two physical sectors between logical registers 1 and 3. This improves system throughput when there is heavy input/output overhead, as often occurs with foreground/background monitor.



## DUMP AND COPY UTILITY COMMANDS

### NOTE

Use the SECTOR INCREMENT function to specify sector sequencing.

- **DUPLICATE — DUP**

The DUPLICATE utility command enables the operator to make a duplicate copy of a diskette. The function prompts for a source drive unit number and a destination drive unit number. For each possible sector address, the function performs a READ SOURCE SECTOR, WRITE DESTINATION SECTOR, READ DESTINATION SECTOR, and COMPARE DATA.

- **COMPARE**

The COMPARE utility command enables the operator to compare two diskettes starting at a specific address. The function prompts for: SOURCE UNIT, STARTING TRACK, STARTING SECTOR, NUMBER OF SECTORS, and DESTINATION UNIT.

- **OCTAL DUMP BY SECTORS — DUMPO**

This utility command enables the operator to cause an octal dump of specified sectors to the console terminal. The function prompts for: UNIT, STARTING TRACK, STARTING SECTOR, and NUMBER OF SECTORS.

- **BINARY DUMP BY SECTORS — DUMPB**

This utility command enables the operator to cause a binary dump of specified sectors to the console terminal. This function prompts for: UNIT, STARTING TRACK, STARTING SECTOR, and NUMBER OF SECTORS.

- **ASCII DUMP BY SECTORS — DUMPA**

This utility command enables the operator to cause an ASCII dump of specified sectors to the console terminal. The function prompts for: UNITS, STARTING TRACK, STARTING SECTOR, and NUMBER OF SECTORS.

For more details on the above listing, refer to the program listing from the source files included on the diagnostic diskette.

### FRD440 Control Characters

There are several control characters to which FRD440 responds even during test execution. These characters and their responses are listed below.

CTRL R Restarts the program at the "MODE:" prompt.

CTRL S Stops output to terminal until another character typed.

- CTRL O Throws away output until another character typed.
- CTRL P Throws away all output except for errors until another char typed.
- <LF> Types track, sector and status counts for current unit.
- CTRL D Transfers control to ODT (Octal Debugging Tool) if ODT is still resident in memory. ODT will be overlayed if your system has less than 20K of memory. In this case, "CTRLD" will simply transfer control to the beginning of FRD440. If you do get into the ODT monitor, a "CTRL C" can be used to return to the program.
- CTRL Y Restarts the last test specified to mode.
- CTRL L Toggles expanded RX02 error printout mode.
- FRD440 will respond to these control characters at any time.

#### FRD440 PROGRAM STATUS AND ERROR DISPLAYS

FRD440 types out error and status information under a wide variety of circumstances. All printouts to the console terminal are sent to a circular buffer in memory as well. The buffer size is determined by available memory. The circular buffer is useful if a hard copy console terminal is not being used and you need to examine error printouts no longer on the face of the CRT screen. The display output buffer (DUMPC) function is used to examine messages in the circular buffer. Each of the status variables that might appear on the console terminal is explained below:

- |                         |   |
|-------------------------|---|
| DEV<XXX>                | Is printed only when running multiple controllers. XXX are the last 3 octal digits of the RXCS address for the system whose error/status data is being displayed. |
| UN<U>                   | U represents the logical drive unit number for which the error/status data is being displayed.  |
| TRACK=<TK>              | Track address at time of status/error printout.   |
| SECTOR=<SC>             | Sector address at time of status/error printout.  |
| RXCS=<XY>               | Shows the contents of the command and status register.  |
| RXDB=<XY>               | Shows the contents of the data buffer register. It should normally be 0 or 214 octal following an INIT.   |
| INTERRUPT<br>ERROR: <X> | If X is less than 0. This indicates that an expected interrupt failed to occur. If X is greater than 0, this indicates that more than one interrupt occurred.     |
| #BAD=<XX>               | This variable indicates the number of status errors detected.   |

#RD/WRT=<XX> This variable indicates the number of sectors that were transferred error-free.

#XFERS=<XX> This variable indicates the number of fill/empty command cycles that were completed successfully.

B-DATA=<XX> Number of data errors where a byte or word of data did not compare with the value the program was expecting. This is different than a CRC error, which would be counted as bad status. There can be up to 128 data errors in 1 sector.

DEFSTT=  
<DEFINITIVE  
ERROR STATUS> Error code associated with the error currently being displayed. The meaning of each error code can be found in Table 5-2.

If in RX02 compatible mode, and CTRL L has been typed to select expanded error printout mode, the following additional status variables appear in the error printout.

DO@TK=<TK> Track Address of drive 0

D1@TK=<TK> Track Address of drive 1

CURTK=<TK> Track address of the current selected logical unit

CSCT=<SC> Sector address of the current selected logical unit

DSTT=<XX> Drive status byte - each of the bits in this status byte is used to encode some information about one or both of the flexible disk drives and/or the media presently installed. The bits get decoded into words which are displayed with the other status. These words are explained below.

USO Drive 0 is currently selected

US1 Drive 1 is currently selected

DN0L Drive 0 currently contains a low density diskette

DNOH Drive 0 currently contains a high density diskette

DN1L Drive 1 currently contains a low density diskette

DN1H Drive 1 currently contains a high density diskette

HDUP Head on currently selected unit is up (unloaded)

HDL D Head on currently selected unit is loaded

TRKRD=<TK> Track address read from a sector header. This number would only be useful following a DEFSTT=150 error.

DEF-RXDB=<XX> Contents of the RXDB following a definitive error status command.

A number of 2-character activity codes are displayed in the context of error printouts. The codes listed below indicate what the diagnostic was doing when the error was detected.

<u>ACTIVITY</u>	<u>CODE</u>	<u>MEANING</u>
FILL-EMPTY	FB	Problem loading sector buffer
FILL-EMPTY	E1,E2	Sector buffer data did not check during an empty buffer operation
FILL-EMPTY	FL,EL	DMA fill or empty error to low mem. buffer
FILL-EMPTY	FD,ED	DMA fill or empty error to CTR. mem. buffer
FILL-EMPTY	FH,EH	DMA fill or empty error to high mem. buffer
SEQ. WRITE	SW,CW	Problem during sequential write
SEQRD	SR	Problem during sequential read
RANDOM	RW,RC,RR	Random (write, check, read) activity when error was detected
ANY READ RETRY	XE	Empty buffer check before retrying read
DUP UTILITY	IN	Error reading the source diskette
DUP UTILITY	CW	Error checking what was just written
DELETED DATA	DW, DR	Deleted data flag failure

The following printouts are examples of what the FRD440 diagnostic program outputs to the console under varying circumstances.

EXAMPLE 1: -

Operator requests status of currently selected drive during a test by typing <LF>.

```
UN 0 TRACK=0 SECTOR=4 BAD=0 RD/WRT=0 XFERS=0 B-DATA=0
```

EXAMPLE 2: -

Operator requests status of both drives using the "STATUS" command.

```
UN 0 BAD=0 RD/WRT=0 XFERS=0 B-DATA=0
UN 0 BAD=0 RD/WRT=0 XFERS=0 B-DATA=0
```

EXAMPLE 3: -

An error occurs during a diagnostic test.

```
UN 1 TRACK=10 SECTOR=27 RXCS=104040 RXDB=100060 BAD=1 RD/WRT=208
      XFERS=0 B-DATA=0 *DENSITY ERROR* D0@TK=010 D1@TK=000
      CURTK=010 CSCT=027 DSTT=100 USO HDUP DN1H DN0L TRKRD=010
      DEF-RXDB=100040
```

In example 3, a density error is reported because the ACCEPTANCE TEST was started while the test diskettes were not both the same density. For this particular error, a literal translation of the error is typed. For all the other errors, the definitive error status code is reported so that the specific error can be identified. Table 5-5 lists these codes and their meanings.

Table 5-5. Error Register Codes (RX2ER)

<u>OCTAL CODE</u>	<u>MEANING</u>
000	No error
010	No drive 0 or drive 0 failed to find track 0 on INIT
020	No drive 1 when Dip Switch indicates there should be a drive 1, or drive 1 failed to find track 0 on INIT
030	Track 0 found while stepping in on initialize
040	Track address passed to controller was invalid (>76)
050	Track 0 found before desired track while stepping
070	Requested sector not found in two revolutions
100	Write protect violation
110	No read data signal present
120	No preamble found
130	Preamble found, but no address mark within window
140	CRC error on what appeared to be a header
150	Address in good header did not match desired track
160	Too many tries for an ID address mark
170	Data address mark not found in allotted time
200	CRC error on data field; RXES bit 0 also set
210	Parity error on interface cable; RXES bit 1 also set
220	READ/WRITE controller failed maintenance mode test
230	Invalid word count specified
240	Density error
250	Wrong key for set media density or format command
260	Indeterminate density
270	READ/WRITE controller write-format failure
320	READ/WRITE controller detected write circuit failure
330	READ/WRITE controller timed out on reset
340	Master controller out of Sync with READ/WRITE controller
350	Non-existent memory error during DMA
360	Drive not ready during format command
370	AC power low caused ABORT of write activity

## DSD 440-8 FLOPPY DISK DIAGNOSTIC (VEP210)

VEP210 is a comprehensive floppy disk diagnostic and formatting program written for the PDP-8 minicomputer family. This is a stand-alone program requiring only 8K of program memory. It allows the user to determine whether or not the floppy disk system is malfunctioning when a problem occurs, and if so, quickly and accurately determine which module is responsible for the malfunction.

VEP210 is an OS/8 compatible program, and can be called from the diagnostic disk using either the user's OS/8 monitor, or the DSD monitor included on the diagnostic disk. The DSD monitor can be loaded and run by the following procedure:

1. Place the 440 in RX01 mode by setting switch number 4 on the controller board in the open position. Note: Diskettes used during diagnostics must be single density.
2. Place the diagnostic disk in drive 0 (left-hand drive).
3. Load the following bootstrap sequence through the front panel.

33/	6755
34/	5054
35/	5045
45/	7326
46/	6751
47/	4053
50/	3002
51/	2050
52/	5047
53/	6753
54/	6753
55/	5033
56/	6752
57/	5453
60/	7004

4. Start at location 33. IF the diskette is write-protected, the computer will halt at location 7604. Pressing the CONTINUE (or run) switch on the front panel will then start the monitor. Additionally, if at any time the computer is halted and you wish to restart in the monitor, this can be done by placing the diagnostic diskette in drive 0 and restarting at location 7605.

When the bootstrapping operation has been successfully completed, the computer will respond by typing a period (.), indicating that the DSD monitor is running. Once in the monitor, the diagnostic can be called by typing:

R VEP 210 <CR>

After typing the carriage return <CR>, the system will access the diagnostic disk and respond with:

```
DSD VEP-8 V1.1      /program name and version
DEV750              /device to be tested.
DRIVE #'S:
```

This indicates that the diagnostic has been successfully loaded and is running. The diagnostic diskette should now be removed, a blank, formatted diskette should be placed in each drive, and the PDP-8 switch register should be set to 0000. This last step is very important, as these switches will affect the operation of the diagnostic. Use of these switches will be discussed at a later time.

When these steps have been completed, enter the drive numbers that you wish to test, in the following manner:

```
1-Drive system:      Type 0 <CR>
2-Drive system:      Type 01 <CR>
```

If you wish to test only one drive in a system, enter only that drive number, at this time. (Note: A single pass deleted data test will occur on drive zero if drive 1 only is selected.) After typing the carriage return <CR>, the diagnostic will respond with:

A, B, F, S, R, V, or D:

This is a list of some of the tests available in the diagnostic. These tests, the interrupt commands that will allow you to control their execution, and the effects of the switch register on the operation of the diagnostic will be explained in the following paragraphs.

**A - Acceptance Test.**

The acceptance test is the basic tool used for fault diagnosis of the DSD 440 system. It uses the sequential, random, and buffer fill-empty tests in a variety of sequences to test the operation of the floppy system. A better understanding of this test sequence can be obtained by reading the sections of these tests, the interpretation of error messages and the error register codes. Once these are understood, almost any problem with the DSD 440 can be identified and isolated to a single module for repair.

**B - Buffer Fill-Empty Test.**

This test checks the sector buffer on the controller for proper operation by first filling the buffer 1000 times with incrementing data and then reading (emptying) the buffer 1000 times. This test is done in 12- and 8-bit modes, and the TRANSFER REQUEST FLAG (TRREQ.) is checked during each transfer.

**F - Format Option.**

This routine allows the user to rewrite the format onto a floppy disk. It is run in the following manner:

1. Type "F." The diagnostic will respond with:

FORMAT DRIVE:

2. Type the drive number you wish to reformat (0, 1, or 2) and space. The system will respond with:

SEQUENTIAL FORMAT?

3. Type Y for a sequential format, N for an interleaved format. The sequential format is used with all DEC processors (PDP- 8/E, F, M, & A), and the interleaved format is used with slower processors, such as the Intersil IM6100 microprocessor. A block diagram of the track format is included in Appendix F of the user's manual.

S - Sequential Read/Write Test.

This test sequentially writes data on all sectors from tracks 1 through 76 (it does not write on track 00) using the OS/8 block mapping, then sequentially reads all the blocks written and checks the data. The diagnostic will print "W" while writing and "R" while reading. This test will repeat continuously when called using the "S" command.

R - Random Read/Write Test. Note: A sequential write must be done to initialize data prior to executing Random Read/Write.

The random test randomly chooses a drive number and OS/8 block number, then randomly reads or writes onto the disk. The diagnostic only prints error messages during this test, and will run continuously without exiting when called using the "R" command. This test is especially useful when attempting to spot and diagnose head positioning problems.

V - Verify.

This test is identical to the acceptance test (A), with one exception: the first pass is restricted to 77 OS/8 blocks, allowing the user to verify system operation more quickly than could be done using the normal acceptance test. After the first pass, the diagnostic prints out "PROCEEDING" and returns to the normal acceptance test. This test has been specially written for testing over extended periods of time.

D - Device Code Change.

The device code change option allows the user to test a system having a device code other than 75, and also to test up to four systems in sequence. This can be set up in the following manner:

Type D. The system will respond with: DEV750:

If at this time you wish to change the device code, the new code should be entered in the following manner:

1. Type the new device code (I.E. - 74 71, etc.)



2. Type the number of drives in the floppy system (1, 2, 3)
3. Type a space. The system will respond with "DEV0:" an example of the results would be:

DEV 750: 742 DEV0:

This indicates that the diagnostic will now test device 74, which is a 2-drive system. Up to four systems can be tested in this manner.

If you wish to merely examine the device code register without changing the contents, type a space after the colon (:). When you wish to exit from the routine, or delete the contents of the register shown, type a carriage return.

## INTERRUPT RESTART COMMANDS

The following commands are used to exit from various routines and tests, restart the program in various locations, and list and reset the cumulative status registers. A thorough understanding of these commands is necessary for efficient use of the diagnostic as a troubleshooting tool.

- Control R - This command will interrupt whatever routine is currently being run and allow the user to select another routine (system prints "A,B,F,S,R,V or D:"). This is equivalent to a restart at location 201 if the processor is halted.
- Control D - This command interrupts the current routine, resets the status registers, and restarts the diagnostic. (System prints "VEP-8 V1.1, DEV 750, DRIVE#'S, allowing user to reset the drives to be tested). This is equivalent to a restart at location 200.
- Control C Control C interrupts the current routine, and causes the system to attempt to restart in the DSD monitor. As this requires reading the monitor from the diagnostic diskette, the diagnostic should be interrupted using either Control R or Control D, and the diagnostic diskette should be placed in Drive 0 before using this command.
- Control T - Control T causes the cumulative status registers to be displayed. If the diagnostic is running a test routine, only the status for the drive selected when the command is given will be displayed, and the test will then continue. If, however, the diagnostic is waiting for a command, (at A,B,F,S,R,V, or D:), the status for all drives and systems set into the diagnostic will be displayed. Status readouts will display the following information:

Drive Number.  
Total Reads.  
Total Writes.  
Total Data Comparison Errors.  
Total Hardware Based Errors.  
Error Register Status and Number of Each  
Hardware Error Type.

Additionally, if the command is given during execution of a test routine, the display will include the current OS/8 block number, track, and sector of the selected drive. The following is an example of the status display:

```
DSK0 BLK=14T,S=116
TOT  READ=10246   TOT WRITES=11437 DATA ERRS=0001
      HDWERRS=0001 ST=100#1
```

This would indicate that drive 0 was interrupted at OS/8 block number 14, track 11, sector 6; the diagnostic had completed 10246 (octal) reads, 11437 writes, and had detected 1 data comparison error and 1 hardware based error, which was error register code 100, a write protect error. Again, the status register can be cleared by executing a control D.

#### EFFECTS OF SWITCH REGISTER SETTINGS ON THE DIAGNOSTIC TESTS.

The PDP-8 switch register can be used to control the operation of the various diagnostic tests. The following will lists some of the switch functions and explain briefly their effects on the operation of the diagnostic.

- Switch 0 - Suppresses the printout of data comparison errors. Only hardware-related errors will be printed.
- Switch 1 - Suppresses the use of the interrupt system. Also suppresses all interrupt-related errors.
- Switch 2 - Suppresses the printout of all hardware-related errors. When used with Switch 0, this effectively suppressed all error messages.
- Switch 4 - Inhibits all other switches. Effectively, this switch disables all the others.
- Switch 5 - Prevents device cycling when more than one floppy system is being tested. When this switch is up, the diagnostic will only test the system currently selected.
- Switch 11 - Prevents floppy disk system from writing on either drive. This is effectively a software write protect switch, and does not effect any read functions.

#### INTERPRETATION OF VEP210 ERROR MESSAGES

Errors detected in VEP210 are all recorded in the status registers and reported in the following manner:

- Type of error (Read, Write, or Data)
- Drive number that the error occurred on (DSK0, 1, or 2)
- OS/8 block number
- Track and sector
- Error register status

A good example of this would be:

WRITE ERR DSK1 BLK=232T,S=21 23 STATUS=100

This error message would indicate that a write error was detected on drive 1 at OS/8 block number 232, track 21, sector 23. The error register status was 100, or a write protect error. \*(Error register codes will be covered more thoroughly in another section). Additionally, data comparison errors (reading the wrong data from the disk) will be reported in the following manner:

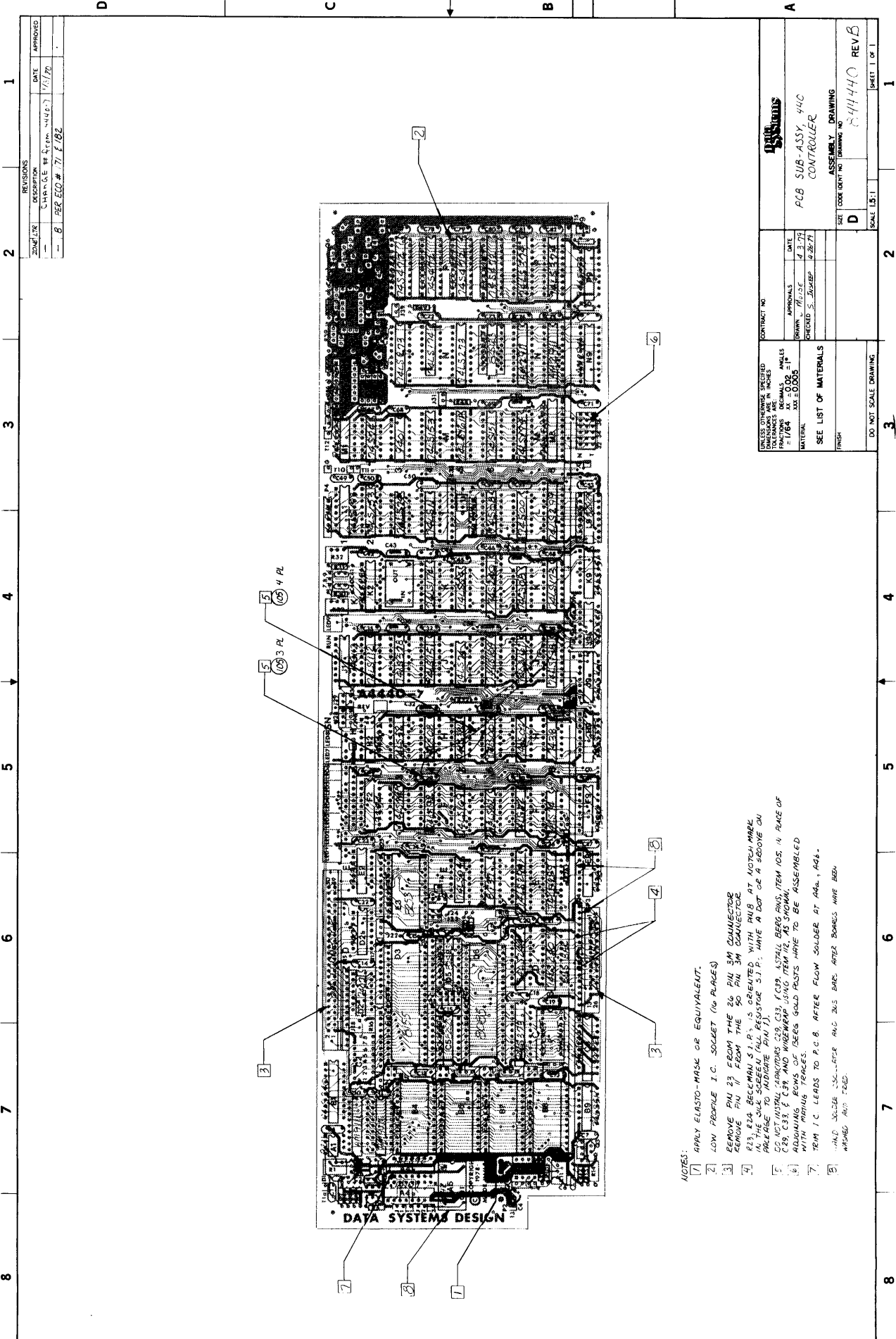
DATA ERR DSK0 BLK=25T,S=5 17 ADDR=3600 WROTE=25 READ 320

This would indicate that a data error was detected on drive 0, OS/8 block 25, track 5, sector 17. The system read 320 from buffer address 3600 and expected to find 25.

\* See Appendix I



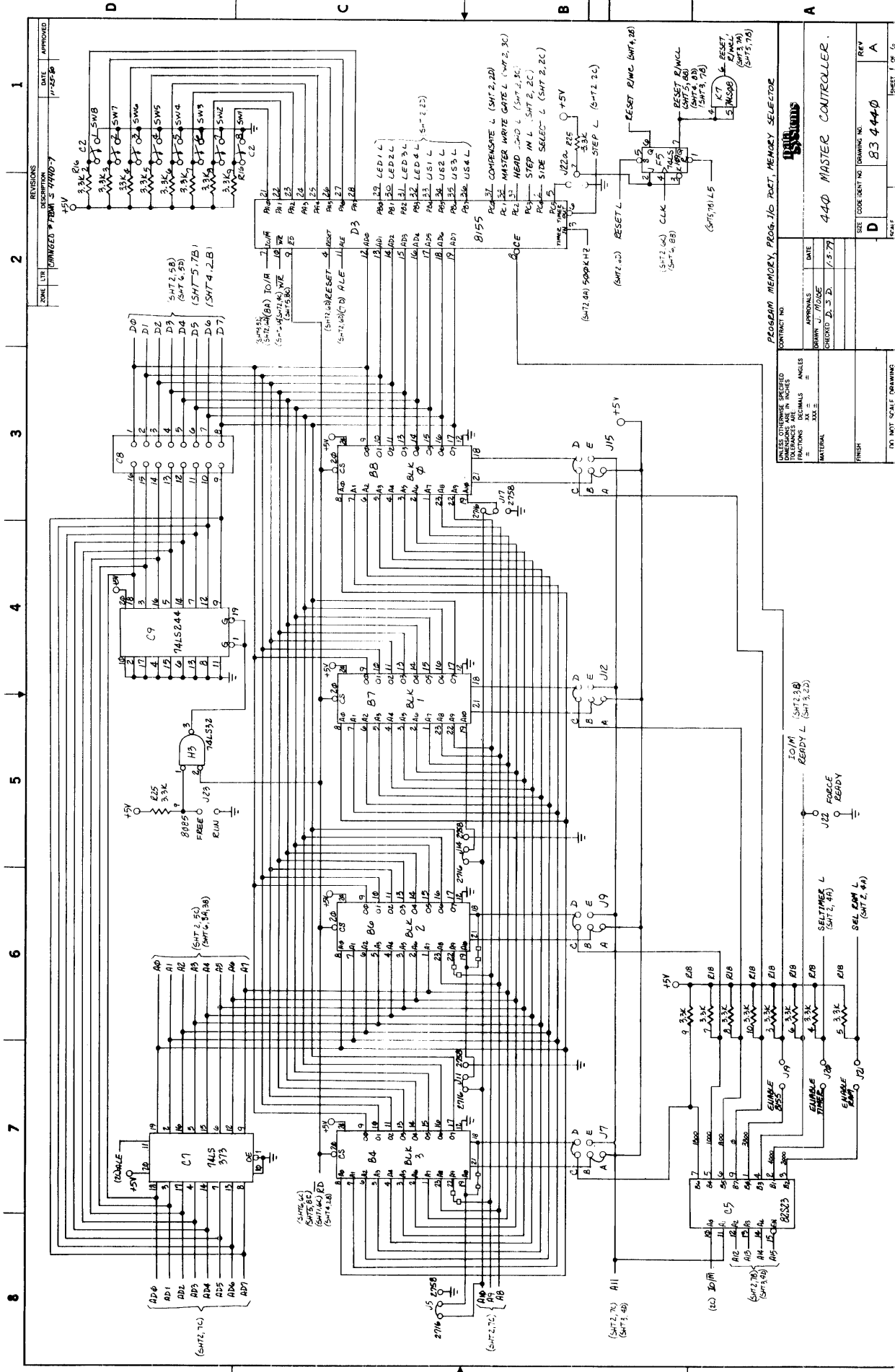
**APPENDIX A**  
**CONTROLLER AND INTERFACE**  
**MODULE SCHEMATICS**



- NOTES:
1. APPLY ELASTO-MASK OR EQUIVALENT.
  2. LOW PROFILE I.C. SOCKET (16 PLACES)
  3. REMOVE PIN 29 FROM THE 26 PIN 3M CONNECTOR
  4. REMOVE PIN 7 FROM THE 50 PIN 3M CONNECTOR
  5. R13, R24 BECEMBUS I.P. IS ORIENTED WITH PA18 AT NOTCH MARK IN THE SILK SCREEN. ALL RESISTOR S.I.P.'S HAVE A DOT OR A SCRIBED ON PLEASE TO INDICATE PIN 1).
  6. 100% VISUAL INSPECTION REQUIRED FOR ALL COMPONENTS.
  7. 100% VISUAL INSPECTION REQUIRED FOR ALL COMPONENTS WITH MOUNTING TRACES.
  8. 100% VISUAL INSPECTION REQUIRED FOR ALL COMPONENTS WITH MOUNTING TRACES.

REVISIONS		DATE	APPROVED
1	CHANGE #2 FROM 11/17/70	11/17/70	
2	PER ECO # 71, 7/1/82		

CONTRACT NO.		DATE	
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		APPROVALS	
FRACTIONS DECIMALS ANGLES		DESIGNER: M.J.S.F.	
1/16 1/32 1/64 1/8 1/4 1/2 3/4 1 1 1/2 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100		CHECKED: S. JAWBEP	
MATERIAL		DATE	
SEE LIST OF MATERIALS		4/3/71	
FINISH		APPROVED BY: JAWBEP	
DO NOT SCALE DRAWING		DATE	
SCALE 1:5:1		4/28/71	
ASSEMBLY DRAWING		DATE	
PCB SUB-ASSY, 44C		4/28/71	
CONTROLER		DATE	
ASSEMBLY DRAWING		DATE	
CODE IDENT NO.		DATE	
D		DATE	
SCALE 1:5:1		DATE	
SHEET 1 OF 1		DATE	
REV B		DATE	
844440		DATE	



1  
2  
3  
4  
5  
6  
7  
8

1  
2  
3  
4  
5  
6  
7  
8

REV	DATE	APPROVED
1	11-27-70	

DATE	DESCRIPTION
11-27-70	CHANGED FROM S 77400-7

APPROVALS	DATE
	11-27-70

CONTRACT NO.	PROJECT NO.

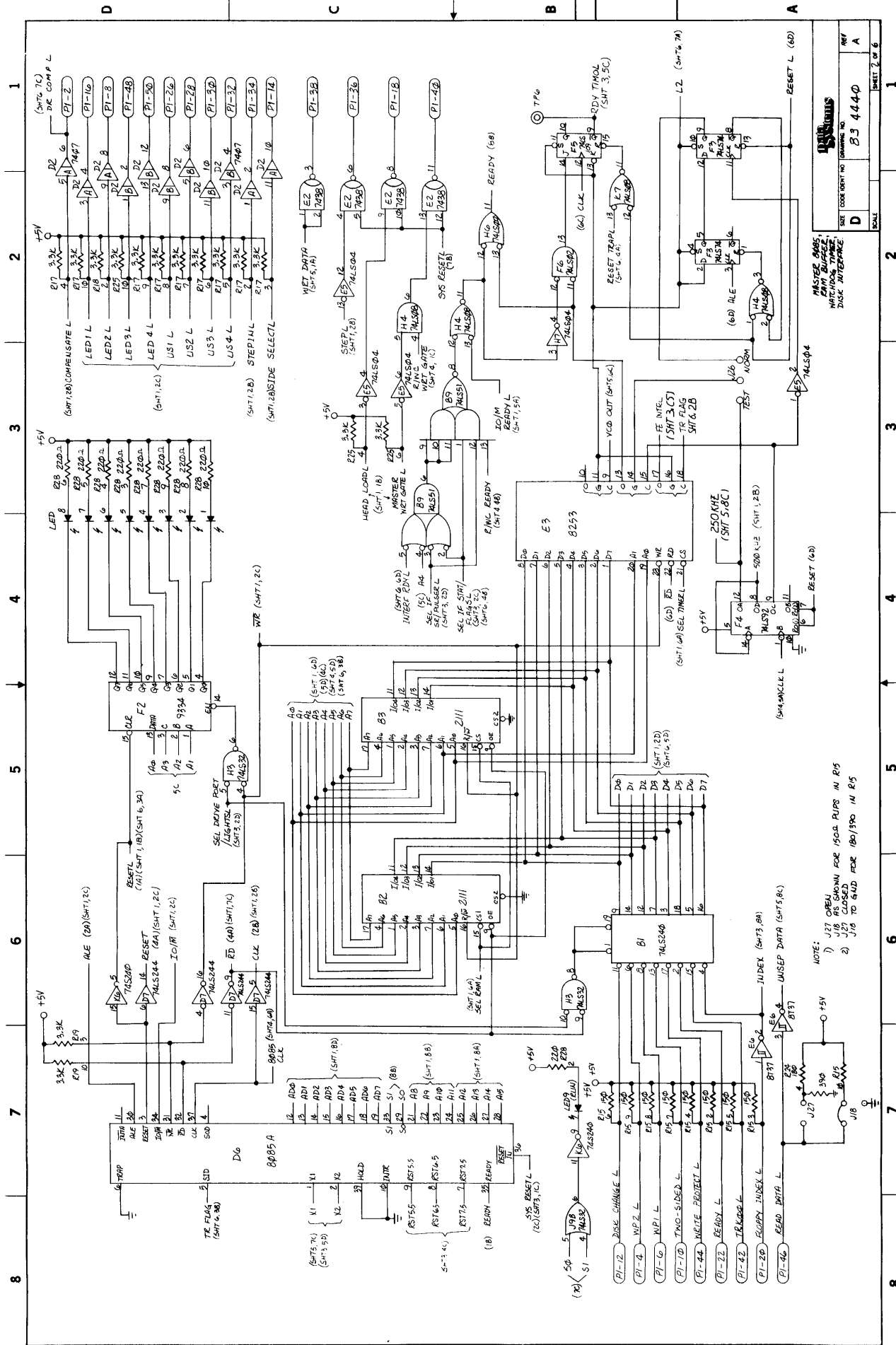
UNLESS OTHERWISE SPECIFIED	UNLESS OTHERWISE SPECIFIED
TOLERANCES ARE IN INCHES	ANGLES
FRACTIONS DECIMALS	DECIMALS
MATERIAL	MATERIAL

SIZE	COORDINATE NO.	REV
D	83 4440	A

CONTRACT NO.	PROJECT NO.

CONTRACT NO.	PROJECT NO.

CONTRACT NO.	PROJECT NO.



**8** 1  
**7** 2  
**6** 3  
**5** 4  
**4** 5  
**3** 6  
**2** 7  
**1** 8

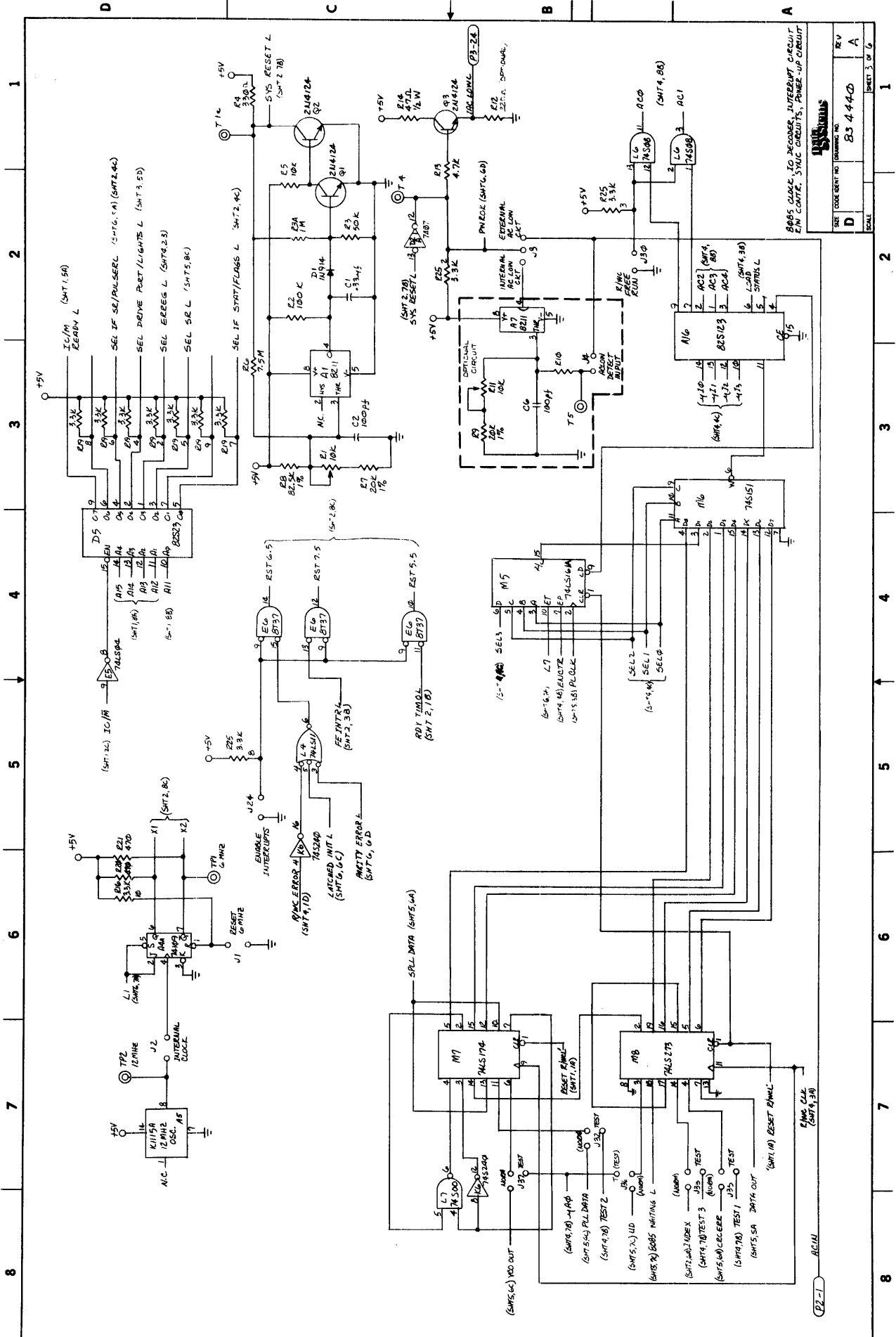
MASTER DISK  
 WATCHDOG TIMER  
 DISK INTERFACE

SET CODE IDENT NO (DRAWING NO)  
**D** 8.3 444-D  
 REF A

SCALE  
 SHEET 2 OF 6

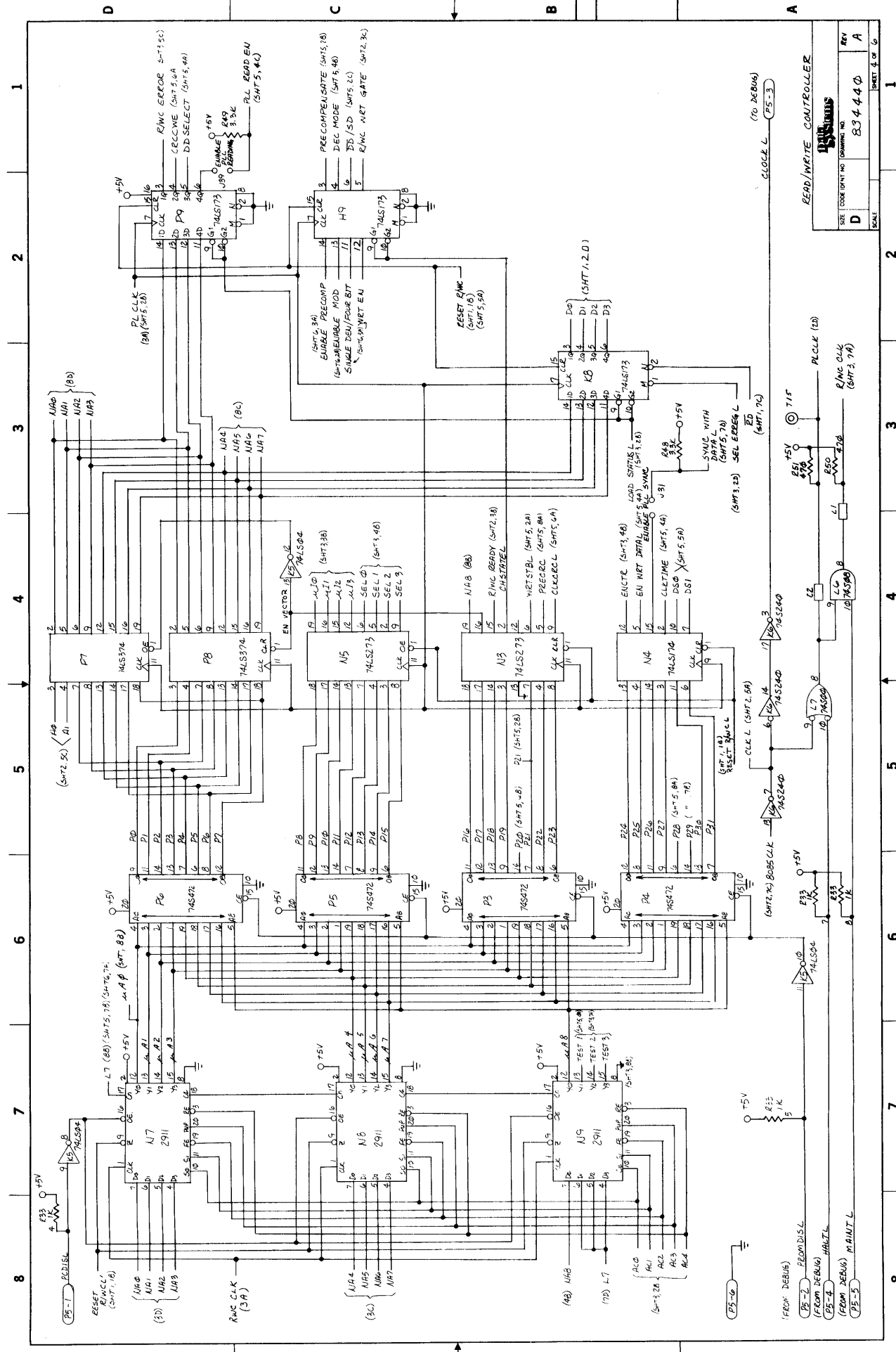
NOTE:  
 1) J27 OPEN/SHUN FOR 1500 PULS IN R15  
 J27 CLOSED  
 2) J16 TO GND FOR 180/390 IN R15



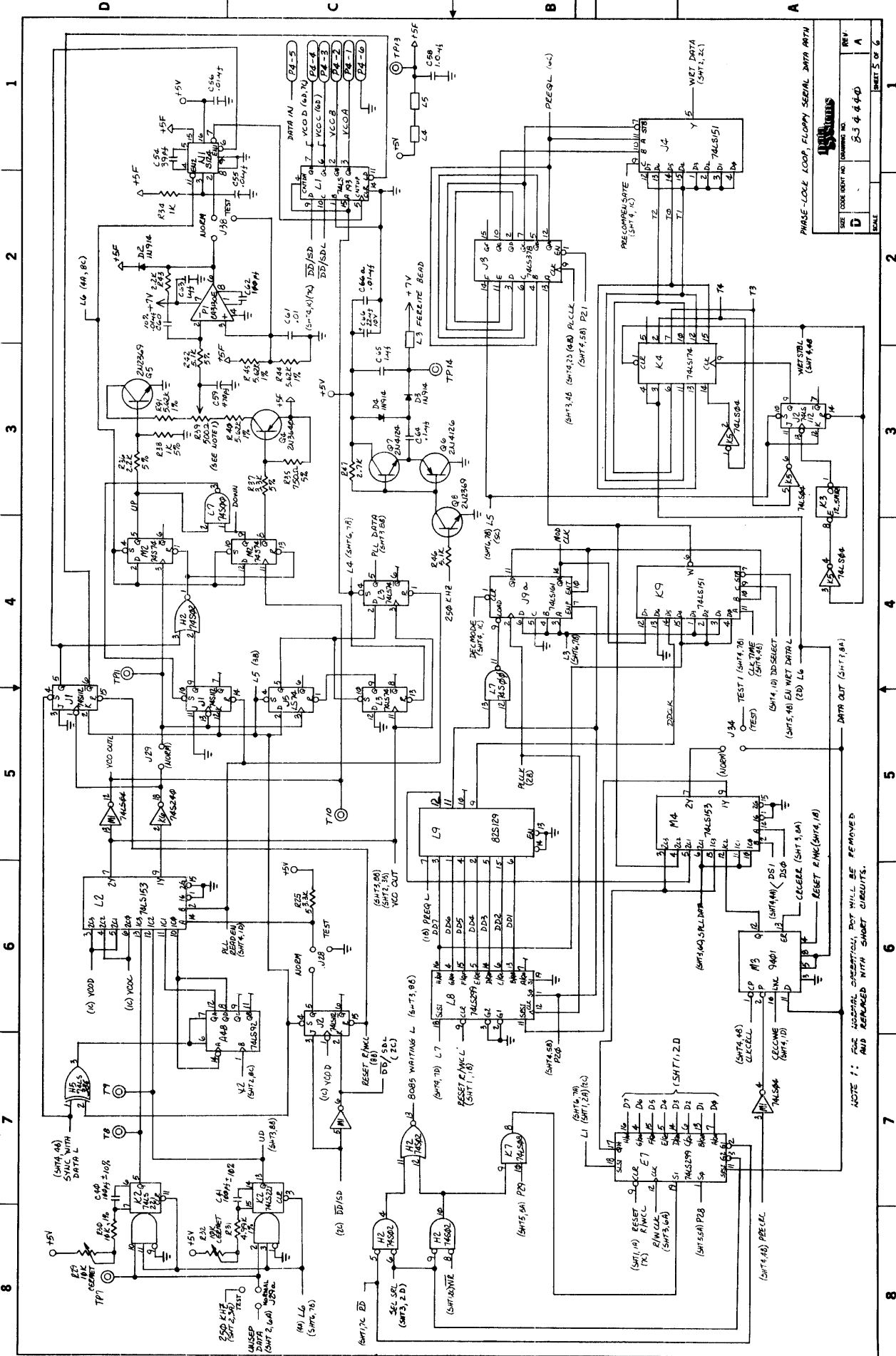


REV	A
CODE	83 444-D
DATE	
SCALE	SHEET 3 OF 6

8005 CLOCK TO RECOVER INTERRUPT CIRCUIT  
 82503 SYNC CIRCUITS, POWER-UP CIRCUIT



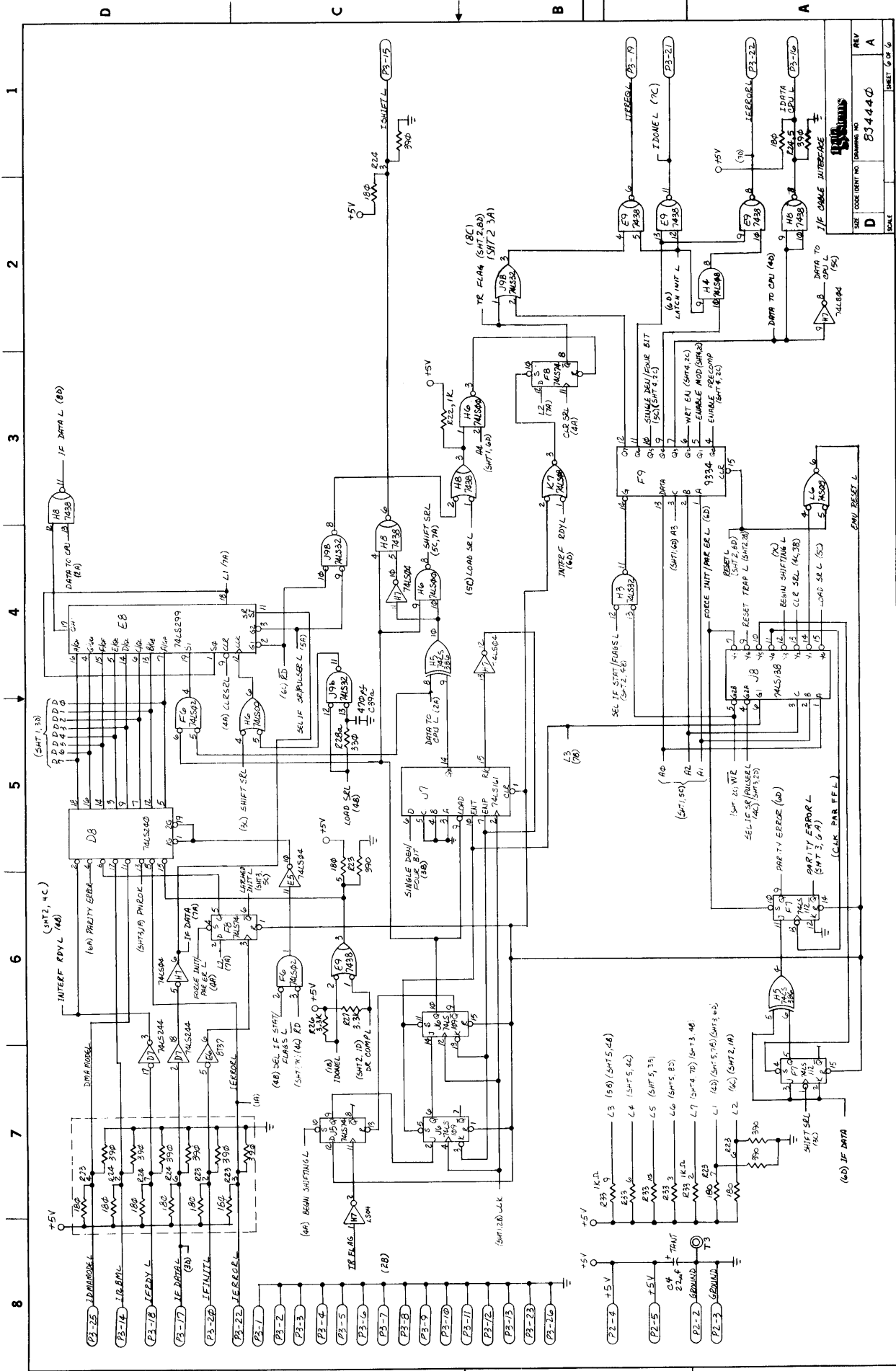
REV	A
SIZE	CODE 83444
SCALE	1
SHEET	1 OF 1



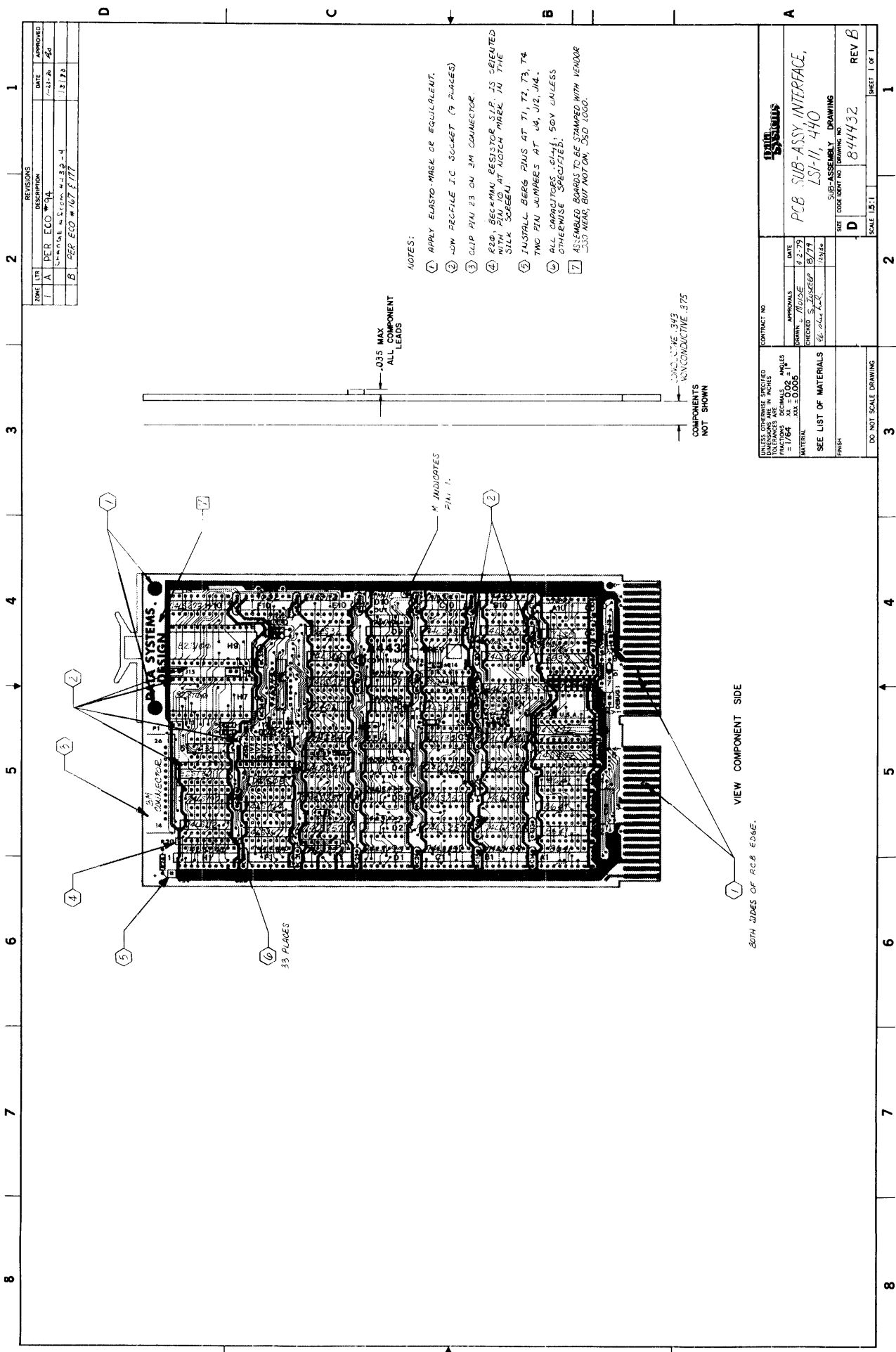
NOTE 1: FOR ADDITIONAL DETAIL, REFER TO THE DRAWING SHEET 5 OF 6.

