

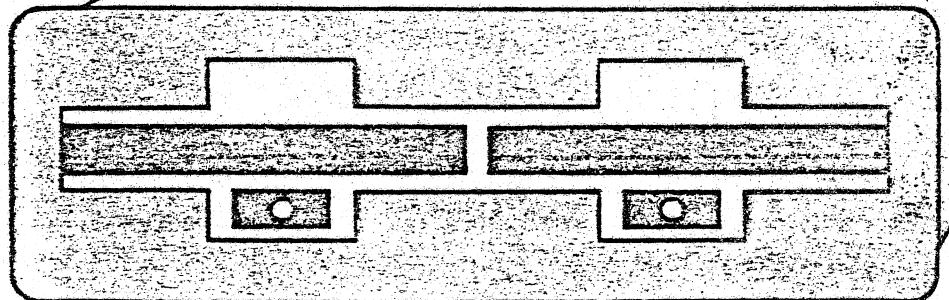
# Data Systems



# DSD-440

## FLEXIBLE DISK MEMORY SYSTEM

### User's Guide



**Data  
Systems**

Data Systems Design, Inc. • 3130 Coronado Drive • Santa Clara, CA 95051  
408/249-9353 • TWX 910-338-0249 DSP SNTA



February 23, 1979

NOTE TO READER:

This is a preliminary copy of the DSD 440 USER'S MANUAL. Many of the illustrations are rough sketches, only intended to serve as a guide to the professional graphic artists. We would very much appreciate your comments and suggestions regarding improvements or any errors that you might find. Please feel free to write to DATA SYSTEMS, or phone us at: (408) 249-9353 EXT. 457.

THANK YOU



T. OLSEN

**NOTICE**

The information contained in this manual is subject to change without notice. Data Systems Design makes no commitment to update nor to keep current the information contained in this manual. Data Systems Design assumes no responsibility for any errors that may appear in this manual. Data Systems Design makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

## CONTENTS

-----  
PREFACE

CHAPTER 1	GENERAL INFORMATION
1-1	PRODUCT INTRODUCTION
1-2	PURPOSE OF EQUIPMENT
1-3	DISKETTE SYSTEM DESCRIPTION
1-4	FEATURES
1-5	SPECIFICATIONS
CHAPTER 2	OPERATING MODES AND SYSTEM CONFIGURATION
2-1	OPERATIONAL MODES
2-2	DSD 440 SYSTEM CONFIGURATION
2-2.1	FLOPPY DISK DRIVES
2-2.2	MASTER CONTROLLER PCB
2-2.3	DC POWER SUPPLY
2-2.4	POWER DISTRIBUTION PCB ASSEMBLY
2-2.5	AC POWER SWITCH
2-2.6	FAN
2-2.7	CORCOM CONNECTOR AND FUSING
2-2.8	INTERFACE CABLE
2-2.9	INTERFACE MODULE
CHAPTER 3	SYSTEM INSTALLATION AND ACCEPTANCE
3-1	ENVIRONMENTAL CONSIDERATIONS
3-2	UNPACKING THE SYSTEM
3-3	DMOUNTING THE DSD 440 CHASSIS
3-4	INSTALLING THE INTERFACE MODULE AND CABLE
3-5	INPUT POWER CONSIDERATIONS
3-6	CHANGING THE OPERATING MODE
3-7	FINAL INSTALLATION
3-8	POWERING UP
3-9	INITIALIZATION RESPONSE CHECK
3-10	SYSTEM BOOTSTRAP
3-11	ACCEPTANCE TESTING ON PDP-11 AND LSI-11
3-12	ACCEPTANCE TESTING ON PDP-8
CHAPTER 4	MAINTENANCE FEATURES
4-1	OVERVIEW OF MAINTENANCE FEATURES
4-2	NORMAL VS. "HYPER-DIAGNOSTIC" MODE
4-3	INDICATOR LEDES, DRIVE LEDES, DIP-SWITCH
4-4	LED MEANINGS DURING "NORMAL" MODE
4-5	DIP-SWITCH SETTINGS DURING "NORMAL" MODE
4-6	HARDWARE SELF-TEST ROUTINES
4-7	"HYPER-DIAGNOSTIC" MODE
4-8	DIP-SWITCH SETTINGS DURING "HYPER-DIAGNOSTIC" MODE
4-9	LED MEANINGS DURING "HYPER-DIAGNOSTIC" MODE
4-10	DESCRIPTION OF "HYPER-DIAGNOSTIC" TESTS
4-11	HOW TO READ THE INDICATOR LEDES