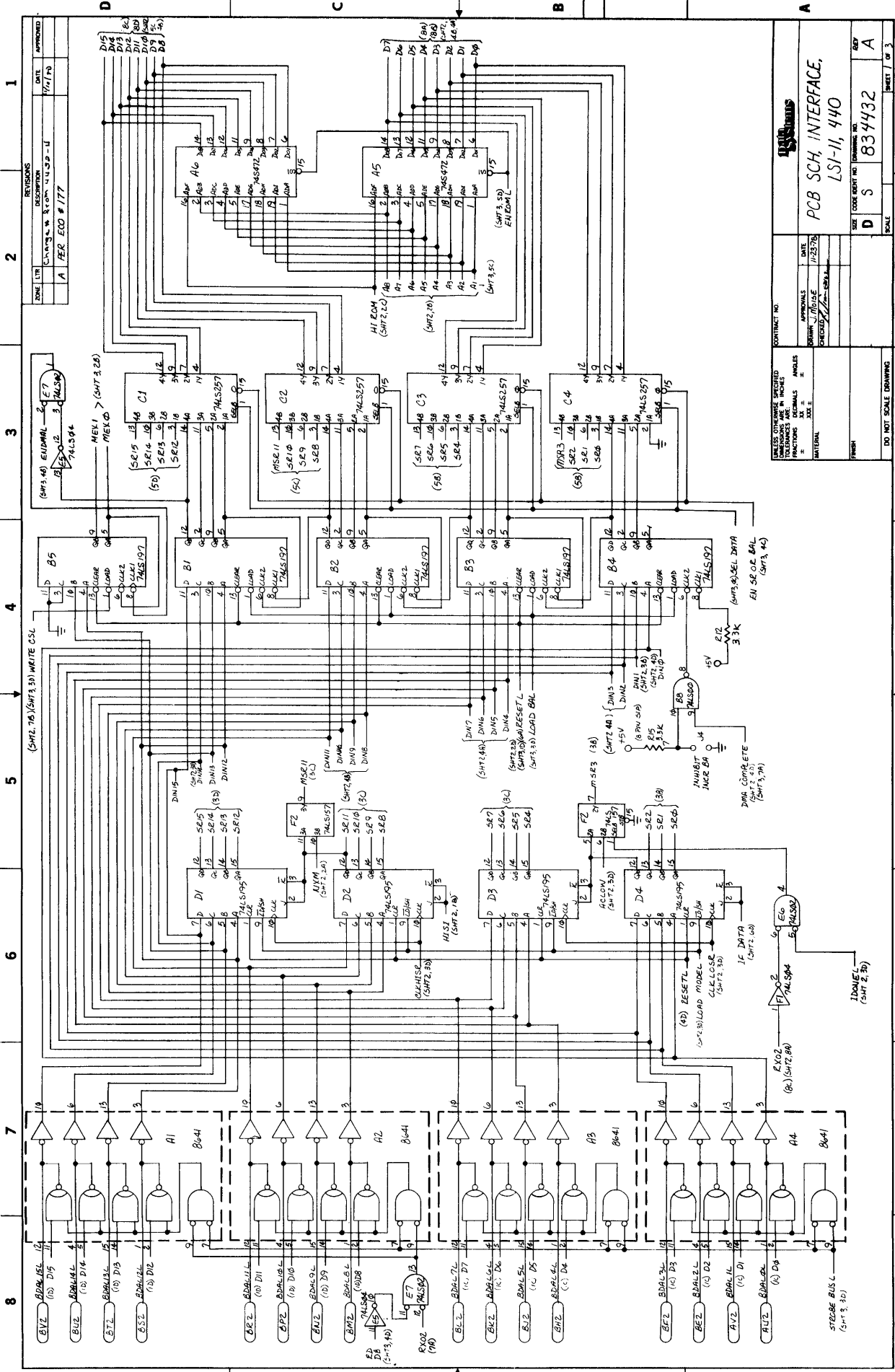
PHASE-LOCK LOOP FLOPPY SERIAL DATA PATH

REV	DATE	BY
A	8.3.64	...



REV	A
DATE	8/3/44
SCALE	1
SHEET	6 OF 6





DATE	BY	APPROVED
1/17/78		

REV	DESCRIPTION	DATE
1	PER ECD #177	

CONTRACT NO.	DATE
	1/23/78

APPROVALS	DATE

SCALE	REV	SHEET	OF
D S	834432	1	3

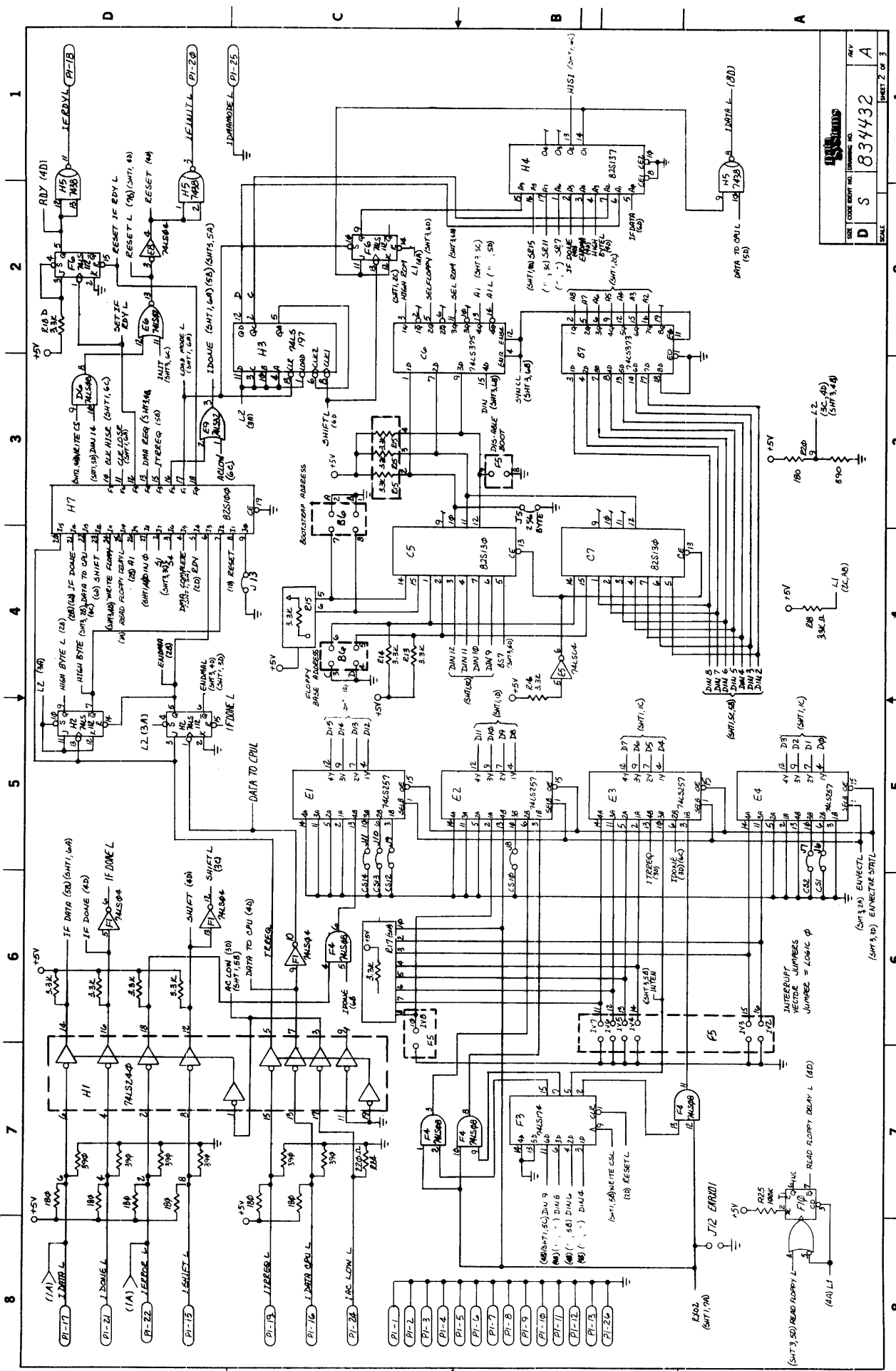
UNITS	INCHES
DECIMALS	1/16
FRACTIONS	1/32

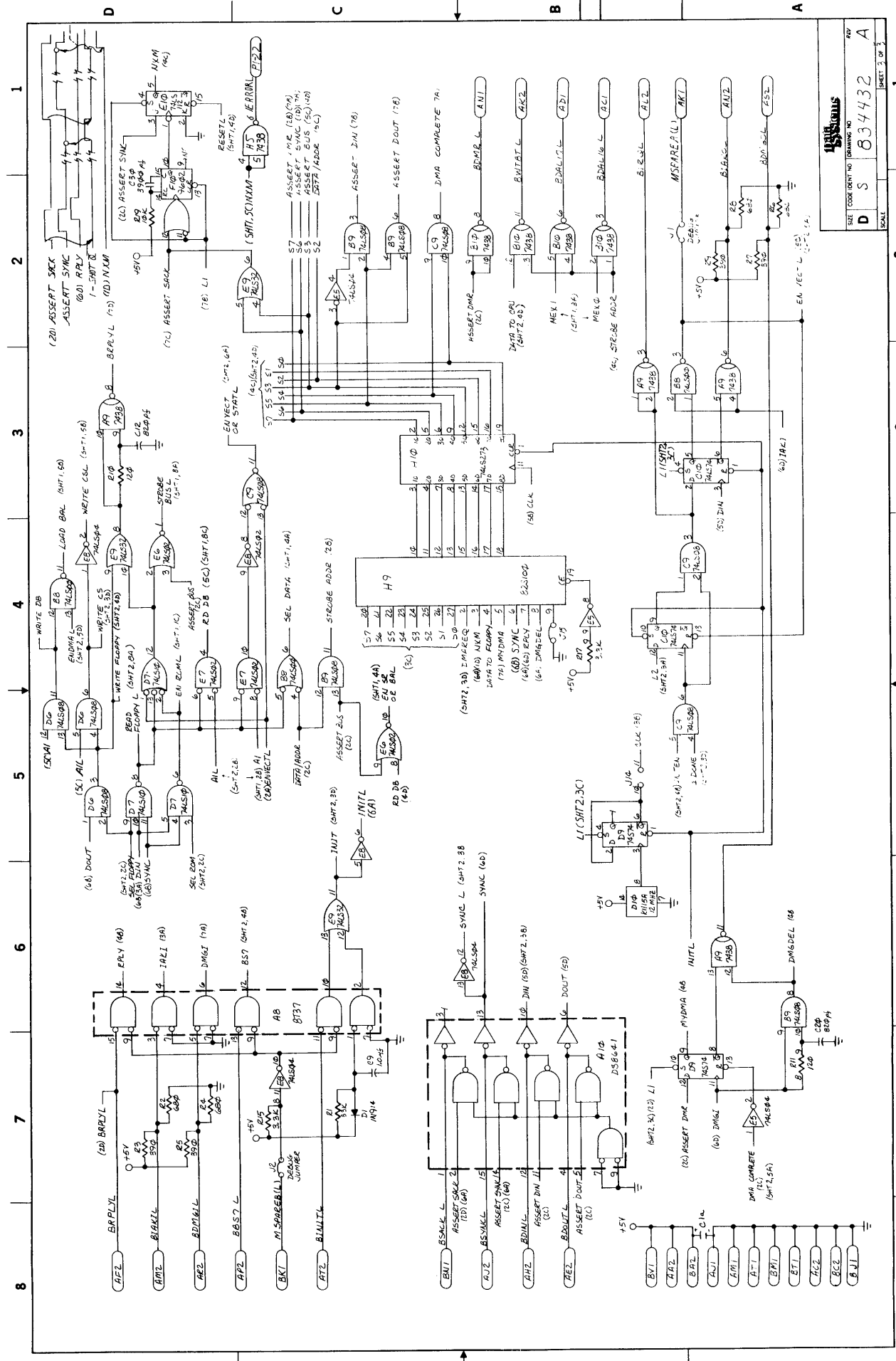
MATERIAL	FRONT

DO NOT SCALE DRAWING

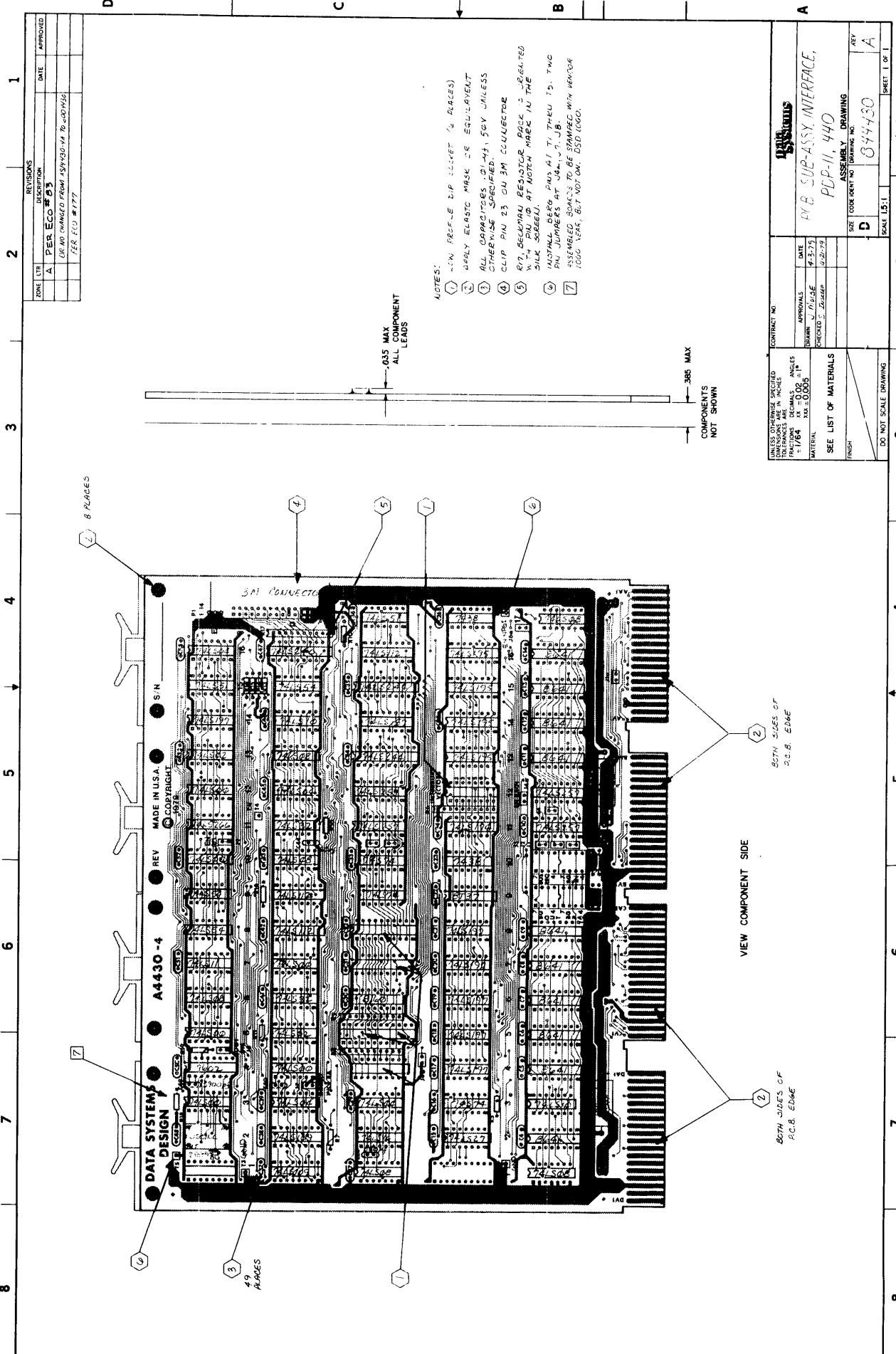


D S 834432 A  
 SCALE  
 SHEET 2 OF 3



SIZE CODE IDENT NO DRAWING NO  
**D S 834432**  
 SCALE  
 SHEET 3 OF 3

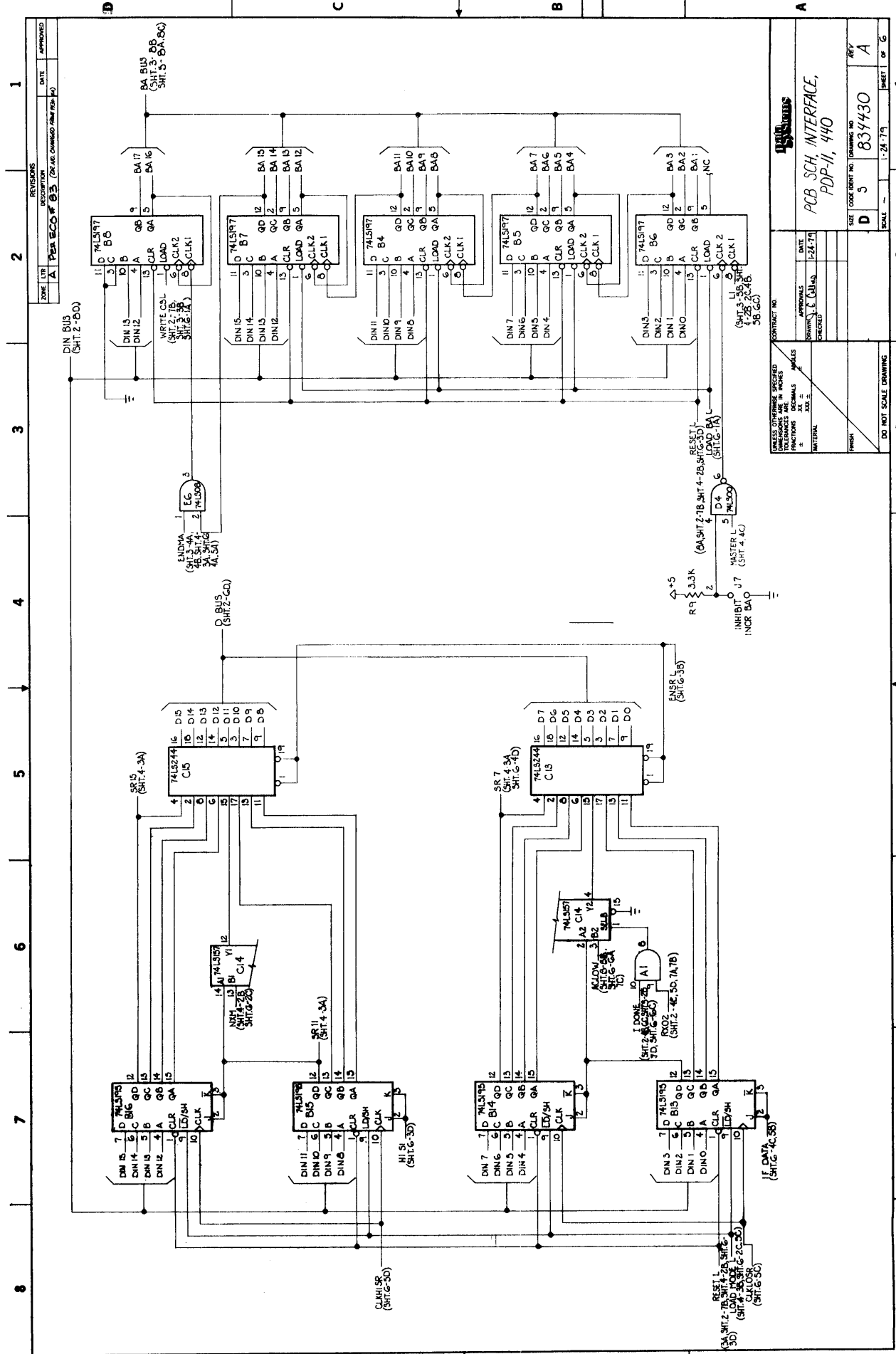




DATE	REV	REVISIONS	APPROVED
	A	PER ECO #03 CR. 800 CHANGED FROM 15V/50-14 TO 10V/50; PER ECO #1777	

- NOTES:
1. IN PROFILE DIP SOCKET (1/2 PAGES)
  2. APPLY ELASTIC MASK TO EQUIVALENT
  3. ALL COMPONENTS MUST BE 50V UNLESS OTHERWISE SPECIFIED
  4. CLIP PIN 23 ON 3M COLLECTOR
  5. 47K OHM RESISTOR PACK IS ORIENTED
  6. INSTALL BEGG. PINS AT T1 THRU T5. TMC PIN JUMBERS AT J4, J5, J6.
  7. ASSEMBLED BOARDS TO BE STAMPED WITH MFGOR 1000 YEAR, BUT NOT ON. DSD 1000.

CONTRACT NO.		DATE	
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ARE: FRACTIONS .002 MIN DECIMALS .0005 MATERIALS SEE LIST OF MATERIALS		APPROVALS	
DRAWN BY: J. J. GALE		DATE: 4-2-73	
CHECKED BY: J. J. GALE		DATE: 4-2-73	
MATERIALS SEE LIST OF MATERIALS		FINISH	
SCALE: 15:1		AS-1	
DRAWING NO. 074730		REV. A	
ASSEMBLY DRAWING		SHEET 1 OF 1	

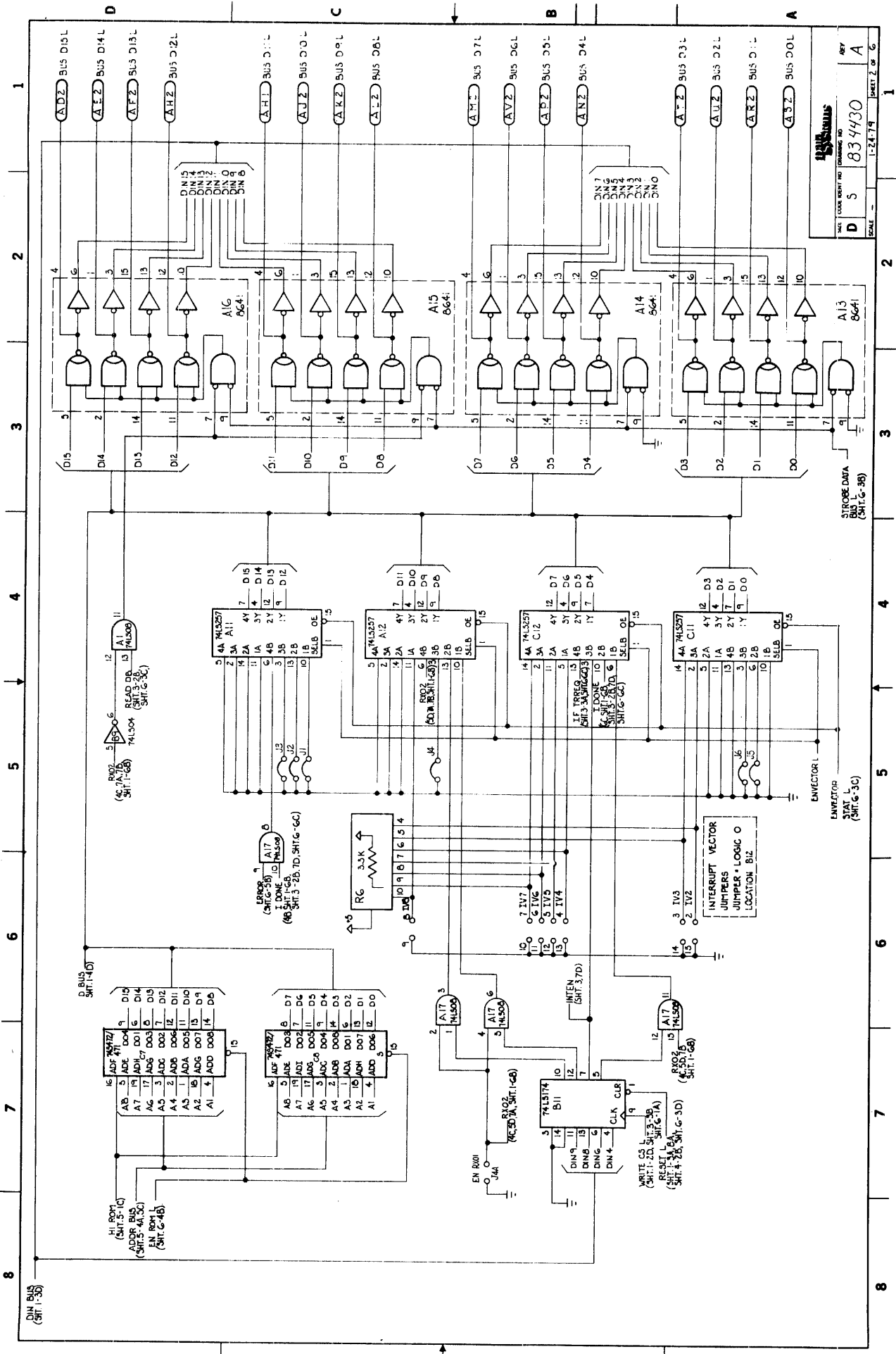


REVISIONS		DATE	APPROVED
ZONE	LTR	DESCRIPTION	
A	PER	ECO # 83 (DR. JAC. CHANGING NAME FROM PDP-11)	

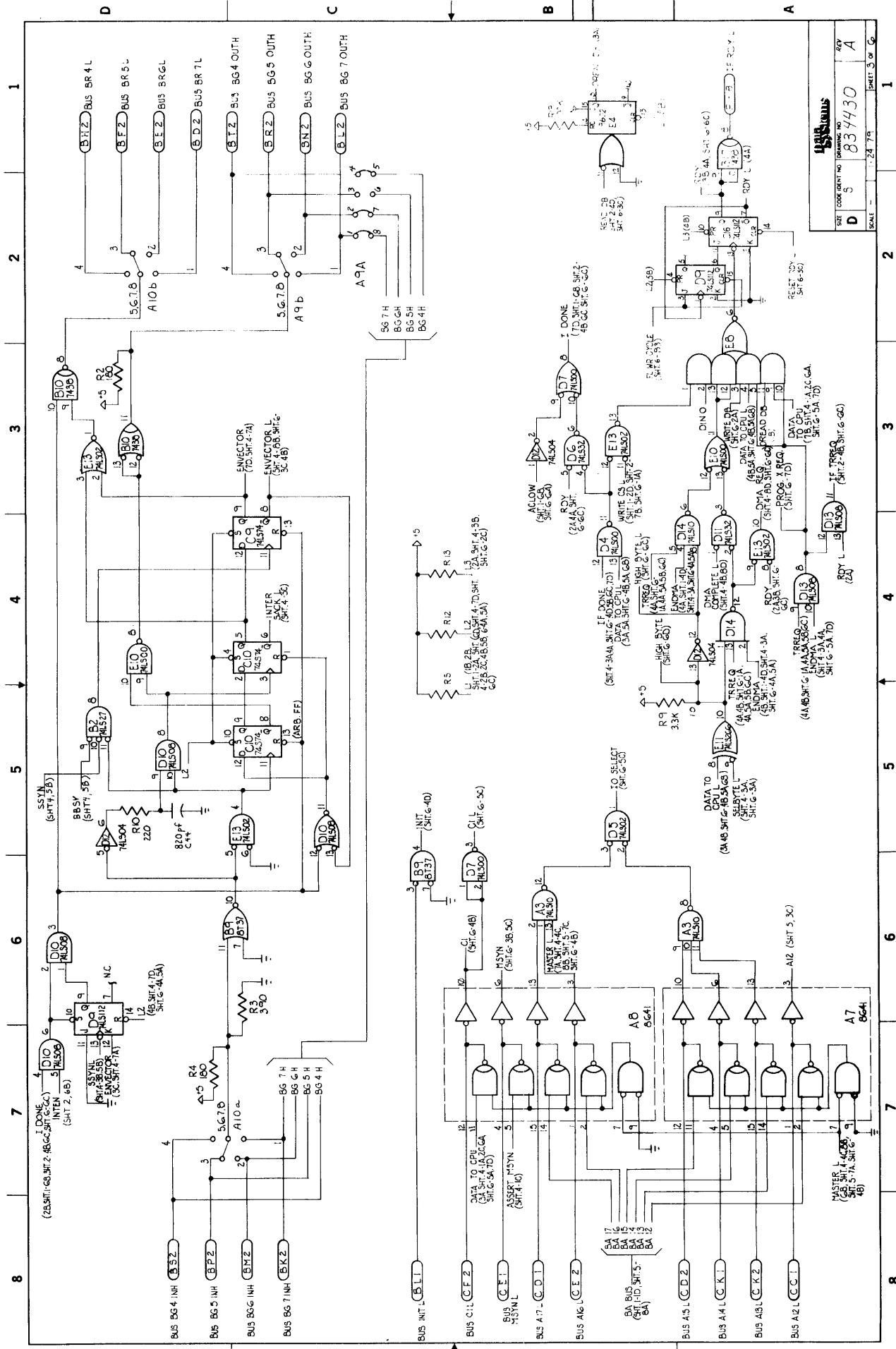
CONTRACT NO.		DATE	APPROVED
834430		1-24-79	
SIZE	CODE IDENT NO.	DRAWING NO.	REF.
D	S	834430	A
SCALE		1-24-79	SHEET OF 6

UNLESS OTHERWISE SPECIFIED  
DIMENSIONS ARE IN INCHES  
FRACTIONS DECIMALS  
XXX .XXX  
MATERIAL  
FINISH

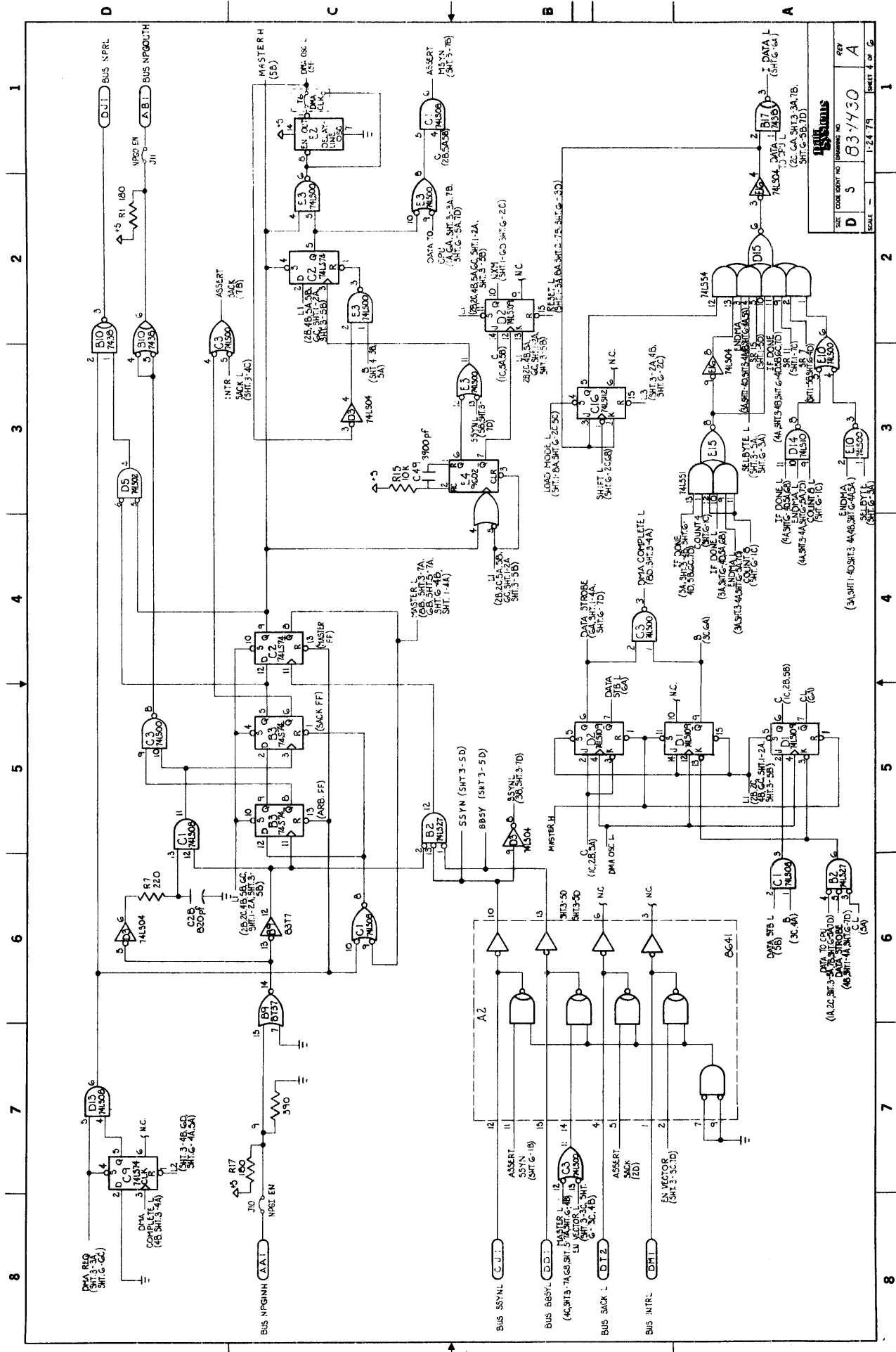
DO NOT SCALE DRAWING

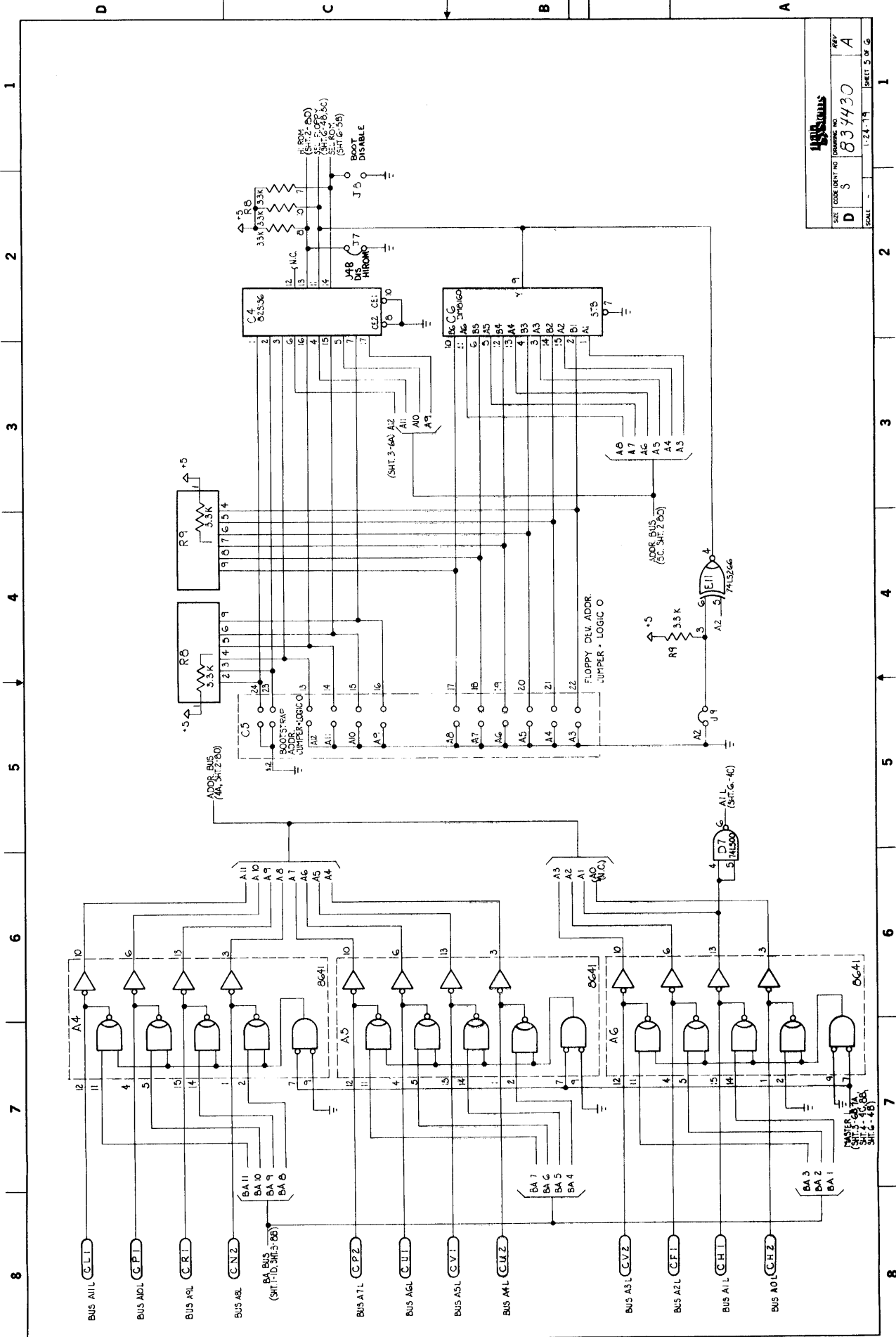


**SCALE** —  
**D** 5  
**834430**  
**REV** A  
**SHEET** 2 OF 6  
**DATE** 1-24-79  
**PROJECT** 834430

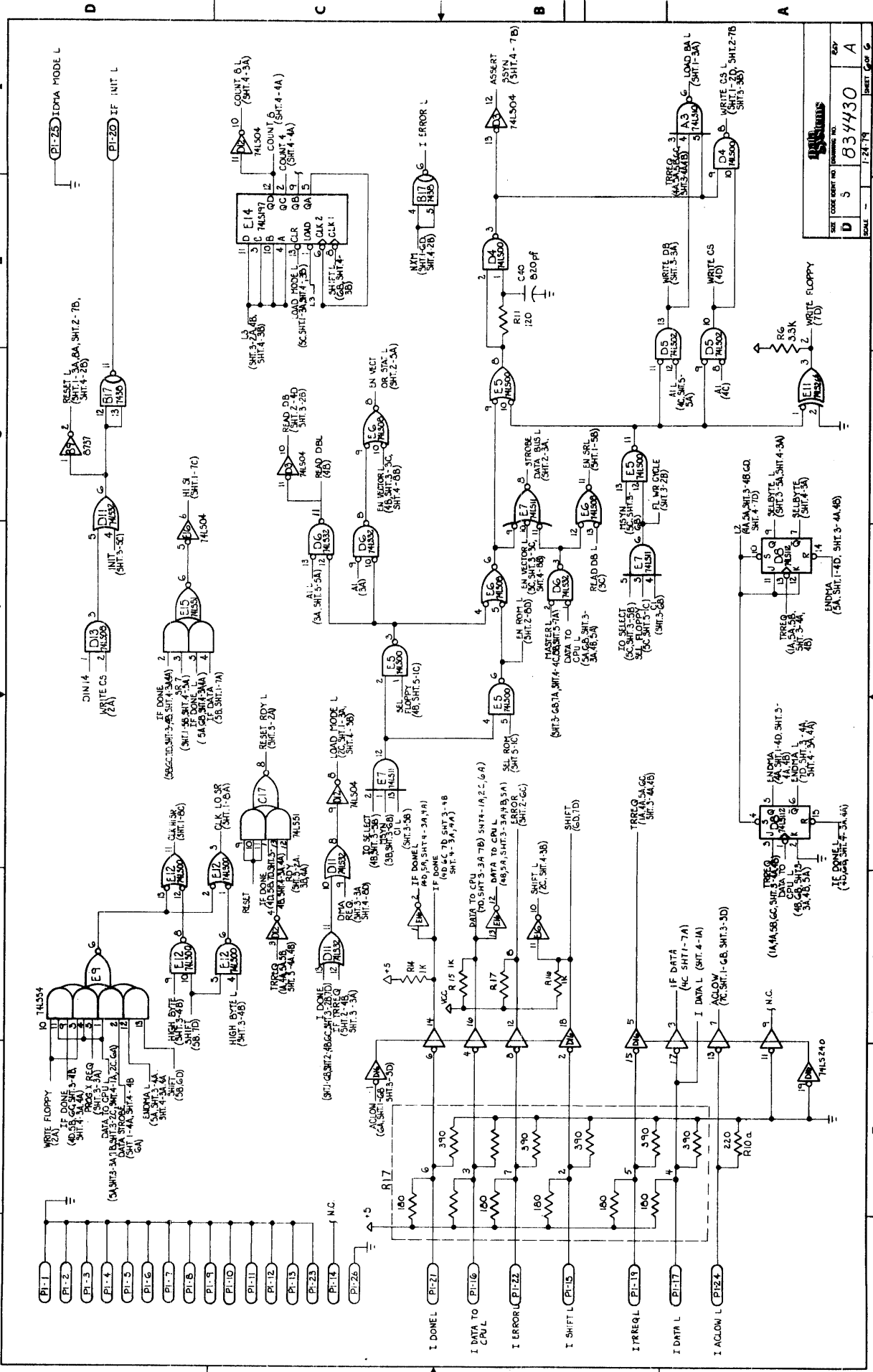


<b>HEATSHIELD</b>		
SIZE	COORDINATE NO.	DRAWING NO.
D	3	834430
REV	A	
SCALE	1:24 7/8	
SHEET	3 OF 6	





SIZE	CODE IDENT NO	DRAWING NO	REV
D	S	837430	A
SCALE	T-24-719 SHEET 5 OF 6		



SIZE	CODE	REV	NO	DATE	BY
D	5		834430	1-24-79	A
SCALE					SHEET
					6

1 2 3 4 5 6 7 8

REVISIONS		
DATE	BY	APPROVED
10-17-68	...	...
10-17-68	...	...
10-17-68	...	...

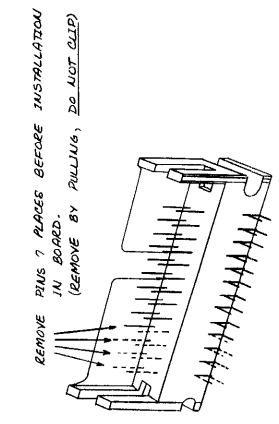
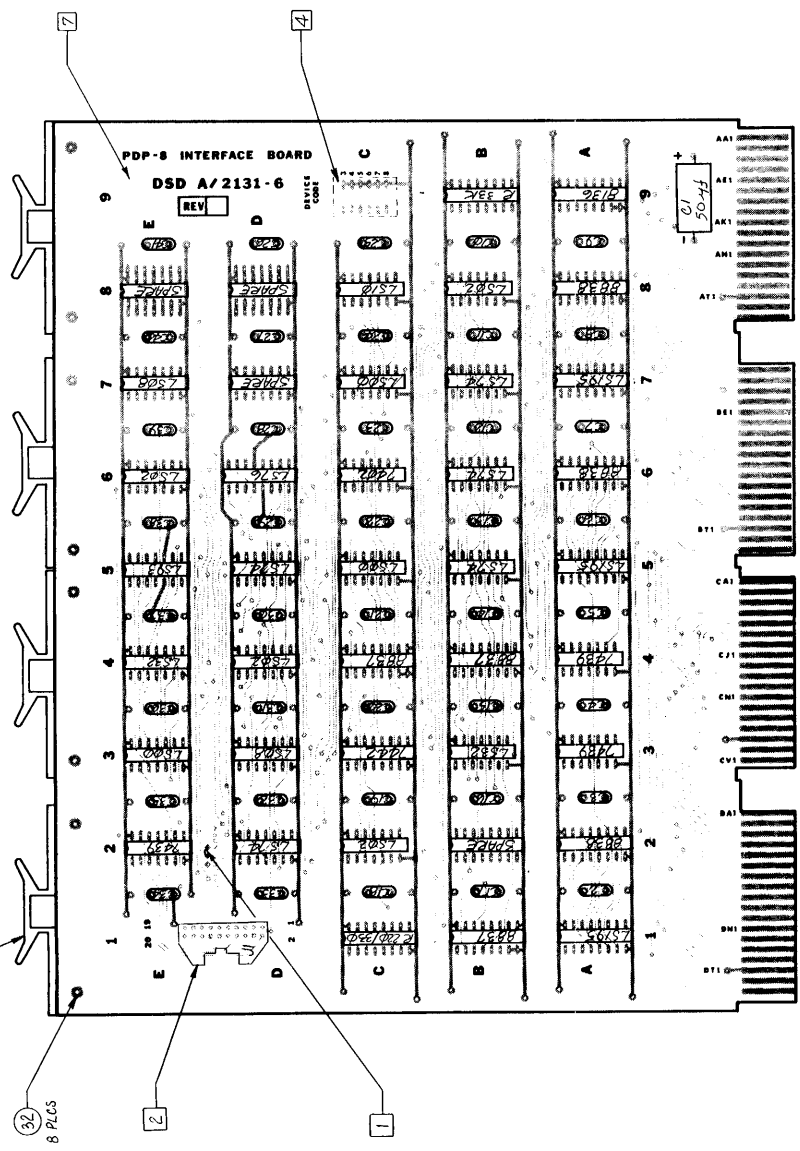


FIG. 1

ASSEMBLY NOTES:

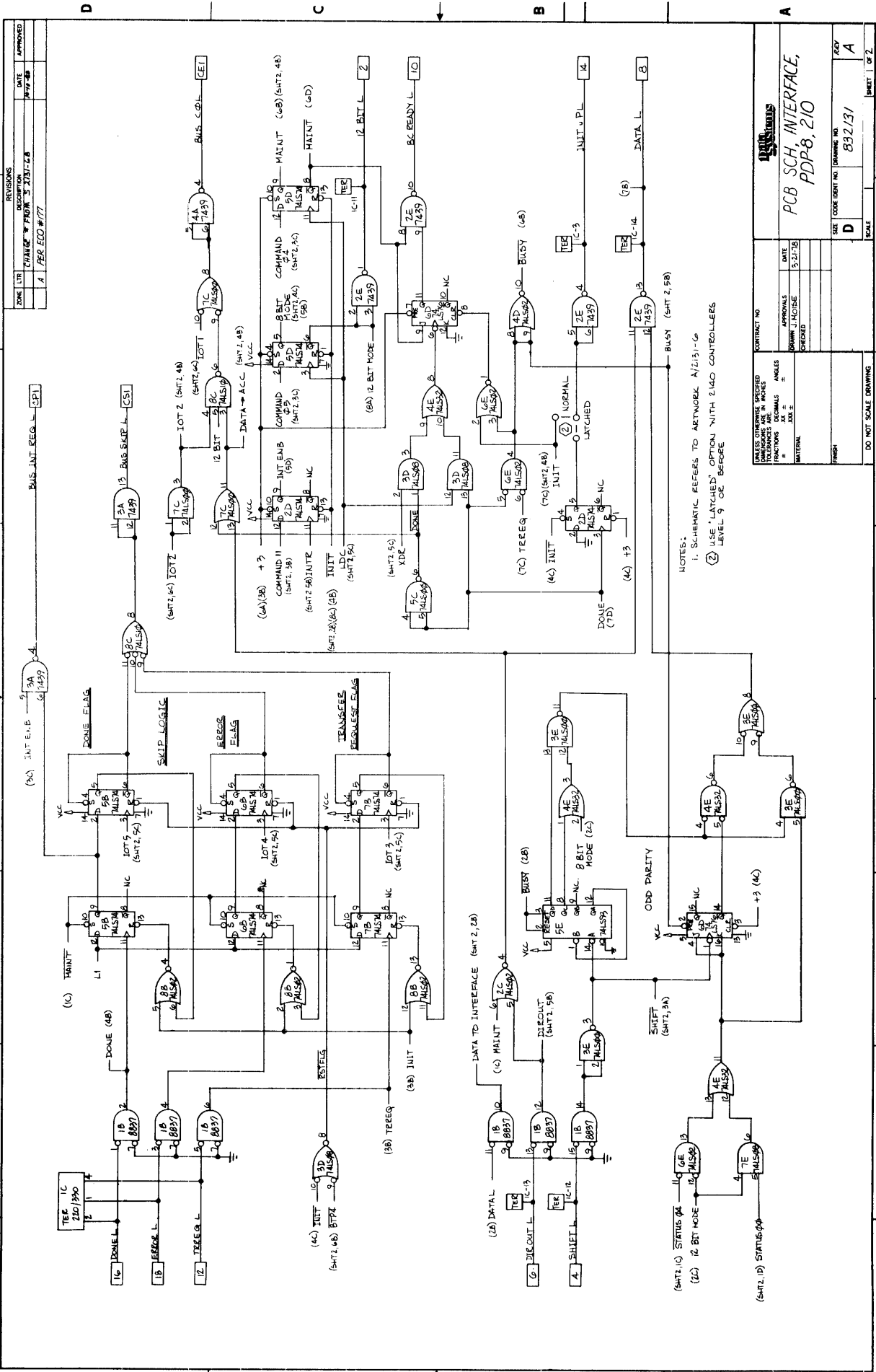
1. INSTALL JUMPER BETWEEN THE CENTER POSITION AND "VOIDM".
2. CUT PIN 20 OF J1 (CABLE SIDE OF CONNECTOR) BEFORE SOLDERING TO BOARD.
3. ALL CAPACITORS .01-μf UNLESS OTHERWISE SPECIFIED.
4. DEVICE CODE FEED-THROUGHS ARE TO BE SOLDER FREE.
5. BOARDS TO BE USED WITH 210 CONTROLLER ARE SPECIFIED AS 802131-01. BOARDS TO BE USED WITH THE 440 CONTROLLER ARE SPECIFIED AS 802131-02.
6. WHEN ASSEMBLING A 802131-02, J1 IS PREPARED WITH A 20 PIN COLLECTOR PINS 1, 2, 13, 23, 24, 25 AND 26 ARE REMOVED COMPLETELY BEFORE INSTALLING THE CONNECTOR. SEE FIG. 1.
7. ASSEMBLED BOARDS TO BE STAMPED WITH VENDOR LOGO WEAR, BUT NOT ON "DSD LOGO".

CONDUCTIVE SURF.  
NON-CONDUCTIVE SURF.  
COMPONENTS NOT SHOWN

VIEW COMPONENT SIDE

CONTRACT NO.		802131	
DATE	APPROVALS	PCB ASSY, INTERFACE, PDP-8, 210	
10-17-68	J. MOYSE	REV B	
MATERIAL		802131	
SEE LIST OF MATERIALS		SCALE 1.5:1	
DO NOT SCALE DRAWING		SHEET 1 OF 1	





REVISIONS		DATE	APPROVED
ZONE	DESCRIPTION		
1	CHANGE # FROM 3 210-210	1/17/77	
2	REV ECO #177		

CONTRACT NO.		DATE	APPROVALS
UNLESS OTHERWISE SPECIFIED			
ALL DIMENSIONS ARE IN INCHES		3/2/76	
FRACTIONS ARE TO BE IN 16ths			
DECIMALS ARE TO BE IN 10ths			
ALL DIMENSIONS ARE TO BE TO CENTER UNLESS OTHERWISE SPECIFIED			
USE STANDARD SYMBOLS FOR ELECTRICAL SYMBOLS			
USE STANDARD SYMBOLS FOR MECHANICAL SYMBOLS			
USE STANDARD SYMBOLS FOR PNEUMATIC SYMBOLS			
USE STANDARD SYMBOLS FOR HYDRAULIC SYMBOLS			
USE STANDARD SYMBOLS FOR THERMAL SYMBOLS			
USE STANDARD SYMBOLS FOR FLUID SYMBOLS			
USE STANDARD SYMBOLS FOR ELECTRICAL SYMBOLS			
USE STANDARD SYMBOLS FOR MECHANICAL SYMBOLS			
USE STANDARD SYMBOLS FOR PNEUMATIC SYMBOLS			
USE STANDARD SYMBOLS FOR HYDRAULIC SYMBOLS			
USE STANDARD SYMBOLS FOR THERMAL SYMBOLS			
USE STANDARD SYMBOLS FOR FLUID SYMBOLS			

DATE	DESCRIPTION	DATE	APPROVED

DATE	DESCRIPTION	DATE	APPROVED

DATE	DESCRIPTION	DATE	APPROVED

DATE	DESCRIPTION	DATE	APPROVED

DATE	DESCRIPTION	DATE	APPROVED

DATE	DESCRIPTION	DATE	APPROVED

1. SCHEMATIC REFERS TO NETWORK A12151-6  
 2. USE 'LATCHED' OPTION WITH 2140 CONTROLLERS  
 LEVEL 9 OR BEGEE

CONTRACT NO. \_\_\_\_\_  
 DRAWN BY: J. MOISE  
 CHECKED: \_\_\_\_\_  
 DATE: 3/2/76  
 APPROVALS: \_\_\_\_\_  
 DATE: \_\_\_\_\_

PCB SCH. INTERFACE,  
 PDP-8, 210

SIZE: 11.000 X 17.000  
 SHEET NO. 832/131  
 DRAWING NO. \_\_\_\_\_  
 SCALE: \_\_\_\_\_  
 DO NOT SCALE DRAWING

DO NOT SCALE DRAWING

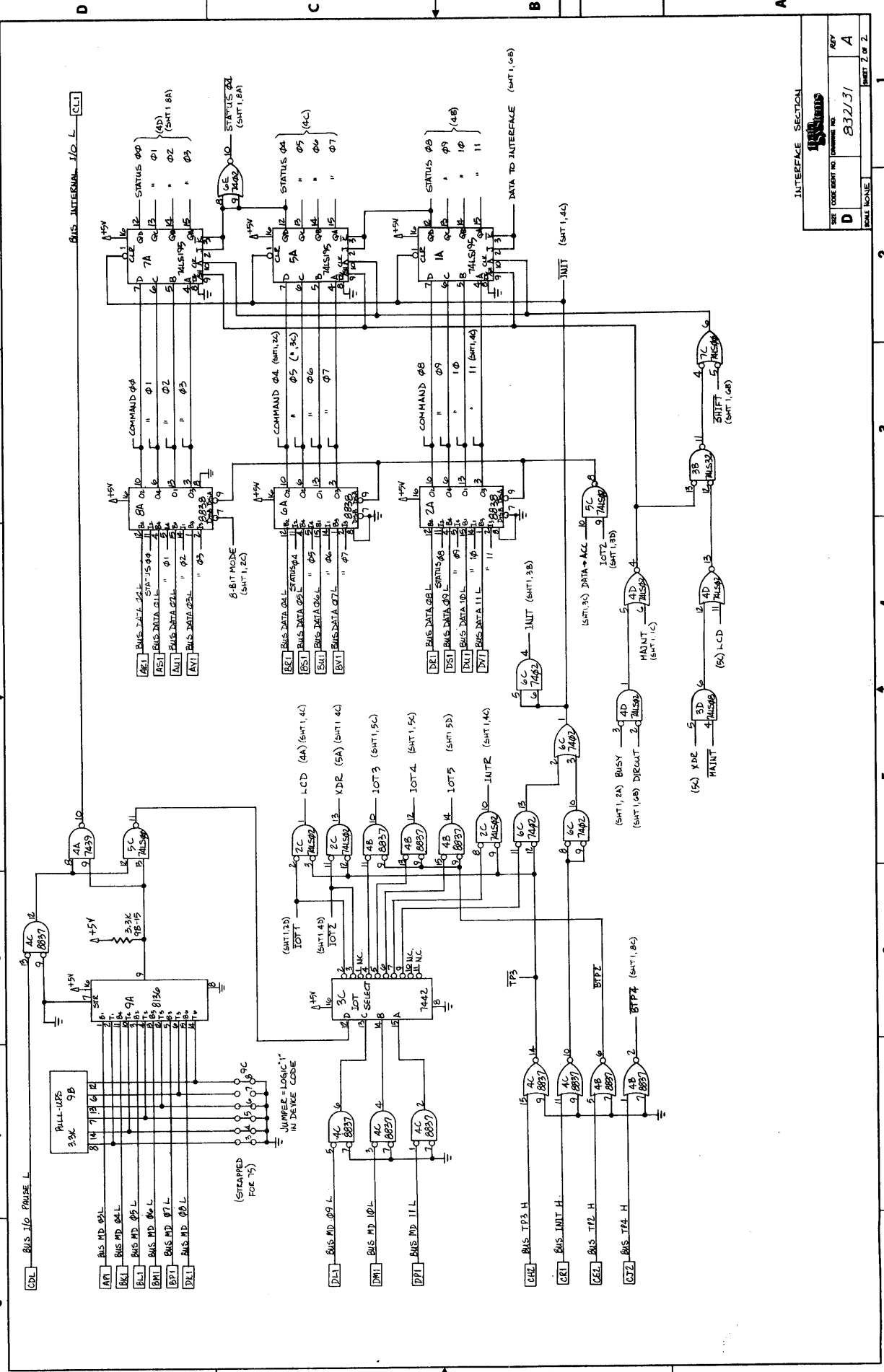
DO NOT SCALE DRAWING

DO NOT SCALE DRAWING

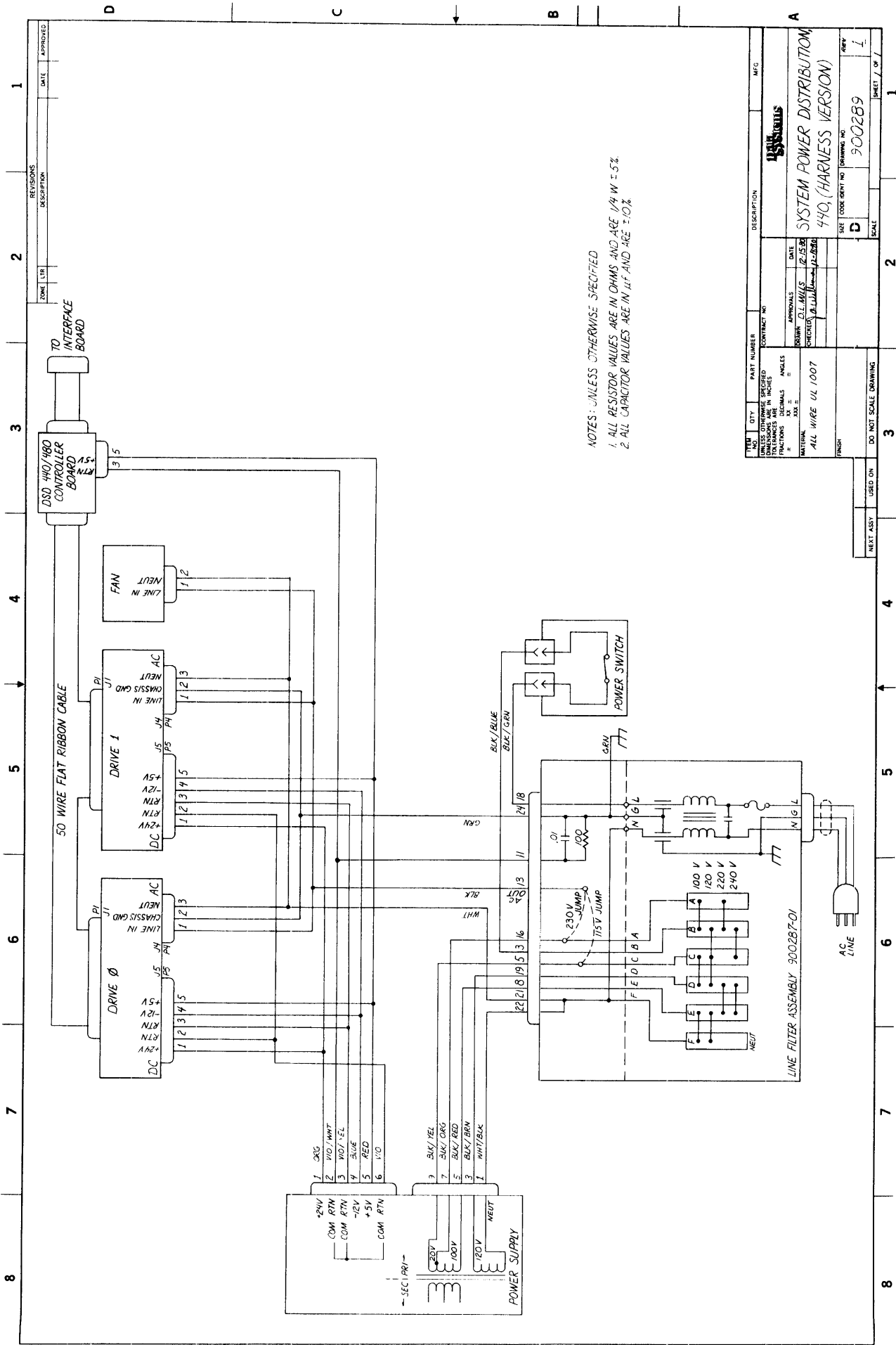
DO NOT SCALE DRAWING

DO NOT SCALE DRAWING

8 7 6 5 4 3 2 1

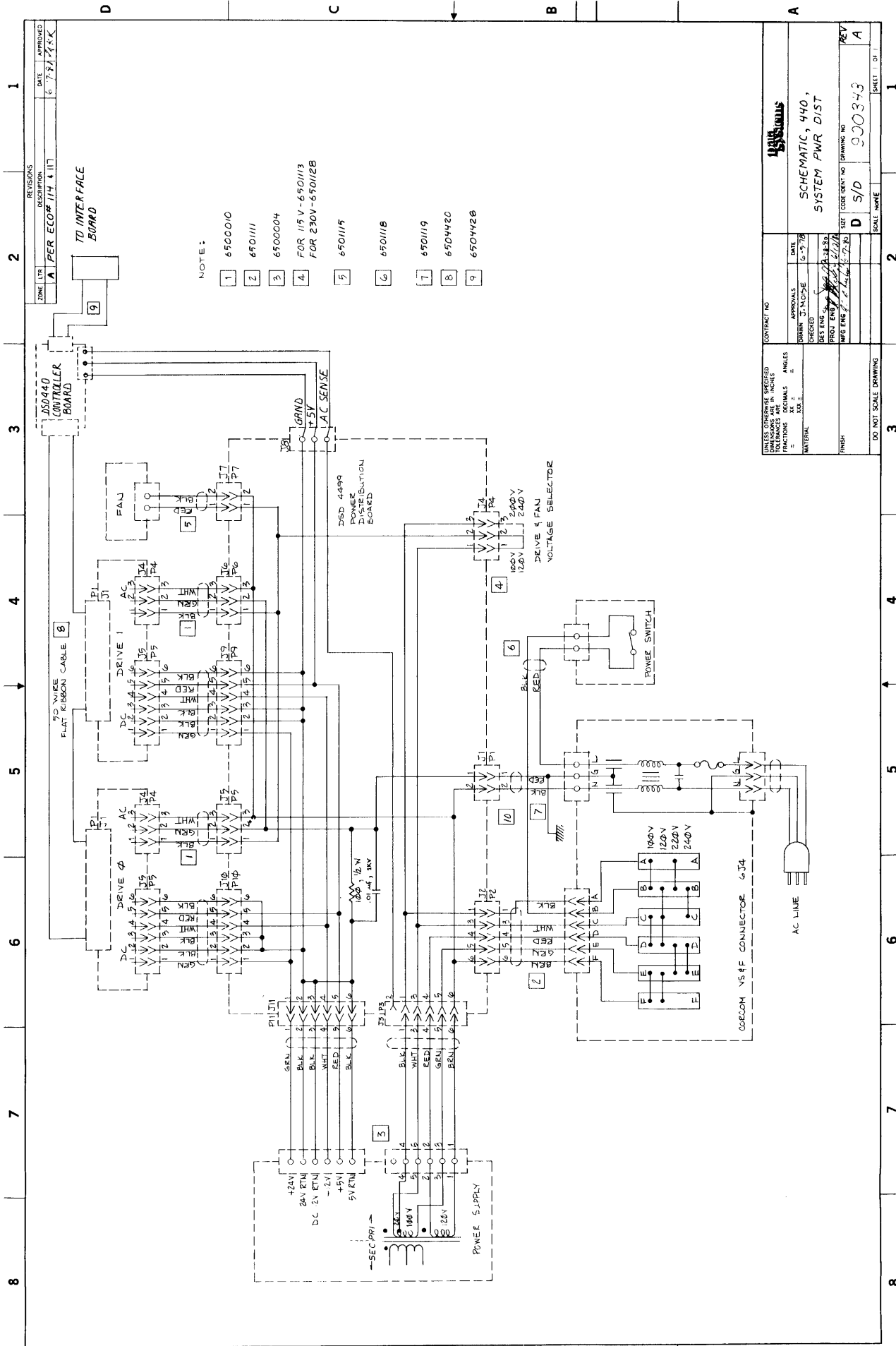


INTERFACE SECTION  
 PART CODE SHEET NO. 832/31  
 REV. A  
 SHEET 2 OF 2



NOTES: UNLESS OTHERWISE SPECIFIED  
 1. ALL RESISTOR VALUES ARE IN OHMS AND ARE  $\pm 5\%$   
 2. ALL CAPACITOR VALUES ARE IN  $\mu F$  AND ARE  $\pm 10\%$

REV. NO.	QTY.	PART NUMBER	DESCRIPTION	MFG.
1		300289	SYSTEM POWER DISTRIBUTION 440, (HARNESS VERSION)	RAY
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		CONTRACT NO.		
DIMENSIONS IN PARENTHESES ARE IN INCHES		APPROVALS	DATE	
DRAWN BY: J.L. HILLS		CHECKED BY: J.L. HILLS	DATE: 12/15/68	
MATERIALS: ALL WIRE UL 1007		FINISH		
SIZE: CODE 8071 NO. 300289		DRAWING NO.	300289	
DO NOT SCALE DRAWING		SCALE	1	
NEXT ASSY. USED ON		SHEET 1 OF 1		

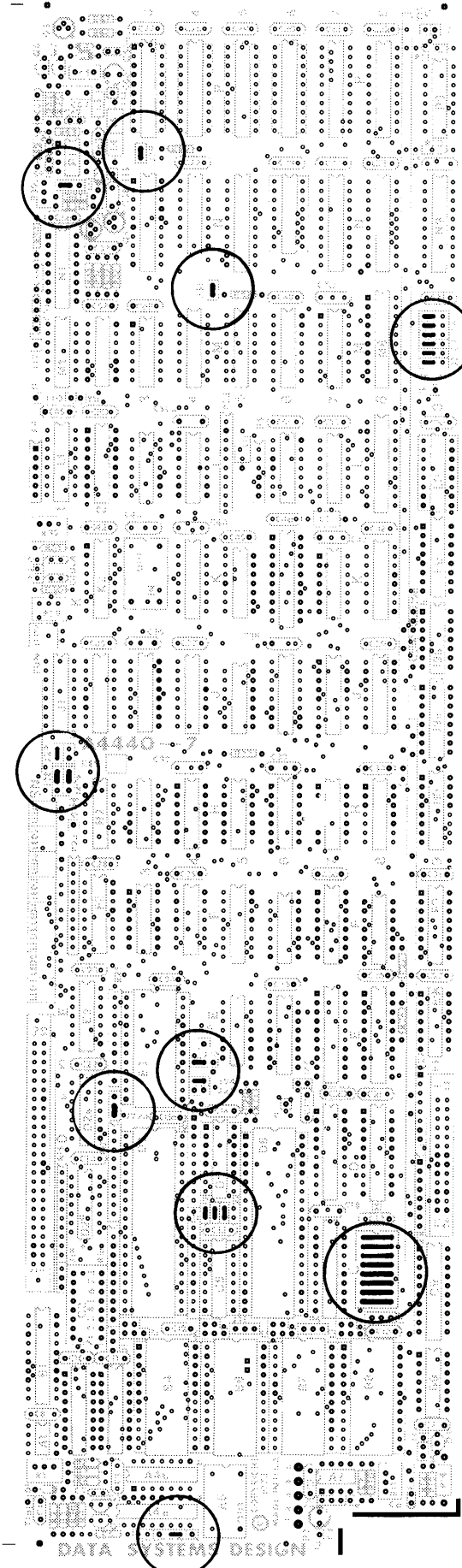


ZONE	LTR	REVISIONS	DATE	APPROVED
A	PER ECO# 114 & 117		6-7-72	JFK

CONTRACT NO.		DATE		SCALE	
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES FRACTIONS DECIMALS ANGLES		APPROVALS		DRAWING NO.	
DRAWN BY: JAL/DSE		DATE: 6-7-72		SHEET 1 OF 1	
CHECKED BY: JAL/DSE		DESIGN: 440V-115V		SCALE: NAME	
PROJECT NO. 440V-115V		PROJ. ENG. JAL/DSE		D S/D 900343	
MATERIAL		SIZE: CODE: QTY: NO. DRAWING NO.		REV. A	
FINISH		MFG. ENG. JAL/DSE		DO NOT SCALE DRAWING	



**APPENDIX B**  
**STANDARD JUMPER POSITIONS ON**  
**INTERFACE AND CONTROLLER MODULES**

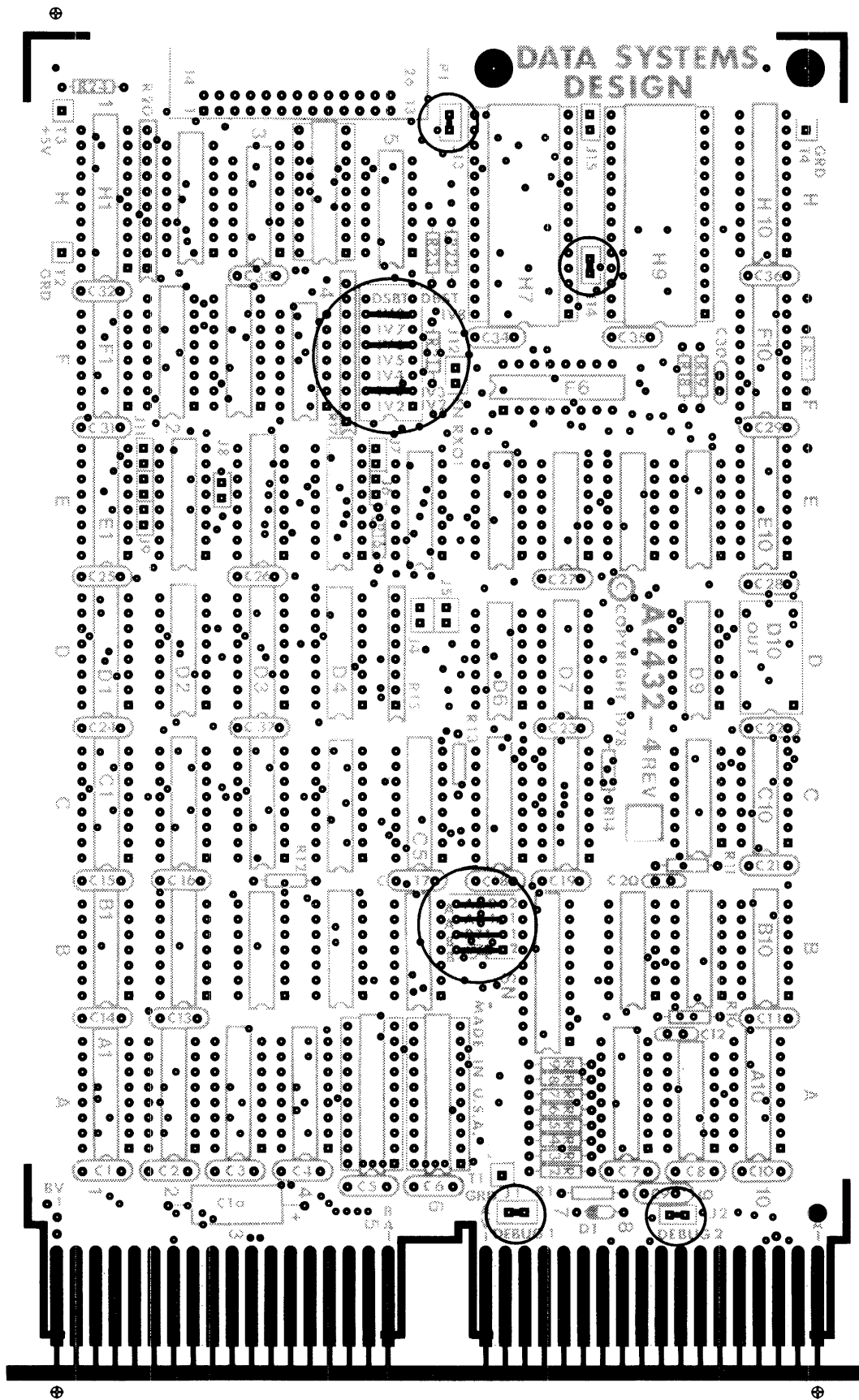


DSD 440 Controller Module

Normal jumper configuration used for the following:

1. Normal Configuration
2. Extended Self Test (with test cables)
3. SA3

DATA SYSTEMS.  
DESIGN

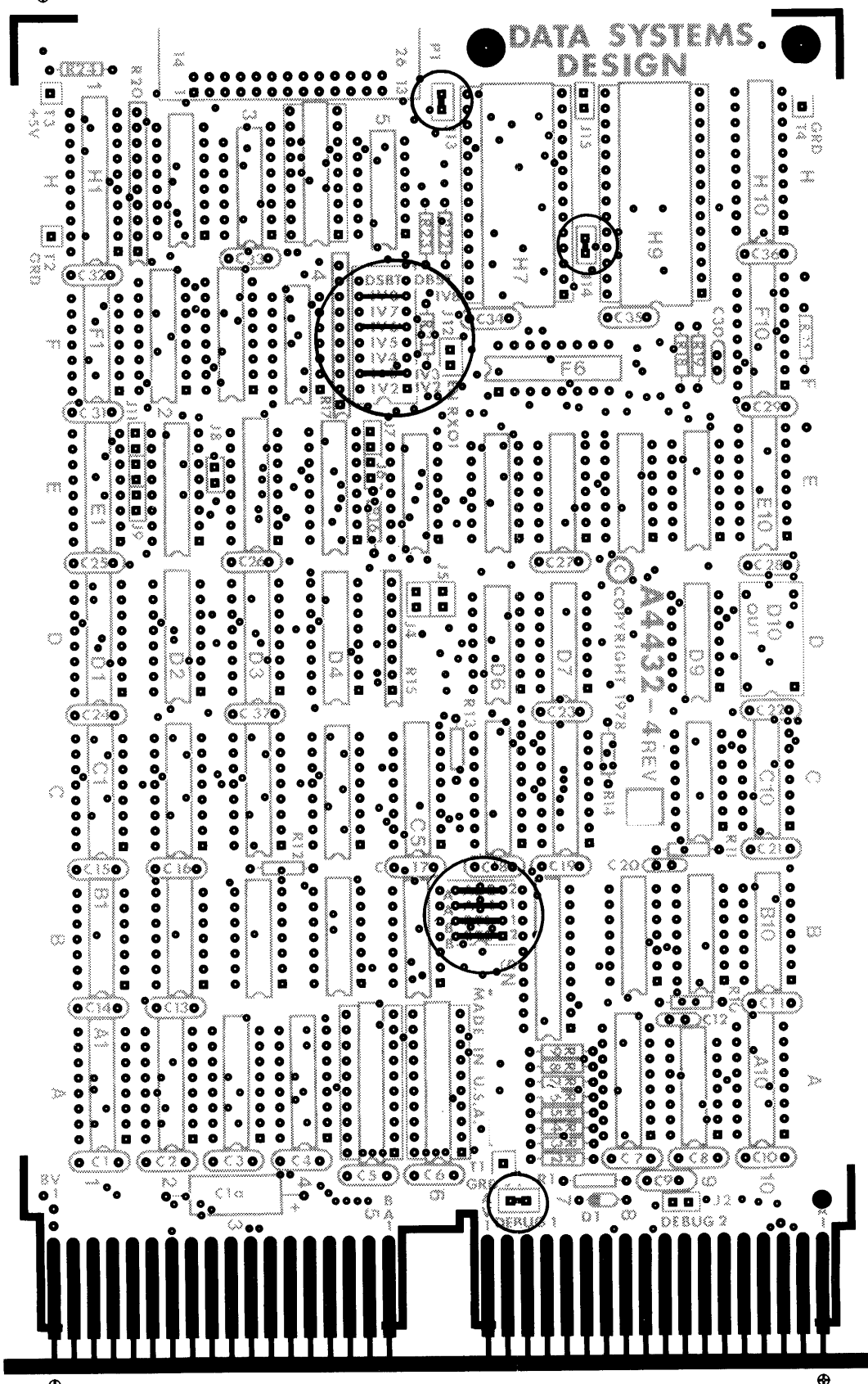


LSI-11 INTERFACE CARD

RX02 MODE



DATA SYSTEMS  
DESIGN



LSI-11 INTERFACE CARD  
RXO1 MODE

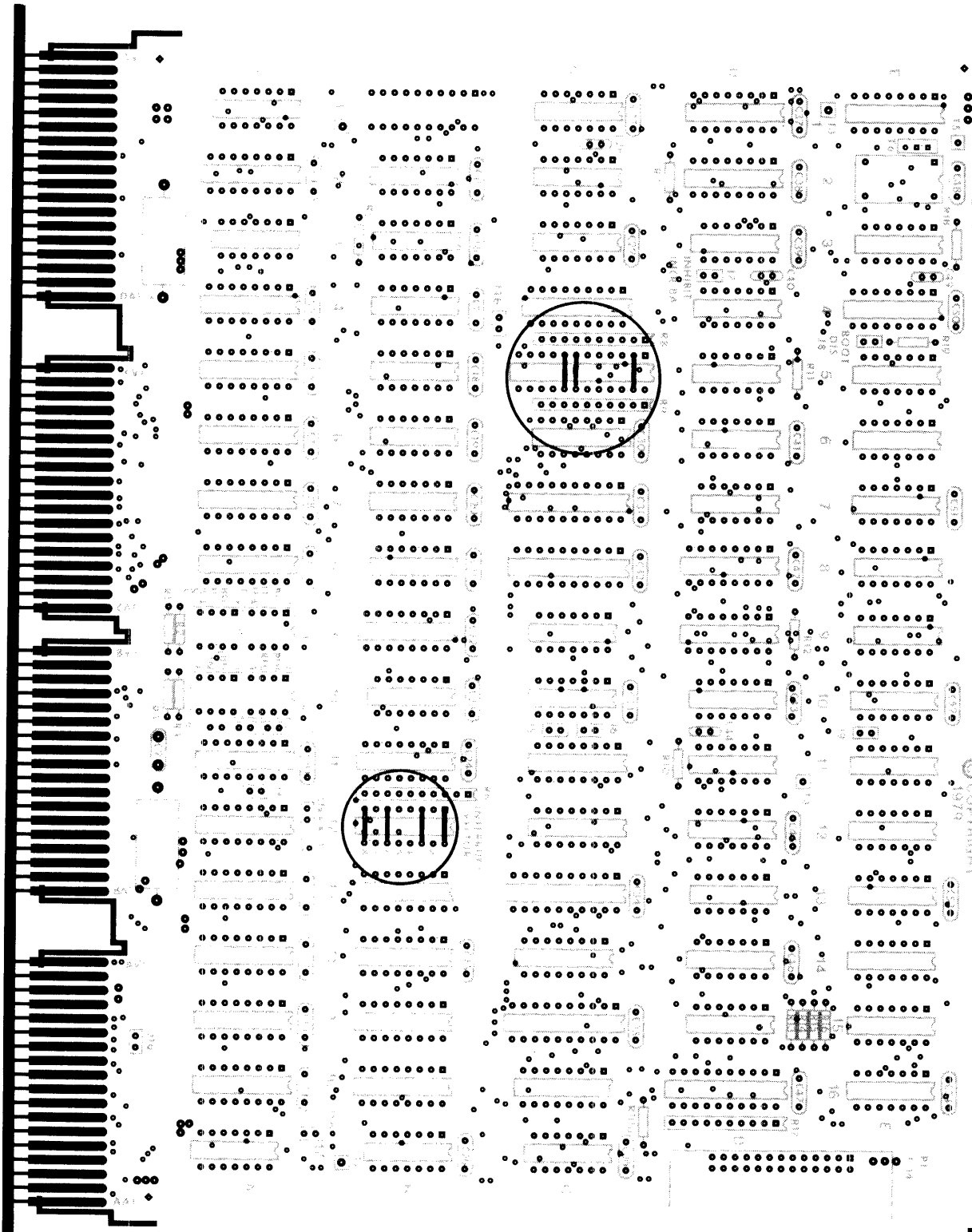
DATA SYSTEMS  
DESIGN

A4430-4

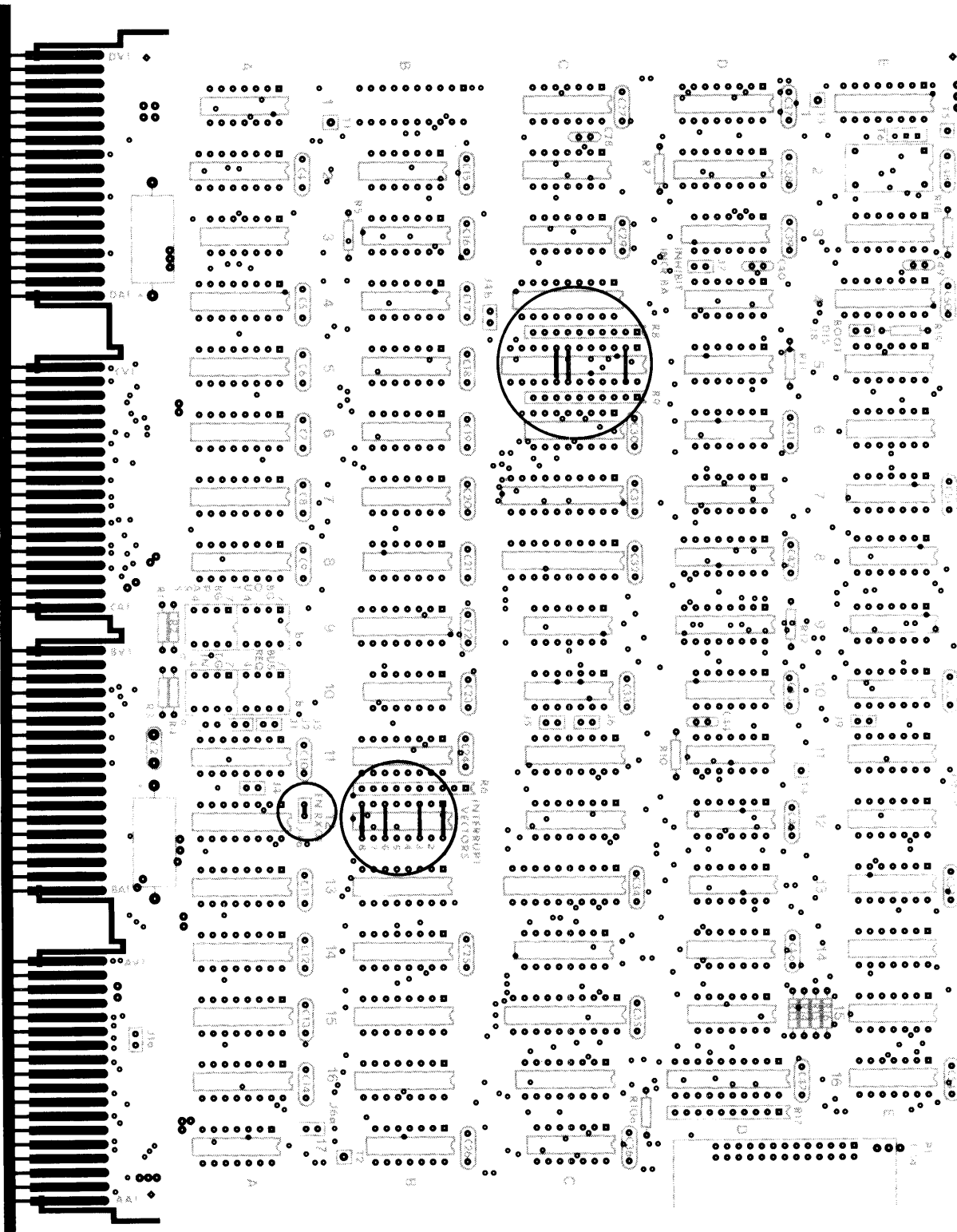
REV

MADE IN U.S.A.  
COPYRIGHT  
1979

S N



**PDP-11 INTERFACE CARD  
RX02 MODE**



**PDP-11 INTERFACE CARD  
RX01 MODE**



**APPENDIX C**  
**DISK DRIVE MAINTENANCE MANUAL**



DSD MODEL 110  
DSD MODEL 210  
DSD MODEL 440

This change package is applicable to all DSD Flexible Disk Systems utilizing the Shugart Associates SA 800/801 drives.

Some Shugart drives are being delivered with a new PCB, part number 25136-0 replacing previous PCB, part number 25103-4.

This package provides a change description, board layout, schematic diagram and timing diagram of the new LSI PC board. Although physical locations of jumper terminals may be different, the configuration table in your current manual is still accurate.

This package should be placed in front of the section on Shugart drives in your current DSD equipment manual. Since the two versions of the PCB are completely interchangeable, the schematics and board layouts that are in your manual should be retained to provide coverage of both configurations.

## CHANGE DESCRIPTION

### 1. Interface

- a) The LSI PCB is completely interchangeable with the current 800/801 board. The interface is identical at each connector.
- b) The LSI PCB does not use the -5V/-15V pins at J5. The pin at J5 is not terminated to anything.

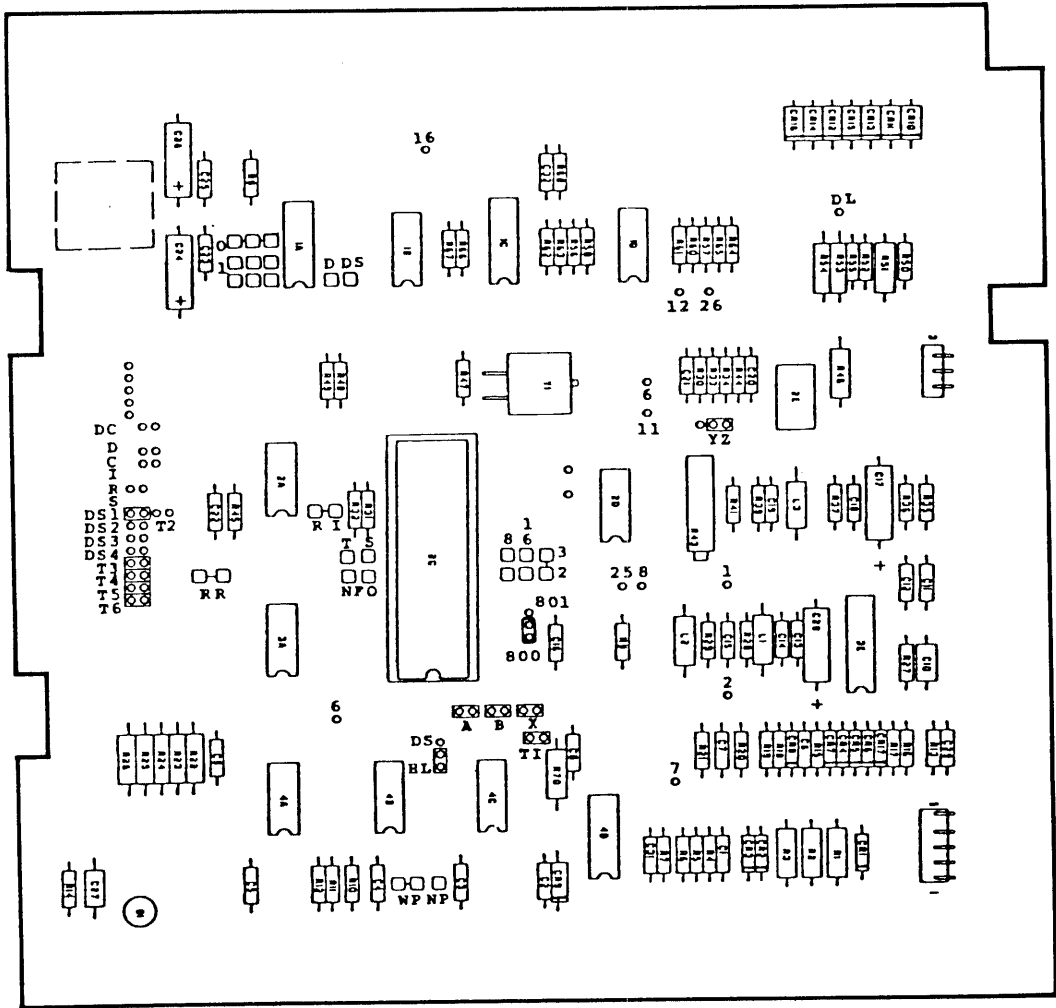
### 2. Optional Features

- a) An add-trace option "NFO" prevents the head from being forced out past track  $\emptyset$ ,
- b) An add-trace option "TS" enables true FM data separation, maintaining synchronization during address marks.

### 3. Test Points

- a) Most test points are retained:
  - 1, 2 Amplified read signal
  - 5, 6, 7 Ground
  - 10 -Index
  - 11 +Head Load
  - 12 -Index/Sector Pulses
  - 16 +Read Data
  - 25 +Write Protect
  - 26 +Detect Track  $\emptyset$
  - 27 +Step Pulse (No longer gates with read gate)
- b) Test Points Not Needed
  - 3, 4 Differentiated read signal (buried inside read chip)
  - 21, 24-Data separator timing (no pot alignment is necessary)
- c) Test Points Added
  - 8 +Data Window (for troubleshooting FM data separation)
- d) The terminal pin "DL" is provided to supply +5 volts to the Door Lock Option PCB.
- e) No adjustment of the sector separator or data separator is necessary.

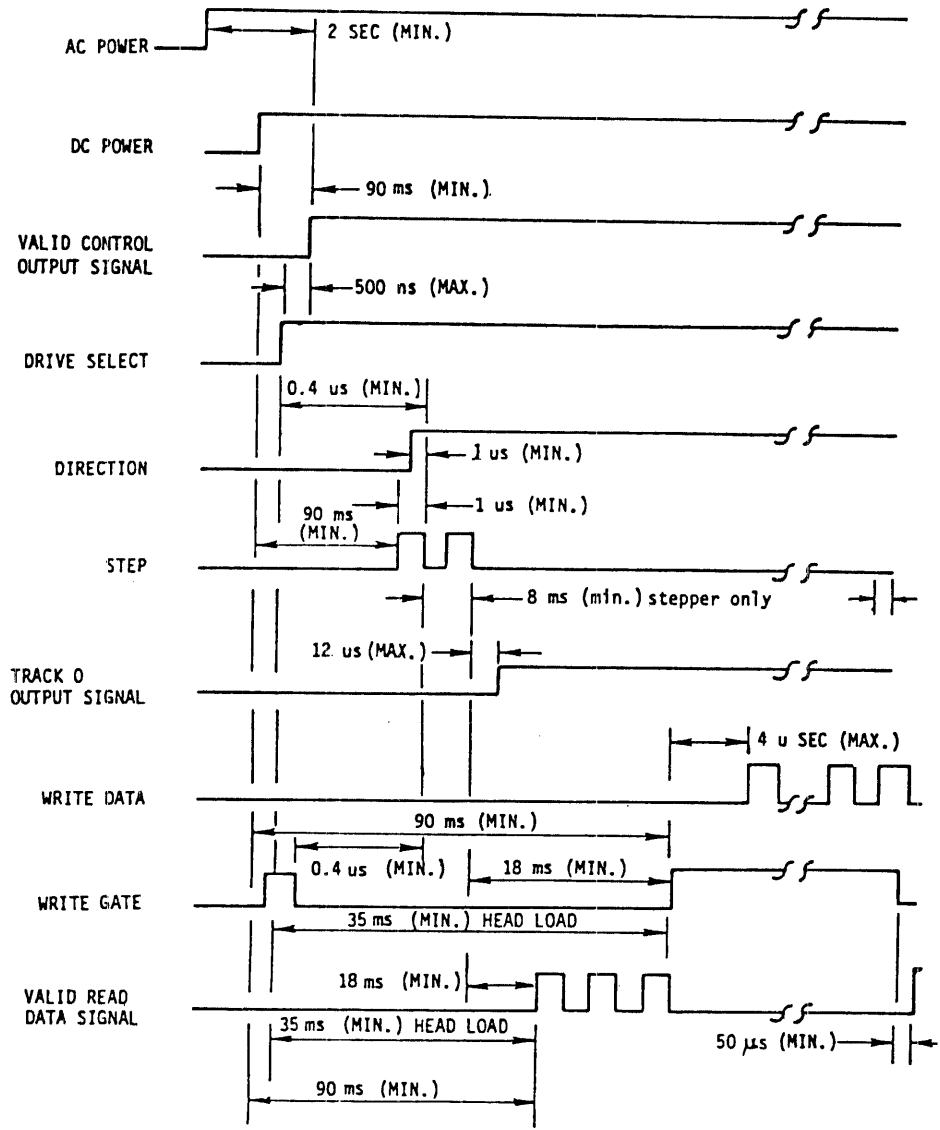




Jumper Plug Installed as Shipped  
 Test Point

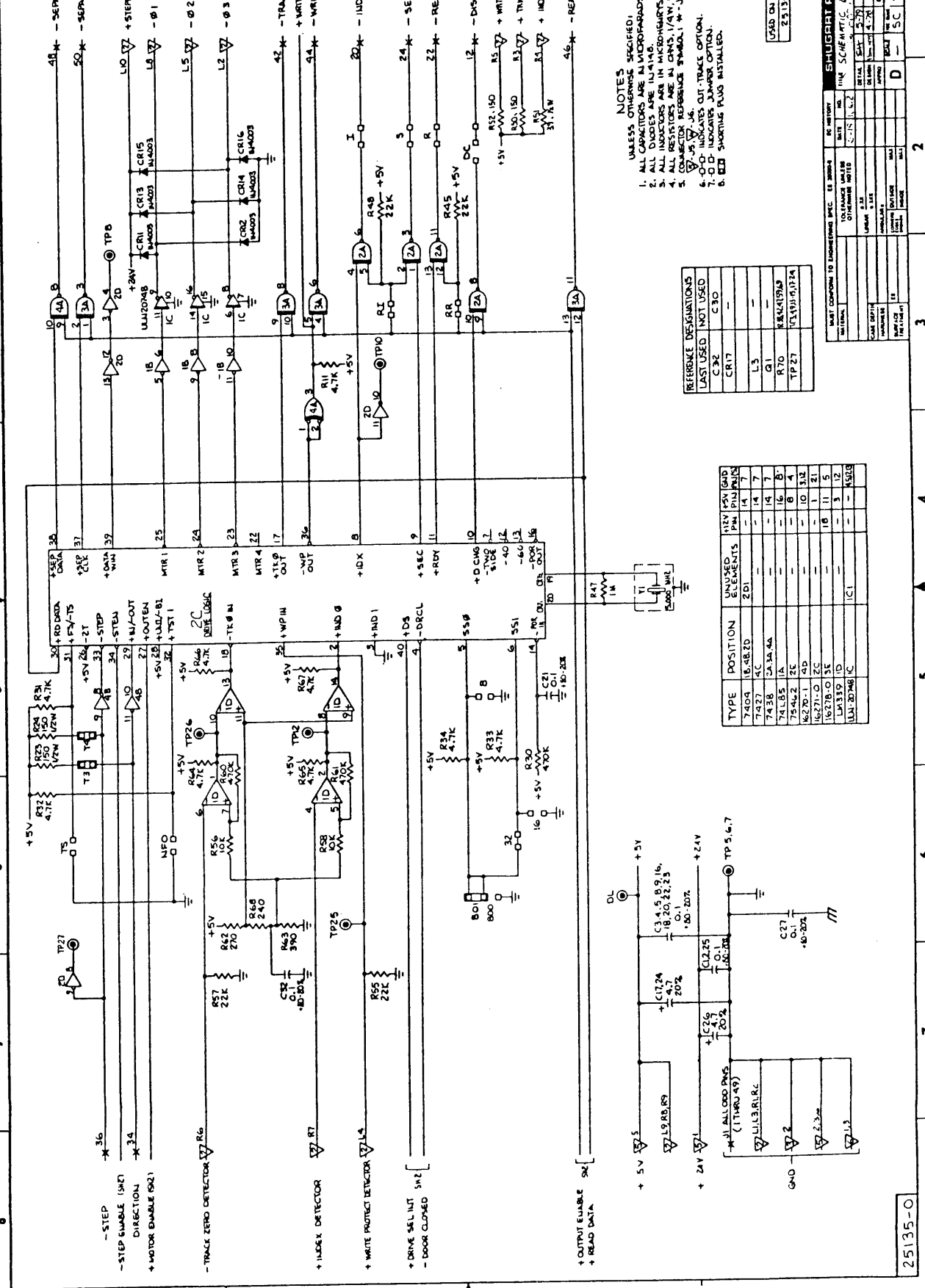
SA800LSI PCB COMPONENT LOCATIONS

800 LSI TIMINGS



GENERAL CONTROL  
DATA TIMING REQUIREMENTS (POSITIVE LOGIC)

25135-0



- NOTES
1. ALL CAPACITORS ARE MICROFARADS, 50V.
  2. ALL INDUCTORS ARE IN MICROHENRS, 107A.
  3. ALL RESISTORS ARE IN OHMS, 1/4W, 5%. UNLESS OTHERWISE SPECIFIED.
  4. CONNECTOR REFERENCE SYMBOLS: \* - J1, J2, J3, J4, J5, J6.
  5. O-D INDICATES OUT-TURNS OPTION.
  6. O-D INDICATES OUT-TURNS OPTION.
  7. O-D INDICATES OUT-TURNS OPTION.
  8. RED SHORTING PLUGS INSTALLED.