CHAPTER 5

DSD 440 PROGRAMMING INFORMATION  
DEC 11 FAMILY PROGRAMMERS' INTERFACE  
MODE 1 OPERATION

5-1

5-1.1

5-1.1.1

5-1.1.1.1

5-1.1.1.2

5-1.1.1.3

5-1.1.1.4

5-1.1.1.5

5-1.1.1.6

5-1.1.2

5-1.1.2.1

5-1.1.2.2

5-1.1.2.3

5-1.1.2.4

5-1.1.2.5

5-1.1.2.6

5-1.1.2.7

5-1.1.2.8

5-1.1.2.9

5-1.1.3

5-1.1.3.1

5-1.1.3.2

5-1.1.3.3

5-1.1.3.4

5-1.1.4

5-1.2

5-1.2.1

5-1.2.1.1

5-1.2.1.2

5-1.2.1.3

5-1.2.1.4

5-1.2.1.5

5-1.2.1.6

5-1.2.1.7

5-1.2.2

5-1.2.2.1

5-1.2.2.2

5-1.2.2.3

5-1.2.2.4

5-1.2.2.5

5-1.2.2.6

5-1.2.2.7

5-1.2.2.8

5-1.2.2.9

5-1.2.2.10

5-1.2.3

5-1.2.3.1

5-1.2.3.2

5-1.2.3.3

5-1.2.3.4

5-1.2.4

PERIPHERAL DEVICE REGISTERS  
COMMAND AND STATUS REGISTER  
DATA REGISTER  
DATA BUFFER REGISTER  
FLOPPY DISK TRACK ADDRESS  
FLOPPY DISK SECTOR ADDRESS  
SYSTEM ERROR AND STATUS REGISTER

MODE 1 PROTOCOLS  
FILL SECTOR BUFFER  
EMPTY SECTOR BUFFER  
WRITE SECTOR  
READ SECTOR  
READ STATUS  
WRITE DELETED DATA SECTOR  
READ ERROR REGISTER  
POWER FAIL OR INIT COMMAND  
DISKETTE FORMATTING

TYPICAL SEQUENCES OF OPERATIONS  
READ/WRITE BUFFER  
READ/WRITE/WRITE D. D.  
STATUS READ  
COMMON PROGRAMMING PITFALLS

INTERRUPTS

MODE 2 OPERATION

PERIPHERAL DEVICE REGISTERS  
COMMAND AND STATUS REGISTER  
DATA REGISTER  
FLOPPY DISK TRACK ADDRESS  
FLOPPY DISK SECTOR ADDRESS  
WORD COUNT REGISTER  
BUS ADDRESS REGISTER  
SYSTEM ERROR AND STATUS REGISTER

MODE 2 PROTOCOLS  
FILL SECTOR BUFFER  
EMPTY SECTOR BUFFER  
WRITE SECTOR  
READ SECTOR  
SET MEDIA DENSITY  
READ STATUS  
WRITE DELETED DATA SECTOR  
READ EXTENDED STATUS  
POWER FAIL  
DISKETTE FORMATTING IN MULTIPLE DENSITIES