REFERENCE DESIGNATIONS

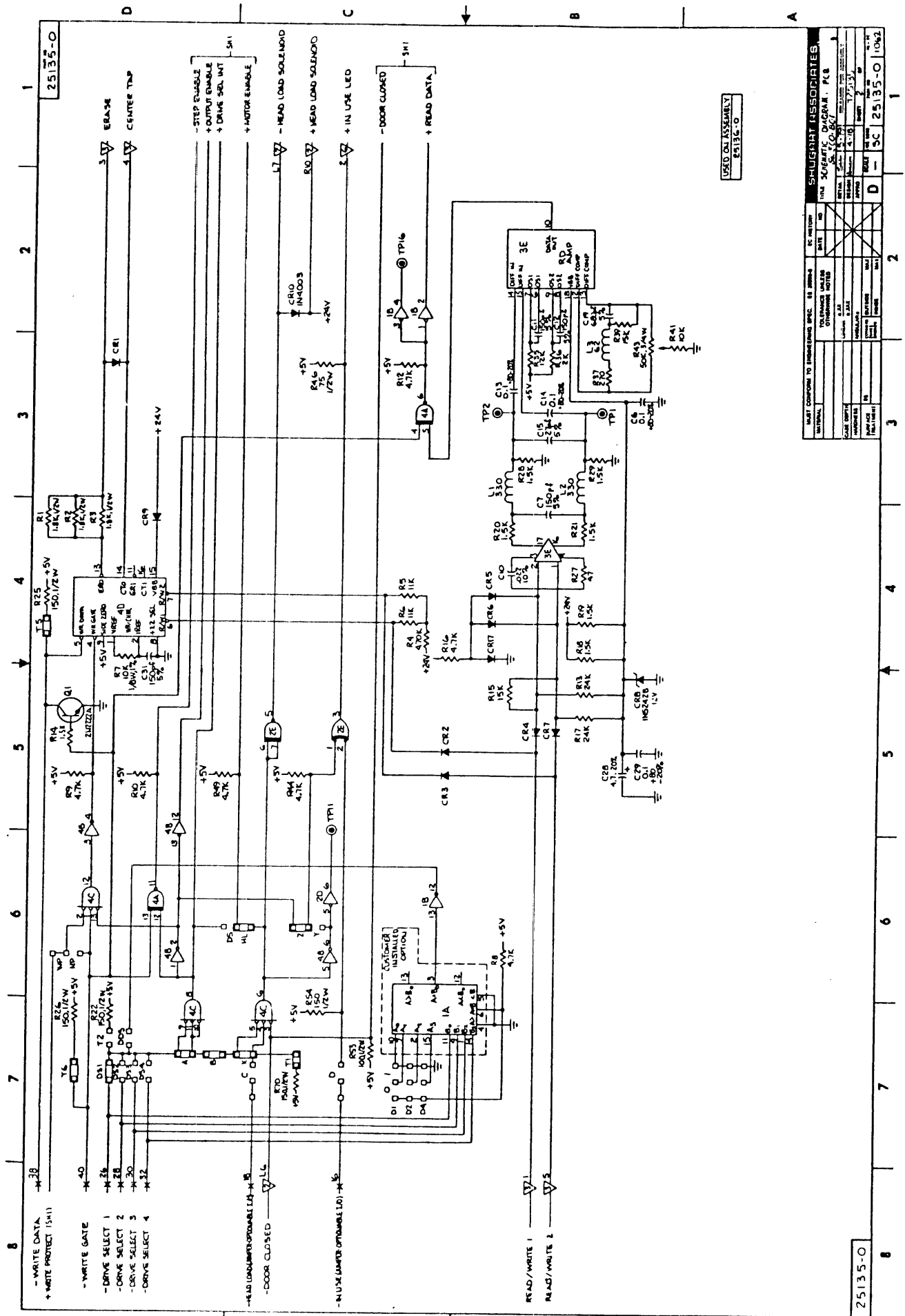
LAST USED	NOT USED
C 32	C 30
CR17	
L3	
O1	
R10	
R17	
TP21	

TYPE	POSITION	UNUSED ELEMENTS	MIN. PIN	MAX. PIN
7409	16, 4E, 2D	Z01	1	7
7427	4C		14	7
7438	2A, 3A, 4A		14	7
74LS5	1A		16	5
7546	2E		6	4
8270-1	4D		10	13
16271-0	5C		1	31
16271-0	5B		18	11
16271-0	5A		3	12
16271-0	5D		3	12
16271-0	5E		3	12
16271-0	5F		3	12
16271-0	5G		3	12
16271-0	5H		3	12
16271-0	5I		3	12
16271-0	5J		3	12
16271-0	5K		3	12
16271-0	5L		3	12
16271-0	5M		3	12
16271-0	5N		3	12
16271-0	5O		3	12
16271-0	5P		3	12
16271-0	5Q		3	12
16271-0	5R		3	12
16271-0	5S		3	12
16271-0	5T		3	12
16271-0	5U		3	12
16271-0	5V		3	12
16271-0	5W		3	12
16271-0	5X		3	12
16271-0	5Y		3	12
16271-0	5Z		3	12

SHUGART ASSOCIATES

DATE	BY	REV.	DESCRIPTION
11/15/78	J. S. J.	1	INITIAL RELEASE
11/15/78	J. S. J.	2	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	3	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	4	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	5	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	6	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	7	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	8	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	9	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	10	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	11	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	12	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	13	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	14	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	15	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	16	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	17	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	18	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	19	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	20	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	21	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	22	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	23	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	24	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	25	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	26	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	27	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	28	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	29	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	30	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	31	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	32	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	33	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	34	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	35	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	36	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	37	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	38	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	39	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	40	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	41	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	42	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	43	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	44	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	45	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	46	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	47	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	48	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	49	REVISIONS FOR ASSEMBLY
11/15/78	J. S. J.	50	REVISIONS FOR ASSEMBLY

25135-0



MATERIAL		PC HISTORY		REVISION	
REV	DATE	BY	CHKD	NO	DESCRIPTION
1	10/10/82	...	...	1	...
2	...	...	...	2	...
3	...	...	...	3	...
4	...	...	...	4	...
5	...	...	...	5	...
6	...	...	...	6	...
7	...	...	...	7	...
8	...	...	...	8	...
9	...	...	...	9	...
10	...	...	...	10	...

25135-0

# SA800/801 Diskette Storage Drive

# Maintenance Manual



# TABLE OF CONTENTS

<b>Section</b>		<b>Page</b>
1	Maintenance Features . . . . .	1
2	Diagnostic Techniques . . . . .	3
3	Preventive Maintenance . . . . .	5
4	Removals Adjustments . . . . .	7
5	Physical Locations . . . . .	17
6	SA809 Exerciser Connections . . . . .	21
7	SA800/801 PCB Component Locations . . . . .	22
8	SA800/801 Logic Manual . . . . .	23
9	SA800/801 Schematic Diagrams . . . . .	31





## 1.0 MAINTENANCE FEATURES

### 1.1 Alignment Diskette

The SA120 Alignment Diskette is used for alignment of the SA800/801. The following adjustments can be made using the SA120.

1. R/W Head radial alignment using track 38.
2. R/W Head azimuth alignment using track 76.
3. Index Photo-Detector Adjustment using tracks 01 and 76.
4. Track 00 is recorded with standard IBM 3740 format.
5. TK 75 has 1f + 2f signal for load pad adjustment.

Caution should be exercised in using the SA120 Alignment Diskette. Tracks 00, 01, 36, 37, 38, 39, 40, 75, and 76 should not be written on. To do so will destroy pre-recorded tracks.

### 1.2 SA809 Exerciser

The SA809 Exerciser is built on a PCB whose dimensions are 8" x 8". The Exerciser PCB can be used in a stand alone mode or it can be built into a test station or used in a tester for field service.

The Exerciser is designed to enable the user to make all adjustments and check outs required on the SA800/801 drives, when used with the SA120 alignment diskette.

The exerciser has no intelligent data handling capabilities but can write both 1f and 2f frequencies. The exerciser can enable read in the drive to allow checking of read back signals.

Refer to Section 6 for illustration.

### 1.3 Special Tools

The following special tools are available for performing maintenance on the SA800/801.

Description	Part Number
Alignment Diskette	SA120-1
Cartridge Guide Adj. Tool	50377-1
Head Penetration Gauge	50380-0
Load Bail Gauge	50391-0
Exerciser	50619-0
Spanner Wrench	50752-0



## **2.0 DIAGNOSTIC TECHNIQUES**

### **2.1 Introduction**

Incorrect operating procedures, faulty programming, damaged diskettes, and "soft errors" created by airborne contaminants, random electrical noise, and other external causes can produce errors falsely attributed to drive failure or misadjustment.

Unless visual inspection of the drive discloses an obvious misalignment or broken part, attempt to repeat the fault with the original diskette, then attempt to duplicate fault on second diskette.

### **2.2 "Soft Error" Detection and Correction**

Soft errors are usually caused by:

1. Airborne contaminants that pass between the read/write head and the disk. Usually these contaminants can be removed by the cartridge self-cleaning wiper.
2. Random electrical noise that usually lasts for a few  $\mu$  sec.
3. Small defects in the written data and/or track not detected during the write operation that may cause a soft error during a read.
4. Worn or defective load pad.

The following procedures are recommended to recover from the above mentioned soft errors:

1. Reread the track ten (10) times or until such time as the data is recovered.
2. If data is not recovered after using step 1, access the head to the adjacent track in the same direction previously moved, then return to the desired track.
3. Repeat step 1.
4. If data is not recovered, the error is not recoverable.

### **2.3 Write Error**

If an error occurs during a write operation, it will be detected on the next revolution by doing a read operation, commonly called a "write check." To correct the error, another write and write check operation must be done. If the write operation is not successful after ten (10) attempts have been made, a read operation should be attempted on another track to determine if the media or the drive is failing. If the error still persists the diskette

should be replaced and the above procedure repeated. If the failure still exists, consider the drive defective. If the failure disappears, consider the original diskette defective and discard it.

### 2.4 Read Error

Most errors that occur will be "soft" errors. In these cases, performing an error recovery procedure will recover the data.

### 2.5 Seek Error

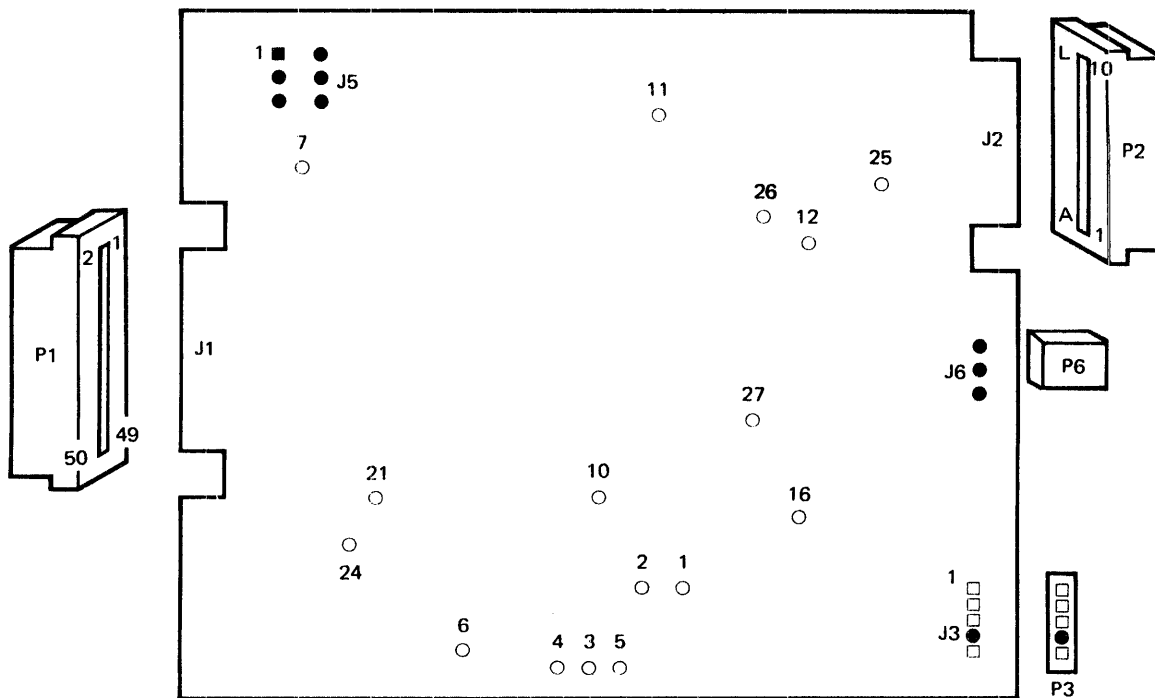
1. Stepper malfunction.
2. Improper carriage torque.

To recover from a seek error recalibrate to track 00 and perform another seek to the original track.

### 2.6 Test Points- 800/801

TP	1	Read Data Signal
	2	Read Data Signal
	3	Read Data (Differentiated)
	4	Read Data (Differentiated)
	5	Signal Ground
	6	Signal Ground
	7	Signal Ground
	10	- Index
	11	+ Head Load
	12	- Index and 801 Sector Pulses
	16	+ Read Data
	21	- Data Separator Time + 1
	24	- Data Separator Time + 2
	25	+ Write Protect
	26	+ Detect Track 00.
	27	+ Gated Step Pulses

### 2.7 Test Point Locations



### 3.0 PREVENTIVE MAINTENANCE

#### 3.1 Introduction

The prime objective of any preventive maintenance activity is to provide maximum machine availability to the user. Every preventive maintenance operation should assist in realizing this objective. Unless a preventive maintenance operation cuts machine downtime, it is unnecessary.

Visual inspection is the first step in every scheduled maintenance operation. Always look for corrosion, dirt, wear, binds, and loose connections. Noticing these items during PM may save downtime later.

Remember, do not do more than recommended preventive maintenance on equipment that is operating satisfactorily.

#### 3.2 Preventive Maintenance Procedures

Details of preventive maintenance operations are listed in Figure 1. During normal preventive maintenance, perform only those operations listed on the chart for that preventive maintenance period. Details on adjustments and service checks can be found in the maintenance manual. Observe all safety procedures.

#### 3.3 Cleanliness

Cleanliness cannot be overemphasized in maintaining the SA800/801. Do not lubricate the SA800/801; oil will allow dust and dirt to accumulate. The read/write head should be cleaned only when signs of oxide build up are present.

UNIT	FREQ MONTHS	CLEAN	OBSERVE
Read/Write Head	12	Clean Read/Write Head ONLY IF NECESSARY	Oxide build up
R/W Head Load Button	12*	Replace	
Stepper Motor and Lead Screw	12	Clean off all oil, dust, and dirt	Inspect for nicks and burrs
Belt	12		Frayed or weakened areas
Base	12	Clean base	Inspect for loose screws, connectors, and switches
Read/Write Head	12		Check for proper alignment

\*Assumes normal usage

Figure 1 PM Procedures



## **4.0 REMOVALS, ADJUSTMENTS**

For parts location, see Section 5.

### **4.1 Motor Drive**

#### **4.1.1 Drive Motor Assembly: Removal and Installation**

- a. Extract 3 contacts to disconnect motor from AC connector.
- b. Loosen two screws holding capacitor clamp to the base. Remove rubber boot and disconnect motor leads from capacitor.
- c. Remove connectors from PCB and remove PCB.
- d. Remove belt from drive pulley.
- e. Remove 4 screws holding the motor to the base casting and remove motor.
- f. Reverse the procedure for installation.

Note: Insure ground lead is installed between capacitor clamp and base.

#### **4.1.2 Motor Drive Pulley**

- a. Loosen set screw and remove pulley.
- b. Reverse procedure for installation.

Note: When installing a new pulley, the drive pulley must be aligned with the spindle pulley so that the belt tracks correctly.

### **4.2 Side Cover: Removal**

- a. Retract screw from upper casting wall sufficiently to allow the side cover to be rocked out.
- b. Lift cover off screw in lower casting wall.

### **4.3 Cartridge Guide Access**

- a. Remove side cover (Section 4.2).
- b. Position head to approximate center of head load bail (to prevent load arm damage).
- c. Loosen 2 screws holding cartridge guide to door latch plate.
- d. Swing cartridge guide out.
- e. When the guide is swung in, it must be adjusted as per Section 4.9.2.

#### 4.4 Sector/Index LED Assembly: Removal and Installation

- Remove side cover (Section 4.2).
- Disconnect the wires to the LED terminals (solder joints).
- Remove the screw holding the LED assembly to the cartridge guide.
- Reverse the procedure for installation.
- Check index timing and readjust if necessary.

#### 4.5 Write Protect Detector: Removal and Installation

- Remove connectors from PCB and remove PCB.
- Extract wires from P2 connector, pins L3, L4, R5 (E), and R8 (S).
- Remove cable clamps.
- Remove side cover (Section 4.2).
- Remove screw holding the detector bracket and remove assembly.
- Reverse procedure for reinstalling. Connect the wires to P2 by the following: Red to '3' (L3), Grey to '4' (L4), Black to 'E' (R5) and White to 'J' (R8).

#### 4.5.1 Write Protect Detector Adjustment

- Insert SA101 diskette into drive. Write protect hole must be open.
- Set oscilloscope to AUTO sweep, 2V/div. and monitor TP25.
- Loosen screw on detector assembly and adjust until maximum amplitude is achieved. Tighten screw.

#### 4.6 Head Load Actuator

##### 4.6.1 Head Load Actuator: Removal and Installation

- Remove side cover (Section 4.2).
- Disconnect the wires to the actuator terminals (solder joints).
- Swing out the cartridge to guide assembly (Section 4.3).

- Remove screw holding the actuator to the cartridge guide.

**CAUTION:** Restrain the head load arm to prevent its impact with the head.

- Reverse the procedure for installation.

##### 4.6.2 Head Load Actuator Adjustment

- Remove side cover.
- Energize Head Load Coil.
- Place Head Load Actuator adjustment tool, P/N 50391, on platen.
- Adjust down stop so that the top of Head Load Bail is flush with top of tool within  $\pm .005$ " at track 76. Reference Figure 3.
- Step carriage to track 38.
- De-energize Head Load Coil.
- Place adjustment tool onto R/W Head and place load button in cup of tool.
- Adjust up stop on actuator so that bail just touches Head Load Arm or has  $.005$ " clearance or lifts Load Arm  $.005$ ". Reference Figure 2.
- Energize Head Load Coil and step carriage between track 00 and 76. Insure that there is a clearance of a minimum of  $.010$ " between Head Load Bail and Head Load Arm.
- Replace side cover.

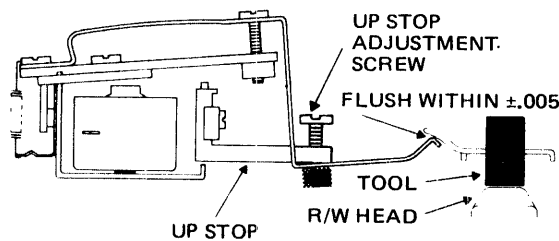


FIGURE 2 HEAD LOAD ACTUATOR UPSTOP ADJUSTMENT

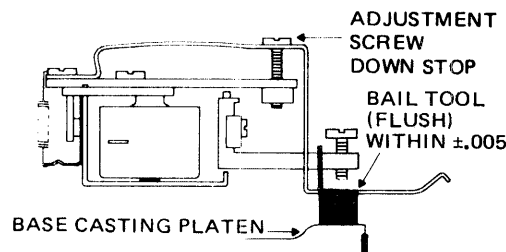


FIGURE 3 HEAD LOAD ACTUATOR DOWN STOP ADJUSTMENT



#### 4.6.3 Head Load Actuator Timing

- a. Insert Alignment Diskette (SA120).
- b. Step carriage to track 00.
- c. Sync oscilloscope on TP11 (+ Head Load). Set time base to 10MSEC/division.
- d. Connect one probe to TP1 and the other to TP2. Ground probes to the PCB. Set the inputs to add and invert one input
- e. Energize the Head Load solenoid and observe the read signal on the oscilloscope. The signal must be at 50% of full amplitude by 35Msec. Reference Figure 4.
- f. If this is not met, continue on with the procedure.
- g. Check adjustments outlined in paragraph 4.6.2.
- h. If item 'g' is ok, adjust down stop screw (Figure 6) clockwise until timing is met.

Note: Not to exceed  $\frac{1}{4}$  turn.

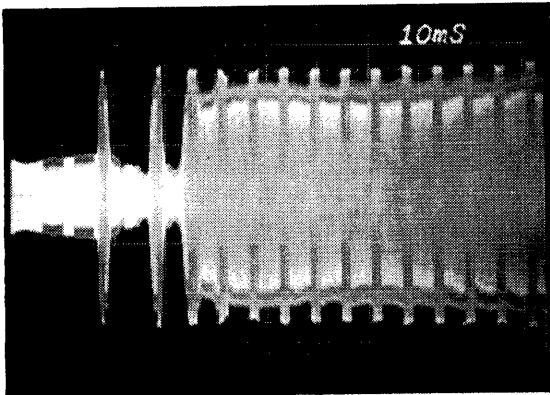


FIGURE 4 HEAD LOAD ACTUATOR TIMING

#### 4.7 Index/Sector Photo Transistor Assembly

##### 4.7.1 Index/Sector Photo Transistor Assembly: Removal and Installation

- a. Disconnect P2 connector from PCB.
- b. Remove wires from Door Closed switch and extract wires from P2 connector pin 9 (L9) Black, H (R7) Brown, 6 (L6) Red and B (R2) Orange.
- c. Remove cable clamp holding wires from detector.

- d. Remove screw holding detector to the base plate and remove assembly.
- e. To install reverse procedure.

##### 4.7.2 Index/Sector Photo Transistor Potentiometer Adjustment

- a. Insert Alignment Diskette (SA120).
- b. Using oscilloscope monitor TP-12 (- Index), sync internal negative, DC coupled, set vertical scale to 2 V/cm.
- c. Adjust the potentiometer on the Sector/Index Phototransistor to obtain a pulse of 1.7 msec.  $\pm .5$  msec. duration.
- d. Continue adjustment in Section 4.7.3.

##### 4.7.3 Index/Sector Adjustment

- a. Insert Alignment Diskette (SA120).
- b. Step carriage to track 01.
- c. Sync oscilloscope, external negative, on TP 12 (- Index). Set time base to 50  $\mu$ sec/division.
- d. Connect one probe to TP 1 and the other to TP 2. Ground probes to the PCB. Set the inputs to AC, Add and invert one channel. Set vertical deflection to 500 MV/division.
- e. Observe the timing between the start of the sweep and the first data pulse. This should be  $200 \pm 100 \mu$ sec. If the timing is not within tolerance, continue on with the adjustment. Reference Figure 5.

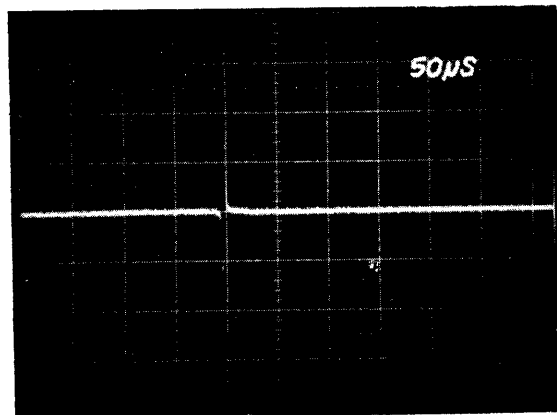


FIGURE 5 INDEX TIMING

- f. Loosen the holding screw in the Index Transducer until the transducer is just able to be moved.
- g. Observing the timing, adjust the transducer until the timing is  $200 \pm 100 \mu\text{sec}$ . Insure that the transducer assembly is against the registration surface on the base casting.
- h. Tighten the holding screw.
- i. Recheck the timing.
- j. Seek to track 76 and reverify that the timing is  $200 \pm 100 \mu\text{sec}$ .

#### 4.8 Spindle Assembly

- a. Remove side cover (Section 4.2).
- b. Swing out cartridge guide (Section 4.3).
- c. Remove the nut and washer or 2 spring washers holding the spindle pulley. On late level drives, Spanner Wrench 50752 may be used to hold spindle.

CAUTION: The pre-loaded rear bearing may fly out when spindle pulley is removed.

- d. Withdraw spindle hub from opposite side of baseplate.
- e. Reverse the procedure for installation.
- f. Tighten nut to 20 in./lbs. If spring washers are used, insure they are compressed. Add a drop of LOCTITE® #290 to threads.

#### 4.8.1 Clamp Hub Removal

- a. Remove hub clamp plate. Reference Figure 6.
- b. Remove clamp hub and spring.
- c. To install, reverse the procedure. No adjustment necessary.

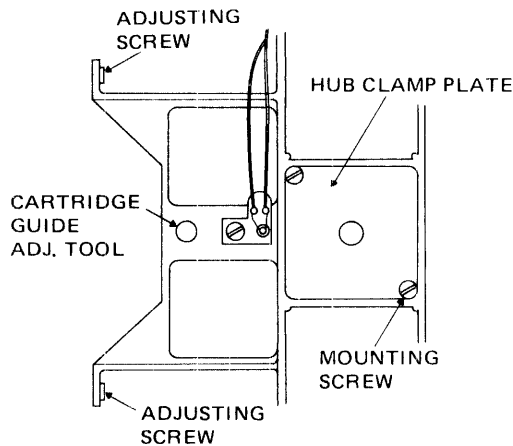


FIGURE 6 CARTRIDGE GUIDE ADJUSTMENT

#### 4.9.1 Cartridge Guide Removal

- a. Perform steps 4.3 through 4.6.1.
- b. Remove C-clip from pivot shaft. Reference Figure 7.
- c. Remove pivot shaft.
- d. Tilt the cartridge guide slightly, and remove it from the upper pivot.
- e. To install the cartridge guide, reverse the procedure.

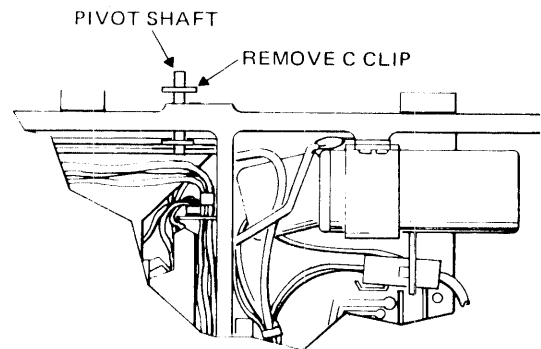


FIGURE 7 CARTRIDGE GUIDE REMOVAL

#### 4.9.2 Cartridge Guide Adjustment

- a. Insert the shoulder screw (tool P/N 50377-1) through the adjustment hole in the cartridge guide and screw completely into the base casting (hand tight). Reference Figure 6.
- b. Move the handle into the latched position and hold it lightly against the latch.
- c. Tighten two screws holding the cartridge guide to the latch plate.
- d. Remove the tool and check to determine the flange on the clamp hub clears the cartridge guide when the spindle is rotating. If the clamp hub rubs on the cartridge guide, repeat the adjustment procedure.
- e. Check index alignment per Section 4.7.3.
- f. Insert diskette, close and open door, then check for proper operation.

#### 4.10 Front Plate Assembly: Removal

- a. Remove side cover (Section 3.4.2).
- b. Swing out the cartridge guide assembly (Section 4.3).

- c. Remove 4 screws holding the front plate assembly to the base casting.
- d. To install, reverse the procedure.
- e. Check Index adjustment Section 4.7.3.

#### 4.11 Head Amplitude Check

These checks are only valid when writing and reading back as described below. If this amplitude is below the minimum specified, the load pad should be replaced and the head should be cleaned if necessary before re-writing and re-checking. Insure the diskette used for this check is not "worn" or otherwise shows evidence of damage on either the load pad or head side.

- a. Install good media.
- b. Select the drive and step to TK 76.
- c. Sync the oscilloscope on TP-12 (- Index) connect one probe on TP-2 and one on TP-1, on the drive PCB. Ground the probes to the PCB add and invert one input. Set volts per division to 50mv and time base to 20 M sec. per division.
- d. Write the entire track with 2F signal (all one's).
- e. The average minimum read back amplitude, peak to peak, should be 110 millivolts.

If the output is below minimum and a new load pad and different media is tried and the output is still low, it will be necessary to install a new head and carriage assembly.

##### 4.11.1 Stepper/Carriage Assembly; Removal and Installation

- a. Remove cable clamp holding R/W head cable on PCB side of drive.
- b. Remove side cover (Section 4.2).
- c. Extract stepper cable contacts from P2 connector. Black 10 (L10), Red 2 (L2), Brown 5 (L5), and Orange 8 (8).

Note: This step is only necessary if the stepper motor is to be replaced.

- d. Loosen (2) screws and swing clamp down to allow withdrawal of motor.

**CAUTION: DO NOT LOOSEN THREE SCREWS COATED WITH GLYPTOL.**

- e. Remove the grommet on the cable that is inserted into a slot on the Track 00 Detector bracket.
- f. Turn stepper shaft until the carriage runs off the end of the lead screw.
- g. To install stepper/carriage assembly, reverse procedure. Note steps "h" and "i".
- h. If installing a new carriage, set the pre-load nut in the #2 notch. Reference Figure 8.
- i. When threading lead screw into carriage assembly, press the pre-load nut slightly against spring in order to start thread. After threading, insure there is a gap between pre-load nut and rear of carriage.

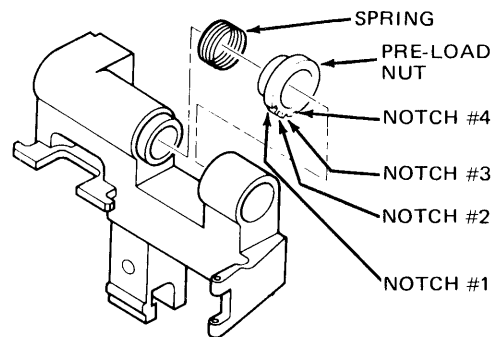


FIGURE 8 CARRIAGE ASSEMBLY

##### 4.11.2 Carriage Assembly Readjustment After Replacement

- a. Loosen Track 00 stop collar and manually move the carriage towards the stepper by rotating the lead screw until the carriage load arm tab is near the edge of the load bail. Tighten the collar set screw.
- b. Position the Track 00 flag approximately in the center of its slot and tighten the screw. Move the carriage towards the spindle by rotating the lead screw until the flag is clear of the detector.
- c. Insert the SA120 alignment diskette and load the head. Set the scope as explained in Section 4.11.3 steps c and d.
- d. Step the carriage towards track 00 until the track 00 signal is detected on the interface pin 42.

- e. Loosen the 2 stepper motor mounting screws slightly and slowly rotate the stepper motor case until a read data signal off of track 00 appears. Continue rotation until maximum amplitude is obtained. This is only a rough adjustment.

**CAUTION: DO NOT LOOSEN THREE SCREWS COATED WITH GLYPTOL.**

- f. Step the carriage to TK 38 and proceed with head radial adjustments. Refer to Section 4.11.3.
- g. Adjust Track 00 stop (Section 4.11.7).
- h. Adjust Track 00 flag (Section 4.11.8).
- i. Adjust index (Section 4.7.3).
- j. Adjust Azimuth (Section 4.11.9 ).

#### 4.11.3 Head Radial Alignment

Note: Head radial alignment should be checked prior to adjusting index/sector, Track 00 flag or carriage stop.

- a. Load alignment diskette (SA120).  
Note: Alignment diskette should be at room conditions for at least twenty minutes before alignment.)
- b. Step the carriage to track 38.
- c. Sync the oscilloscope, external negative, on TP 12 (- CE Index). Set the time base to 20 Msec per division. This will display over one revolution.
- d. Connect one probe to TP 1 and the other to TP 2. Ground the probes on the PCB. Set the inputs to AC, Add and invert one channel. Set the vertical deflection to 100 MV/dev.
- e. The two lobes must be within 70% amplitude of each other. If the lobes do not fall within the specification, continue on with the procedure. Reference Figure 9.
- f. Loosen the two mounting screws which hold the motor clamp to the mounting plate.

**CAUTION: DO NOT LOOSEN THREE SCREWS COATED WITH GLYPTOL.**

- g. Rotate the stepper motor to radially move the head in or out. If the left lobe is less than 70% of the right, turn the stepper motor counter-clockwise as viewed from the rear. If the right lobe is less than 70% of the left lobe, turn the stepper motor clockwise as viewed from the rear.

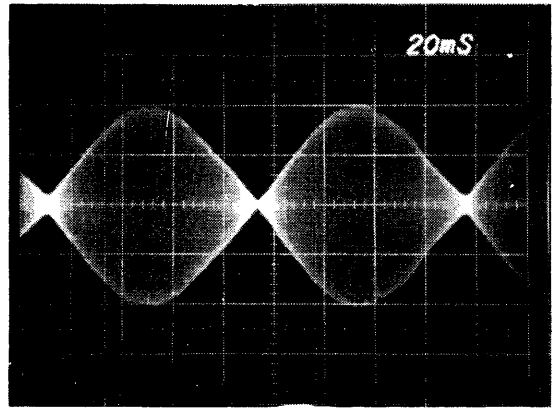


FIGURE 9 HEAD RADIAL ALIGNMENT

- h. When the lobes are of equal amplitude, tighten the motor clamp mounting screws. Reference Figure 9.
- i. Check the adjustment by stepping off track and returning. Check in both directions and readjust as required.
- j. Whenever the Head Radial Alignment has been adjusted, the Track 00 flag adjustment (Section 4.11.8), Track 00 stop (Section 4.11.7) and R/W head azimuth (Section 4.11.9 ) must be checked.

#### 4.11.3 Read/Write Head Load Button: Removal and Installation

- a. Remove side cover if installed.
- b. To remove the old button, hold the arm out away from head, squeeze the locking tabs together with a pair of needle nose pliers and press forward.
- c. To install load button, press the button into the arm, from the head side, and it will snap in place. Reference Figure 10.

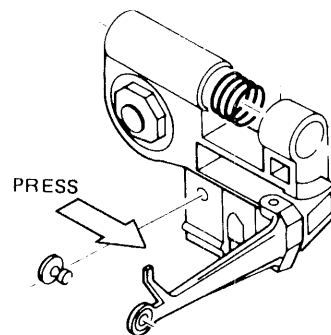


FIGURE 10

d. Adjust according to Section 4.11.4.

Note: The load arm should never be opened over 90° from carriage assembly or while at track 00 to prevent possible damage to the torsion spring.

#### 4.11.4 Read/Write Head Load Button Adjustment

- Insert Alignment Diskette (SA120).
- Connect oscilloscope to TP 1 and 2, added differentially and sync negative external on TP 12 (- INDEX).
- Step carriage to track 75.

d. Observing read signal on oscilloscope, rotate the load button counter-clockwise in small increments (10°) until maximum amplitude is obtained.

#### 4.11.5 Head Penetration Adjustment

Note: This adjustment is not normally done in the field. The only time that this adjustment need be done is when the stepper mounting plate has been loosened or removed.

- Place the penetration tool (P/N 50380) on the gauge block and insure that the gauge reads .030 (3 on the small hand) and zero the dial for the large hand. This results in a reading of .030”.

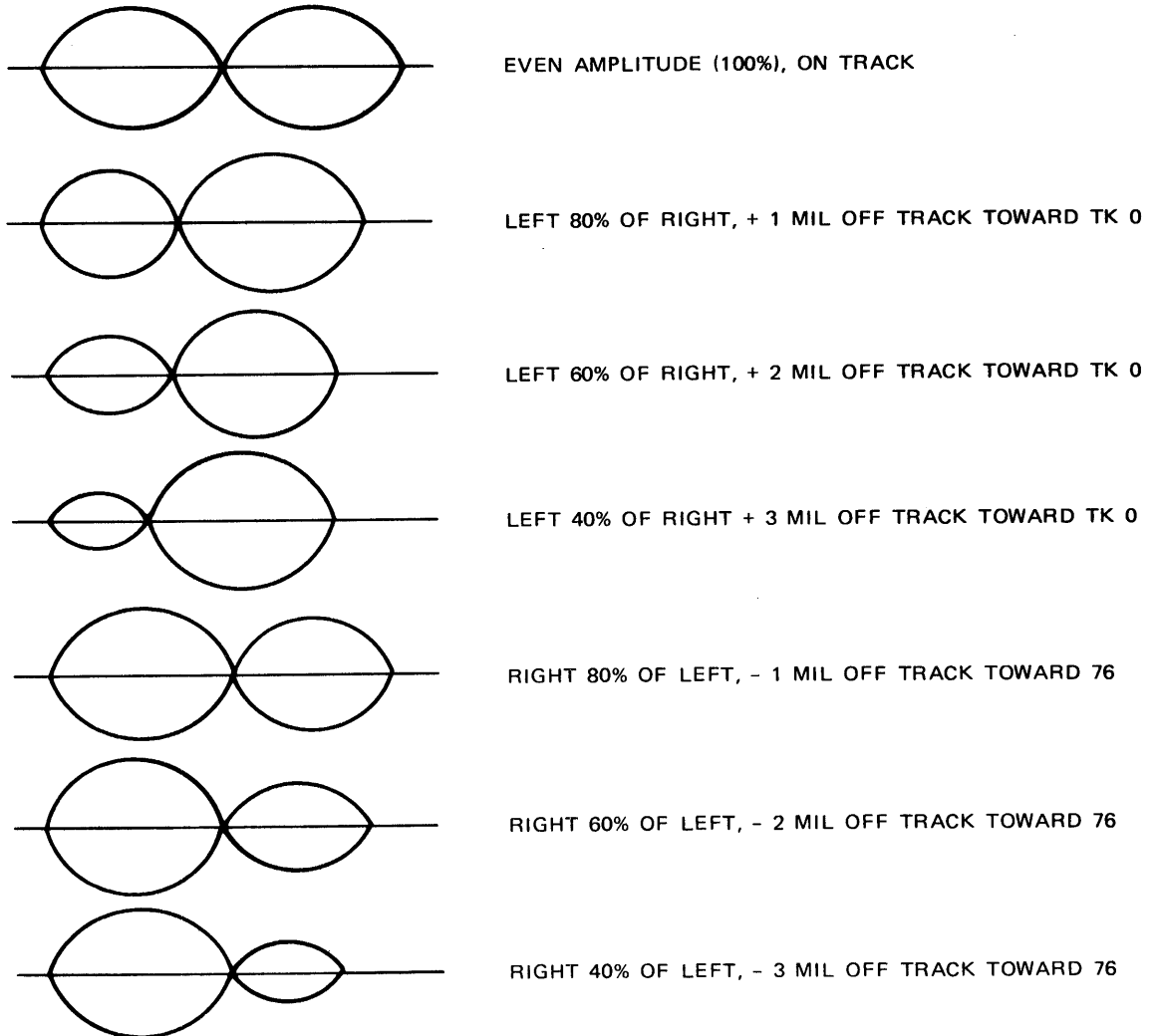


Figure 11 R/W Head Radial Alignment

- b. Swing open the cartridge as per Section 4.3.
- c. Place the penetration tool on the base assembly with the short leg on the platen, the long leg on the carriage guide bar, and the plastic tip in the center of the R/W head.
- d. The head penetration should be  $.030'' \pm .003''$  read on the gauge.
- e. If the head does not meet this adjustment, move the stepper plate laterally until the gauge reads  $.030''$ .
- f. Tighten the screws and recheck the adjustment.
- g. Return cartridge guide and adjust as per Section 4.9.2.
- h. Adjust Azimuth (Section 4.11. 9 ).
- d. If clearances are not within tolerance, continue on with adjustment procedure.
- e. Turn DC power ON.
- f. Step carriage to track 02.
- g. Loosen Track 00 stop collar.
- h. Grasp end of lead screw, in back of stepper motor, with a pair of pliers and manually turn lead screw clockwise to the track -01 position. (Next detent position on stepper motor.)
- i. Position the stop collar axially along the lead screw so there is  $.020'' \pm .010''$  between collar and carriage. Rotate the collar toward inside until the stop on the collar contacts the carriage stop surface. Tighten screw.

#### 4.11.6 Track 00 Detector: Removal and Installation

- a. Remove side cover (Section 4.2).
- b. Swing cartridge guide open (Section 4.3).
- c. Manually rotate stepper shaft and move carriage all the way in.
- d. Remove 2 screws holding bracket to base casting and remove bracket and detector.
- e. Remove PCB connector and remove PCB.
- f. Extract cable from P2 connector; Brown, A (R1); Black, C (R3); Red, F (R6); and Orange K (R9).
- g. Remove cable clamps and remove Detector assembly.
- h. To install, reverse the procedure.
- i. Adjust according to Section 4.11.8.

#### 4.11.7 Track 00 Stop Adjustment

- a. Remove side cover (Section 4.2).
- b. Step carriage to track 00. Verify that carriage is at 00 by checking P1 pin 42 is minus (ground).
- c. Check that stop is  $.040'' \pm .020''$  between collar and carriage. Turn DC power OFF, and manually rotate lead screw clockwise until carriage stops. Check that stop is  $.020'' \pm .010''$  between collar and carriage.

- j. Turn DC Power OFF and back ON. Carriage should move to track 00. Verify that there is data at track 00.
- k. Step carriage between track 00 and 76 and check for any binding or interference between the carriage, lead screw, stop and head cable.

#### 4.11.8 Track 00 Flag Adjustment

- a. Remove side cover (Section 4.2).
- b. Check head radial alignment and adjust if necessary before making this adjustment.
- c. Connect oscilloscope probe to TP 26. Set vertical deflection to 1 v/division and sweep to continuous.
- d. Step carriage to track 01. TP 26 should be high (+5 volts).
- e. If TP 26 is not high, loosen screw holding Track 00 flag and move flag towards stepper until TP 26 just goes high.
- f. Step carriage to track 2. TP 26 should go low. Adjust flag towards spindle if not low.
- g. Check adjustment by stepping carriage between tracks 00 and 02, observing that TP 26 is low at track 02 and high at tracks 01 and 00.
- h. Replace side cover.

#### 4.11.9 R/W Head Azimuth Alignment

This adjustment can only be made on SA800/801's at MLC 3 or higher with a new style stepper plate which has 50112-4 stamped on it. This adjustment is only necessary when the stepper or carriage assembly has been replaced or if the stepper plate has been loosened.

- If stepper plate has been loosened or replaced adjust head penetration, Section 4.11.5.
- Align R/W head, 4.11.3.
- Install C.E. alignment diskette SA 120-1. Select the drive and step to track 76.
- Sync the scope external negative on TP 12, set time base to .5 MSec per DIV.
- Connect one probe to TP 1 and the other to TP 2. Invert one channel and ground the probes to TP 5 & 6. Set the inputs to AC, ADD and 50 MV per division.
- Compare the wave form to Figure 13. If not within the range shown the head Azimuth will require adjustment. If required, proceed to next step.
- Slightly loosen the 2 R.H. stepper plate mounting screws only. Reference Figure 12. Do not loosen the L.H. screw as this will effect the head penetration adjustment.

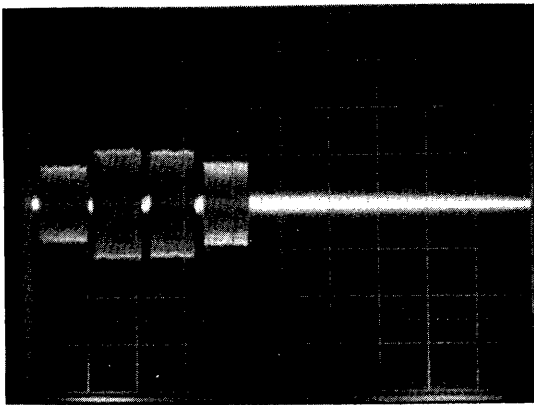


FIGURE 13

- Push the stepper down towards the A.C. drive motor until the 1st sector is larger than the 2nd sector.
- Pry the R.H. side of the stepper plate up with a medium screw driver until the 1st and 4th sectors have equal to or less amplitude than the middle 2 sectors. Reference Figure 13.

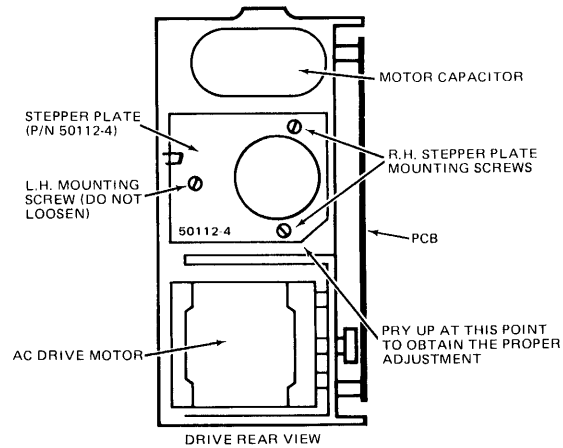


FIGURE 12

- Re-tighten the 2 R.H. screws. If either of the outside 2 sectors increase in amplitude greater than the inside 2 after re-tightening the screws, perform the adjustment again.
- Check and re-adjust the index timing and head radial adjustment if required.

#### 4.11.10 Stepper Plate Removal and Adjustment

- Remove PCB.
- Remove head and carriage assembly from stepper lead screw, section 4.11.1.
- Pull the stepper motor out thru the stepper plate until the lead screw is completely clear of the plate.
- Remove the 3 stepper plate mounting screws.
- Reinstall the stepper plate.

NOTE: If the stepper plate is P/N 50112-4, there must be a nylon bushing in the L.H. hole and all 3 screws must have a flat washer and a black spring washer.

- Reinstall head and carriage and stepper motor assemblies.
- Adjust penetration, Section 4.11.5. If the stepper plate is P/N 50112-4, there will remain a gap between the bottom of the stepper plate and the machined surface on the casting. All other style stepper plates must remain flush with machined surface.
- Readjust carriage assembly, Section 4.11.2.
- Check and adjust Azimuth alignment, Section 4.11.10.

#### 4.12 Activity Light Removal and Installation (Standard)

- Remove P6 connector from PCB.
- Remove cable clamp holding the cable and remove cable from clamp.
- Remove the 2 screws holding the push button.
- Remove push button and activity light from the front as an entire assembly.
- Install the light and push button assembly by reversing the removal procedure.
- No special orientation is required when installing P6 onto the PCB. No adjustments are required to the push button assembly.

- Remove front plate (Section 4.10).
- Remove two screws holding assembly to front plate.
- Remove two allen head screws holding assembly to push button.
- Grasp both ends of push button and bow outwards to remove LED.
- Reverse procedure to assemble.
- Adjustment of the door lock should not be necessary. If it has to be, the gap between the armature tab and the latch should be  $.015 \pm .010$ . This adjustment can be made by loosening the two screws on the armature.

#### 4.13 Door Lock

- Disconnect P6 connector.
- Disconnect red wire near IC 2G.

#### 4.14 Activity Light (with Door Lock Option)

- Follow procedure for door lock (4.13).

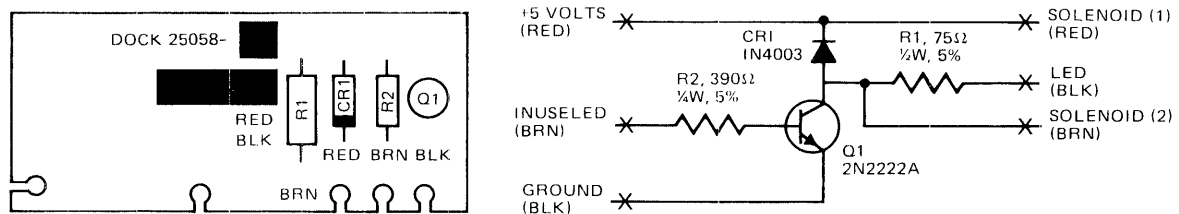
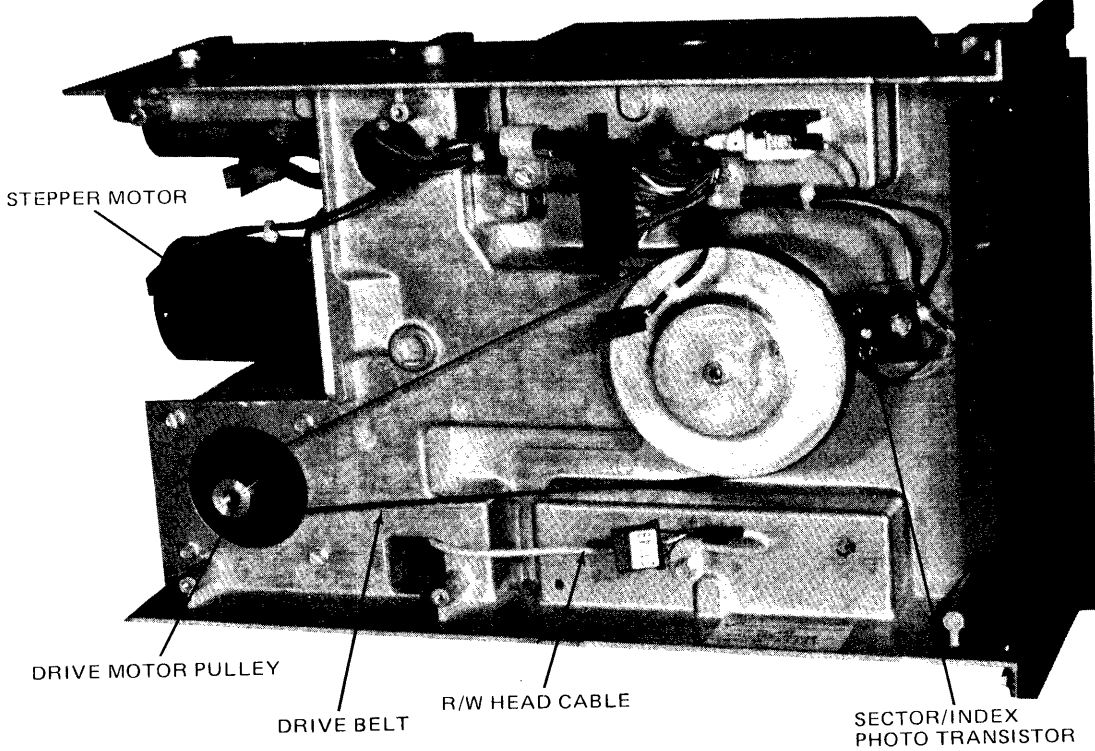
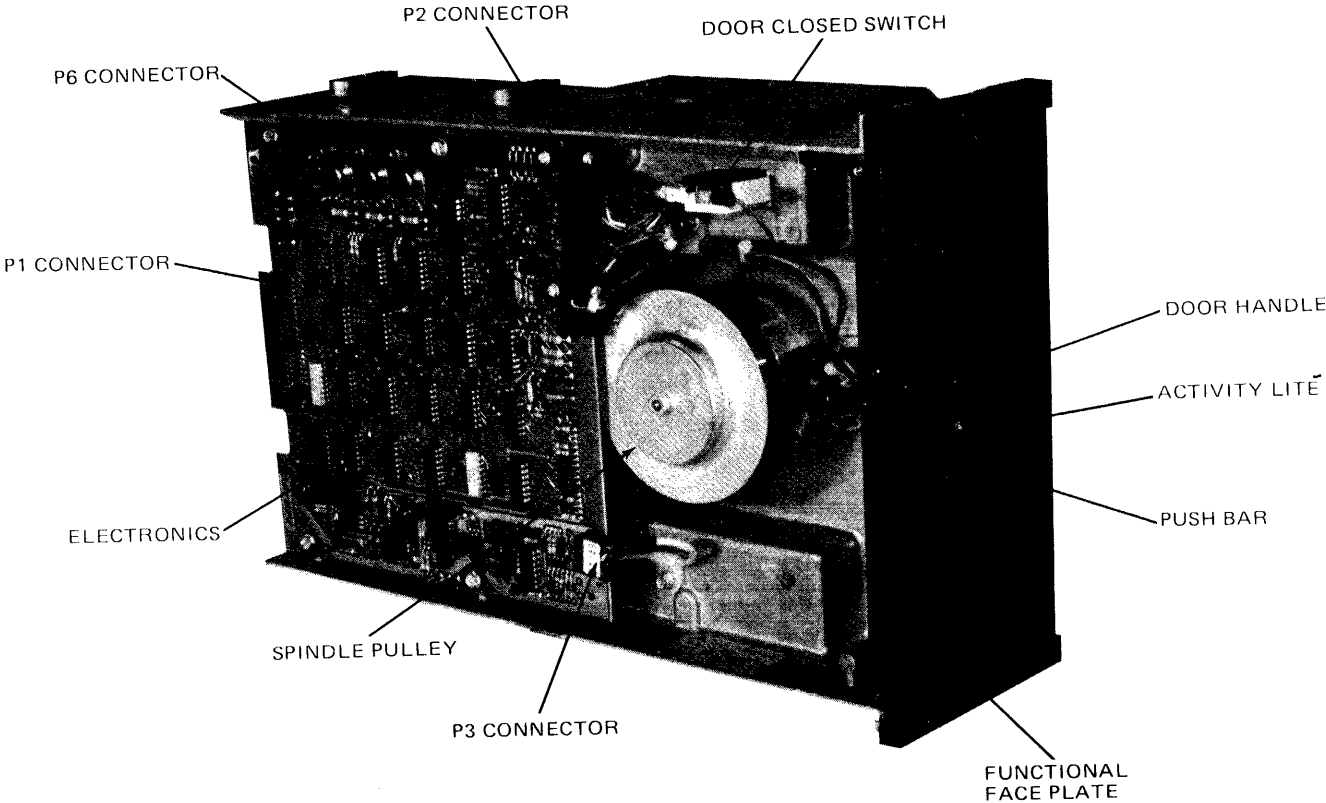
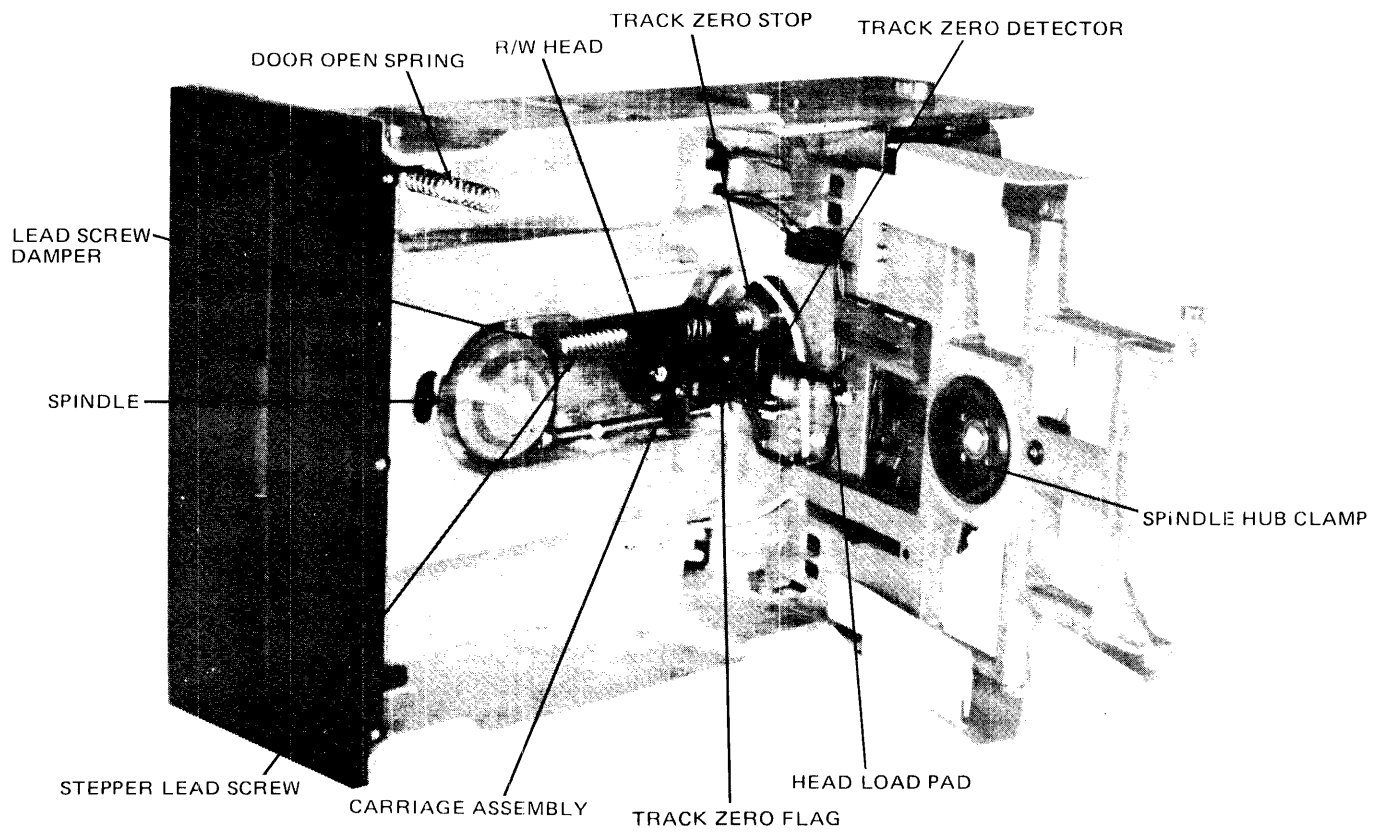
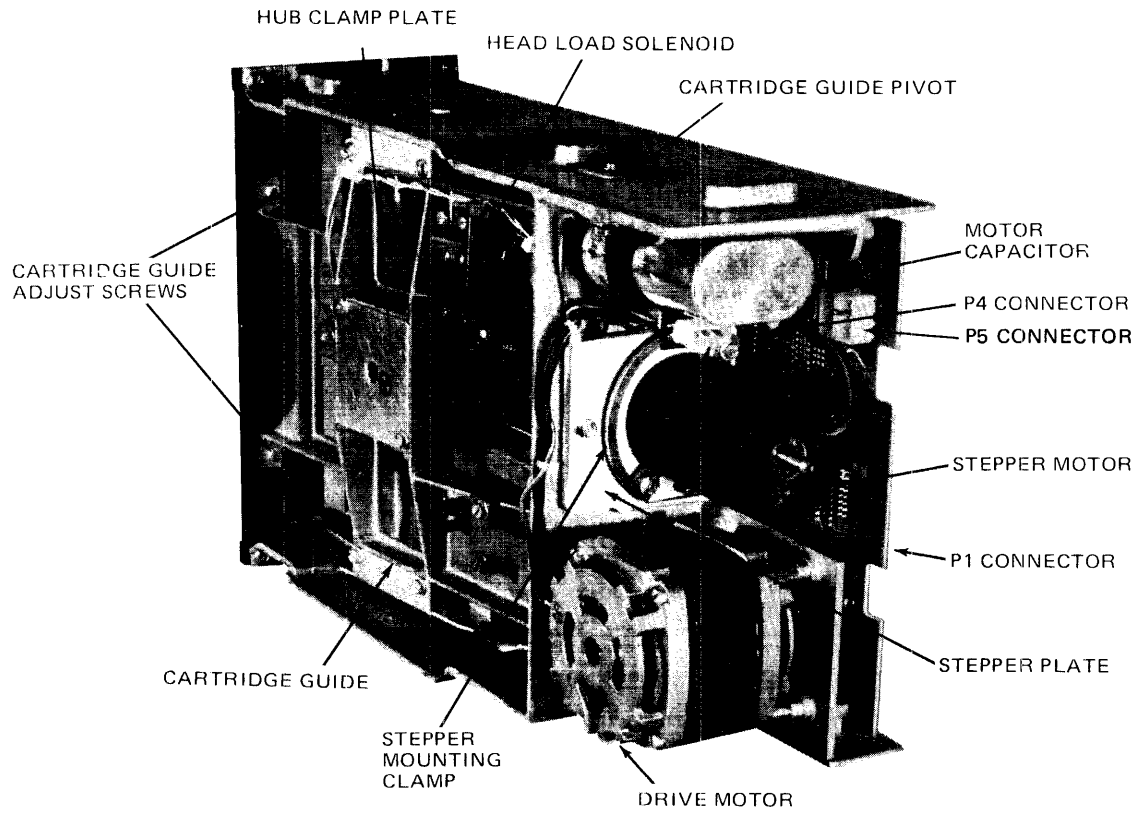


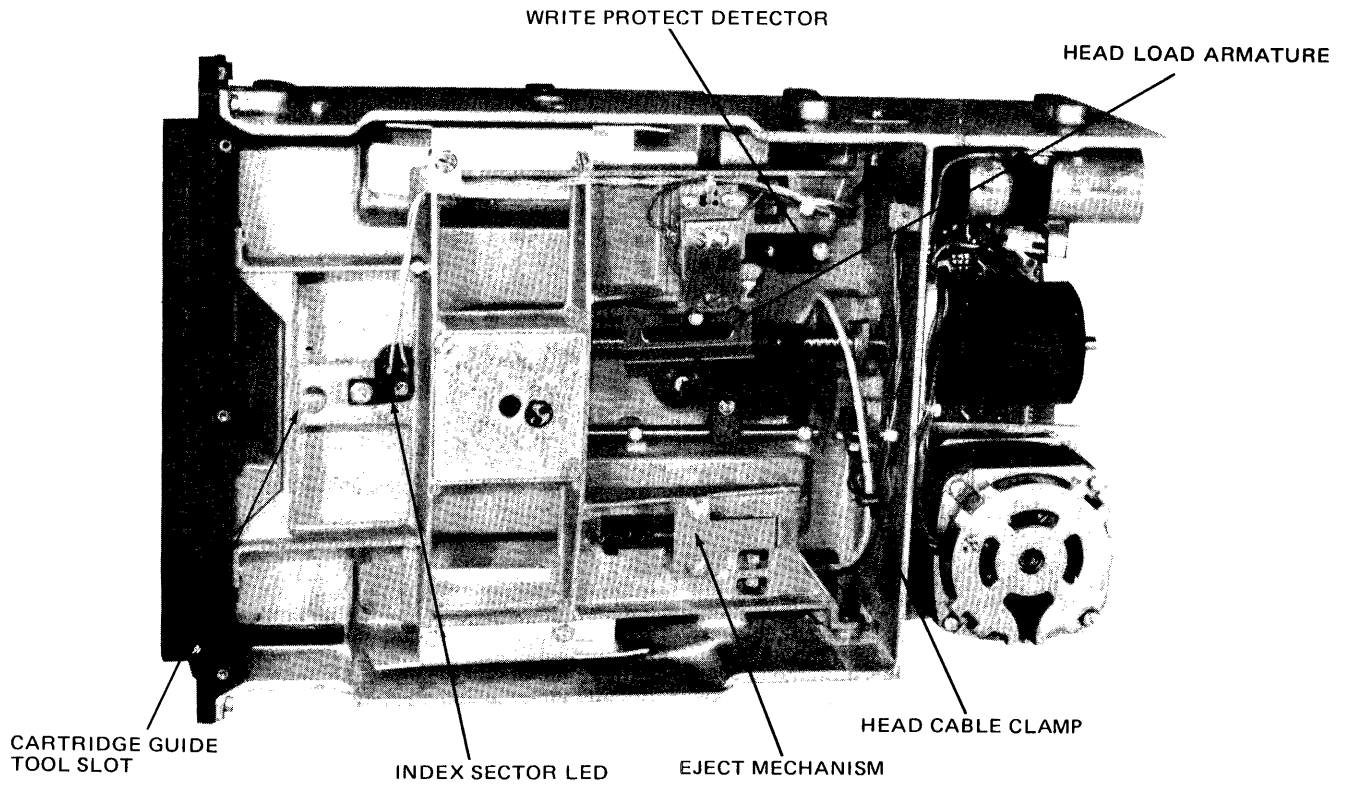
FIGURE 14 DOOR LOCK SCHEMATIC



5 PHYSICAL LOCATIONS

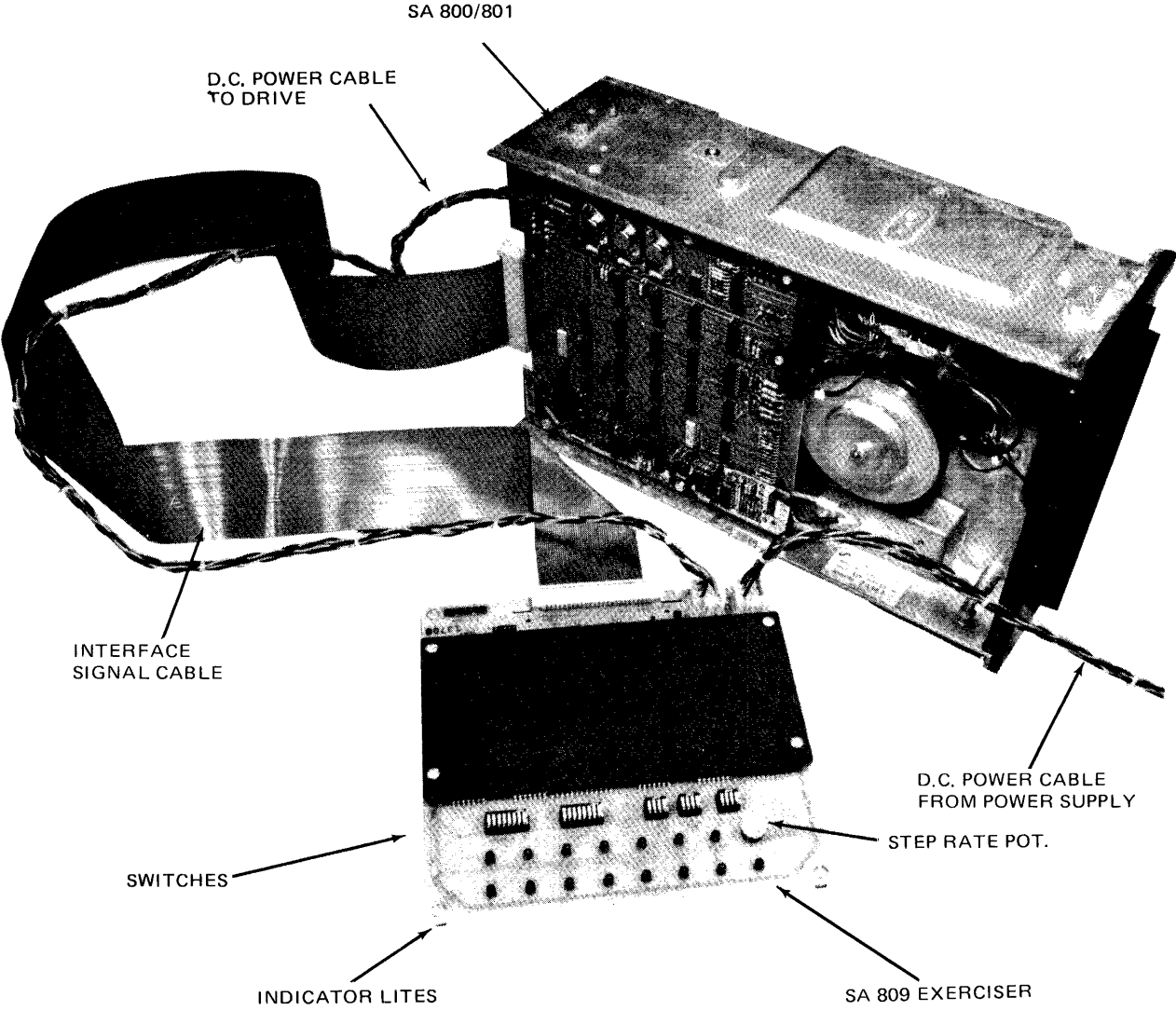


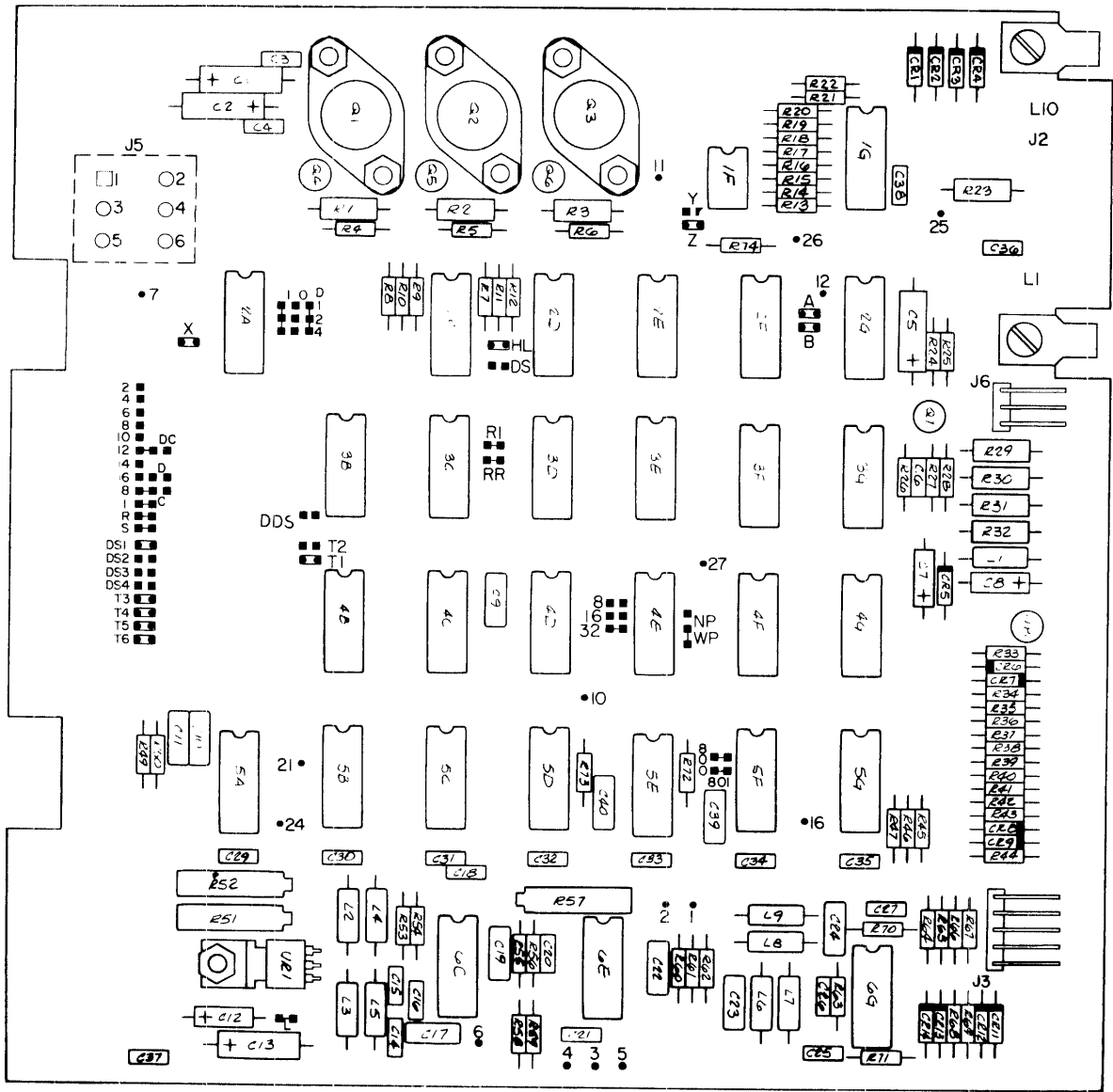






6 SA809 EXERCISER CONNECTION





SA800/801 PCB Component Location







LOGIC MANUAL DRIVE SN \_\_\_\_\_

INDEX

AA001 INDEX  
 ABO10 PCB TRACE AND TEST POINT LOCATIONS  
 DAO10 READ/WRITE/INDEX/SECTOR  
 DAO20 STEPPER CONTROL  
 DAO30 DETECTORS  
 DAO40 MOTORS SOLENOID SWITCH

TABLE III  
 CUSTOMER CUT TRACE OPTIONS

DESIGNATOR	DESCRIPTION	SHIPPED FROM FACTORY
		OPEN SHORT
T3,T4,T5 & T6	TERMINATIONS FOR MULTIPLEXED INPUTS	<input type="checkbox"/> X
T2	SPARE TERMINATOR FOR RADIAL HEAD LOAD	X
T1	TERMINATION FOR DRIVE SELECT	<input type="checkbox"/> X
DS1	DRIVE SELECT INPUT - ALTERNATE PINS: DS2, DS3, DS4	<input type="checkbox"/> X
RR	RADIAL READY	X
RI	RADIAL INDEX AND SECTOR	X
R,I,S	READY, INDEX, SECTOR ALTERNATE OUTPUT PROVISION	X
A,B,X	RADIAL HEAD LOAD	<input type="checkbox"/> X
HL	STEPPER POWER FROM HD LD	<input type="checkbox"/> X
DS	STEPPER POWER FROM DRIVE SELECT	X
WP	INHIBIT WRITE WHEN WRITE PROTECTED	X
NP	ALLOW WRITE WHEN WRITE PROTECTED	X
8,16,32	8,16 OR 32 SECTORS	X
D	ALTERNATE INPUT - IN USE	X
2,4,6,8,10,12,14,16,18	NINE ALTERNATE I/O PINS	X
D1, D2, D4, DDS	CUSTOMER INSTALLABLE DECODE DRIVE SELECT OPTION	X
C	ALTERNATE INPUT - HEAD LOAD	X
Z	INUSE LED FROM DRIVE SELECT	<input type="checkbox"/> X
Y	INUSE LED FROM HD LD	X
DC	ALTERNATE OUTPUT - DISK CHANGE	X

FACTORY CUT TRACE OPTIONS AND HISTORY CHART

L	-5 OR -7 TO -16V	SEE TABLE II
800	INDEX ONLY (800)	SEE TABLE II
801	INDEX AND SECTOR(801)	SEE TABLE II

OPEN TRACE WITH SHORTING PLUG INSTALLED.

TABLE I  
 OPTIONAL FEATURES

PCB ASM NO.	OPTIONAL FEATURES	
	-5V -7 TO -16V	DATA SEP SECTOR
25102	X	
25103		X
25104	X	X
25105		X
25106	X	X
25107	X	X

WRITE PROTECT CAN BE ORDERED WITH ANY OF THE ABOVE

TABLE II  
 FACTORY CUT TRACE OPTIONS

PCB ASM NO	TRACE "L"	TRACE "801"	TRACE "800"
25102	SHORTED	OPEN	SHORTED
25103	OPEN	▲	▲
25104	SHORTED	▲	▼
25105	OPEN	OPEN	SHORTED
25106	SHORTED	SHORTED	OPEN
25107	OPEN	SHORTED	OPEN

LOGIC MANUAL P/N 50590 - 1

PART NO  
 50591-0

MUST CONFORM TO ENGINEERING SPEC ES 3000A-0		EC HISTORY		SHUGART ASSOCIATES	
MATERIAL	TOLERANCE UNLESS OTHERWISE NOTED	DATE	NO	TITLE	INDEX
		6-75	0398		
		7-76	0509	DETAIL	RELEASED FOR ASSEMBLY
				DESIGN	50590
				APPRO	SHEET 1 OF 1
				SCALE	DOC CODE
				SCALE	00
				REV EC	5059-0
				REV EC	0509

AA001

D

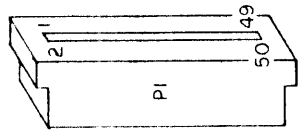
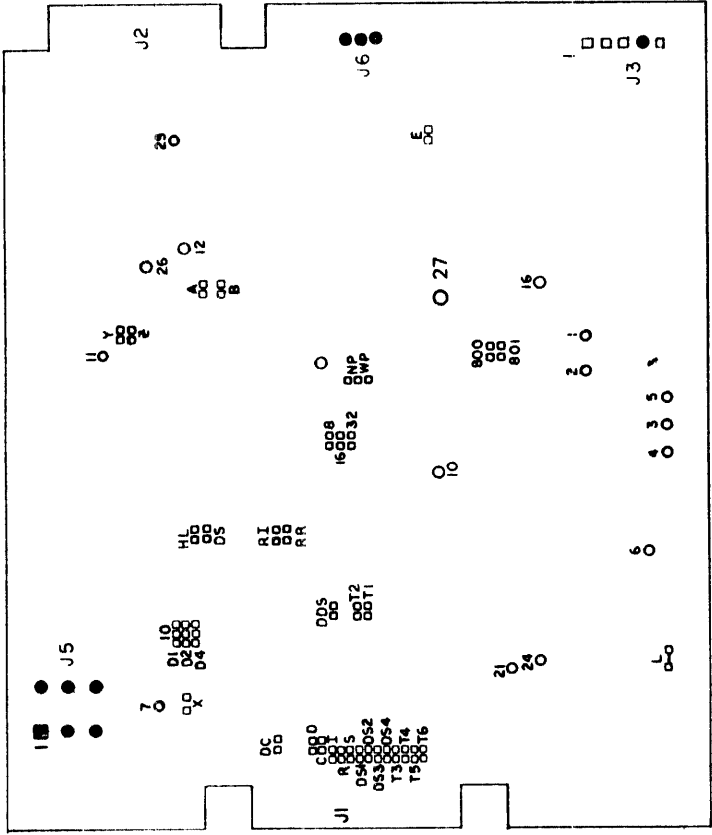
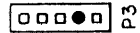
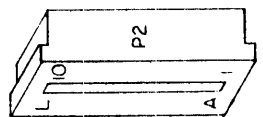
C

B

A

TEST POINTS

TP1, TP2	FILTERED READ SIGNAL
TP3, TP4	DIFFERENTIATED READ SIGNAL
TP5, TP6, TP7	SIGNAL GROUND
TP10	- INDEX (TTL)
TP11	- LOAD HEAD (TTL)
TP12	- DETECTED INDEX / SECTOR HOLES (TTL)
TP16	+ READ DATA (TTL)
TP21	DATA SEPARATOR TIME t1 (TTL)
TP24	DATA SEPARATOR TIME t2 (TTL)
TP25	+ WRITE PROTECT
TP26	+ DETECTED TRACK ZERO (TTL)
TP27	+ GATED STEP PULSE (TTL)



MUST CONFORM TO ENGINEERING SPEC. ES 30000-0		EC HISTORY	SHUGART ASSOCIATES	
MATERIAL	TOLERANCE UNLESS OTHERWISE NOTED	DATE	PCB TRACE AND TEST POINT LOCATION	
	LINEAR ± .XX	6-75	NO.	RELEASED FOR ASSEMBLY
CASE DEPTH	ANGULAR: CORNERS BROKEN	7-76	0509	DESIGN 6432
HARDNESS	ES SURFACE TREATMENT			APPRO 447
	OUTSIDE MAX			SHEET 1 OF 1
	INSIDE MAX			SCALE 1:1
				POC CODE 00
				PART NO 50592-0
				REV'S 0509

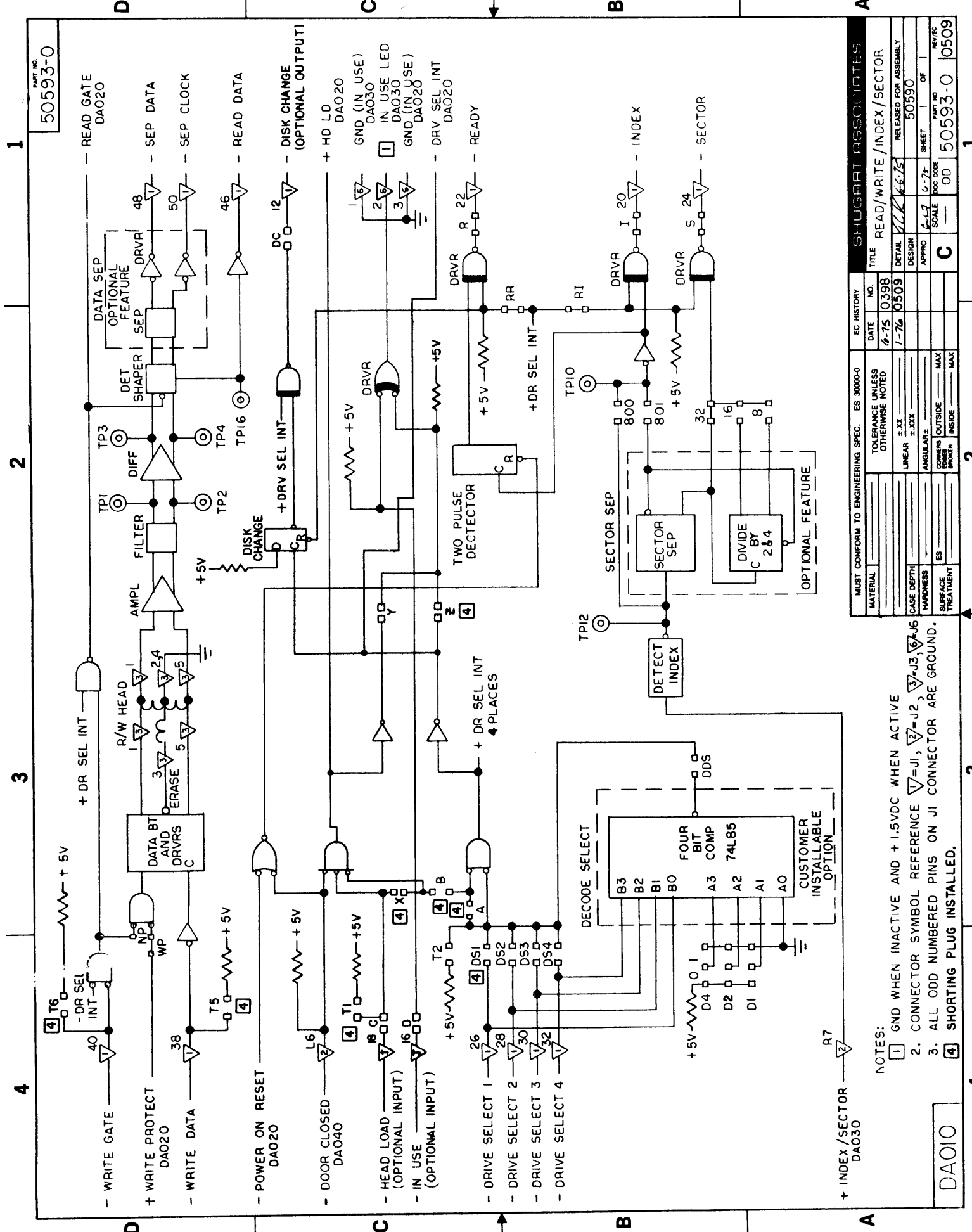
NOTES:  
1. □/□/□□ DENOTES CUT TRACE/JUMPER OPTION.  
2. ○ DENOTES TEST POINTS.

4

3

2

1



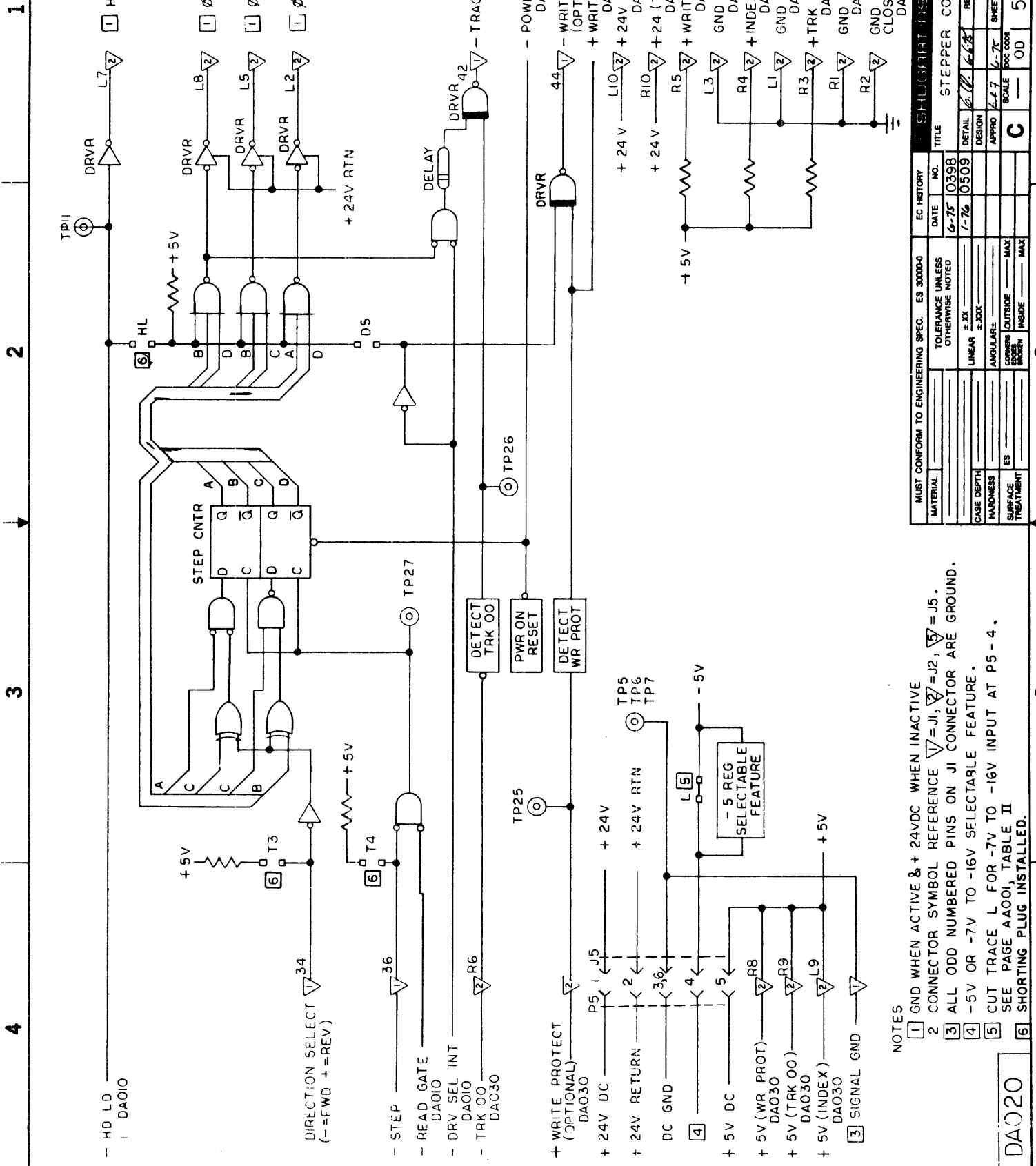
PART NO  
50593-0

MUST CONFORM TO ENGINEERING SPEC. ES 30000-0		EC HISTORY	
MATERIAL	TOLERANCE UNLESS OTHERWISE NOTED	DATE	NO.
CASE DEPTH	LINEAR ± .XX	4-75	0398
HARDNESS	ANGULAR ± .XXX	7-76	0509
SURFACE TREATMENT	CONFORMS TO MIL-C-13000	DESIGN	4-6-75
	BROKEN INSIDE	APPRO	6-6-77
	MAX	SCALE	1:1
	MAX	DOC CODE	50590
		SHEET	OF
		50593-0	1
		REV/EC	0509

NOTES:  
 1. GND WHEN INACTIVE AND +1.5VDC WHEN ACTIVE  
 2. CONNECTOR SYMBOL REFERENCE ▽=J1, ▽=J2, ▽=J3, ▽=J6  
 3. ALL ODD NUMBERED PINS ON J1 CONNECTOR ARE GROUND.  
 4. SHORTING PLUG INSTALLED.

DAO10

SHUGART ASSOCIATES  
 TITLE READ/WRITE / INDEX / SECTOR  
 DETAIL 4-6-75  
 DESIGN 4-6-75  
 APPRO 6-6-77  
 C

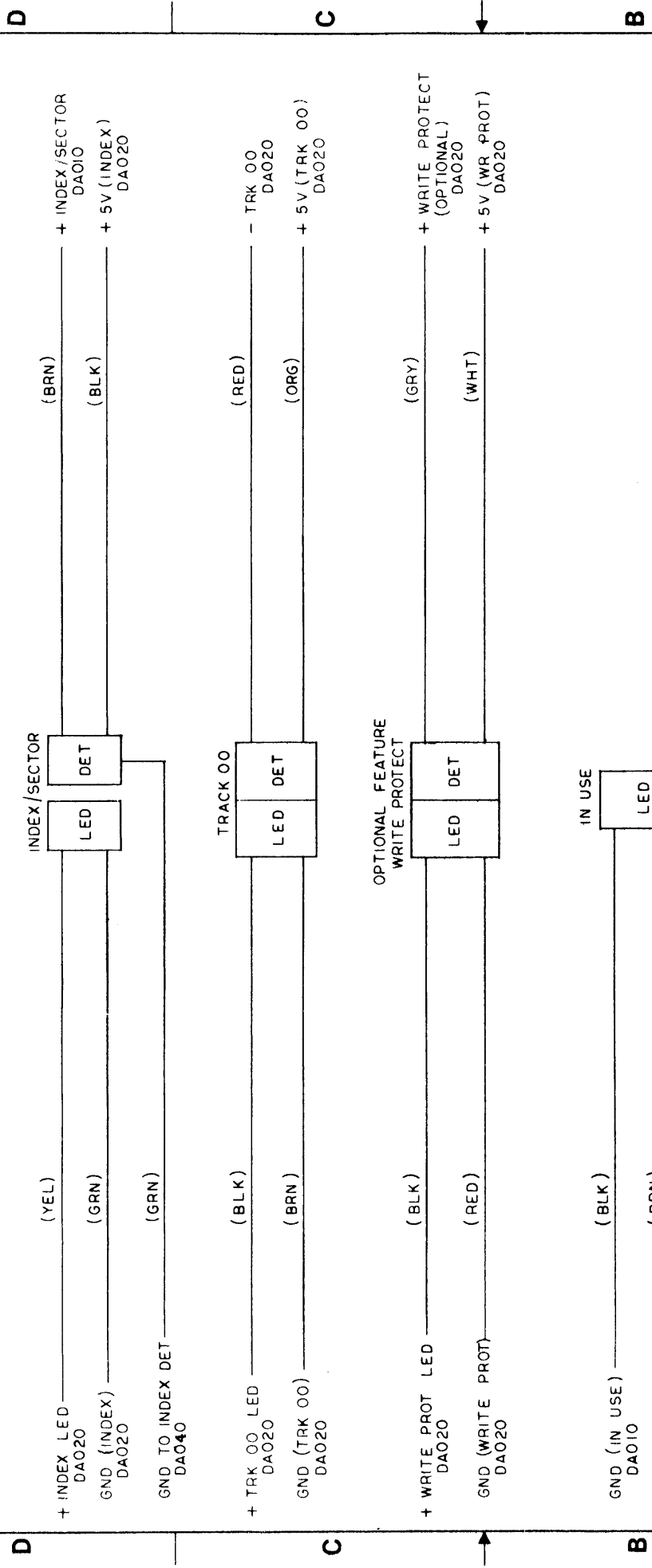


- NOTES
- 1 GND WHEN ACTIVE & + 24VDC WHEN INACTIVE
  - 2 CONNECTOR SYMBOL REFERENCE ▽=J1, ▽=J2, ▽=J5.
  - 3 ALL ODD NUMBERED PINS ON J1 CONNECTOR ARE GROUND.
  - 4 -5V OR -7V TO -16V SELECTABLE FEATURE.
  - 5 CUT TRACE L FOR -7V TO -16V INPUT AT P5 - 4.
  - 6 SEE PAGE AAO01, TABLE II SHORTING PLUG INSTALLED.