TYPICAL SEQUENCES OF OPERATIONS  
READ/WRITE BUFFER  
READ/WRITE/WRITE D. D.  
STATUS READ  
COMMON PROGRAMMING PITFALLS

INTERRUPTS



5-2	DEC PDP-8 FAMILY PROGRAMMERS' INTERFACE
5-2.1	INSTRUCTION SET
5-2.1.2	THE TRANSFER DATA REGISTER (XDR) INSTRUCTION
5-2.1.3	THE SKIP ON TRANSFER REQUEST (STR) INSTRUCTION
5-2.1.4	THE SKIP ON ERROR (SER) INSTRUCTION
5-2.1.5	THE SKIP ON DONE (SDN) INSTRUCTION
5-2.1.6	THE INTERRUPT ENABLE/DISABLE (INTR) INSTRUCTION
5-2.1.7	THE INITIALIZE (INIT) INSTRUCTION
5-2.2	REGISTER DESCRIPTIONS
5-2.2.1	COMMAND REGISTER
5-2.2.2	TRACK ADDRESS REGISTER
5-2.2.3	SECTOR ADDRESS REGISTER
5-2.2.4	DATA BUFFER REGISTER
5-2.2.5	ERROR AND STATUS REGISTER (MODE 1 OPERATION)
5-2.2.6	ERROR AND STATUS REGISTER (MODE 2 OPERATION)
5-2.3	FUNCTION CODE DESCRIPTIONS
5-2.3.1	FILL BUFFER
5-2.3.2	EMPTY BUFFER
5-2.3.3	WRITE SECTOR
5-2.3.4	READ SECTOR
5-2.3.5	SET MEDIA DENSITY
5-2.3.6	READ STATUS
5-2.3.7	WRITE DELETED DATA SECTOR
5-2.3.8	READ ERROR REGISTER
5-2.3.9	POWER FAIL
5-2.3.10	DISKETTE FORMATTING IN MULTIPLE DENSITIES

CHAPTER 6	HIGH LEVEL SOFTWARE AND THE DSD 440
6-1	GENERATING AN RT11 SYSTEM DISKETTE
6-2	GENERATING DOUBLE DENSITY DISKETTES
6-3	FRD440 SYSTEM DIAGNOSTIC PROGRAM
6-3.1	PROGRAM LOADING AND MONITOR PROTOCOL
6-3.2	FRD440 PROGRAM FUNCTIONAL MODES
6-3.3	PROGRAM STATUS AND ERROR PRINTOUTS

LIST OF ILLUSTRATIONS

---

FIGURE 1-1	SYSTEM BLOCK DIAGRAM
FIGURE 1-2	SINGLE AND DOUBLE SIDED DISKETTE
FIGURE 1-3	SINGLE DENSITY TRACK FORMAT
FIGURE 1-4	DEC DOUBLE DENSITY TRACK FORMAT
FIGURE 2-1	DSD 440 SYSTEM
FIGURE 2-2	MASTER CONTROLLER PCB
FIGURE 2-3	DC POWER SUPPLY
FIGURE 3-1	CHASSIS SLIDES MOUNTING
FIGURE 3-2	BULLET ENTERING HOLE IN REAR OF UNIT
FIGURE 3-3	FRONT VIEW OF CHASSIS WITH FRONT PANEL REMOVED
FIGURE 3-4	LSI-11 INTERFACE MODULE
FIGURE 3-5	OPTION PRIORITY IN DEC BACKPLANES FOR LSI-11
FIGURE 3-6	PDP-11 INTERFACE MODULE
FIGURE 3-7	INTERRUPT PRIORITY LEVELS
FIGURE 3-8	PDP-8 INTERFACE MODULE
FIGURE 3-9	MASTER CONTROLLER BOARD DIP-SWITCH
FIGURE 3-10	INSERTING A DISKETTE INTO A DRIVE
FIGURE 4-1	SWITCH AND LED ORIENTATION ON MASTER CONTROLLER CARD
FIGURE 4-2	DIP-SWITCH SETTING WHEN PRODUCT IS SHIPPED
FIGURE 4-3	EXAMPLE "HYPER-DIAGNOSTIC" DIP-SWITCH CONFIGURATION
FIGURE 4-4	LED INTERPRETATION EXAMPLES
FIGURE 5-1	COMMAND AND STATUS REGISTER; MODE 1 (RX01 COMPATIBLE)
FIGURE 5-2	REGISTER FORMATS, 11 FAMILY; MODE 1 (RX01 COMPATIBLE)
FIGURE 5-3	
FIGURE 5-4	FILL / EMPTY BUFFER PROGRAM EXAMPLE; MODE 1
FIGURE 5-5	READ / WRITE SECTOR PROGRAM EXAMPLE; MODE 1
FIGURE 5-6	COMMAND AND STATUS REGISTER; MODE 2 (RX02 COMPATIBLE)
FIGURE 5-7	REGISTER FORMATS; 11 FAMILY; MODE 2 (RX02 COMPATIBLE)
FIGURE 5-8	FILL / EMPTY BUFFER PROGRAM EXAMPLE; MODE 2
FIGURE 5-9	READ / WRITE SECTOR PROGRAM EXAMPLE; MODE 2
FIGURE 5-10	COMMAND REGISTER FORMAT; PDP-8; MODE 1
FIGURE 5-11	COMMAND REGISTER FORMAT; PDP-8; MODE 2; 12 BIT WORD
FIGURE 5-12	COMMAND REGISTER FORMAT; PDP-8; MODE 2; 8 BIT WORD
FIGURE 5-13	TRACK ADDRESS REGISTER; PDP-8
FIGURE 5-14	SECTOR ADDRESS REGISTER; PDP-8
FIGURE 5-15	DATA BUFFER REGISTER; PDP-8
FIGURE 5-16	ERROR/STATUS REGISTER; PDP-8; MODE 1
FIGURE 5-17	ERROR/STATUS REGISTER; PDP-8; MODE 2
FIGURE 5-18	FILL BUFFER FLOW CHART; PDP-8
FIGURE 5-19	EMPTY BUFFER FLOW CHART; PDP-8
FIGURE 5-20	WRITE/READ/WRITE DEL. DATA FLOW CHART; PDP-8