MUST CONFORM TO ENGINEERING SPEC. ES 30000-0		EC HISTORY		SHOULDN'T ASSUMES	
MATERIAL	TOLERANCE UNLESS OTHERWISE NOTED	DATE	NO.	TITLE	STEPPER CONTROL
		6-75	0398	DETAIL	6-75
CASE DEPTH	LINEAR ±.003	7-76	0509	DESIGN	6-75
HARDNESS	ANGULAR ±.003			APPROD	6-75
SURFACE TREATMENT	CONFORM TO OUTSIDE MAX BROKEN INSIDE MAX			SCALE	1/8" = 1"
ES				DOC CODE	6-75
				SHEET	OF 1
				PART NO.	50594-0
				REV/EC	0509

PART NO  
50595-0

4 3 2 1



NOTES:  
 GND WHEN INACTIVE AND +1.5VDC WHEN ACTIVE

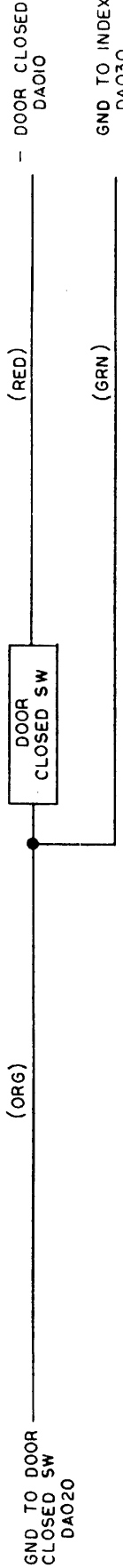
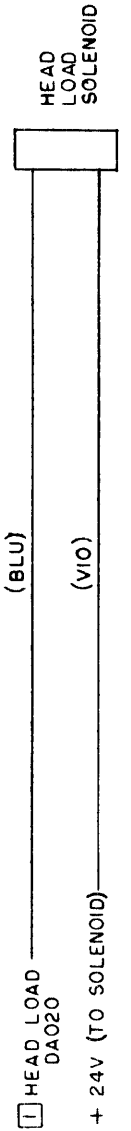
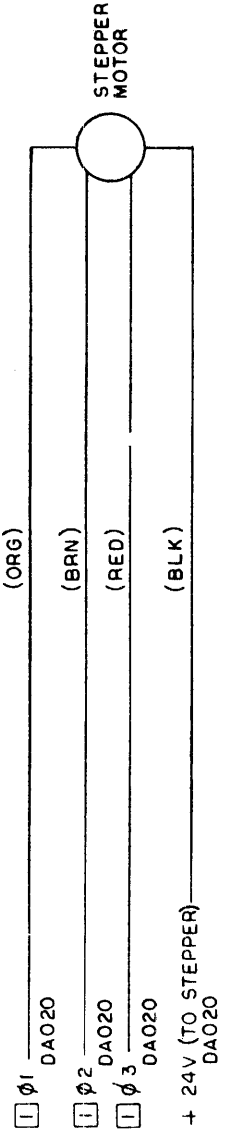
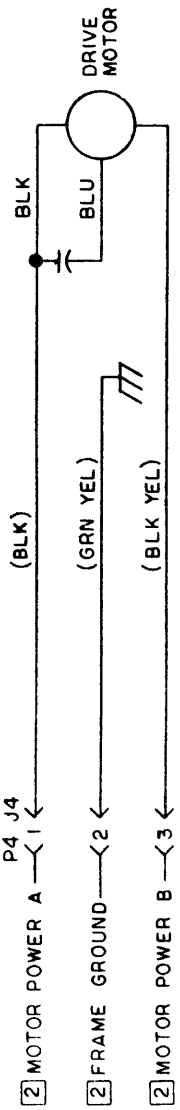
MATERIAL		MUST CONFORM TO ENGINEERING SPEC. ES 30000-0		EC HISTORY		TITLE	
		TOLERANCE UNLESS OTHERWISE NOTED		DATE		NO.	
		LINEAR ±.XX		6-75		0398	
		ANGULAR ±.XXX					
CASE DEPTH		CORNERS		SCALE		SHEET	
HARDNESS		BROKEN		MAX		OF	
SURFACE TREATMENT		INSIDE		MAX		SHEET	
						50590	
						6.61	
						RELEASED FOR ASSEMBLY	
						50590	
						00	
						0398	
						50595-0	
						0398	

DAO30

A

A

PART NO  
50596-0



GND TO INDEX DET  
DAO30

NOTES:  
 [1] GND WHEN ACTIVE & +24V WHEN INACTIVE,  
 [2] 115 OR 230 VAC

MUST CONFORM TO ENGINEERING SPEC. ES 30000-0		EC HISTORY		TITLE		
MATERIAL	TOLERANCE UNLESS OTHERWISE NOTED	DATE	NO.	MOTORS/SOLENOID/SWITCH		
		6-75	0398	DETAIL	2/68	RELEASED FOR ASSEMBLY
	LINEAR ±.001	7-75	0509	DESIGN		
CAGE DEPTH				APPRO	6-75	SHEET 1 OF 1
HARDNESS	ANGULAR: _____			SCALE	6-75	
SURFACE TREATMENT	CONFORM _____					
	BRUSH _____					
	OUTSIDE _____					
	INSIDE _____					

DAO40

4 3 2 1

D D

C C

B B

A A

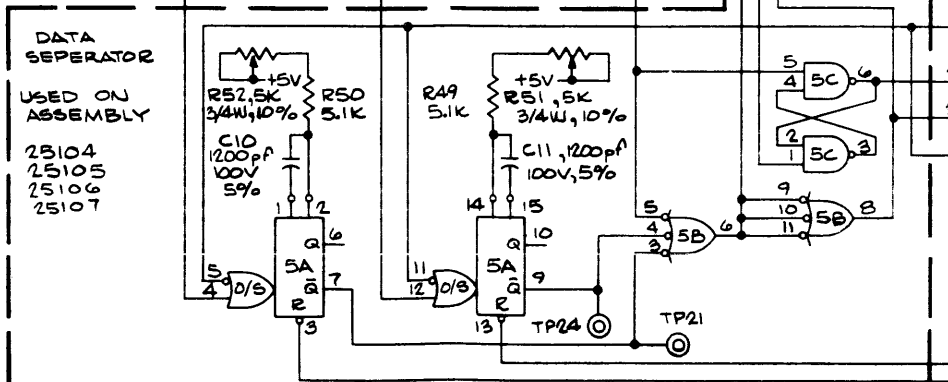
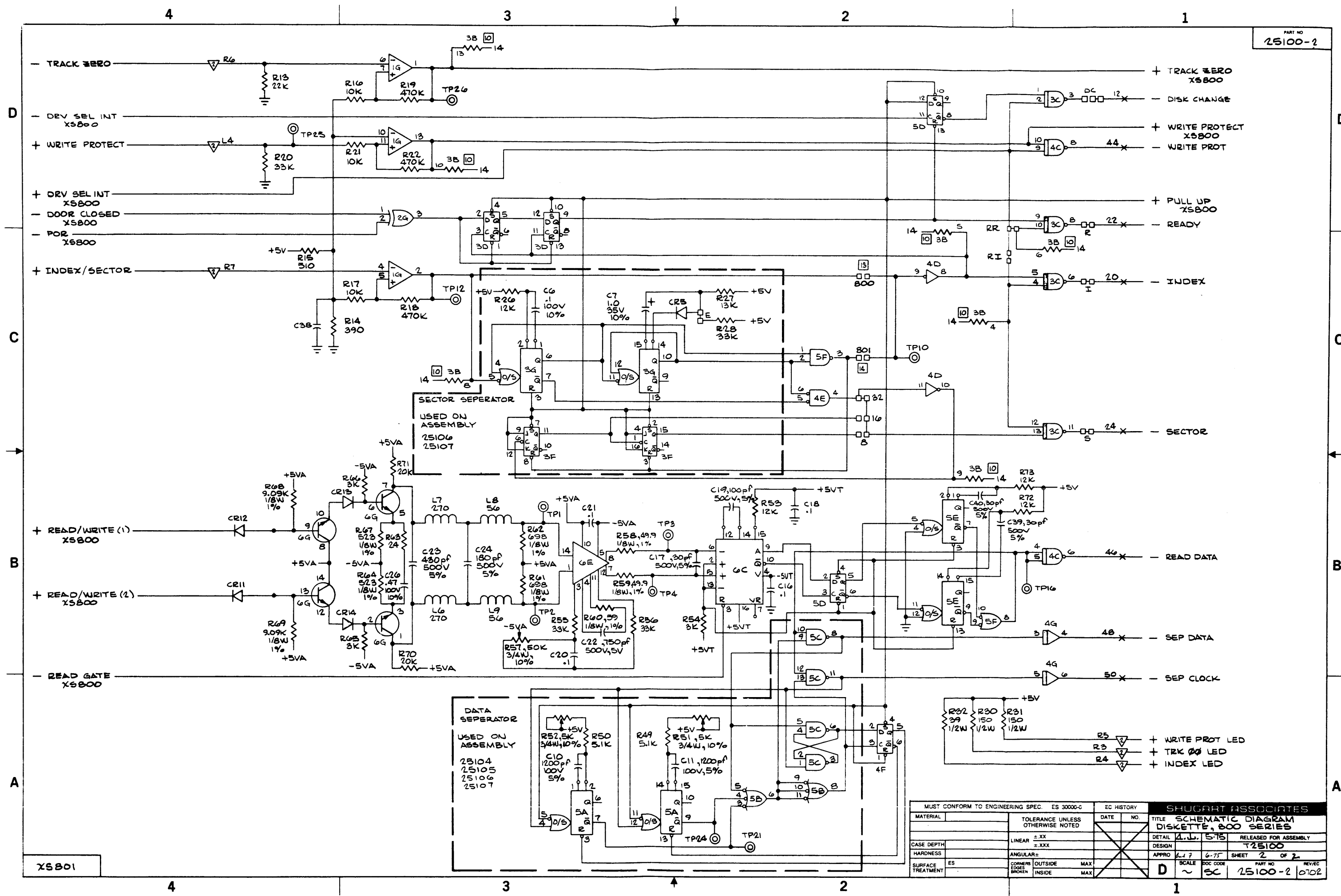
4 3 2 1

**SA800/801 SCHEMATIC DIAGRAMS**



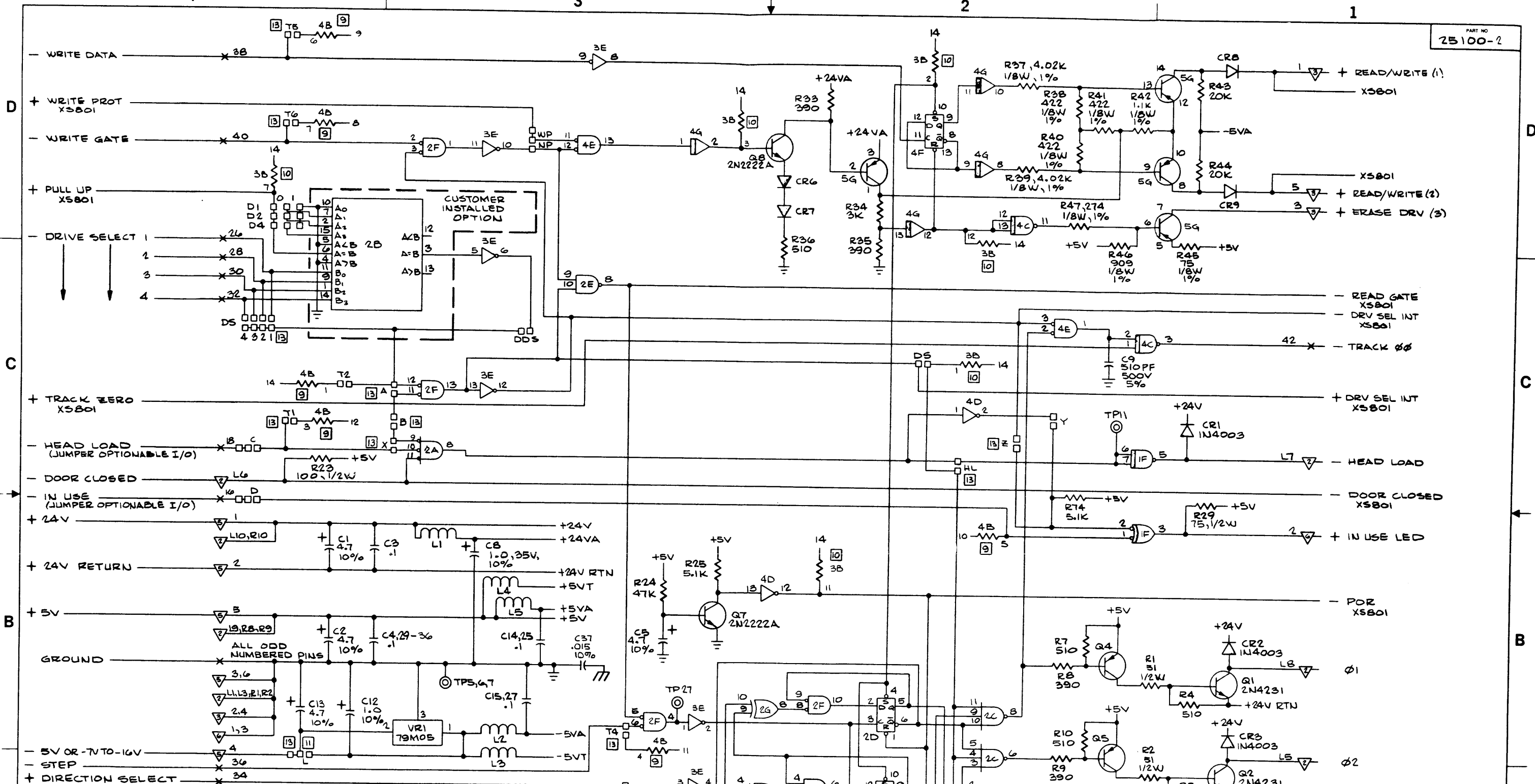
435 Oakmead Parkway, Sunnyvale, California 94086  
Phone: (408) 733-0100 TWX: 910 339 9355 SHUGART SUVL





MUST CONFORM TO ENGINEERING SPEC. ES 30000-0		EC HISTORY		SHUGART ASSOCIATES	
MATERIAL	TOLERANCE UNLESS OTHERWISE NOTED	DATE	NO.	TITLE SCHEMATIC DIAGRAM DISKETTE, 800 SERIES	
	LINEAR ±.XX			DETAIL	4.11.575
CASE DEPTH	ANGULAR ±.XXX			DESIGN	T25100
HARDNESS				APPRO	6.75
SURFACE TREATMENT	CORNERS OUTSIDE MAX			SCALE	6-75
	EDGES OUTSIDE MAX			DOC CODE	25100-2
	INSIDE MAX			PART NO	25100-2
				REV/EC	0702

X5801



NOTES, UNLESS OTHERWISE SPECIFIED:

1. ALL RESISTORS ARE IN OHMS, 1/4W, 5%.
2. ALL CAPACITORS ARE IN MICROFARADS, 50V, ±80-20%.
3. ALL INDUCTORS ARE 100MH, 10%.
4. ALL DIODES ARE 1N4148.
5. ALL TRANSISTORS ARE 2N2907A.
6. D-D INDICATES CUT-TRACE OPTION.
7. D-D INDICATES JUMPER OPTION.
8. CONNECTOR SYMBOL REFERENCES; (X)=J1, (∇)=J2, (∩)=J3, (∪)=J5, (∪)=J6.
9. I.C. LOCATION 4B IS A RESISTOR PACK, 150Ω, 1/2W, 5%.
10. I.C. LOCATION 3B IS A RESISTOR PACK, 2K, 1/4W, 5%.
11. ADD JUMPER L FOR -7V TO 16V ON ASSEMBLIES; 25103, 25105, 25107.
12. PINS 8 AND 9 OF I/O ARE GROUND.
13. SHORTING PLUG INSTALLED.
14. REMOVE JUMPER FROM 800 AND INSTALL IN 801.

REFERENCE DESIGNATION	LAST USED	NOT USED
C40		C28 (RESERVED)
CR14		CR10
L9		
Q8		
R74		R48
TP27		TP8, 9, 10, 15, 17, 20, 22, 25
VR1		

TYPE	POSITION	UNUSED ELEMENTS	VCC PIN	GND PIN	TYPE	POSITION	UNUSED ELEMENTS	VCC PIN	GND PIN
7400	5C, 2E, 5F	2E1, 4, 5F2, 4	14	7	74L85	2B		16	8
7402	4E	4E3			75452P	1F		8	4
7404	3E				8T20	6C		~	8
7405	4D	4D2, 3			9602	5A, 5E, 3G		16	8
7407	4G				LM339	1G	1G3	3	12
7410	5B, 2C	5B1			NES92A	6E		~	~
7438	3C, 4C				2GT2222	6G		~	~
7474	2D, 3D, 5D, 4F				2GT2905	5G		~	~
7476	3F		5	13	7428	2F		14	7
7486	2G	2G4	14	7	7427	2A	2A1, 2	14	7

MATERIAL		TOLERANCE UNLESS OTHERWISE NOTED		DATE HISTORY		SHUGART ASSOCIATES	
CASE DEPTH	±.XX	LINEAR	±.XXX	DATE	NO.	TITLE	SCHEMATIC DIAGRAM
HARDNESS	ANGULAR:	ANGULAR:		10-76	0591	DISKETTE, 800 SERIES	
SURFACE TREATMENT	ES	CORNERS BROKEN	OUTSIDE MAX	11-76	0702	RELEASED FOR ASSEMBLY	
		INSIDE MAX					
SCALE		DOC CODE		PART NO		REV/EC	
D		SC		25100-2		0102	

XSB00

USED ON ASSEMBLY
25102-4
25103-4
25104-4
25105-4
25106-4
25107-4

**APPENDIX D**  
**STANDARD JUMPER POSITIONS**  
**FOR DRIVE ELECTRONICS**

SHUGART DRIVE JUMPER CONFIGURATION FOR DSD 440  
AS SHIPPED FROM SHUGART

TRACE DESIGNATOR	DESCRIPTION	SHIPPED FROM SHUGART	
		OPEN	SHORT
T3,T4,T5,T6	Terminations for Multiplexed Inputs		Plugged
T1	Terminator for Drive Select		Plugged
T2	Spare Terminator for Radial Head Load	X	
DS1,DS2,DS3,DS4	Drive Select Input Pins	X	DS1 is Plugged
RR	Radial Ready		X
RI	Radial Index and Sector		X
R,I,S,	Ready, Index, Sector Alternate Output Pads		X
HL	Stepper Power From Head Load		Plugged
DS	Stepper Power From Drive Select	X	
WP	Inhibit Write When Write Protected		X
NP	Allow Write When Write Protected	X	
8,16,32,	8, 16, 32 Sectors (SA801 Only)	8 & 16	32
D	Alternate Input-In Use	X	
2,4,6,8,10,12,14,16,18	Nine Alternate I/O Pins	X	
D1,D2,D4,DDS	Customer Installable Decode Drive Select Option	X	
A,B,X	Radial Head Load		Plugged
C	Alternate Input-Head Load	X	
Z	In Use from Drive Slect		Plugged
Y	In Use from HD LD	X	
DC	Alternate Output-Disk Change	X	

AS SHIPPED IN DSD 440

TRACE DESIGNATOR	DESCRIPTION	SHIPPED FROM DSD	
		OPEN	SHORT
T3,T4,T5,T6	Terminations for Multiplexed Inputs		Note 1
T1	Terminator for Drive Select		Note 1
T2	Spare Terminator for Radial Head Load		X
DS1,DS2,DS3,DS4	Drive Select Input Pins .	X	Note 2
RR	Radial Ready		X
RI	Radial Index and Sector		X
R,I,S,	Ready, Index, Sector Alternate Output Pads		X
HL	Stepper Power From Head Load	X	
DS	Stepper Power From Drive Select		Plugged
WP	Inhibit Write When Write Protected		X
NP	Allow Write When Write Protected	X	
8,16,32,	8, 16, 32 Sectors (SA801 Only)	8 & 16	32
D	Alternate Input-In Use		Note 3
2,4,6,8,10,12,14,16,18	Nine Alternate I/O Pins		Note 3
D1,D2,D4,DDS	Customer Installable Decode Drive Select Option	X	
A,B,X	Radial Head Load	X	A,B
C	Alternate Input-Head Load		X
Z	In Use from Drive Slect	X	
Y	In Use from HD LD		Plugged
DC	Alternate Output-Disk Change		Plugged

NOTE 1: Last drive on daisy chain should have jumper T1, T3, T4, T5 and T6 installed.

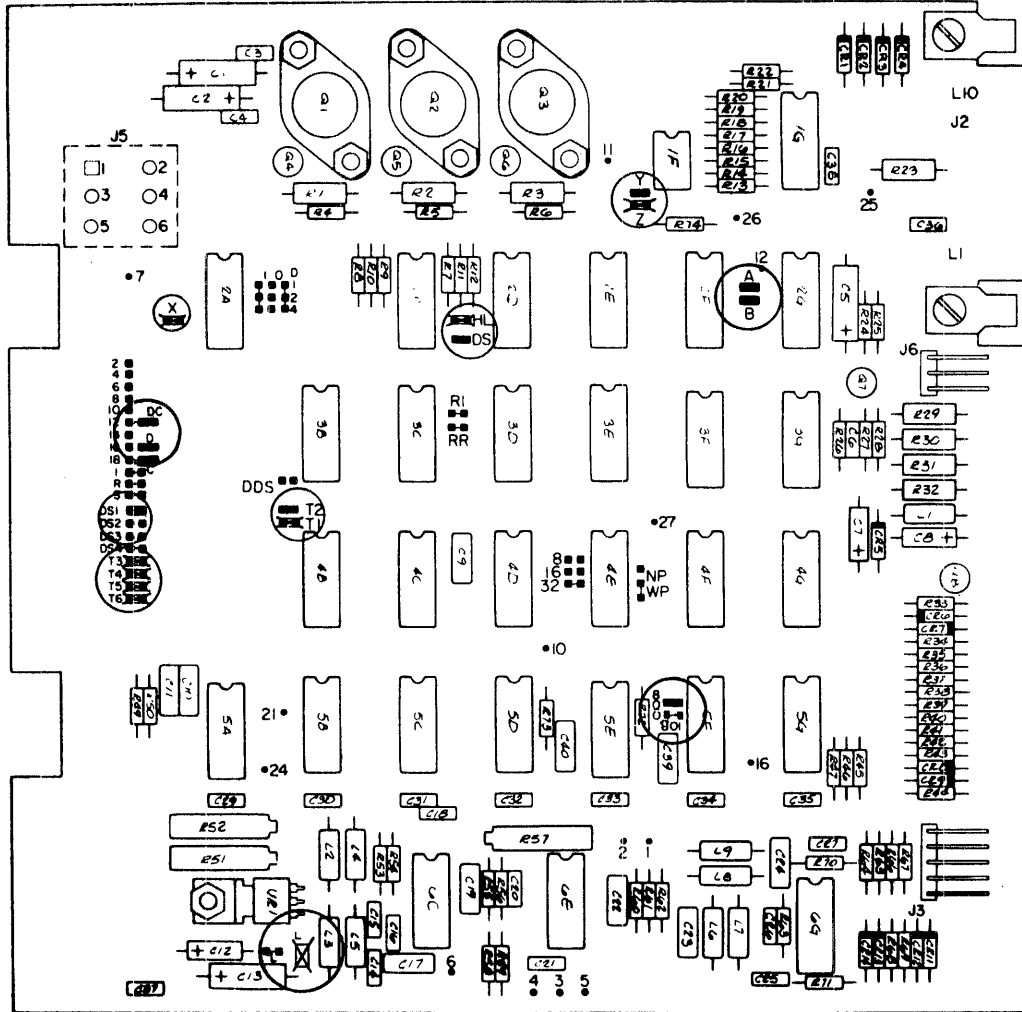
NOTE 2: One jumper installed according to physical drive number.

NOTE 3: Pin D is connected to Pin 16 on physical drive 0 and Pin 8 on physical drive 1.

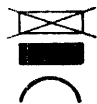
NOTE 4: Jumper L is open and 800 option is shorted.

KEY: X - Specified signal is either open or shorted, depending upon in which column the "X" appears.

Plugged - Specified signal has a pair or wire wrap pins which are shorted together.

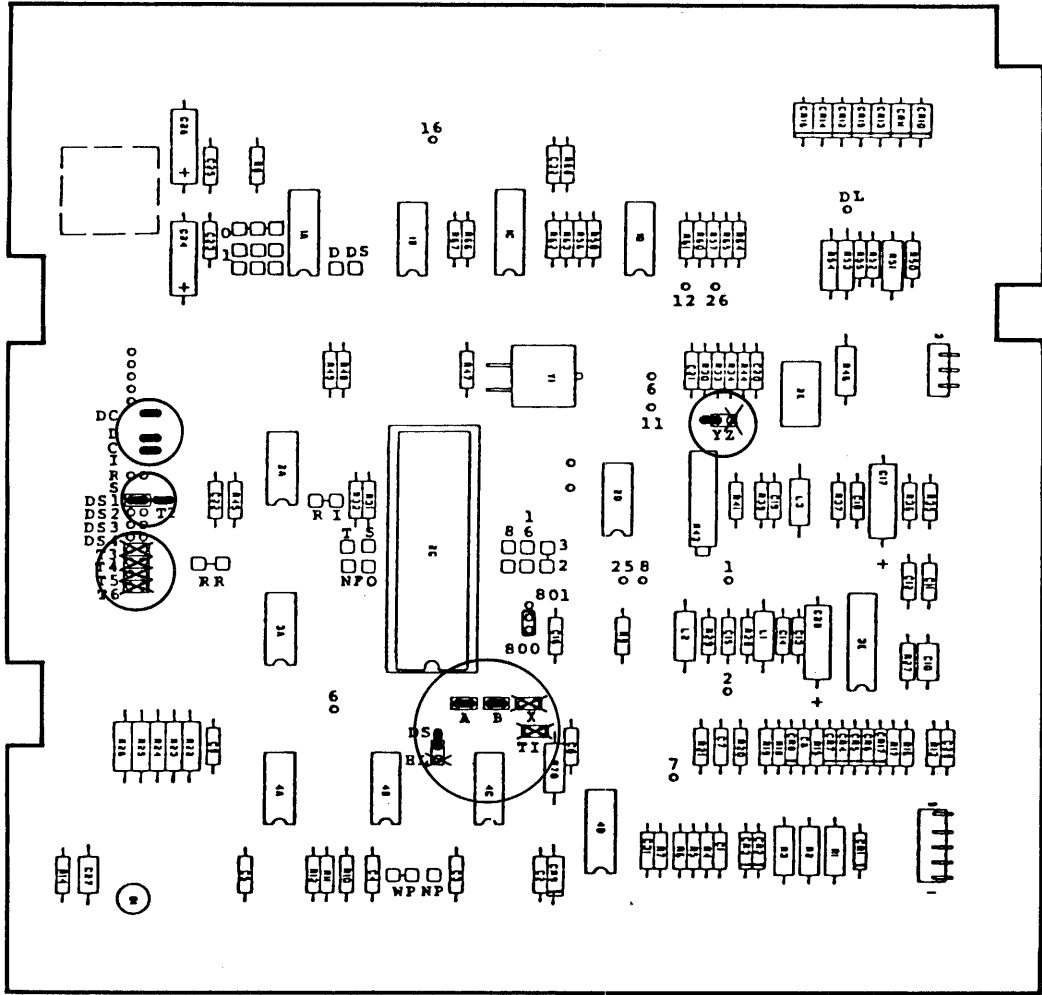


INSTRUCTION SYMBOLS:



REMOVE JUMPER  
 INSTALL JUMPER  
 WIRE WRAP

MAPPING CONFIGURATION, DRIVE 0

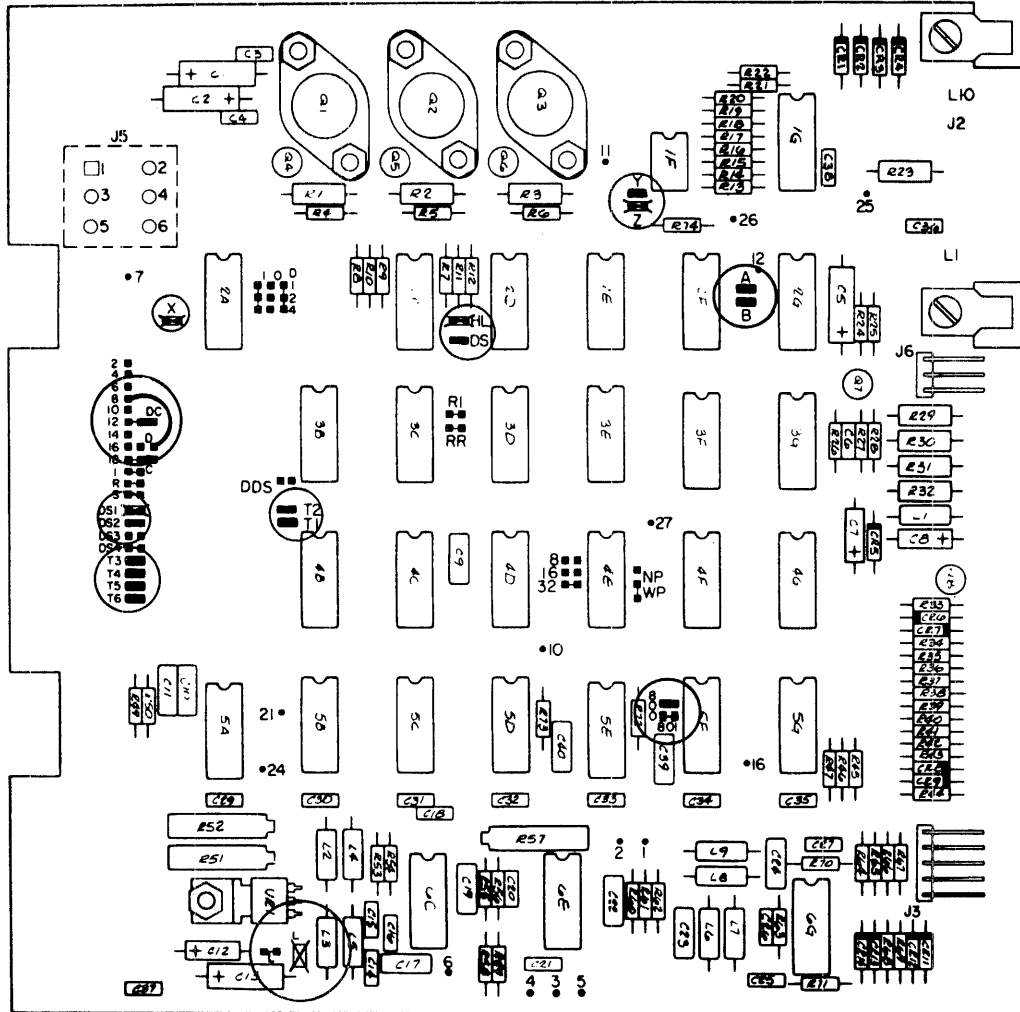


INSTRUCTION SYMBOLS:



REMOVE JUMPER  
 INSTALL JUMPER  
 WIRE WRAP

MAPPING CONFIGURATION, DRIVE 0

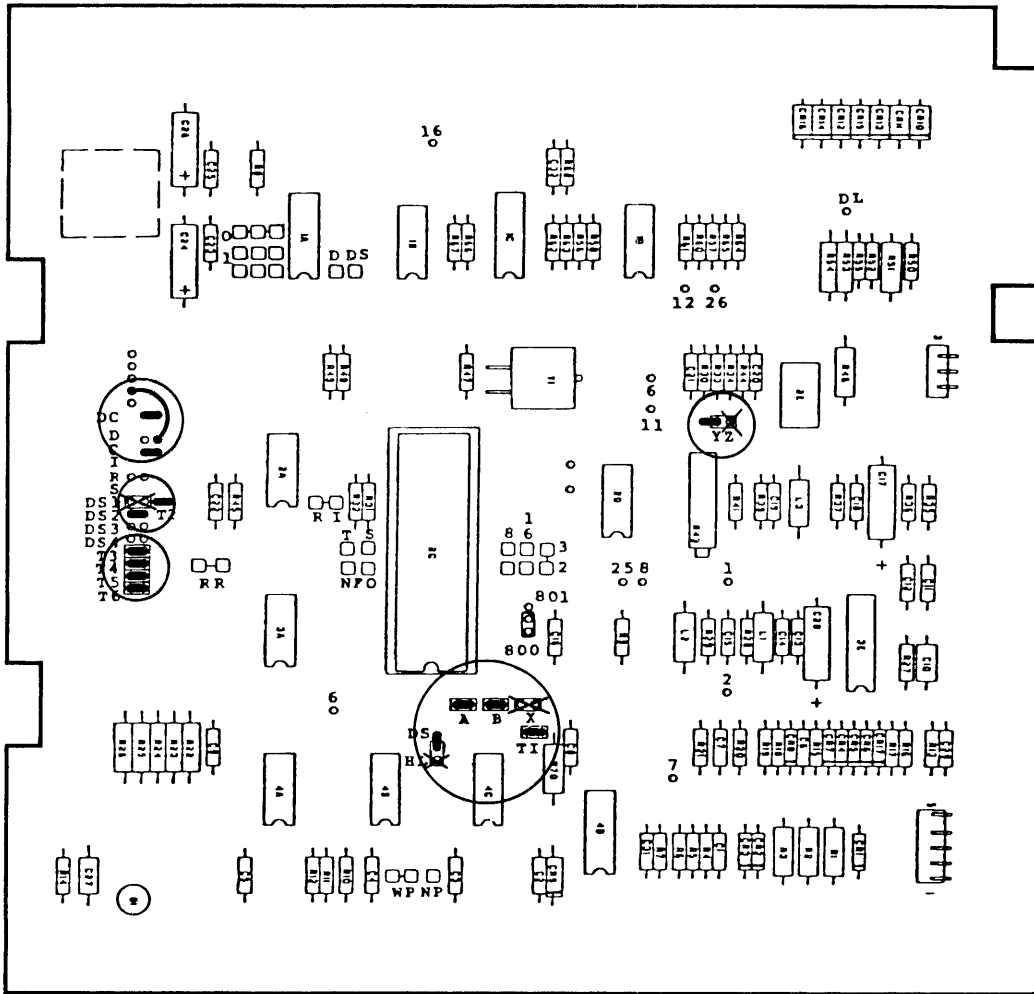


INSTRUCTION SYMBOLS:



REMOVE JUMPER  
 INSTALL JUMPER  
 WIRE WRAP

MAPPING CONFIGURATION, DRIVE 1



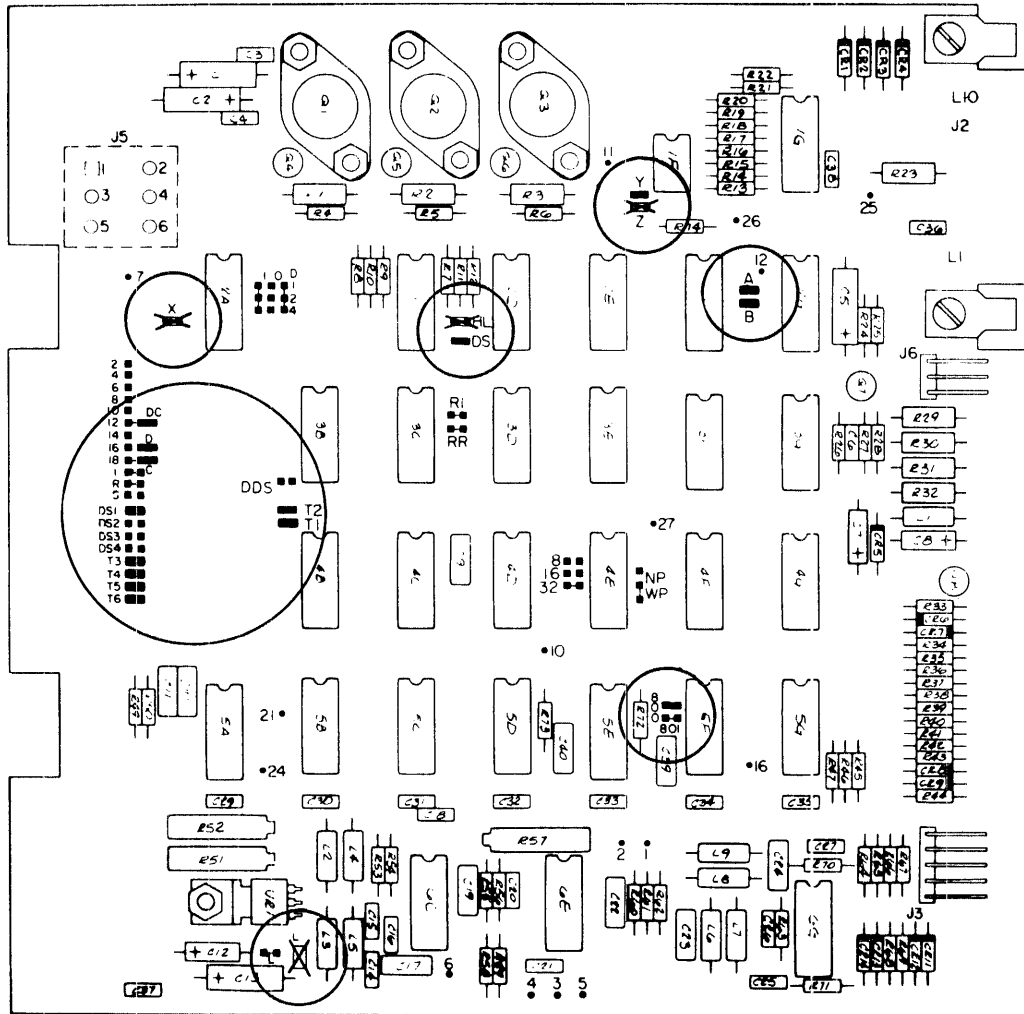
INSTRUCTION SYMBOLS:



REMOVE JUMPER  
 INSTALL JUMPER  
 WIRE WRAP

MAPPING CONFIGURATION, DRIVE 1



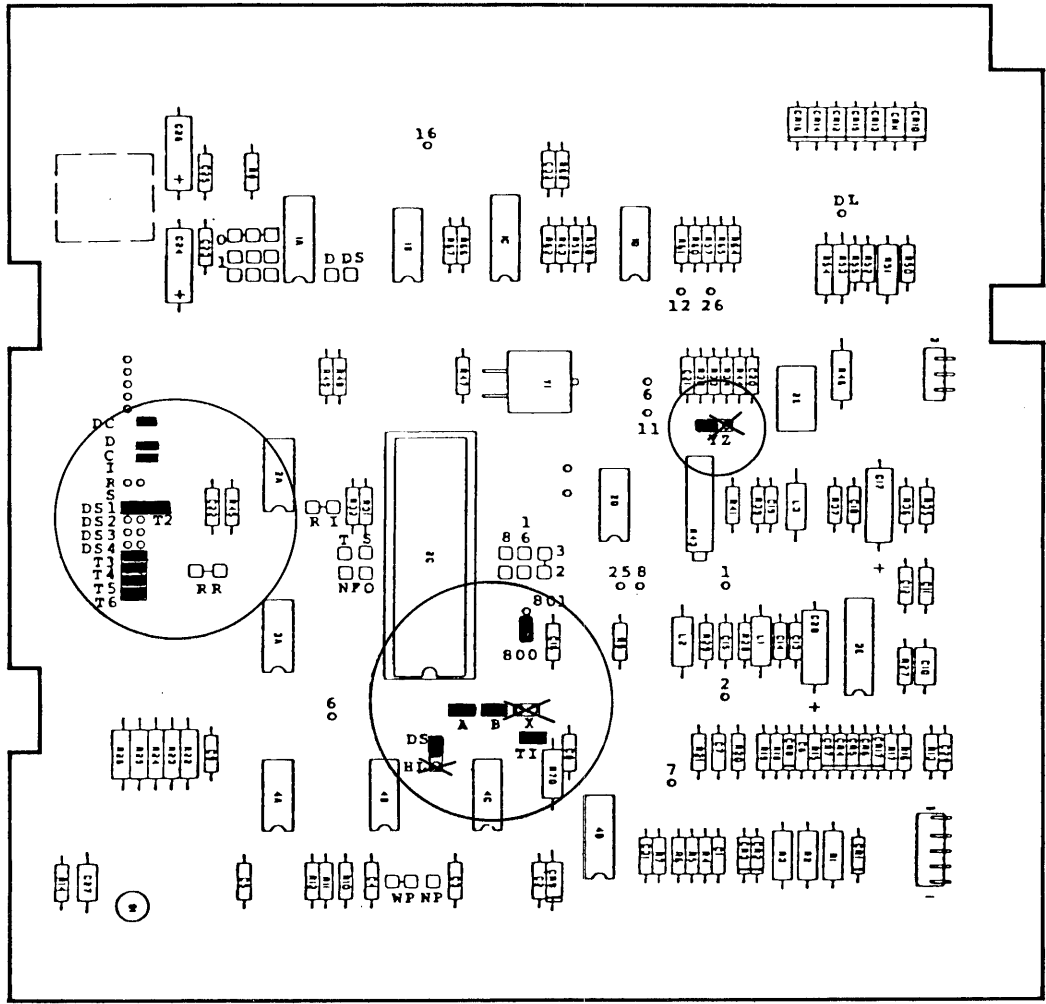


INSTRUCTION SYMBOLS:



REMOVE JUMPER  
 INSTALL JUMPER  
 WIRE WRAP

MAPPING CONFIGURATION, SINGLE DRIVE



INSTRUCTION SYMBOLS:

-  REMOVE JUMPER
-  INSTALL JUMPER
-  WIRE WRAP

MAPPING CONFIGURATION, SINGLE DRIVE



**APPENDIX E**  
**POWER SUPPLY MANUAL**

## SPECIFICATIONS

### CP146 Power Supply

AC Input	100/120/220/240 V AC $\pm$ 10% 47 - 63 Hz
DC Output	+5 Vdc @ 9 A +24 Vdc @ 2.4 A peak -7 to -16 Vdc unregulated @ .5 A
Line Regulation:	+0.05% for a 10% line change (+5 V and +24 V only)
Load Regulation:	+0.10% for a 50% load change (+5 V and +24 V only)
Output Ripple:	10mV pk-pk maximum
Over Voltage Protection:	6.2V $\pm$ .4 Vdc (+5 V only)
Transient Response:	30 $\mu$ sec for a 50% load change
Short Circuit and Overload Protection:	Automatic current limit/fold back from 110% to 150% maximum rated current (+5 V, +24 V only)
Stability:	$\pm$ .5% for a 24 hour warm up
Temperature Rating:	0°C to 50°C ambient at full load (requires >30CFM fan for cooling) Derated linearly to 40% @ 70°C
Temperature Coefficient:	$\pm$ .05% per °C maximum
Efficiency:	Approximately 55% (with full load all outputs at 115 V AC line voltage)

CP146 PARTS LIST

<u>REF. DES.</u>	<u>DESCRIPTION</u>
C1, 2	16000/15 V Capacitor, Elect.
C3	2100/50 V Capacitor Elect.
C9	1000/16 V Capacitor Elect.
C4, 5	470/35 V Capacitor, Elect.
C7	470/16 V Capacitor, Elect.
C6, 10	47/50 V Capacitor, Elect.
C8	.001/100 V Capacitor, Mylar
C11	.005/100 V Capacitor Mylar
CR1	30A 100 V Rectifier, R711A
CR 2-4, 7-10	1A 200 V Rectifier, 1N4003
CR5, 6	3A 100 V Rectifier, MR501
CR11, 12	Zener, 1N752A
Q1-4	Transistor, 2N3005
Q5	Transistor, TIP 31A
SCR1	SCR, 8A 30 V
U1, 2	UA723C, IC, Regulator
J1	AMP Socket 1-380999-0
R2, 6, 9	6.8, $\frac{1}{2}$ W, CF Resistor
R8,10	3.9, $\frac{1}{2}$ W, CF Resistor
R7	2K, $\frac{1}{2}$ W, CF Resistor
R1	47, $\frac{1}{2}$ W, CF Resistor
R4	2.2K, $\frac{1}{2}$ W, CF Resistor
R5	1K, $\frac{1}{2}$ W, CF Resistor
R11, 14	1.1K, $\frac{1}{2}$ W, CF Resistor

CP146 PARTS LIST (continued)

<u>REF. DES.</u>	<u>DESCRIPTION</u>
R16, 18, 19	10K, $\frac{1}{2}$ W, CF Resistor
R15	750, $\frac{1}{2}$ W, CF Resistor
R3, 12	Trim pot, 111-1500
R3	12, 2W, WW Resistor
R17	270, $\frac{1}{2}$ W, CF Resistor

## CP146 POWER SUPPLY TROUBLESHOOTING GUIDE

### SYMPTOM

### POSSIBLE CAUSE

Unit Over-Heating

1. Overload
2. AC input too high
3. Inadequate ventilation
4. Improper transformer primary connection.

Low Output Voltage with High Ripple

1. Output overloaded
2. U1 Faulty (U2)\*
3. CR1, 2 or 3 open (CR4, 5, 6)\*
4. C1, 2 or 9 open (C3, 10)\*
5. Q1, 2 or 3 open (Q4, 5)\*
6. SCR1 shorted or OVP triggered
7. R3, R7 open (R15)\*

High Output Voltage and Ripple, Poor Regulation

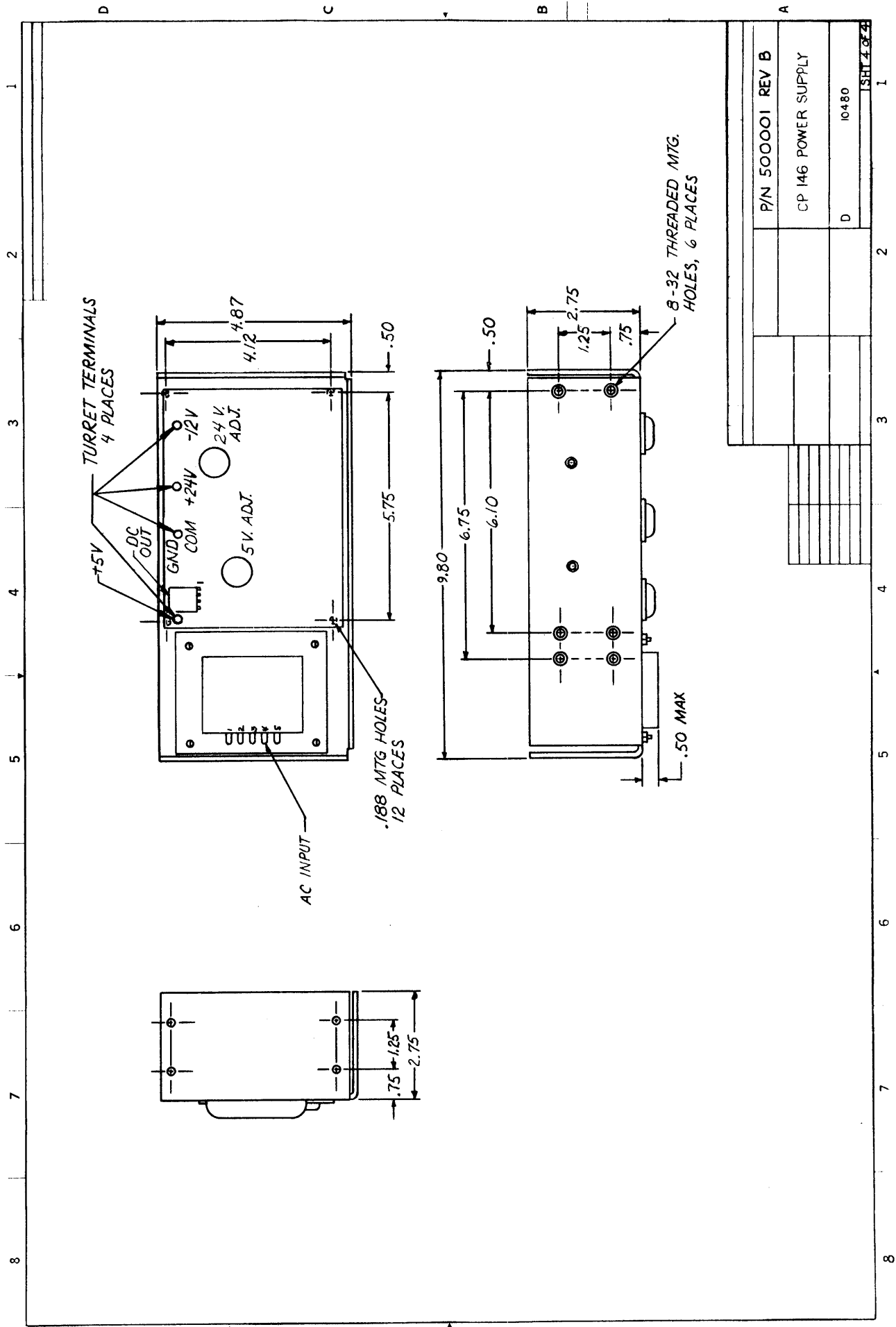
1. Q1, 2 or 3 shorted (Q4,5)\*
2. U1 Faulty (U2)\*

High Input Current Fuse Blows

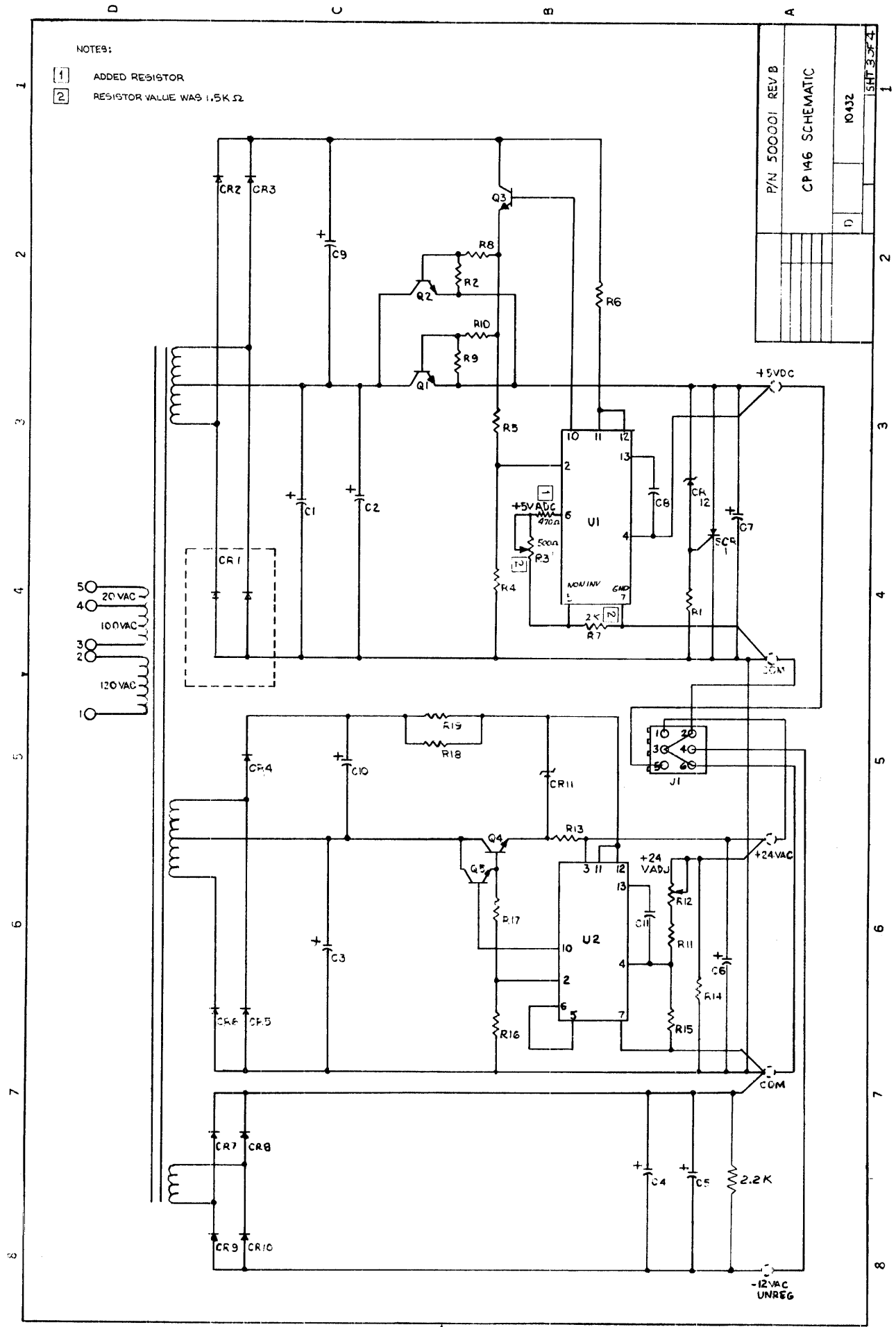
1. Improper input voltage or frequency
2. C1, 2 or 9 shorted (C3, 10)\*
3. CR1, 2 or 3 shorted (CR4, 5, 6)\*
4. CR7-10 shorted
5. C4, 5 shorted

\* Denotes +24V component.