## PREFACE

The purpose of this manual is to provide the user with sufficient information to correctly set up and operate the DSD 440 Double Density Floppy Disk Memory System. Included is a detailed description of what the product is, how to install it, how to program it, and how to maintain it. A familiarity with basic data processing terminology and the DEC PDP-11, LSI-11, and PDP-8\* is assumed in some sections.

\* DEC, PDP, UNIBUS, and OMNIBUS are registered trademarks of Digital Equipment Corporation.

## 1-1 PRODUCT INTRODUCTION

The DSD 440 Floppy Disk Memory System is a random access data storage subsystem. Data is stored in fixed-length blocks on preformatted industry standard 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 average sustainable data transfer rate is 16 kilobytes per second. A complete data storage system consists of a rack-mountable controller/drive subsystem, a computer interface module, and an interconnecting cable.

The DSD 440 can be used with the DIGITAL EQUIPMENT CORPORATION PDP-11, PDP-8, or LSI-11 computers using interface modules supplied by DATA SYSTEMS DESIGN. When used with the appropriate interface module, the DSD 440 is hardware, software, diagnostic, and media compatible with the DEC RX02 double density flexible disk system. In addition, the DSD 440 has many features not found in the DEC RX02 system including:

- \* HARDWARE BOOTSTRAP
- \* SELF-CONTAINED DIAGNOSTICS AND UTILITY PROGRAMS
- \* DISKETTE FORMATTING

The DSD 440 can be used with other types of computers if a user designs an interface module that conforms to the DSD 440 Interface Bus Specification.

## 1-2 PURPOSE OF EQUIPMENT

A Floppy Disk Memory System consists of a set of mechanical and electronic components arranged for the purpose of passing data between a host computer and a flexible disk or, as IBM calls it, a "diskette". Due to their low cost and convenient packaging, flexible disks have become a very popular transportable data storage media for many applications. This popularity will be further increased by flexible disk systems such as the DSD 440 which provide increased capacity and performance over first generation flexible disk systems at only a slight increase in cost.

The DSD 440 was designed to provide a highly reliable, low cost and compact flexible disk system that is totally compatible with the DEC RX02. While satisfying these design goals, advances in LSI circuitry and microprocessors have made possible the incorporation of a number of "BONUS" features. Important among these features are:

- \* The ability to execute test and utility programs on the controller/drive subsystem even when the controller and drives are not connected to a host computer.

- \* The ability to write-format diskettes in two industry standard formats.

High reliability is attained by using LSI circuits, by burning-in and pretesting all components, and by using field-proven Shugart Associates flexible disk drives. Low cost is attained by a design which minimizes parts count and assemblies. Compactness is provided by packaging two disk drives horizontally in a 5 1/4" high rack-mountable chassis. Figure 1-1 shows the System Block Diagram.

42.381 50 SHEETS 5 SQUARE  
42.382 100 SHEETS 5 SQUARE  
42.383 100 SHEETS 5 SQUARE  
42.384 100 SHEETS 5 SQUARE  
NATIONAL

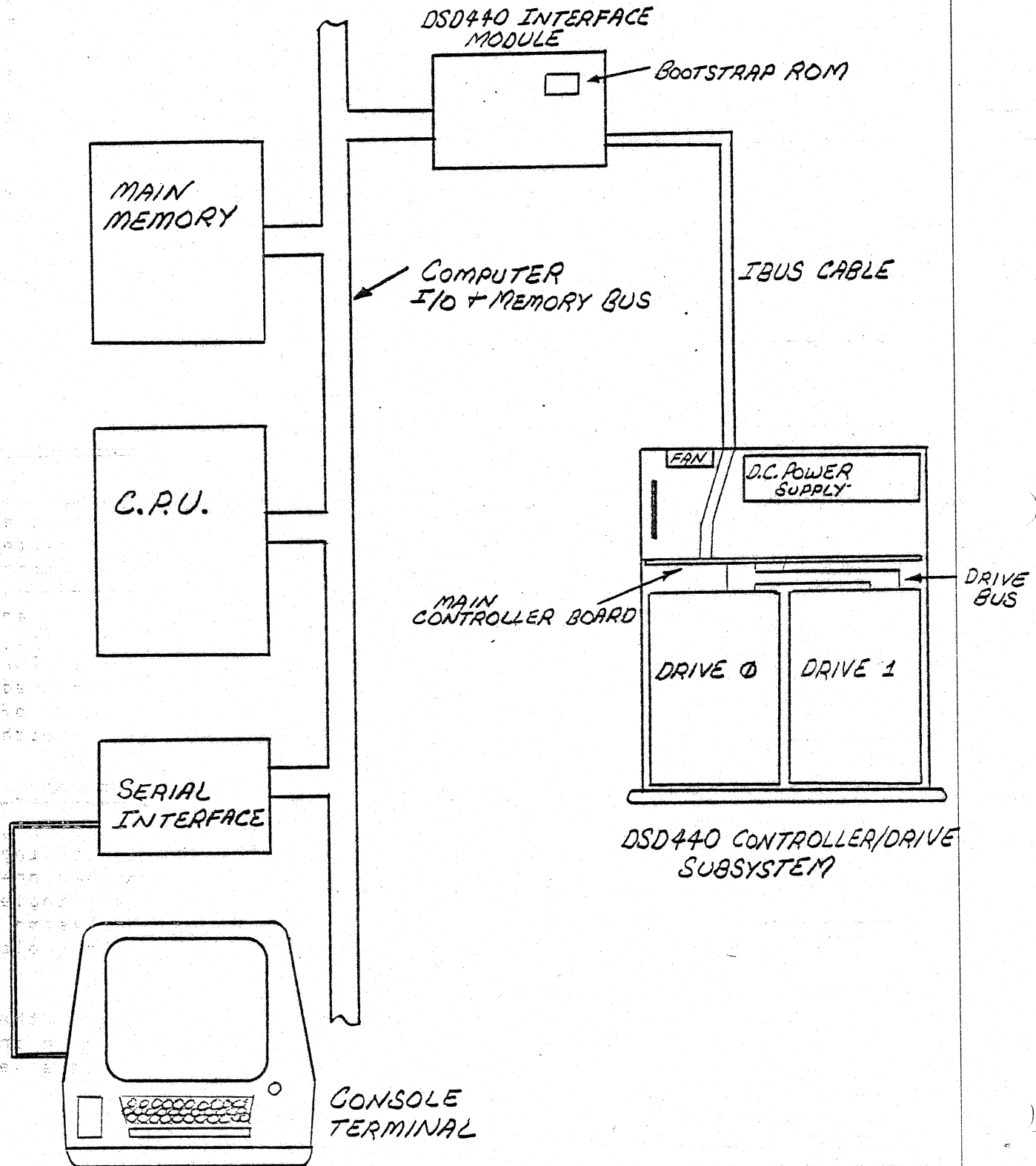


FIGURE 1-1 SYSTEM BLOCK DIAGRAM



## 1-3 DISKETTE SYSTEM DESCRIPTION

A flexible disk is an oxide-coated mylar disk, 7.8 inches in diameter, and .005 inches thick. It is permanently housed in an 8-inch-square flexible envelope. The flexible disk rotates inside the envelope at 360 RPM whenever the diskette is inserted into an operating drive and the drive door is closed. In standard IBM single density format, a diskette can store up to 256,256 bytes of data. In DEC double density format, a diskette can store 512,512 bytes of data. It is important to realize that there is no physical difference between a diskette which contains single density data and one which contains DEC format double density data. The only difference is the data encoding method that is used to record the user data bytes on the diskette. Single density data is encoded using a technique known as "double frequency recording" while double density data is encoded using a technique known as "modified frequency modulation" or MFM, for short. The very same floppy disk drive can write data using both of these data encoding techniques with no problem. Diskettes can commonly differ in the following ways:

- 1) Intended for double or single sided drives
- 2) Hard sectored or soft sectored
- 3) Write protect notch available?

Initially, the DSD 440 will be shipped with the single-sided Shugart model 800 flexible disk drive. It is important to use only diskettes intended for single-sided drives, such as the one shown in the lower half of Figure 1-2. If a diskette intended for a double-sided drive is accidentally loaded into a single sided drive, the photo-sensor will not line up with the index access hole. As a result, the controller will never see an index pulse and will conclude that the drive is simply not ready. Be careful not to try and use an IBM Diskette 2D, Part No. 1766872. These diskettes are explicitly intended for use on double-sided drives. In addition, the format of the recorded data on these diskettes is not compatible with the DEC double density format used on the DSD 440.

The DSD 440 requires the use of soft sectored diskettes. To determine if you have a soft or hard sectored diskette, simply rotate the mylar disk inside the envelope while looking through the index access hole. If you observed more than one hole punched in the mylar disk in the course of a single revolution (maybe 27 holes) you have a hard sectored diskette. Soft sectored diskettes should have only one index hole punched in them.

The need for a write protect notch is completely up to the user. If you have diskettes with this notch, as shown in Figure 1-2, you must cover the notch with an opaque adhesive tab when you want to write on the diskette.