P/N 500001 REV B	
CP 146 POWER SUPPLY	
D	10-480
SHEET 4 OF 4	



NOTES:

- 1 ADDED RESISTOR
- 2 RESISTOR VALUE WAS 1.5KΩ

P/N 500001 REV B	
CP 146 SCHEMATIC	
D	K0432
LSHT 33F-4	

**APPENDIX F**  
**FLEXIBLE DISKETTE DESCRIPTION**  
**AND MAINTENANCE**

## DISKETTE DESCRIPTION

In standard IBM 3740 single density format, a diskette stores up to 256,256 bytes of data. In DEC double density format, a diskette stores up to 512 512 bytes of data. DEC double density format diskettes are made from a standard IBM 3740 format diskette by changing the data address mark on each sector and by writing the data in MFM double density format. Configuration choices for diskettes are:

- 1) Double or single sided.
- 2) Hard sectored or soft sectored.
- 3) With or without a write protect notch.

Since the DSD 440 is shipped with single-sided disk drives, it is important to use only single-sided diskettes, such as the one shown in Figure F-1. If a diskette intended for a double-sided drive is loaded inadvertently into a single sided drive, the drive unit's photo sensor does not line up with the index mark access hole. As a result, the controller never recognizes an index pulse and concludes that the drive is not ready.

### NOTE

Do not use IBM double density diskette Part No. 1766872. These diskettes are intended for use on double-sided drives only.

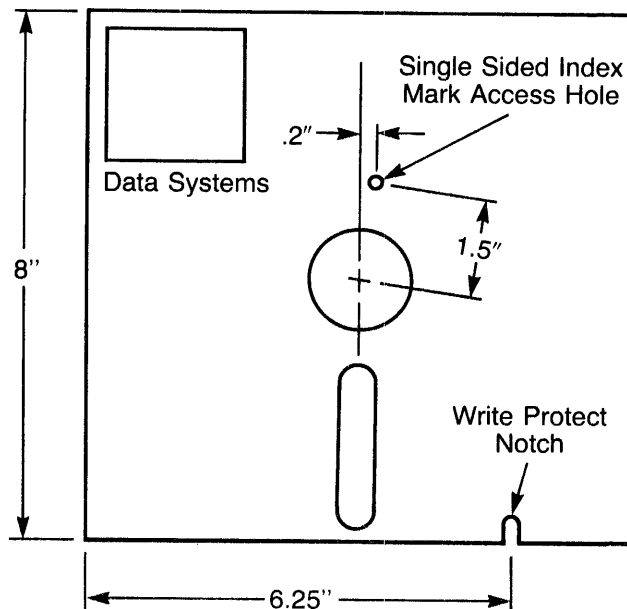


Figure F-1. Single Sided Diskette

The DSD 440 requires the use of soft sectored diskettes. To determine if you have a soft or hard sectored diskette, rotate the mylar diskette inside the envelope while looking through the index mark access hole. If you observe more than one hole punched in the mylar diskette you have a hard sectored diskette. Soft sectored diskettes have only one index hole punched in them.

The data and programs stored on a diskette can be protected from being rewritten. The diskette is "write-protected" by uncovering a notch in the sealed protective jacket at the location shown in Figure F-2. When the write protect notch is uncovered, nothing can be written on the diskette. When the write protect notch is covered, writing is allowed on the diskette. Any opaque tape can be used to cover the write protect notch.

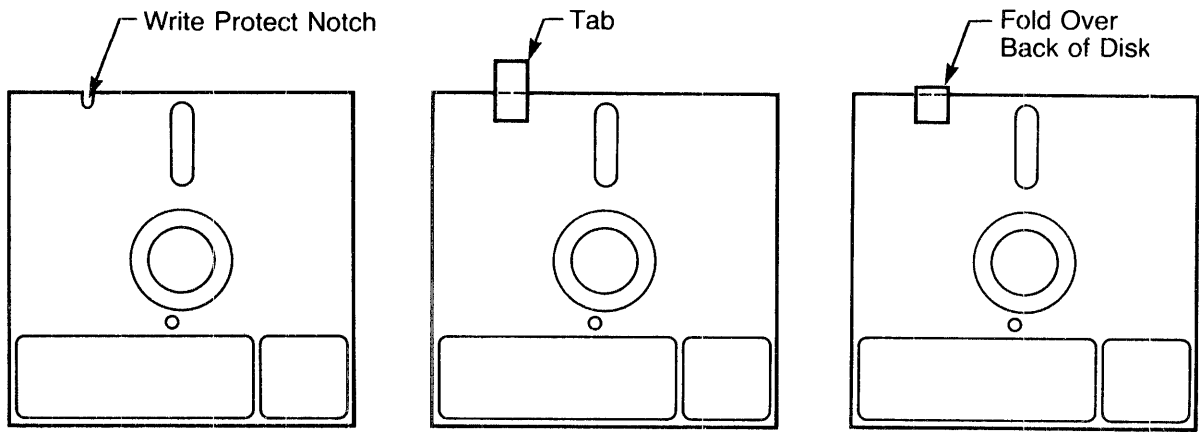


Figure F-2. Write Protect Notch Location

#### INDUSTRY RECORDING AND MEDIA STANDARDS

Unlike rigid diskettes, industry standards have been established for the physical format of the recorded data on diskettes. Each of the formats used by the DSD 440 — IBM 3740 single density and DEC double density — records data on 77 concentric tracks, at a track density of 48 tracks per radial inch. Each track is divided into 26 sectors. Each sector contains 128 eight bit bytes of user data in single density format and 256 eight bit bytes of user data in double density format.

Each sector has an ID (identification) field and a data field. A unique bit pattern known as the ID address mark enables the controller to recognize the start of an ID field. The ID field also contains a track address byte, a head address byte, and a sector address byte. Appended to these diskette address bytes is a pair of CRC (Cyclic Redundancy Check) bytes. These are used to determine if a data error has occurred while reading the diskette address data.

The controller is able to find the sector it wants to read or write by scanning the ID fields. Note that the ID field just described is the same for diskettes containing single density data and those containing double density data. In both cases, all the data bytes contained in the ID field are encoded using the "double frequency" recording technique associated with single density.

Following the ID field of each sector is the data field. The beginning of the data field is identified by another unique bit pattern called the Data Address Mark. After this mark are the 128 or 256 bytes of data and another pair of CRC check bytes. Figure F-3 is a schematic representation of a single density track format. Figure F-4 shows the format of a DEC double density track. Note that only the 256 user data bytes and the 2 CRC bytes following the data are encoded in double density using the MFM ("modified frequency modulation") recording technique. All the other fields (preamble, postamble, and ID) are recorded in the same manner as the single density formatted track.

#### DISKETTE CARE GUIDELINES

The following handling recommendations for diskettes should be followed to prevent unnecessary loss of data or interruptions of system operation.

- 1) Do not touch the exposed diskette surface.
- 2) Keep the diskette away from heat and sunlight.
- 3) Do not use paper clips on the diskette.
- 4) Do not expose the diskette to magnets or tools that may have become magnetized
- 5) Keep diskettes stored in their envelopes. Diskettes not being used should be stored vertically in a file box.
- 6) Do not write on the envelope containing the diskette.

The reliability of your system depends on the care you exercise in handling your diskettes.

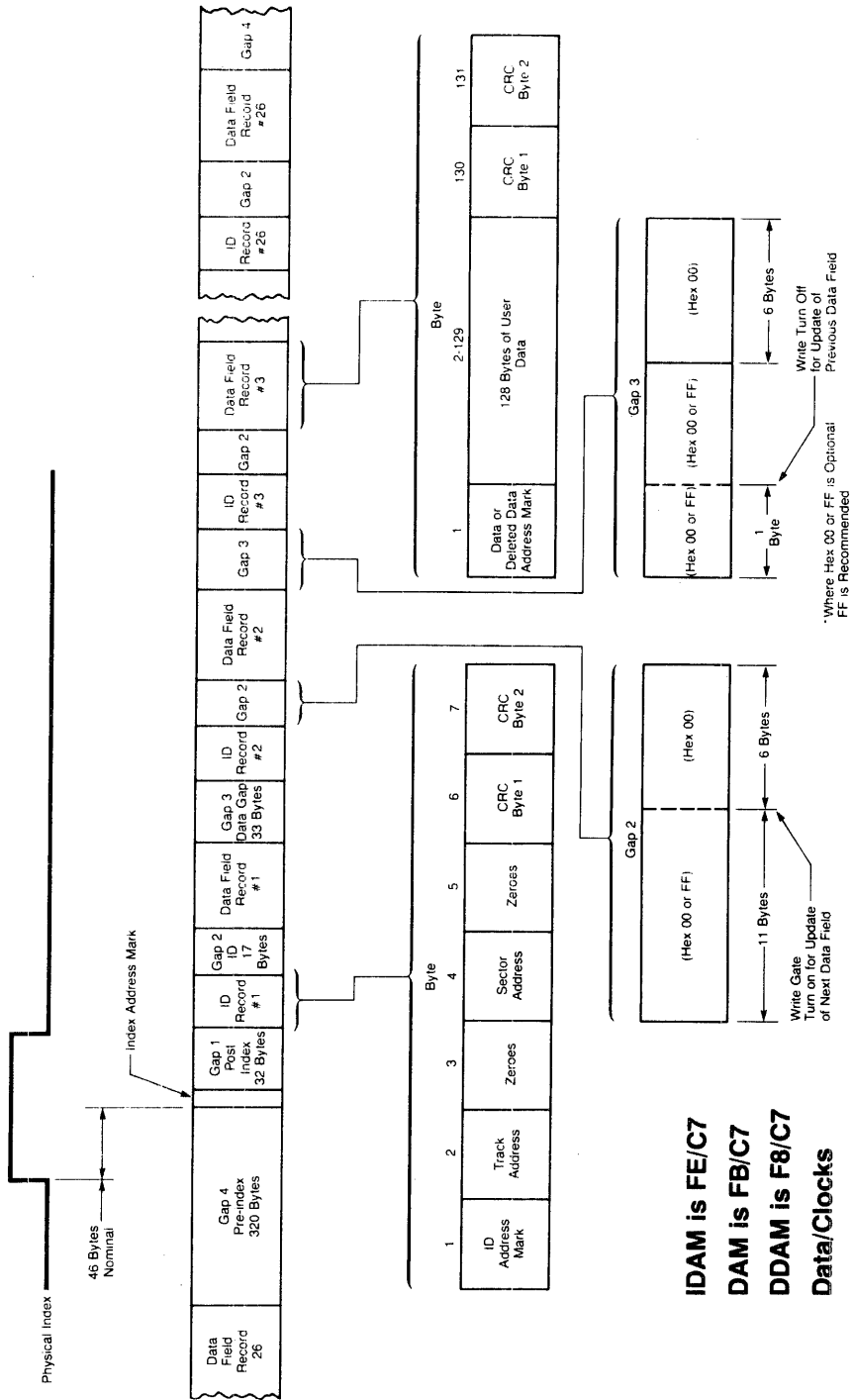


Figure F-3. Single Density Track Format

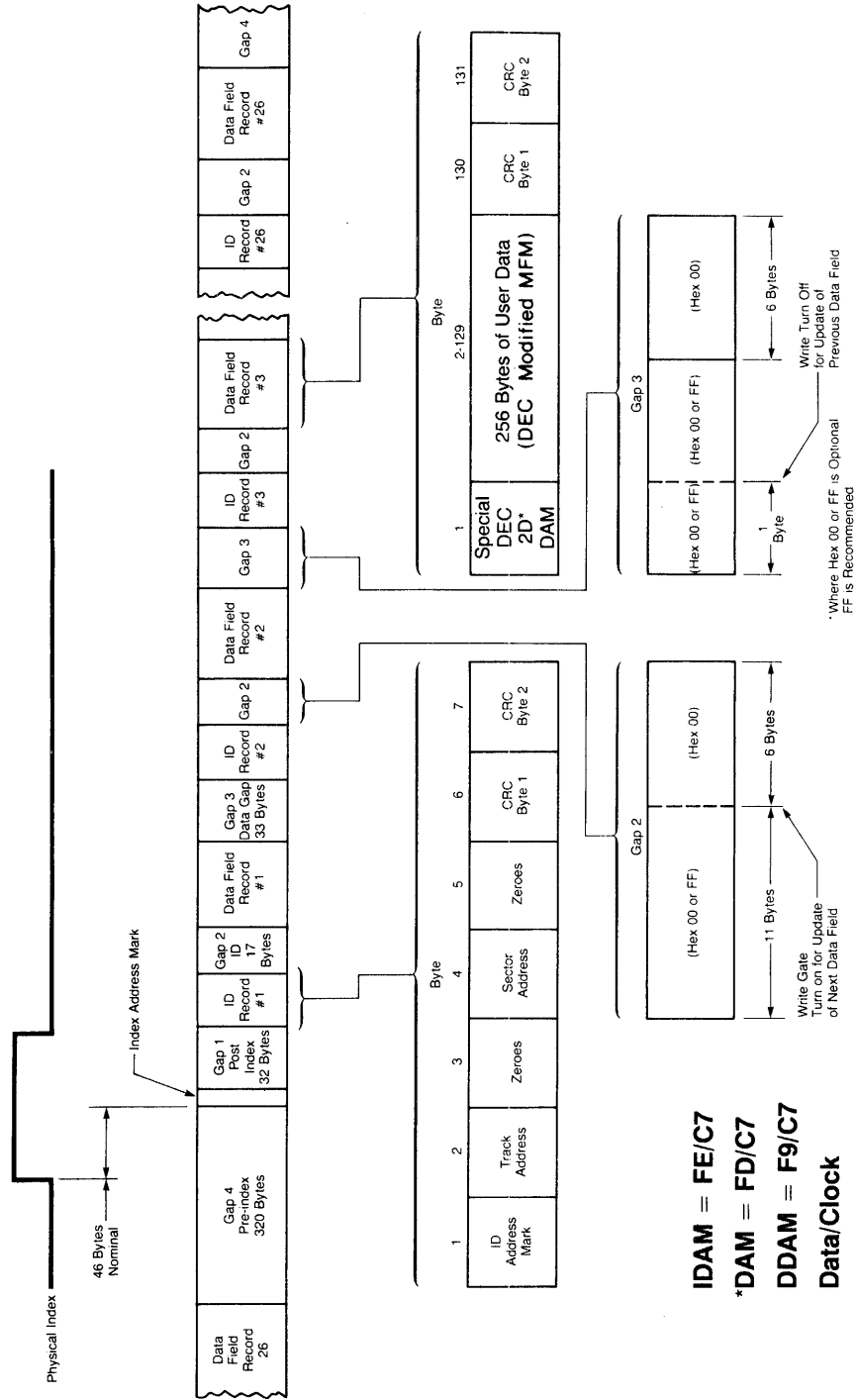


Figure F-4. DEC Double Density Track Format





**APPENDIX G**  
**LSI-11 AND PDP-11**  
**BOOTSTRAP PROGRAM**

;TITLE DSD 440 BOOTSTRAP PROM  
; BOT 440. MAC 06-AUG-79  
;>--BOOTSTRAP FOR DSD 440 FLEXIBLE DISK CONTROLLER--<

I.) CONSTRAINTS AND PARAMETERS

- 1 ) THE DISKETTE BEING BOOTED MUST HAVE THE CORRECT MONITOR FOR THE EXISTING HARDWARE CONFIGURATION.
2. BOOTS EITHER SINGLE OR DOUBLE DENSITY DISKETTES, TYPICALLY: DXMNXX FOR RX01, AND DYMNXX FOR RX02, WHERE XX CAN BE SJ, BL, OR FB
3. NO CHANGE TO THE BOOTSTRAP IS REQUIRED IN RX01 - DSD 210 OR RX02 - DSD 440 MODES.

II. ) BOOTING THE LSI-11 WITH THE LINE TIME CLOCK (LTC) ENABLED

1. ) THIS BOOT CAN BE STARTED WITH THE LTC RUNNING IN ONE OF THE FOLLOWING WAYS:

A) BY INSURING THAT THE STACK IS POINTING TO ANY NONEXISTENT MEMORY LOCATION, THUS FORCING A DOUBLE BUS ERROR ON ANY INTERRUPT. THEN TYPE "(BASE ADDRESS + OFFSET)G". IF A HALT OCCURS DUE TO ATTEMPTED INTERRUPTS TYPE "P", (THIS MAY HAPPEN 3-4 TIMES).  
E.G., FOR A DEVICE WITH RXCS AT 177170 THIS WOULD BECOME "(173000 + 0)G" OR SIMPLY "173000G" SINCE THE OFFSET IS EQUAL TO ZERO.

B) BY SETTING THE PSW PRIORITY BEFORE BOOTING TO DISABLE INTERRUPTS - IN ODT TYPE "\$S/340 <CR> " "R7/(BASE ADDRESS + OFFSET) <CR> " AND HITTING "P".  
E.G., FOR A DEVICE WITH RXCS OF 177170 THIS BECOMES "\$S/340 <CR> " - "R7/173000 <CR> - "P".

III.) THE DSD 440 BOOTSTRAP PROGRAM OUTLINE

- 1.) THE BOOTSTRAP PROCEEDS IN FOUR MAJOR STEPS.

A) SELECT RXCS DETERMINES DEVICE TO BE BOOTED.

B) RAM TEST CHECKS ALL AVAILABLE MEMORY FOR STUCK BITS 0 BOTH DATA AND ADDRESS LINES < 0-30K FOR LSI 0-28K FOR PDP > DOES BOTH ADDRESS AND PATTERN TESTS.

1B)---- CLEARS MEMORY TO ZEROES AND SIZES MEMORY.

2B)---- LOADS MEMORY = ADDRESS AND CHECKS.

3B)---- LOADS MEMORY = COMPLEMENT OF ADDRESS AND CHECKS.

4B)---- LOADS MEMORY WITH THE REPEATING PATTERN OF 131617, 154707, 166343, 173161, and 175470.

- ```

;      C)  FILL EMPTY      FILL EMPTY CHECKS DSD 440 CONTROLLER/INTERFACE
;                               PATH FOR SYNTAX AND DATA ERRORS.  FILL EMPTY
;                               ALSO INSURES ALL AVAILABLE ADDRESS LINES
;                               TOGGLE UNDER DMA.  FILL EMPTY CHECKS BUFFERS
;                               AT LOCATIONS 774, 17700, 37676, 77704 AND
;                               137700 IF MEMORY EXISTS.
;
;      D)  BOOTSTRAP      THE BOOTSTRAP THEN READS IN BLOCK 0 OF THE
;                               SECONDARY BOOTSTRAP FROM THE DISKETTE IN
;                               EITHER RX01 OR RX02 MODES AND RUNS THE
;                               PROGRAM STARTING AT LOCATION 0.
  
```

IV. ) ERROR HALTS AND/OR HANG-UP LOOPS

- ```

;      1)  THE ADDRESSES SHOWN BELOW ARE THE OFFSET LOCATIONS RELATIVE
;           TO THE BASE ADDRESS.  E.G., "173000 + RELATIVE ADDR.", IF THE
;           RELATIVE ADDRESS EQUALS 156 THEN THE ACTUAL LOCATION BECOMES
;           173156.
  
```

<u>ADDRESS</u>	<u>CONDITION</u>	<u>TYPE OF ERROR</u>
156	HALT	MEMORY ERROR AT LOC (R4) DATA READ IN RO EXPECT VALUE IN R4.
204	HALT	MEMORY ERROR AT LOC -2(R4), READ DATA IN RO EXPECT ZERO.
252	HALT	FILL EMPTY ERROR IF R5 =BOOT ADDRESS+522, DATA EXPECTED IN R3, DATA READ IN RO. ERROR AT LOCATION -2(R4) MEMORY ERROR IF R5 =BOOT ADDRESS+112, DATA EXPECTED IN R3, DATA READ IN RO.
314	LOOP	DEVICE ADDRESS SELECTED FOR BOOTING DOES NOT RESPOND.
324	HALT	ERROR FLAG IN RXCS SET AFTER INIT.
342	HALT	RXCS INTERFACE REGISTER STUCK BIT PROBLEM - EXPECT VALUE OF 5460 IN RXCS.
364	HALT	RXDB INTERFACE LATCH PROBLEM, EXPECT VALUE OF 1420 OR 173767 IN RXDB.
400-402	LOOP	DSD 440 TRANSFER REQUEST HANG-UP (FILL EMPTY).
414-416	"	"
452-454	"	"
576-600	LOOP	DSD 440 TRANSFER REQUEST HANG-UP (BOOTSTRAP).
604-606	"	"
652-654	"	"
666-670	"	"
742-746	LOOP	DSD 440 FLAG WAIT ROUTINE HANG-UP.



VI. ) 11/04 AND 11/34 BOOTSTRAPING PARTICULARS.

- 1) A "CONTROL BOOT" ON THE 11/04 OR 11/34 FRONT PANEL PRINTS R0, R4, SP, & R7, ON THE TERMINAL. IF AN ERROR HALT OCCURS AT (BASE ADDRESS + 774) WHILE BOOTING, THEN A REBOOT IS ATTEMPTED, THE FOLLOWING ARE PRINTED OUT:

WHEN ....

R0 = CURRENT DRIVE # BEING BOOTED FROM.

R4 = LOAD ADDRESS WHERE ERROR OCCURRED.

SP = DEFINITIVE ERROR STATUS OF ERROR.

R7 = ERROR HALT ADDR + 2.

- 2) NOTE: A HALT OR HANG-UP OCCURRING BETWEEN (BASE ADDRESS + 742) AND (BASE ADDRESS + 746) THAT WILL NOT RESPOND TO A BREAK OR A HALT IS GENERALLY DUE TO A LACK OF DMA GRANT CONTINUITY ON THE BUS. THE USER SHOULD PLACE THE DSD 440 INTERFACE CARD CLOSER TO THE PROCESSOR, IN ORDER TO INSURE BUS GRANT CONTINUITY.

VII. ) FORMAT OF THE RXCS REGISTER.

- 1) SEE DSD 440 MANUAL CHAPTER 4 FOR FURTHER DETAILS.

BIT# 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0  
 ER--INT--XM--XM--XO2----SDE-DEN-TRG-IEN-DNE-UNT-FNC-FNC-FNC--EX

BIT#	MEANING
15	ER = ERROR FLAG.
14	INT = LOAD INTO RXCS TO INITIALIZE.
13	XM = EXTENDED MEMORY SELECT BITS.
12	XM = EXTENDED MEMORY SELECT BITS.
11	XO2 = 1 IF IN RXO2 MODE.
10	-- = UNUSED.
9	SDE = SIDE OF DISK.
8	DEN = 1 IF IN DOUBLE DENSITY
7	TRG = TRANSFER REQUEST DATA TO/FROM RXDB.
6	IEN = INTERRUPT ENABLE.
5	DNE = DONE FLAG.
4	UNT = UNIT DRIVE SELECTED 0-1.
3	FNC = FUNCTION SELECT.
2	FNC = "
1	FNC = "
0	EX = 1 TO EXECUTE SELECTED FUNCTION.

VIII. ) FORMAT OF THE RXES REGISTER.

BIT#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	--NOT-USED--				NXM-WCO-HDS-UNT-RDY-DLD-DEN-DER-ACL-IND-PAR-CRC											

BIT#	MEANING
11	NXM = 1 FOR NONEXISTENT MEMORY ERROR.
10	WCO = 1 IF WORD COUNT OVERFLOW.
9	HDS = HEAD SELECT, IF 1 THEN SIDE 1.
8	UNT = UNIT SELECT, IF 1 THEN DRIVE 1.
7	RDY = 1 IF DRIVE IS READY.
6	DLD = 1 IF DELETED DATA FOUND.
5	DEN = 1 IF DRIVE DENSITY IS DOUBLE.
4	DER = 1 IF DENSITY ERROR.
3	ACL = 1 IF AC POWER LOSS DETECTED.
2	IND = 1 IF INITIALIZE DONE.
1	PAR = 1 IF PARITY ERROR.
0	CRC = 1 IF CYCLIC REDUNDANCY ERROR.

IX. ) REGISTER ASSIGNMENT FOR MNEMONIC DEFINITIONS.

1) WHEN THE FOLLOWING MNEMONICS ARE USED THEY WILL MEAN THE FOLLOWING REGISTERS.

XCS = R1	POINTER TO RXCS.
XDB = R2	POINTER TO RXDB.
LDP = R4	LOAD POINTER.
SCT = R5	CURRENT SECTOR # (1, 3, 5, 7).

DSD440 - RX02 REGISTER SYNTAX DEFS

RXCS=177170

ERR	INI XM XM X02 ?? SID	DEN TRG IEN	DON UNI FUN	FUN FUN GO
.ERR=	100000	; ERR	ERROR FLAG	
	40000	; INI	LOAD INTO RXCS TO INITIALIZE	
	30000	; XM	EXTENDED MEMORY SELECT BITS	
.DBDMA=	4000	; X02	= 1 IF RX02 MODE SYNTAX	
	400	; DEN	SET = 1 FOR DOUBLE DENSITY	
	200	; TRQ	TRANSFER REQUEST - DATA TO/FROM RXDB	
	16	; FUN	FUNCTION <0-7> - SET "GO" TO EXEC	

RXDB=RXCS+2

	NXM	WCV	SID	DRV	DRV	DEL	DSK	DEN	ACL	INT	SID	CRC							
		OVF	#1	#1	RDY	DAT	DEN	ERR	LOW	DON	RDY	ERR							

REGISTER USAGE IN BOT440 SECTION

XCS=	%1																		
XDB=	%2																		
LDP=	%4																		
SCT=	%5																		

```

; START HERE FOR DEVICE 177170 BOOT
000000 012706 BOT170:: MOV    #-1, SP ; INHIBIT INTERRUPTS IN ONE INSTUCTION
          177777
000004 012700          MOV    #340, RO      ; SET PROCESSOR STATUS WORD
          000340
000010 106400          MTPS   RO          ; FROM REG SINCE READ-MODIFY-WRITE
; ABOVE 5 WORDS BECOME / MOV #340, RO / MOV RO, @#177776 /
;                               / NOP           / IN PDP-11 BOOT
000012 012710          MOV    #177170, (RO) ; SET DEVICE ADDRESS
          177170
000016 000406          BR     BOTCOM
000020 012700 BOT150: MOV    #340, RO      ; SET PROCESSOR STATUS WORD
          000340
000024 106400          MTPS   RO          ; IN ORDER TO DISABLE INTERRUPTS.
000026 000240          NOP           ; MAKE MINIMAL CHANGES TO PDP-11
; ABOVE 2 WORDS BECOME / MOV RO @#177776 / IN PDP-11 BOOT
000030 012710          MOV    #177150, (RO) ; LOAD ALTERNATIVE DEVICE ADDR
          177150
000034 005001 BOTCOM: CLR    R1          ; SET UP MEM TEST PTR
000036 011021          MOV    (RO), (R1)+ ; LOAD DEVICE ADDR INTO LOC 0
; GENERAL ENTRANCE - SET LOC 0 = RXCS VALUE, RO-340, R1=2
000040 012706 BOTGEN:: MOV    #5002, SP      ; INIT STACK
          005002
000044 000005          RESET
000046 004467          JSR    R4, MEMHGH      ; GET POINTER TO TRAP ROUTINE
          000012
; TRAP PROCESSOR FOR NON-EXISTENT MEMORY TIMEOUT
; SETS CARRY AND RETURNS ON NON-EXISTENT MEMORY TRAP
000052 012766 TRAP4: MOV    #341, 2(SP) ; SETS CARRY ON TRAP TO 4
          000341
          000002
000060 000002          RTI          ; ALSO SETS CURRENT PRIORITY HIGH
000062 037177          .WORD   37177      ; LSI-11 CHECKSUM WORD FOR BOTCHK
;                               .WORD   57012   IF PDP-11 BOOT
; NOW TEST FROM 10 TO TOP OF AVAILABLE CONTIGUOUS MEMORY
; INIT VECTORS AND SET LOW TEST LIMIT TO 10
000064 005021 MEMHGH: CLR    (R1)+      ; BUMP TO LOC 4
000066 010421          MOV    R4, (R1)+      ; LOAD TRAP VECTOR
000070 010021          MOV    RO, (R1)+      ; LOAD TRAP PSW VALUE = 340
000072 010102          MOV    R1, R2      ; INIT TO LOW MEMORY = 10

```



```
                ; FIND TOP OF AVAILABLE MEMORY
000074  005022  2$:   CLR   (R2)+           ; FIND TOP OF MEMORY
000076  103403           BCS   4$                 ; CARRY SET BY TRAP TO 4
000100  020227           CMP   R2, #170000      ; AT END OF MEM ADDR SPACE?
                ; STOP AT 160000 IF PDP-11
                ;
000104  103773           BLO   2$
000106  005042  4$:   CLR   -(R2)           ; SET POINTER TO LAST LOCATION+2
000110  004567           JSR   R5, MEMCHK      ; TEST TO TOP OF MEMORY
                ;
                000022
```

```

; FILL EMPTY TEST - DONE AT MULTIPLE BUFFER ADDRESSES IN ORDER
; TO TOGGLE ALL ADDRESS BITS IN SYSTEM MEMORY
000114 004567      JSR      R5, FILEMP      ; DO FILL-EMPTY BUFFER TEST
000150 000150
000120 000774      10+<5*100.>      ; START FILL AT BEGINNING OF
000122 017700      10+<5*1624.>     ; PATTERN REPETITION LEFT BY RAM TEST
000124 037676      10+<5*3262.>     ; DO DMA TEST ACROSS ALL ADDRESS BITS
000126 077704      10+<5*6540.>     ; THAT CAN BE SET IN AVAILABLE MEMORY
000130 137700      10+<5*98 6.>      ; SO ALL BITS TOGGLE OK
000132 000000      0              ; ADDRESS TERMINATOR
000134 000573      BR       BOT440        ; *****
; ROUTINE TO TEST MEMORY FROM C(R1) = LOW LIMIT
; TO C(R2) = UPPER LIMIT BEYOND TEST
; IF ERROR FOUND HALTS WITH R4 POINTING TO ERROR LOC, OR 2 BEYOND.
; RO = DATA READ

000136 010104  MEMCHK:  MOV     R1, R4          ; GET STARTING ADDRESS
000140 010400  2$:      MOV     R4, RO          ; KILL Z FLAG  MOV R4, (R4)+
000142 010024          MOV     RO, (R4)+        ; LOAD CONTENTS =ADDRESS
000144 020402          CMP     R4, R2          ; AT END OF TEST?
000146 103774          BLO
000150 024404  CHKADP:  CMP     -(R4), R4      ; CHECK BACK DOWN TO START ADDR
000152 001402          BEQ     NCKADP
000154 011400          MOV     (R4), RO        ; DATA READ IN ERROR IN RO
000156 000000          HALT          ; STUCK BIT IN DATA OR ADDRESS!!
000160 020401  NCKADP:  CMP     R4, R1
000162 101372          BHI     CHKADP        ; CONTINUE TILL AT START ADDR

000164 005124  SETCOM:  COM     (R4)+        ; MAKE LOC = ADDR COMPLEMENT
000166 020402          CMP     R4, R2          ; AT END OF TEST?
000170 103775          BLO     SETCOM

000172 010104          MOV     R1, R4          ; START AT BEGINNING
000174 060414  CHKCOM:  ADD     R4, (R4)      ; SHOULD BE ALL 1'S
000176 005214          INC     (R4)
000200 012400          MOV     (R4)+, RO      ; DATA SHOULD = ALL ZEROES
000202 001401          BEQ     NCKCOM
000204 000000          HALT          ; STUCK DATA BIT IF NO HALT AT +156
000206 020402  NCKCOM:  CMP     R4, R2
000210 103771          BLO     CHKCOM

; SET UP TO LEAVE A PATTERN OF 1 011 001 110 001 111 B ROTATED
; RIGHT INTO 4 SUCCESSIVE WORDS
; USED AS MEM BACKGROUND AND FILL-EMPTY DATA.

000212 010104          MOV     R1, R4          ; SET INITIAL ADDRESS
000214 012703  SETPAT:  MOV     #131617, R3    ; SET INITIAL PATTERN
000220 020402  4$:      CMP     R4, R2          ; END OF ADDRESS RANGE?
000222 103004          BHIS   CHKPAT        ; GO CHECK DATA IF AT END
000224 010324          MOV     R3, (R4)+    ; CARRY SET BY CMP INSTRUCTION.
000226 006203          ASR     R3          ; ROTATE AND LOAD AGAIN
000230 103773          BCS   4$
000232 000770          BR     SETPAT

```

```

000234 010104  CHKPAT: MOV    R1, R4           ; SET INITIAL ADDRESS
000236 012703  CHKPTL: MOV    #131617, R3
          131617
000242 020324  3$:    CMP    R3, (R4)+         ; DATA OK?
000244 001403          BEQ    4$
000246 016400          MOV    -2(R4), R0           ; SET DATA READ FOR LOOKING
          177776
000252 000000          HALT                    ; PATTERN SENSITIVITY ERROR
000254 020402  4$:    CMP    R4, R2           ; AT END OF ADDRESS RANGE?
000256 103003          BHIS   FILEXT              ; YES - EXIT
000260 006203          ASR    R3                   ; CARRY SET BY CMP INSTRUCTION
000262 103767          BCS    3$
000264 000764          BR    CHKPTL
000266 000205  FILEXT: RTS    R5
          ; FILL - EMPTY BUFFER TEST

000270 012504  FILEMP: MOV    (R5)+, R4       ; GET BUFFER ADDRESS
000272 001775          BEQ    FILEXT
000274 005764          TST    404(R4)            ; DOES MEMORY EXIST?
          000404
000300 103773          BCS    FILEMP              ; NO - STEP TO END OF LIST
000302 005000  FILBUF: CLR    R0
000304 011001          MOV    (R0), XCS          ; GET RXCS ADDR
000306 010102          MOV    XCS, XDB          ; INIT FOR RXDB
000310 004767          CALL   WTFLAG            ; WAIT FOR DONE FLAG UP
          000426
000314 103777          BCS    .                  ; LOOP IF NO BUS RESPONSE
000316 032711          BIT    #.ERR!.DBDMA, (R1) ; ERROR SET OR RX02?
          104000
000322 100001          BPL    .+4                ; HALT IF ERROR
000324 000000          HALT                    ; INTERFACE SETUP ERROR
000326 001417          BEQ    RXFIEM            ; IF RX01 MODE THEN NO LATCH TEST
          ; DSD440 - RX02 INTERFACE LATCHED BIT TEST

000330 012722          MOV    #1420, (XDB)+      ; LOAD INTO RXCS
          001420
000334 022711          CMP    #5460, (XCS)      ; DID THEY LATCH OK?
          005460
000340 001401          BEQ    .+4
000342 000000          HALT                    ; STUCK BITS IN RXCS
000344 022712          CMP    #1420, (XDB)      ; LATCHED OK IN RXDB?
          001420
000350 001005          BNE    RXHALT           ; NO - BAD INTERFACE.

000352 012712  RXDBTS: MOV    #173767, (XDB) ; CHECK RXDB LATCH
          173767
000356 022712          CMP    #173767, (XDB)   ; DID THEY LATCH
          173767
000362 001401          BEQ    .+4
000364 000000  RXHALT: HALT                    ; HALT IF INCORRECT BIT LATCHUP
  
```

```

000366 010102  RXFIEM:  MOV    XCS, XDB      ; SET UP RXDB POINTER
000370 012746      MOV    #200, -(SP)      ; SAVE THE WORD-COUNT
          000200
000374 012722      MOV    #401, (XDB)+    ; DO FILL COMMAND
          000401
000400 105711      TSTB   (XCS)          ; WAIT FOR TRREQ
000402 100376      BPL    .-2
000404 032711      BIT    #.DBDMA, (XCS) ; RX02 STYLE FILL?
          004000
000410 001404      BEQ    FILX01         ; NO - DO RX01 STYLE PROG XFER
000412 011612      MOV    (SP), (XDB)    ; WORDCOUNT (=200)
000414 015711      TSTB   (XCS)          ; WAIT FOR TRREQ
000416 100376      BPL    .-2
000420 010412      MOV    R4, (XDB)      ; BUFFER ADDR
000422 004767  FILX01:  CALL   WTFLLAG        ; WAIT FOR DONE, ERRR, OR TRREQ
          000314
000426 105711      TSTB   (R1)           ; CHECK FOR TRREQ ON RX01
000430 100004      BPL    EMPBFT        ; IF DONE, GO DO EMPTY BUF TEST
000432 112412      MOVB  (R4)+, (XDB)    ; DO ANOTHER BYTE
000434 012716      MOV    #100, (SP)     ; SINGLE DENSITY RX01 COUNT
          000100
000440 000770      BR     FILX01        ; CHECK FOR ANOTHER BYTE

          ; NOW EMPTY SECTOR BUFFER AND CHECK DATA VALIDITY

000442 022424  EMPBFT:  CMP    (R4)+, (R4)+ ; BUMP EMPTY BUFFER ADDR
          ; SO ERROR IF NO DATA TRANSFER.
000444 012711      MOV    #403, (XCS)    ; DO EMPTY BUFFER COMMAND
          000403
000450 010403      MOV    R4, R3         ; SAVE BUFFER START ADDRESS
000452 105711      TSTB   (XCS)          ; WAIT FOR TRREQ
000454 100376      BPL    .-2
000456 032711      BIT    #.DBDMA, (XCS) ; IS TI IN RX02 MODE?
          004000
000462 001404      BEQ    EMPX01         ; NO - DO RX01 STYLE EMPTY
000464 011612      MOV    (SP), (XDB)    ; LOAD WORD COUNT
000466 105711      TSTB   (XCS)          ; WAIT FOR TRREQ
000470 100376      BPL    .-2
000472 010412      MOV    R4, (XDB)      ; AND FILL BUFFER ADDR+2
000474 004767  EMPX01:  CALL   WTFLLAG        ; WAIT FOR ERROR, DONE OR TRREQ
          000242
000500 105711      TSTB   (XCS)          ; TRREQ FROM RX01 TYPE EMPTY?
000502 100002      BPL    CHKEMP        ; NO - CHECK DATA
000504 111223      MOVB  (XDB), (R3)+    ; LOAD THROUGH DATA POINTER
000506 000772      R     EMPX01

000510 006316  CHKEMP:  ASL    (SP)      ; MAKE WORD COUNT INTO BYTE COUNT
000512 010402      MOV    R4, R2
000514 062602      ADD    (SP)+, R2      ; SET R2 = END ADDR TO CHECK
000516 004567      JSR   R5, CHKPTL     ; DO DATA CHECK
          177514
000522 000622      BR     FILEMP        ; DO NEXT FILL-EMPTY
  
```

```

; BOOT THE DEVICE IN LOC 0, REGISTERS USED AS INDICATED BELOW
                                ; R0  UNIT # BOOTED FROM (0, 1)
000001  XCS=    %1                ; R1  POINTER TO RXCS
000002  XDB=    %2                ; R2  POINTER TO RXDB
                                ; R3  READ COMMAND VALUE WITH DENSITY BIT
000004  LDP=    %4                ; R4  LOAD POINTER
000005  SCT=    %5                ; R5  CURRENT SECTOR # (1, 3, 5, 7)
                                ; (SP) WORD COUNT FOR CURRENT DENSITY

000524  005000  BOT440:  CLR      R0                ; SET INITIAL UNIT (0, 1, 2, 3)
000526  011001  MOV      (R0), R1                ; SET RXCS POINTER
000530  000401  BR       BOOTR1                ; ALLOW SAME UNIT

000532  005200  NXTUNT:  INC      R0                ; BUMP DRIVE #
000534  011706  BOOTR1:  MOV      (PC), SP                ; INIT STACK POINTER
000536  005004  CLR      LDP                ; INIT LOAD ADDRESS POINTER
000540  042700  BIC      #RDTBL-UNTDEC, R0 ; ALWAYS INSURE VALID UNIT #.
000544  004367  JSR     R3, UNTDEC                ; GEN A POINTER INTO RDTBL
000550  007      RDTBL:  .BYTE  7, 27 ;47, 67 ; READ SECTOR FUNC FOR DRIVE 0, 1
000551  027
000552  060003  UNTDEC:  ADD      R0, R3                ; POINTER TO READ COMMAND
000554  111303  MOV      (R3), R3                ; GET THE COMMAND
000556  012746  MOV      *100, -(SP)            ; SET LOW DENSITY WORDCOUNT
000562  012705  MOV      #1, SCT                ; INIT SECTOR TO READ
000566  004767  RDLP:   CALL     WTFLAG                ; WAIT FOR DONE FLAG SET?
000572  010150  MOV      XCS, R2                ; COPY RXCS POINTER
000574  010322  MOV      R3, (R2)+              ; LOAD READ COMMAND
000576  105711  TSTB    (XCS)                   ; WAIT FOR TRREQ
000600  100376  BPL     .-2                      ;
000602  010512  MOV      SCT, (XDB)             ; LOAD SECTOR
000604  105711  TSTB    (XCS)                   ;
000606  100376  BPL     .-2                      ;
000610  012712  MOV      #1, (XDB)             ; LOAD TRACK
000614  004767  CALL     WTFLAG                ; WAIT FOR DONE
000620  005711  TST     (XCS)                   ; CLUDGE SINCE DEC RX02 SETS ERROR
                                ; BEFORE IT SETS DONE
000622  100010  BPL     EMPBUF                  ; EMPTY IF NO ERROR
000624  032712  BIT     #20, (XDB)             ; IS ERROR A DENSITY ERROR?
000630  001450  BEQ     DEFNST                  ; NO- DO DEFINITIVE STATUS
000632  052703  BIS     #400, R3                ; SET COMMAND TO DOUBLE DENSITY
000636  012716  MOV     #200, (SP)             ; SET TO D.D. WORD COUNT
000642  000200  BR      RDLP                    ; AND TRY READING AGAIN
000642  000751

```

```

000644 010346 EMPBUF: MOV R3, -(SP) ; GET COMMAND COPY
000646 042716 BIC #4, (SP) ; MAKE INTO AN EMPTY BUFFER COMMAND
000004
000652 012611 MOV (SP)+, (XCS) ; AND EXECUTE
000654 105711 TSTB (XCS) ; WAIT FOR FIRST TRREQ
000656 100376 BPL .-2
000660 032711 BIT #4000, (XCS) ; RX02?
004000
000664 001404 BEQ WTEMDN ; NO - DO TYPE TEMPT
000666 011612 MOV (SP), (XDB) ; LOAD THE WORD COUNT
000670 105711 TSTB (XCS)
000672 100376 BPL .-2
000674 010412 MOV LDP, (XDB) ; AND XFER ADDRESS
000676 004767 WTEMDN: CALL WTFLAG ; WAIT FOR DONE OR TRREQ
000040
000702 105711 TSTB (XCS) ; TRREQ OR DONE?
000704 100003 BPL EMPDON ; BR IF DONE FLAG SET
000706 111224 MOV (XDB), (LDP)+ ; DO RX02 STYLE EMPTY BUFFER
000710 005016 CLR (SP) ; DON'T BUMP LOAD POINTER TWICE
000712 000771 BR WTEMDN

000714 123727 EMPDON: CMPB @#0, #240 ; INSURE FIRST INSTRUCT IS A NOP.
000000
000240
000722 001304 BNE BOOTR1 ; NO - NOT VALID DATA AT LOC 0
; C(SP) = WORD COUNT
000724 061604 ADD (SP), LDP ; BUMP LOAD ADDRESS FOR NEXT SECT
000726 061604 ADD (SP), LDP ; ADD ACTUAL TYPE COUNT
000730 122525 CMPB (SCT)+, (SCT)+ ; BUMP SECTOR # BY 2
000732 020427 CMP LDP, #1000 ; FINISHED LOADING?
001000
000736 002713 BLT RDLP ; READ NEXT SECTOR

000740 005007 CLR PC ; GO DISPATCH

; WAIT FOR DISKETTE FLAGS, DONE, ERROR, TRREQ

000742 032711 WTFLAG: BIT #240, (XCS) ; WAIT FOR DONE OR TRREQ
000240
000746 001775 BEQ WTFLAG ; CAN'T TEST RX02 ERROR HERE
000750 000207 RETURN

```

```

; LOADS DEFINITIVE ERROR CODE INTO STACK POINTER = SP
; THEN HALTS. A PROCEED WILL ATTEMPT TO BOOT THE NEXT DRIVE.

000752 012711 DEFNST: MOV #17, (XCS) ; DO DEFINITIVE ERROR STATUS
000017
000756 105711 DEFNWT: TSTB (XCS) ; WAIT FOR TRREQ OR DONE
000760 001776 BEQ .-2
000762 100003 BPL DEFNRD
000764 010412 MOV LDP, (XCB) ; STATUS UPWARDS FROM LOAD ADDR
000766 010402 MOV LDP, R2 ; SET FOR STATUS READ FROM MEM
000770 000772 BR DEFNWT

000772 011206 DEFNRD: MOV (R2). SP ; SHOW DEFINITIVE STATUS IN SP.
000774 000000 HALT ; EXAMPLE SP VALUE IF HERE
000776 000655 BR NXTUNT ; ALLOW PROCEED IF AVAILABLE
; TO BOOT TRY TO BOOT ON OTHER DRIVE

001000 BOTLST::

000000' .END BOT170

```

BOOTR1	000534R	002	EMPBUF	000644R	002	RXDB =	177172	
BOTCOM	000034R	002	EMPDON	000714R	002	RXDBTS	000352R	002
BOTGEN	000040RG	002	EMPX01	000474R	002	RXFIEM	000366R	002
BOTLST	001000RG	002	FILBUF	000302R	002	RXHALT	000364R	002
BOT150	000020R	002	FILEMP	000270R	002	SCT	=%000005	
BOT170	000000RG	002	FILEXT	000266R	002	SETCOM	000164R	002
BOT440	000524R	002	FILX01	000422R	002	SETPAT	000214R	002
CHKADP	000150R	002	LDP =%	000004		TRAP4	000052R	002
CHKCOM	000174R	002	MEMCHK	000136R	002	UNTDEC	000552R	002
CHKEMP	000510R	002	MEMHGH	000064R	002	WTEMDN	000676R	002
CHKPAT	000234R	002	NCKADP	000160R	002	WTFLAG	000742R	002
CHKPTL	000236R	002	NCKCOM	000206R	002	XCS	=%000001	
DEFNRD	000772R	002	NXTUNT	000532R	002	XDB	=%000002	
DEFNST	000752R	002	RDLP	000566R	002	.DBDMA	= 004000	
DEFNWT	000756R	002	RDTBL	000550R	002	.ERR	= 100000	
EMPBFT	000442R	002	RXCS	= 177170				
. ABS.	000000	000						
	000000	001						
BOOT	001000	002						
ERRORS DETECTED:	0							

VIRTUAL MEMORY USED: 304 WORDS (2 PAGES)  
 DYNAMIC MEMORY AVAILABLE FOR 51 PAGES





**APPENDIX H**  
**PDP-8 BOOTSTRAP PROGRAM**

The PDP-8 interface board manufactured by Data Systems Design does not include bootstrap hardware. PDP-8 users may bootstrap their system manually by following the instructions given in Part II of this appendix.

#### PART I - PDP- 8/A BOOTSTRAP PROM PROCEDURE

PDP-8A users may toggle in the bootstrap manually or they may replace the PROMs on their DEC M8317 bootstrap module. These PROMs require replacement because they are programmed to bootstrap only DEC RX01 compatible systems. They can be replaced with the RX02 compatible bootstrap PROMs included in every PDP-8 interface shipment by Data System Design. The following is the procedure for replacing the PROMs installed on the M8317 module with the DSD bootstrap PROMs:

1. Install the PROM labelled A0089A in socket EB2 with the indentation near Pin 1 pointing toward the handle of the M8317 board.
2. Install the PROM labelled A0090A in socket EB7 with the indentation near Pin 1 pointing toward the handle of the M8317 board.
3. Set the start address to 64 by changing module switches SW1 and SW2 to the positions described in Table H-1.

The PDP-8A may now be bootstrapped by pressing the "Boot" switch on the front panel. For further information refer to Appendix J-119 in your PDP-8A user's manual.