Flexible disk systems are ideal for applications that require a low cost, medium speed, random access memory device. Flexible disk systems provide faster access times than magnetic tapes, and cost less than the bigger rigid disk systems. They can replace paper tape or punch card storage methods. Because the diskettes can be quickly removed, the amount of data that is immediately accessible with a flexible disk system is far greater than the capacity of a single diskette. Diskettes can be exchanged in a drive within seconds, the limiting factor being the dexterity of the user.

Unlike rigid disks, industry standards have been established for the physical format of the recorded data on diskettes. Each of the formats possible on the DSD 440 record data on 77 concentric tracks, at a track density of 48 tracks per inch. Each track is divided up into 26 sectors. Each sector contains 128 bytes of user data in single density format, and 256 bytes of user data in double density format. Associated with each sector is an ID field and a data field. The ID field contains a unique bit pattern, known as the ID address mark, that enables the controller to recognize the start of an ID field. This ID field contains a track address byte, a head address byte, and a sector address byte. Appended to these disk address bytes is a pair of CRC (cyclic redundancy check) bytes which are used to determine if a data error has occurred while reading the disk 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 exactly 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. Following this mark are the 128 or 256 bytes of data and another pair of CRC check bytes. Figure 1-3 is a schematic representation of the format of a single density track. Figure 1-4 shows the format of a DEC compatible double density track. Note that only the 256 user data bytes and the 2 CRC bytes following the data are encoded using the "modified frequency modulation" recording technique. All the other fields (preamble, postamble, ID) are recorded the same as in the single density track format.

TECHNICAL COMPATIBILITY NOTE:

The "modified frequency modulation" encoding algorithm which DEC chose to use when implementing the RX02 is not exactly the same as the MFM encoding algorithm one would find described in a communications theory textbook. For this reason (and others), both the RX02 and DSD 440 are not compatible with the IBM double density recording technique which uses a "textbook" MFM encoding algorithm to record data. The IBM format does not mix "double frequency" encoding and "modified frequency modulation" recording on the same track. The fact that the DEC double density format DOES mix these two encoding algorithms is the basis for why a standard MFM encoding algorithm could not be used on the RX02 and DSD 440. These technical details in no way relate to the reliability or performance of the DSD 440 machine. They were included here to give the reader a better understanding of why what we call "DEC double density format" is not compatible with "IBM double density format".