Table H-1. Setting the Bootstrap Address

Switch Position	OFF/ON	Selected Bootstrap Address Bit Value
SW1 - 1	OFF	10(8)
SW1 - 2	ON	4(8)
SW1 - 3	ON	2(8)
SW2 - 5	ON	200(8)
SW2 - 6	OFF	100(8)
SW2 - 7	OFF	40(8)
SW2 - 8	ON	20(8)

#### NOTE

The complete configuration for a useable RX01/RX02 bootstrap is as follows:

	1	2	3	4	5	6	7	8
SW1	OFF	ON	ON	ON	XX	ON	OFF	ON
SW2	ON	OFF	OFF	OFF	ON	OFF	OFF	ON

PDP-8/A BOOTSTRAP PROGRAM

The following listing is the bootstrap program executed by the PDP-8A when these PROMs are installed:

```
      / RX01 - RX02 MODE PDP-8 BOOTSTRAP PROGRAM
      / START AT LOC 33 TO BOOT IN EITHER MODE ON EITHER DRIVE
      /
20 1061 READ,   TAD UNIT           /TRY NEXT SEQUENCE OF DENSITY, UNIT
21 1046         TAD CON360        /ADD IN 360
22 0060         AND CON420        /KEEPING ONLY 420 BITS
23 3061         DCA UNIT          /CYCLES 400, 420, 0, 20, 400
24 7327         AC6               /COMMAND TO READ DISK
25 1061         TAD UNIT          /UNIT + DENSITY
26 6751         LCD               /COMMAND TO CONTROLLER
27 7301         AC1               /TO SET SECTOR, TRACK TO 1
30 4053         JMS LOAD          /SECTOR TO CONTROLLER, AC UNCHANGED
31 4053         JMS LOAD          /AND TRACK
32 7004 LITRAL, 7004             /LEAVING A 2 IN AC,
      /
      /FOLLOWING IS PART OF WAIT LOOP, SAME SECONDARY BOOTS
      / OLD PRIMARY BOOT.
33 6755 START, SDN               /HAS DONE COME UP - STARTS HERE
34 5054         JMP LOAD+1        /NO GO CHECK IF READY TO XFER
35 6754         SER               /SKIP ON AN ERROR, TRY NEXT DENSITY
36 7450         SNA               /AC=2 FOR EMPTY, 0 ON START UP
37 5020         JMP READ          /STARTUP - SET UP UNIT THEN DO READ
40 1061         TAD UNIT          /AC = 2+ UNIT DENSITY
41 6751         LCD               /TO EMPTY SILO
42 1061         TAD UNIT          /SET UP LOC 60 FOR OLD BOOT
43 0046         AND CON360        /KEEPING ONLY DENSITY BIT
44 1032         TAD LITRAL        /ADDING IN 7004 FOR RX01 SYS
45 3060         DCA RX1SAV        /OLD SECONDARY BOOT MOVES THIS
      / TO HANDLER
46 0360 CON360, 360             /LITRAL - EXECUTES AS NOOP
47 4053         JMS LOAD          /READ NEXT WORD FROM BUFFER
50 3002         DCA 2             /TRADITIONAL BOOT LOADING ADDRESS
51 2050         ISZ 50            /INCREMENT LOAD ADDRESS.
52 5047         JMP              47 /GO BACK FOR ANOTHER WORD.
      /
      / SECONDARY BOOT LOADS OVER PRIMARY BOOT UNTIL LOC 47 IS LOADED,
      / THEN CONTROL PASSES TO SECONDARY BOOT
53 0000 LOAD, 0                 /SUBR TO GIVE AND TAKE DATA
54 6753         STR               /WILL IT TALK TO US?
55 5033         JMP START        /NO, PERHAPS DONE WITH EMPTY OR ERROR
56 6752         XDR               /YES-DATA IN OR OUT, IF IN AC UNCHANGED
57 5453         JMP I LOAD       / EXIT.
      /
      / 60 GOES TO OLD SECONDARY BOOT
      / 61 HAS DENSITY AND UNIT THAT ACTUALLY BOOTED
      /
60 0420 RX1SAV, 420             /UNIT*20+7004 FOR SYS HANDLER
61 0020 UNIT, 20                /<DENSITY*400>+<UNIT *20> THAT BOOTED
```

This bootstrap program determines the density of the diskette in Drive 0 and configures itself accordingly. If an error occurs, it switches to the other drive and tries again.

PART II - MINIMAL PDP-8 BOOTSTRAP PROGRAM

The following is a minimal length boot program for both RX01 and RX02 mode systems. This boot tries only the specified drive with the specified density. Start at location 32 to BOOT drive 0. Start at location 22 to BOOT drive 1.

```

/ * THIS FIRST SECTION IS NECESSARY ONLY WHEN BOOTING ONTO DRIVE 1.
/ READS IN SECTOR 1 TRACK 1 ON SPECIFIED DRIVE
/
22*6755 BOTDV1, SDNF /START HERE TO BOOT DRIVE 1.
23*7000 NOP /SKIP THIS WHEN CLEARING FLAG
24*7327 AC6 /SET AC=6
25*1061 TAD UNIT /MAKE INTO READ SECTOR COMMAND
26*6751 LCD /COMMAND = CONTROLLER
27*7301 CLA IAC /SET AC TO 1 FOR SECTOR, TRACK
30*4053 JMS LOAD /SEND SECTOR TO CONTROLLER
31*4053 JMS LOAD /SEND TRACK TO CONTROLLER
/ DOES NOT USE LOCATIONS 22-31 WHEN STARTED AT 32.
/
/ START AT LOCATION 32 TO BOOT DRIVE 0.
/ USES INIT TO READ DRIVE 0 TRACK 1 SECTOR 1
/
32 7305 BOTDVO, CLA CLL IAC RAL /GENERATE THE EMPTY BUFFER COMMAND
33 6755 CHKFLG, SDNF /WAIT FOR DONE FLAG UP
34 5054 JMP LOAD+1 /NO - CHECK FOR READY TO XFER
35 1061 TAD UNIT /YES-PUT IN READ UNIT, DENSITY
36 6751 LCD /SEND EMPTY BUFFER COMMAND
37 5047 JMP BOTLP /START TO LOAD SECTOR BUFFER

////////
////////
47 4053 BOTLP, JMS LOAD /READ NEXT WORD FROM SILO
50 3002 DCA 2 / START LOADING AT LOC. 2
51 2050 ISZ .-1 / BUMP LOAD ADDRESS
52 5047 JMP BOTLP /CONTINUE EMPTYING BUFFER

/
53 **** LOAD, 0 /DATA TRANSFER SUBROUTINE
54 6753 STRF / SKIP IF CONTROLLER WILL SPEAK
55 5033 JMP CHKFLG / NO - CHECK IF FINISHED
56 6752 XDR / TRANSFER DATA IN OR OUT
57 5453 JMP I LOAD / RETURN TO CALLER

/
60 7004 (DY0) OR 7024 (DY1) / USED BY SECONDARY BOOT
/ TO SELECT DRIVE 0 OR DRIVE 1
61 0000 OR 0400 OR 0020 OR 0420 / DX0-SD, DX0-DD, DX1-SD, DX1-DD

```

This bootstrap requires different values in locations 60 and 61 for single or double density, and Drive 0 or Drive 1 bootstrapping. These values are listed below.

```

Drive 0, Single Density - Start at Location 32
60/ 7004
61/ 0000

```

Drive 0, Double Density - Start at Location 32  
60/ 7004  
61/ 0400

Drive 1, Double Density - Start at Location 22  
Locations 22 - 31 must be entered  
60/ 7024  
61/ 0420

**APPENDIX I**  
**CONTROLLER ERROR CODE**  
**DESCRIPTIONS AND CAUSES**

## ERROR CODE DESCRIPTIONS AND CAUSES

### Error Code: 010

Drive 0 failed to home on INIT

Possible Causes:

- No Drive 0 in system.
- Bad track 0 sensor. This is especially possible if noise is noted during drive stepping operation and if the head is at the outside of the diskette.
- Drive electronics could be incorrectly jumpered causing the stepper motor to be actuated by the Headload Signal instead of the drive select signal. (HL is jumpered instead of DS.)

Check Next:

- Drive 0 jumpers.
- Track 00 sensor on Drive 0.
- Drive flash indicating the error. (If not then the system could consist of two Drive 1's.)

### Error Code: 020

Drive 1 failed to home on INIT. (Track 0 not found while head was stepping out.)

Possible Causes:

- No Drive 1 in system when the configuration switch is set for a 2 drive system.
- Bad track 0 sensor (open) on Drive 1.
- Two Drive 0's.
- Stepper motor failure.

Check Next:

- Head position on Drive 1. If the head is at the outside of the diskette and noise accompanies the stepping operation, then the track 00 sensor probably failed on Drive 1.
- If both drive activity lights flash together then there are probably two drive 0's.
- Check if there is a Drive 1. (This could be a single drive system with the controller module is set for a two-drive system.)

Error Code: 030

Track 0 found while stepping in on INIT. (This can happen on either Drive 0 or Drive 1.)

Possible Causes:

- Track 00 sensor is shorted.
- The STEP-IN L line is stuck-at-low.
- The drive could have been out beyond track zero during INIT.
- Drive is wired incorrectly so that "headload" enables the stepper instead of "drive select".
- Drive cable is installed backwards.
- Stepper motor failure

Check Next:

- Drive flash indicating INIT error.
- Drive signal STEPIN L is shorted.
- Check if drive was at negative track and try the INIT operation again.
- Check jumpers on drive electronics.
- If the drive cable is backwards then both drivers would have heads loaded and activity lights would be on.

Error Code: 040

Invalid drive or track address specified.

Possible Causes:

- Software operation requesting Drive 1 in a one-drive system.
- DIP-Switch on the controller/formatter module set incorrectly.
- There is a software error.

Check Next:

- Immediately after the error, use the "Extended Status Dump" command (#7) to find the requested drive and track.

Error Code: 050

Track 0 encountered unexpectedly



Possible Causes:

- This could be a problem in the controller module but it must be intermittent. Normally, the controller would find 30 error on INIT.
- Stepper failure (over step outwards toward track 0).
- Direction line failed.

Check Next:

- This is a rare failure which implies a seek problem.

Error Code: 070

Requested sector not found in two revolutions.

Possible Causes:

- Desired ID sector has a hard CRC error.
- Improperly formatted diskette (missing sector requested).
- R/WC time out while looking for a byte in header (controller failure).

Check Next:

- Check if a bad diskette is inserted in the system.
- Check if the drive is generating bad diskettes because of excessive wear.

Error Code: 100

Write-protect violation (attempt to write on write-protected diskette).

Possible Causes:

- Error occurs during write, format or set media density commands.
- Diskette write-protect tab is missing.
- Diskette write-protect tab is not an opaque sticker.
- Shorted write-protect sensor.
- Format operation attempted on a drive not containing a diskette. (This could not occur during normal write on the missing diskette because a 260 error would occur first.)

Check Next:

- Write-protect tab.
- Write-protect sensor on related disk drive.

Error Code: 110

Drive read signal lost (48 microseconds elapsed with no read pulses from the drive).

Possible Causes:

- Headload problem (headload solenoid).
- Weak head load solenoid.
- Defective component installed in the drive electronics.
- Bad negative supply. (This is only used for analog read circuits.)
- Head is trying to read on non-existent Drive 1.
- Drive electronics could be jumpered incorrectly (headload select).
- The L jumper may be inserted on the drive resulting in a shorted regulator on the drive electronics board.

Check Next:

- Check if the head is loaded.
- Check the negative power supply at the drive.
- Check that Drive 1 is jumpered correctly by using the INIT cycle.
- Check the jumpering on the drive electronics.

Error Code: 120

No preamble found (R/W Controller could not identify preamble-independent of phase-lock loop).

Possible Causes:

- Damaged media (track erased).
- Drive pulley size is incorrect.
- Read signal is weak or fading.

Check Next:

- Check if DSD 440 RX2ES indicates the drive is ready.
- Check drive read signal strengths.

Error Code: 130

Preamble found but no IDAM within window (preamble seems to continue forever).

Possible Causes:

- Down-level microcode on DSD 440 cannot find IDAM.
- Diskette is misformatted.
- Although mark not found, read/write controller and PLL were able to continue finding valid preamble so that circuitry must be OK.

Check Next:

- Use FRD440 or scan hyper-diagnostic to verify that error occurs on multiple tracks.
- Check another diskette.

Error Code: 140

CRC error on what appeared to be a header. (This error code is not accompanied by the error flag in the RXCS.)

Possible Causes:

- Head-load problem has occurred. This includes head bounce problems as well as more common head-load problems.
- This error can only show up in the "non-0" error status print-out of FRD440 as it does not generate an error abort.
- Suspected header is internally inconsistent with its CRC.

Error Code: 150

Track or head in good header did not match expected. (The CRC code on the ID sector field was correct; the track or head within the ID sector field did not match expected value.)

Possible Causes:

- Bad drive exists.
- Stepper motor is malfunctioning.

- Bad head guide shaft bearing exists.
- A step circuit on drive PCB is defective.
- Invalid diskette format.

**Check Next:**

- Use seek test on FRD440.
- Use butterfly seek test in "HYPERDIAGNOSTICS".

**Error Code: 160**

Too many tries for an IDAM. (Can find a preamble so R/WC is OK but overflows bad IDAM counter.)

**Possible Causes:**

- Phase-lock loop has problem in controller.
- Read channel in drive is weak.

**Check Next:**

- Use good diskette in drive under test to check drive signal amplitude.
- Check DSD 440 controller with PLL "HYPERDIAGNOSTIC" test.

**Error Code: 170**

Preamble found but no Data Address Mark followed. (Correct ID sector found, valid data preamble found, but no DAM followed.)

**Possible Causes:**

- Diskette is mis-written.
- Damaged media.

**Check Next:**

- Verify read operation on DSD 440 using good diskettes.

**Error Code: 200**

Data CRC error

**Possible Causes:**

- Diskette is defective.

- Read head in drive is weak.
- Read channel on drive PCB is weak.
- Electrical interference (EMI) is occurring.
- Diskette is contaminated with airborne particles.

Check Next:

- See if error was hard or soft, depending if in FRD440 the error occurred once or twice.
- Attempt to read diskette on a different drive to check drive alignment problem (possibly diskette was read or written misaligned).
- Examine diskette for signs of damage and wear from head load pad.
- Note that 200 errors are generally intermittent in nature. They can be caused by a variety of unpredictable events including lightning and power glitches.

Error Code: 210

Parity error on interface cable.

Possible Causes:

- Electro-magnetic interference is occurring.
- Interface to Controller cable is wearing out or is near ac power lines.
- Interface board is defective
- Controller board is defective.

Check Next:

- Check if parity error occurs often. If not, it is most likely a result of sporadic interference.
- Use extended self-test to trouble-shoot DSD 440 controller.

Error Code: 220

Read/write controller self-test failure (unexpected R/WC error).

Possible Causes:

- DSD 440 board has a hardware problem. Under no conditions should R/WC interrupt 8085 with error code equal 0.

Check Next:

- Check interrupt circuit on DSD 440 controller board.

Error Code: 230

Invalid word count specified

Possible Causes:

- This code results from a programming error.

Check Next:

- Immediately use Extended Status Dump to determine diskette media density and requested word count.

Error Code: 240

Density error (density mismatch).

Possible Causes:

- This is a normal occurrence in system software.
- Write gate or other write circuit on disk drive failed.

Check Next:

- Check to see if alternate drive will execute software properly. May be a write circuit problem.
- Note that 240 error does not cause flashing of activity LEDs as 240 error is common occurrence.

Error Code: 250

Wrong key word for Set Media Density.

Possible Causes:

- Set Media Density command has programming error.

Error Code: 260

Indeterminate density (system was unable to determine density of selected diskette).

Possible Causes:

- Any read error while checking density could cause 260 error. If read signal is lost 110 error could occur instead.

- Diskette has bad data, possibly caused by drive write problem.
- Two Drive 0's exist, both of which have diskettes inserted.
- Selected drive has head-load problem.

Check Next:

- Check if diskette is readable on alternative drive.
- Check if system INITs properly, and if there are 2 Drive 0's.
- Check if head loads at all.
- Check for head-load problem on DSD 440 controller module.

Error Code: 270

R/WC write format failure (time-out waiting for index mark at beginning of format operation or at end of format operation).

Diskette must be in system as earlier index pulse was required to get past 360 error time-out.

Possible Causes:

- Disk is spinning too slowly, caused by 60 Hz pulleys on drive powered by 50Hz AC.

Check Next:

- Use Read Status Command to see if selected diskette is spinning at correct speed.

Error Code: 320

Read/write controller write failure (R/WC time-out failure during write or format).

Possible Causes:

- DSD 440 controller has failed.

Check Next:

- Use Extended Self Test or SA 4 to check hand shake between 8085 and R/WC.
- This error indicates that a read/write controller command took much longer than it normally should have.

Error Code: 330

Read/write controller time-out on the reset (R/WC did not return ready after INIT).

Possible Causes:

- The DSD 440 board has a hardware problem.

Check Next:

- Check R/WC and 8085 handshake using extended self-test or SA 4.

Error Code: 340

Master controller out of sync with R/WC on controller module.

Possible Causes:

- 4440 hardware failure has occurred.

Check Next:

- Check R/WC to 8085 handshake using extended self-test or SA 4.

Error Code: 350

Non-existent memory error during DMA (after error occurs system is locked-up and it is impossible to retrieve this error code).

Possible Causes:

- Interface board (bus address register) is bad.
- A programming error exists (invalid buffer address specified).
- Memory board has failed.

Check Next:

- Use hardware bootstrap to execute RAM test in host computer memory.
- Try booting in RX01 mode in order to avoid use of DMA circuit. This pins problems down to DMA circuit and G-bus or UNIBUS.

Error Code: 360

Drive not ready during format command (error indicates missing index pulse at beginning of format command).



Possible Causes:

- Diskette is not spinning or spinning too slow (60 Hz pulley, 50 Hz power).
- 800 jumper is missing on drive.

Check Next:

- Use Read Status to check to see if selected drive is spinning at correct speed.
- Verify selected drive, being Drive 0, can use INIT operation to check if drive is ready. If ready, read will occur on drive 0. If not ready, read will not occur on drive 0.

Error Code: 370

AC power caused abort of write activity.

Possible Causes:

- Power OK is sensed before write operation or format operation. If AC power is not OK at that point the write operation is aborted. Since the AC power-low detect circuit is not stuffed in a normal controller board this error should never occur.

Check Next:

- Check primary AC power if AC detect circuit is stuffed on the 4440 board.



APPENDIX J

FLPEXR DIAGNOSTIC PROGRAM USER'S MANUAL

---

---

# FLPEXR Users Manual

---

- INTRODUCTION
- PROGRAM LOADING
- PROGRAM EXIT
- PROGRAM COMMANDS
- PROGRAM INPUT/OUTPUT
- PROGRAM STATUS AND ERROR DISPLAYS
- DETAILED DESCRIPTION OF COMMANDS

- Comprehensive Tests
- Individual Tests
- Media Modification
- Program Control Values
- Program Status
- Data Utilities

## INTRODUCTION

All DSD flexible disk systems with an LSI-11 or PDP-11 interface board are shipped with a diskette containing an interactive diagnostic program called FLPEXR. The manual explains the operation of this comprehensive set of tests and utility programs. This manual assumes the user is familiar with floppy diskette operations and terminology.

FLPEXR supports the full product line of floppy disk drive products and multiple drive systems with 1 through 4 drives per system. It is a standalone program, capable of being bootstrapped into the processor. It performs auto configuration of certain control parameters, determining both disk and CPU characteristics. It supports both hard copy and video display terminals with full x-on, x-off output control. In order to facilitate unattended testing, terminal output is also retained in a circular buffer autoconfigured to the full

available memory; commands are also provided to display and reset the circular buffer. Commands are also provided for diskette formatting, examination, duplication, and comparison. Test commands fully exercise system capabilities with operational parameters being user selectable through commands. The acceptance test and verify commands are suitable for both incoming quality control checks and system exercise/burn-in.

## PROGRAM LOADING

FLPEXR requires a standard console device, an LSI-11 or PDP-11 computer and at least 12K words of memory. Loading FLPEXR can be accomplished by two methods. One method is to bootstrap the diagnostic diskette. This loads FLPEXR into memory automatically. The other method requires an RT-11 operating system. The FLPEXR diagnostic diskette has an RT-11 compatible directory and file space. The files on the diagnostic diskette can be accessed using standard RT-11 procedures. For example, FLPEXR can be run from an RT-11 system by typing.

```
RU DEV: FLPEXR <CR>
```

where <DEV:> might be DX0:, DX1:, DY0:, DY1: as appropriate.

On a system running other operating systems (e.g., RSX11M, IAS, RSTS, etc.), the distribution diskette must be bootstrapped into memory.

Since both bootstrap and diagnostic programs handle RX01 and RX02 protocols, FLPEXR diagnostic diskette may be used with any DEC compatible disk system.

Once the FLPEXR diagnostic program has been loaded into memory, the diagnostic diskette may be used with any DEC compatible disk system.

Once the FLPEXR diagnostic program has been loaded into memory, the diagnostic diskette should be removed from the drive so it is not erased.

Two high quality, write-enabled formatted diskettes of the same type (density and number of sides) should be installed in the FLPEXR drives before proceeding with any of the tests.

After FLPEXR is loaded into memory, a brief description is displayed on the terminal which includes a memory map, preliminary usage instructions, and a prompt for selection of device type.

The memory map indicates the ranges of the address space which responds with SSYNC (or BRPLY) when accessed by the host computer. The figure below shows the text initially output:

```

<Memory map>
Remove distribution diskette.
DSD floppy disk diagnostic with format capability.
Type 'V' to do verify/acceptance test on two drives.
    This will do a set media and short verify.
    Then go into a regular acceptance test.
Type 'H' for a list of valid commands.
Type 'FO' to format a diskette.
CTRL-C returns to mode.
CTRL-R aborts function and returns to mode.
All numeric inputs/outputs are in octal.
Insert test diskettes (both must be of same density).
Enter device type (0 to 7) or 'H' for list of types.
    
```

The device type specification is used by FLPEXR to set up internal control values that tailor the program's operation to specific DSD product capabilities. An input of 0 will select a default value that is applicable for all products. The device flag (which is the major control value set by the device type specification) can be modified during program operation by the 'SET DEVICE' command. An 'H' input in response to the device type prompt will output the list of types as shown below:

<i>Type</i>	<i>Device</i>
0	Default
1	110
2	210
3	430
4	440
5	470
6	480
7	880

Which type of device? (0 to 7):

After the device type is selected, FLPEXR will output the device flag being used, as shown below.

Device flag being used is: XXXX  
Use set device command to modify flag

FLPEXR then outputs the name and version number of the program.

DSD FLPEXR V2A

FLPEXR types "<CRLF> #" when starting, and the program then attempts an INIT (initialize) instruction. When the INIT cycle is successful, the program types the prompt word: "DD COMMAND:" or "COMMAND:". This prompt string allows the operator to input a command. The "DD" indicates that the program is accessing double density diskettes. A list of all the available commands may be obtained by typing an 'H' (HELP).

### PROGRAM EXIT

If FLPEXR was loaded via the bootstrap, the operating system must be rebooted.

If FLPEXR was loaded via the RT-11 operating system, direct return to the operating system may be possible. A control input of 'CTRL C' will cause FLPEXR to output 'EXIT TO RT-11?'. A 'Y' response will cause the return to the RT-11 monitor. Exit to the monitor may not function if:

1. There is insufficient memory available.
2. The system device is not located at 177170.
3. The system device or diskette is not available.

If the direct monitor exit is not possible, the operating system must be rebooted.

### PROGRAM COMMANDS

Legal responses to "COMMAND:" are listed in Table 1, grouped by class of command. Only the characters enclosed in parenthesis need to be typed. The parenthesis should NOT be typed. When the typed string is recognized, the terminal "BELL" will sound at which time <CR> should be typed. The program will fill in the remaining characters and then proceed to execute the function.

FLPEXR also recognizes various control inputs. Table 2 lists the control input and the associated action. This input can be performed at any time, even while a test is in progress.

**Table 1.** FLPEXR Commands

<i>Command</i>	<i>Description</i>
<b>Comprehensive Tests</b>	
• (V)ERIFY	General Exerciser
• (SH)ORT VERIFY	Short Exerciser
<b>Individual Tests</b>	
• (F)ILL EMPTY	Fill/Empty Buffer Test
• (SEQW)/R	Sequential Write/Read Test
• (SEQ)READ	Sequential Read
• (RA)NDOM R/W	Random Read/Write
• (REA)D RANDOM	Read Random
• (SC)AN	Scan
• (SEE)K RANGE	Seek Range
<b>Media Modification</b>	
• (SET M)EDIA DENSITY	Set Media Density
• (FO)RMAT	Format Diskette
<b>Program Control Values</b>	
• (SET U)NIT	Set Unit
• (SET T)RACK	Set Track Limits
• (SEC)TOR INCREMENT	Specify Sector Inteleave
• (I)NTERRUPT	Set Interrupt Status
• (DE)NSITY LOCKUP	Lock Density to Current Density
• (SET D)EVICE	Set Device
• (H)ELP	Output List of Commands
<b>Program Status</b>	
• (M)AP ADDRESS	Memory and Device Map
• (ST)ATUS	Display Status Information
• (RES)ET STATUS	Change Status
• (SA)VE STATUS	Save Status on Diskette
• (DUMP C)IR BUFFER	Display Circular Output Buffer
• (REC)OVER STATUS	Retrieve
<b>Data Utilities</b>	
• (DUP)LICATE	Duplicate
• (CO)MPARE	Compare by Sector
• (DUMP O)CTAL	Data Dump in Octal Format
• (DUMP B)YTE	Data Dump in Byte Format
• (DUMP A)SCII	Data Dump in ASCII Format

FLPEXR has several restart addresses that can be used to restart the program if necessary. They are:

1104	—Normal start-restart address
1110	—Start address from monitor call
1114	—Start at command prompt, without performing INIT on device
1100	—Return address from ODT after CTRL D dispatch

## PROGRAM INPUT/OUTPUT

All data input and output is in octal format unless otherwise specified.

The 'DEL' or 'RUB' key may be used during input to remove the previously input character. On some output devices, the cursor will be backspaced one position for each 'DEL'; on other devices, a '/' will be output followed by the characters being deleted. Normal input may be resumed at any time.

**Table 2.** Control Inputs

<i>Input</i>	<i>Meaning</i>	<i>Notes</i>
CTRL R	Aborts current test, restarts at command	
CTRL S	Freezes terminal output until another character is typed	
CTRL O	Throws away all output until another character is typed	
CTRL P	Throws away all output except errors until another character is typed	
CTRL Q	Causes output to resume	1
<LF>	Types current track and sector and status counts	4
CTRL C	Asks 'EXIT TO RT-11?' if RT-11 monitor is available. Type Y to exit. If RT-11 monitor not available, action is similar to CTRL R. If in ODT, may return control to program	
CTRL D	Causes control transfer to ODT	2,3
CTRL T	Causes control transfer to ODT with stack trace	2,3
CTRL L	Toggles extended error printout formats	
RUB or DEL	Deletes previous character in input string	

### Notes:

1. Actually, any character being input will perform this function.
2. Exit to monitor and control transfer to debug may not function if there is not enough memory available or if booted from a device other than 177170.
3. Control transfer from ODT back into FLPEXR is accomplished by 'CTRL C'. If this does not work, the program may be restarted by 'XXXX'G, where XXXX is the appropriate restart address (see below).
4. This command always functions; however, for some tests, the track and sector information should be disregarded (e.g., fill-empty test).

The program fully supports X-on, X-off protocol (i.e., CTRL S, CTRL O and CTRL O) to enable output to be suspended and restarted.

Diskette data is accessed via a combined address unit #, side #, track #, and sector #. Various commands are provided to specify the limits of the address components to be used for tests. These values are set to default values when the device type is selected following initial program load.

Input is typically terminated by either a <CR> or <SP>. Validation input (e.g., Y or N) typically does not require termination.

## PROGRAM STATUS AND ERROR DISPLAYS

FLPEXR types out error and status information under a wide variety of circumstances. All printouts to the console terminal are sent to a circular buffer in memory as well. The buffer size is determined by available memory. The circular buffer is useful if a hard copy console terminal is not being used and error printouts no longer on the face of the CRT screen need to be examined. The display output buffer (DUMP C) function is used to examine messages in the circular buffer. The status

variables that might appear on the console terminal are explained below:

DEV XXX	Is printed only when running multiple controllers. XXX are the last 3 octal digits of the RXCS address for the system whose error/status data is being displayed.
UN U	U represents the logical drive unit number for which the error/status data is being displayed.
TRACK = TK	Track address at time of status/error printout.
SECTOR = SC	Sector address at the time of status/error printout.
RXCS = XY	Shows the contents of the command and status register.
RXDB = XY	Shows the contents of the data buffer register. It should normally be 0 or 214 octal following an INIT.
INTERRUPT ERROR: X	If X is less than 0, this indicates that an expected interrupt failed to occur. If X is greater than 0, this indicates that more than one interrupt occurred.
#BAD = XX	This variable indicates the number of status errors detected.
#RD/WRT = XX	This variable indicates the number of sectors that were transferred error-free.
#XFERS = XX	This variable indicates the number of fill/empty command cycles that were completed successfully.
B-DATA = XX	Number of data errors where a byte or word of data did not compare with the value the program was expecting. This is different than a CRC error, which would be

DEFSTT = DEFINITIVE  
ERROR STATUS

SIDE 1

counted as bad status. There can be up to 128 data errors in 1 sector.

Error code associated with the error currently being displayed. The meaning of each error code can be found in the unit users manual.

Indicates an error has occurred on side 1 (second side of a diskette). Error messages not specifying side 1 relate to side 0. Single sided products display only side 0.

## EXPANDED ERROR DISPLAYS

If in RX02 compatible mode, and CTRL L has been typed to select expanded error printout mode, the following additional status variables appear in the error printout:

D0@TK = TK	Track address of drive 0
D1@TK = TK	Track address of drive 1
CURTK = TK	Track address of the current selected logical unit
CSCT = SC	Sector address of the current selected logical unit
DSTT = XX	Drive status byte—each of the bits in this status byte is used to encode some information about one or both of the flexible disk drives and/or the media presently installed. The bits get decoded into words which are displayed with the other status. These words are explained below.
US0	Drive 0 is currently selected
US1	Drive 1 is currently selected
DN0L	Drive 0 currently contains a single density diskette
DN0H	Drive 0 currently contains a double density diskette
DN1L	Drive 1 currently contains a single density diskette
DN1H	Drive 1 currently contains a double density diskette
HDUP	Head on currently selected unit is up (unloaded)
HDLN	Head on currently selected unit is loaded

TRKRD = TK Track address read from a sector header. This number would only be useful following a DEFSTT = 150 error.

DEF - RXDB = XX Contents of the RXDB following a definitive error status command.

## ERROR ACTIVITY CODES

A number of 2-character activity codes are displayed in the context of error printouts. The codes listed below indicate what the diagnostic was doing when the error was detected.

<i>Activity</i>	<i>Code</i>	<i>Meaning</i>
FILL-EMPTY	FB	Problem loading sector buffer
FILL-EMPTY	E1, E2	Sector buffer data did not check during an empty buffer operation
FILL-EMPTY	FL, EL	DMA fill or empty error to low mem. buffer
FILL-EMPTY	FD, ED	DMA fill or empty error to cir. mem. buffer
FILL-EMPTY	FH, EH	DMA fill or empty error to high mem. buffer
SEQ. WRITE	SW, CW	Problem during sequential write
SEQRD	SR	Problem during sequential read
RANDOM	RW, RC, RR	Random (write, check, read) activity when error was detected
ANY READ RETRY	XE	Empty buffer check before retrying read
DUP UTILITY	IN	Error reading the source diskette
DUP UTILITY	CW	Error checking what was just written
DELETED DATA	DW, DR	Deleted data flag failure

## EXAMPLES OF ERROR OUTPUT

The following printouts are examples of what the FLPEXR diagnostic program outputs to the console under varying circumstances.

- EXAMPLE 1: Operator requests status of currently selected drive during a test by typing LF.
- ```
UN 0 TRACK=0 SECTOR=4
BAD=0 RD/WRT=0 XFERS=0
B-DATA=0
```
- EXAMPLE 2: Operator requests status of both drives using the "STATUS"
- ```
UN 0 BAD=0 RD/WRT=0
XFERS=0 B-DATA=0
UN 1 BAD=0 RD/WRT=0
XFERS=0 B-DATA=0
```
- EXAMPLE 3: Disk was write protected. Error detected on drive #1 at track #1, sector #1 error code was 100
- ```
#BAD=1 #RD/WRT=2002
#XFERS=0 B-DATA=0
```
- EXAMPLE 4: Read on drive with no disk installed. Error detected on drive #0 at track #1, sector #11 error code was 110
- ```
#BAD=3 #RD/WRT=2049
XFERS=0 B-DATA=0
```

## COMPREHENSIVE TEST COMMANDS

### • VERIFY—(V)ERIFY

The VERIFY test does one pass of a SHORT ACCEPTANCE TEST, on the first 7 tracks and then resets the limit variables back to the normal default values. It then induces an automatic "CTRL P" to inhibit all but error printout and initiates the long verify test. This test will run until terminated by a "CTRL R."

#### EXAMPLE

```
#DD COMMAND : VERIFY
SCRATCH DISKS INSTALLED? (Y, N) : Y
SET DENSITY TO (S, D) : S
ARE YOU SURE? (Y, N) : Y
VERIFY TEST NOW STARTING
SCAN CRC CHECKED WRITING READING
INTERRUPTS ENABLED
WRITING READING
```

### • SHORT VERIFY—(SH)ORT VERIFY

This interactive program changes the track range used by the VERIFY TEST so that only the first 9 tracks of each selected drive are tested. This test will run until terminated by a CTRL R.



## INDIVIDUAL TESTS

- SCAN—(SC)AN

The SCAN test reads all sectors on all selected drives sequentially and checks for CRC errors. It also determines media density. No direct data checking takes place in this test. Only status is checked. After all units are scanned once, the "COMMAND:" prompt is displayed on the console.

### EXAMPLE

```
#COMMAND: SCAN
CRC CHECKED
#COMMAND:
```

- SEEK RANGE—(SE)EK RANGE

The SEEK RANGE function is a versatile drive test that performs all possible seeks within the operator specified track and seek length boundaries. It specifies a read on the first sector that can be read on the destination track after compensating for step and head load times. Thus it is a worst case test of the drive stepper motor and head setting. Status information will be continuously displayed during execution of this test indicating the seek length currently being used ( x ) and direction of seek ( [ ^ ] = outward). An '!' will be output at the conclusion of each pass. This test will run continuously until terminated by a CTRL R.

### EXAMPLE

```
#DD COMMAND SEEK RANGE
NOTE: ALL TIMES ARE GIVEN IN 'OCTAL'
      TENTHS OF MSEC
SEEK LENGTH ( 1 ): 3 THROUGH ( 27 )
: 7
850 SEEK TIME ( 36 ):
850 SECTOR OFFSET: ( 4 ):
COVERING TRACKS ( 1 ): THROUGH
( 114 ): [ 3 ][ ^ ][ 4 ][ ^ ][ 5 ][ 6 ]
[ ^ ][ 7 ][ ^ ] ! [ 3 ][ ^ ][ 4 ][ ^ ]...
```

- FILL-EMPTY—(FI)LL EMPTY

The FILL-EMPTY test checks the FILL BUFFER and EMPTY BUFFER controller commands. If the controller under test is configured in the RX01 compatible mode, then the test involves only programmed I/O. If the controller is configured as an RX02, the controller does FILL/EMPTIES into three different buffers so as to verify proper operation of all possible address bits. FILL/EMPTIES are done in both densities covering all possible word counts. Since this test does not manipulate the drives, the system will oper-

ate in silence. This test continues until the operator types a 'CTRL R'.

- SEQUENTIAL WRITE/READ—(SEQW)/R

The SEQUENTIAL WRITE / READ test writes pseudo-random data sequentially on all selected drives. The test then reads all the data and checks it. The message "WRITING" is typed on the console terminal when the test first starts writing. The message "READING" is typed when the test starts reading. This test continues until the operator types "CTRL R". It also performs a set media density operation if the diskette is not of the expected density.

---

### Note

---

*The following three tests require a SEQUENTIAL WRITE pass be done first in order to initialize the pseudo-random data. Data compare errors are reported if this is not done. FLPEXR prompts 'IS DISKETTE SEQUENTIAL WRITTEN? (Y, N)' at the start of each test. A 'Y' response will initiate the test; a 'N' response will return to the command prompt.*

---

- SEQUENTIAL READ—(SEQ) READ

The SEQUENTIAL READ test reads the data on all selected drives sequentially and compares the data pattern against what was written. The program types "READING" at the beginning of each pass. This test continues until the operator types "CTRL R".

- RANDOM READ/WRITE—(RA)NDOM R/W

The RANDOM READ/WRITE test selects a random sector of a selected drive, then reads or writes it. It checks data when appropriate. This test continues until the operator types "CTRL R".

- READ RANDOM—(REA)D RANDOM

The READ RANDOM test reads randomly selected sectors. Data is checked following each read. This test continues until the operator types "CTRL R".

## MEDIA MODIFICATION COMMANDS

- REFORMAT—(FO)RMAT

This function is used to rewrite diskette headers, as well as all the other data on a particular diskette. It also prompts for confirmation, unit, and sequential or interleaved format. Either the entire diskette (Formats 2 through 8) or just a portion of the diskette (Format 0 through 1) may be formatted. If a partial format is selected, the track range to be formatted is specified by the set track command. The sides to be formatted can also be specified.

## EXAMPLE (for 480)

```

#COMMAND: FORMAT
SEQUENTIAL SECTOR FORMAT?
(Y OR N) : Y

Density                Type  Supported
                        On
DEC SD (IBM SD 2-128)  0    480
                        440
                        210
                        110
DEC DD                  1    480
                        440
DEC SD (ALL OF DISK)   2    880, 480,
                        470, 430,
                        4140
DEC DD (ALL OF DISK)   3    880, 480,
                        470, 430,
                        4140
IBM SD (92-256)        4    480
IBM SD (2-512)         5    480
IBM DD (2D-256)        6    480
IBM DD (2D-512)        7    480
IBM DD (2D-1024)       8    480
DESIRED SELECTION? (0 to 8) : 4
DO YOU WISH TO DO SIDE #0? (Y OR N) : Y
DO YOU WISH TO DO SIDE #1? (Y OR N) : Y
ARE YOU SURE? (Y OR N) : Y
# COMMAND:

```

FLPEXR is designed to support the full range of formats available throughout the product line. However, not all units are capable of writing all formats. If an inappropriate format is selected, an error message will be output. If the unit is not capable of IBM format modes, they will not be output in the selection menu.

Typically, the operator should format new diskettes by Formats 2 for single density diskettes and 3 for double density diskettes.

- SET MEDIA DENSITY (SET M)EDIA DENSITY

This function enables the operator to initialize a diskette to single density or double density format. The function prompts for function confirmation, unit, and desired density. To select single density, respond with an "S". Type "D" to select double density.

The SET MEDIA DENSITY command is used to implement this function, therefore, no headers are rewritten. The prompt is issued when this function is complete. This function causes any status saved on track 0, sector 1 to be erased.

```

#COMMAND: SET MEDIA DENSITY
DO A SET MEDIA ON ALL DEVICES? (Y OR N) :
N
UNIT: 1: SET DENSITY TO (S,D) : S
ARE YOU SURE? (Y, N) : Y

```

## PROGRAM CONTROL VALUE COMMANDS

- SET UNIT—(SET U)NIT

This command enables the operator to specify which drives are to be accessed by the various test functions. The default drives are units 0 and 1. The currently selected units are printed first. It prompts with "UNIT:", expecting a number between 0 and 3, inclusive. Unit numbers are accepted as long as they are valid. When a non-number is typed to a unit request, the units currently selected are prompted and FLPEXR returns to command prompt.

---

### Note

- 1 If using a two drive system, then selection of units 2 and 3 is invalid and may cause an error.
  - 2 If units are selected by "SET DEVICE", they will override "SET UNIT". See the "SET DEVICE" command for more information.
-

## EXAMPLE

```

"SET DEVICE" overriding "SET UNIT"
#DD COMMAND: SET UNIT
— LOADED BY SET DEVICE FLAGS
  UNITS SELECTED 1

```

- SET TRACK—(SET T)RACK

This command enables the operator to specify lower and upper track limits for all other test functions. The default lower track limit is track 1 and upper track limit is track 76. The "COMMAND" prompt is issued after the entry of valid new limits. The lower limit must not exceed the upper limit.

## EXAMPLE

```

"SET TRACK" used to set track range from
track 1 to track 10
#COMMAND: SET TRACK
FROM 1: THROUGH 14: 10

```

- SECTOR INCREMENT—(SEC)TOR INCREMENT

This command enables the operator to specify the sector increment value. The number is added to the present sector address to determine the next sector address in the functions that read multiple sectors on a single track. If this number were 1 and the diskette did not have an interleaved format, an entire revolution would be required to read each sector. On LSI-11 processors, the default increment value is 3. On PDP-11 processors, the default increment value is 2. The "MODE:" prompt is issued after the new value has been entered.

```

#DD COMMAND: SECTOR INCREMENT
= 3 - 2
#DD COMMAND: SECTOR INCREMENT
= 2 - 3

```

- SET INTERRUPT STATUS—(I)NTERRUPT

The SET INTERRUPT STATUS command enables the operator to test the disk system with interrupts either enabled or disabled. If interrupts are enabled, the FLPEXR ensures that an interrupt occurs whenever it is appropriate. The operator enters a D to disable interrupts and an E to enable interrupts. This function is also used in ACCEPTANCE and VERIFY to set "Interrupts Enabled" and "Interrupts Disabled".

## EXAMPLE

```

#DD COMMAND: INTERRUPT
CURRENTLY INTERRUPTS ARE DISABLED
(D)
INPUT NEW STATUS (ENABLE OR
DISABLE)
(E OR D) : D

```

- DENSITY LOCKUP—(DE)NSITY LOCKUP

The "DENSITY LOCKUP" function allows the operator to lock the current disk density during the various tests. This feature is useful when testing for a problem that occurs in one density only, or when the disk density cannot be changed by a SET MEDIA DENSITY function.

## EXAMPLE

```

#DD COMMAND: DENSITY LOCKUP
DENSITY IS CURRENTLY UNLOCKED
DO YOU WISH TO LOCK THE DENSITY (Y
OR N): Y
#DD COMMAND:

```

- SET DEVICE—(SET D)EVICE

This function facilitates testing controllers that are not configured at the standard device I/O address and interrupt vector. It also enables the FLPEXR test program to simultaneously exercise multiple controllers. The function protocol asks you for device address, interrupt vector, and flag word. If a space is typed, the program steps past that field, leaving it intact. Return to "COMMAND:" is by input of a "CR" (carriage return) in response to "RXCS:". The flag word is organized as follows:

15	14	13	12	11	10	09	08
				DMA	D85	DBS	DDN
07	06	05	04	03	02	01	00
	US3	US2	US1	US0			

When set to a 1, the bit labeled:

DMA indicates the device should be tested as an RX02.

D85 indicates 850 timing should be used (else 800 timing).

DBS indicates that double sided operation is enabled.

DDN indicates double density operation is enabled.

US3 indicates this device contains a drive unit 3.

US2 indicates this device contains a drive unit 2.

US1 indicates this device contains a drive unit 1.

US0 indicates this device contains a drive unit 0.

US0, US1, US2, US3 do an implicit "SET UNIT" function when set. The normal flag variable for RX02 mode is 4400 (octal). The normal flag variable for RX01 is 0000 (octal). The normal flag for double sided RX02 operation is 7400 (octal).

## EXAMPLE SET DEVICE

```
#COMMAND: SET DEVICE
SET THE DEVICE FLAGS FOR EACH SYSTEM AS FOLLOWS:
  4000: ENABLES DMA OPERATION IF AVAILABLE
  2000: SETS 850 TIMING (ELSE 800)
  1000: ENABLES DOUBLE SIDED OPERATION IF DOUBLE SIDED DRIVE AND DISK USED
  400:  ENABLE DENSITY SWITCHING IF RX02/440/480
  20:   ENABLE UNIT #1 ON CURRENT DEVICE
  10:   ENABLE UNIT #0 ON CURRENT DEVICE
RXCS @ 177170: INT @ 264 INTVEC = 264
FLAGS: 4400 641Q
RXCS @ 0:
```

- HELP

The HELP command causes all the valid "MODE:" responses to be displayed on the console terminal. The "MODE:" prompt is typed when this function is complete.

## PROGRAM STATUS COMMANDS

- MAP ADDRESS—(M)AP ADDRESS

The MAP ADDRESS command causes a memory and device address map of the system to be displayed on the console terminal. This is the same map displayed when the FLPEXR program is first loaded. In addition, the interrupt vector address associated with each disk interface is displayed. The "COMMAND:" prompt is typed when this function is complete.

## EXAMPLE

```
#DD COMMAND: MAP ADDRESS
( 0 - 157776)
( 160100 - 160106)
( 165000 - 165776)
( 171000 - 171776)
( 172300 - 172316)
( 172340 - 172356)
( 172520 - 172536)
( 173000 - 173776)
( 176700 - 176746)
( 177170 - 177172)
( 177510 - 177516)
( 177546 - 177546)
( 177560 - 177616)
( 177640 - 177656)
( 177776)
DEV: 177170 INT @ 264
```

## Note

*This example indicates that a device is installed at location 177170 with interrupt vector at location 264.*

- STATUS—(S)TATUS

The STATUS function causes all the current status information including hardware errors, data errors, and pass counts to be displayed on the console terminal. Displaying status information does not reset the status counts. The "COMMAND:" prompt is typed when this function is complete.

## EXAMPLE

```
#COMMAND: STATUS
UNIT #0 #BAD = 3 #RD/WRT = 2049
#XFERS = 0 B - DATA = 0 ST = 110 # = 3
```

- RESET STATUS—(RES)ET STATUS

The RESET STATUS function first displays all the available status counts. Next, the display will ask whether all of the status counts need resetting. A "Y" will cause all of the error, pass, etc. counts to be reset to zero. The "COMMAND:" prompt is output when this function is complete.

- SAVE STATUS—(SA)VE STATUS

The SAVE STATUS command causes all the status counts associated with a particular drive to be written on track 0, sector 1 of the diskette in that drive. Only the SET MEDIA DENSITY commands overwrite track 0, so the status data associated with each drive can be safely stored away. This function is used by the acceptance test so that it can survive a loss of main computer CPU memory without any loss of cumulative error data. The "COMMAND:" prompt is typed when this function is complete.

- RECOVER STATUS—(REC)OVER STATUS

The RECOVER STATUS routine performs the opposite function performed by the SAVE STATUS function. The status data stored away on track 0, sector 1 of the diskette in each drive is transferred back from the diskette to the status/counter variables in memory. The "COMMAND:" prompt is displayed when this function is complete.

- DISPLAY CIRCULAR OUTPUT BUFFER—(DUMP C)IR BUFFER

The DUMP C function is used to display the output buffer associated with all console terminal output. This function is useful on systems where the console terminal is CRT. Messages previously output can be re-examined on the console. The buffer can be cleared after it is displayed by this function.

**DATA UTILITIES COMMANDS**

---

*Note*

---

*The SECTOR INCREMENT function may be used to specify sector sequencing for the duplicate and compare commands. For the dump commands, a sector increment of 1 is always assumed.*

---

- **DUPLICATE—(DUP)LICATE**

The DUPLICATE command enables the operator to make a duplicate copy of a diskette. The function prompts for a source drive unit number and a destination drive unit number. For each possible sector address, the function performs a READ SOURCE SECTOR, WRITE DESTINATION SECTOR, READ DESTINATION SECTOR, and COMPARE DATA.

EXAMPLE

```
#DD COMMAND: DUPLICATE
SOURCE UNIT: 0
TO DESTINATION UNIT: 1
#DD COMMAND:
```

- **COMPARE—(CO)MPARE**

The COMPARE command enables the operator to compare two diskettes starting at a specific address. The function prompts for: SOURCE UNIT, STARTING TRACK, STARTING SECTOR, NUMBER OF SECTORS, and DESTINATION UNIT. Any differences in data will be output.

EXAMPLE

```
#DD MODE: DUMP OCTAL
SOURCE UNIT: 0 TRACK: 0 SECTOR: 1 #
SECTORS: 2
[DDEN DRIVE #0 AT TRACK 0, SECTOR 1,
SIDE 0]
SC = 1
```

- **OCTAL DUMP BY SECTORS—(DUMP O)CTAL**

This command enables the operator to cause an octal dump of specified sectors to the console terminal. The function prompts for: UNIT, STARTING TRACK, STARTING SECTOR, SIDE, and NUMBER OF SECTORS.

```
0: 00037760000
20: 00000000
40: 00000000
60: 00000000
100: 00000000
120: 00000000
140: 00000000
160: 00000000
200: 00037220000
220: 00000000
240: 00000000
260: 00000000
300: 00000000
320: 00000000
340: 00000000
360: 00000000
```

[DDEN DRIVE #0 AT TRACK #0, SECTOR #2 SIDE #0]  
SC = 2

```
0: 00000000
20: 00000000
40: 00000000
60: 00000000
100: 00000000
120: 00000000
140: 00000000
160: 00000000
200: 00000000
220: 00000000
240: 00000000
260: 00000000
300: 00000000
320: 00000000
340: 00000000
360: 00000000
```

- **BYTE DUMP BY SECTORS—(DUMP B)YTE**

This command enables the operator to cause an octal dump of specified sectors to the console terminal. The function prompts for: UNIT, STARTING TRACK, STARTING SECTOR, SIDE, and NUMBER OF SECTORS.

- **ASCII DUMP BY SECTORS—(DUMP A)SCII**

This utility command enables the operator to cause an ASCII dump of specified sectors to the console terminal. The function prompts for: UNIT, STARTING TRACK, STARTING SECTOR, SIDE, and NUMBER OF SECTORS.