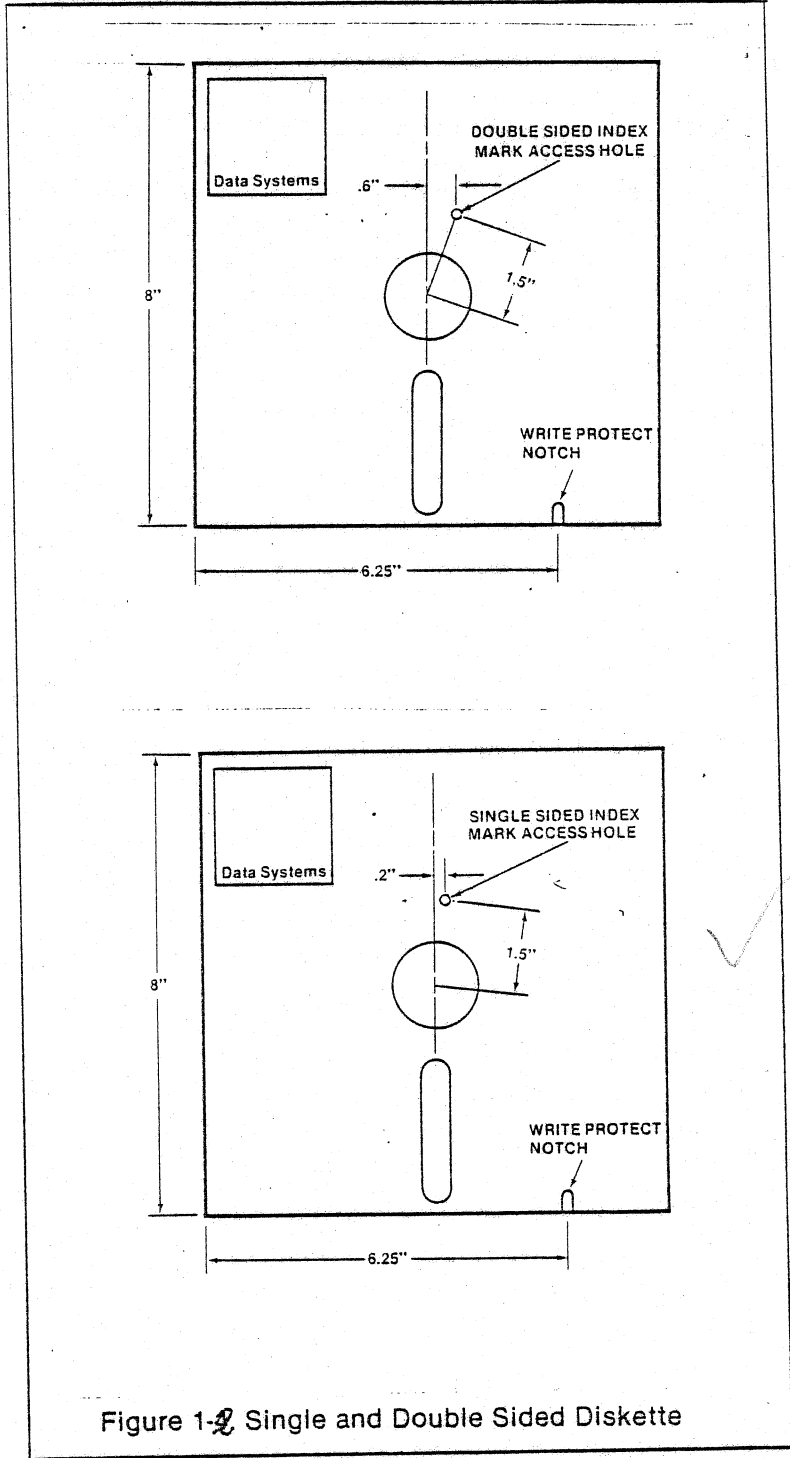


Figure 1-4 Single and Double Sided Diskette

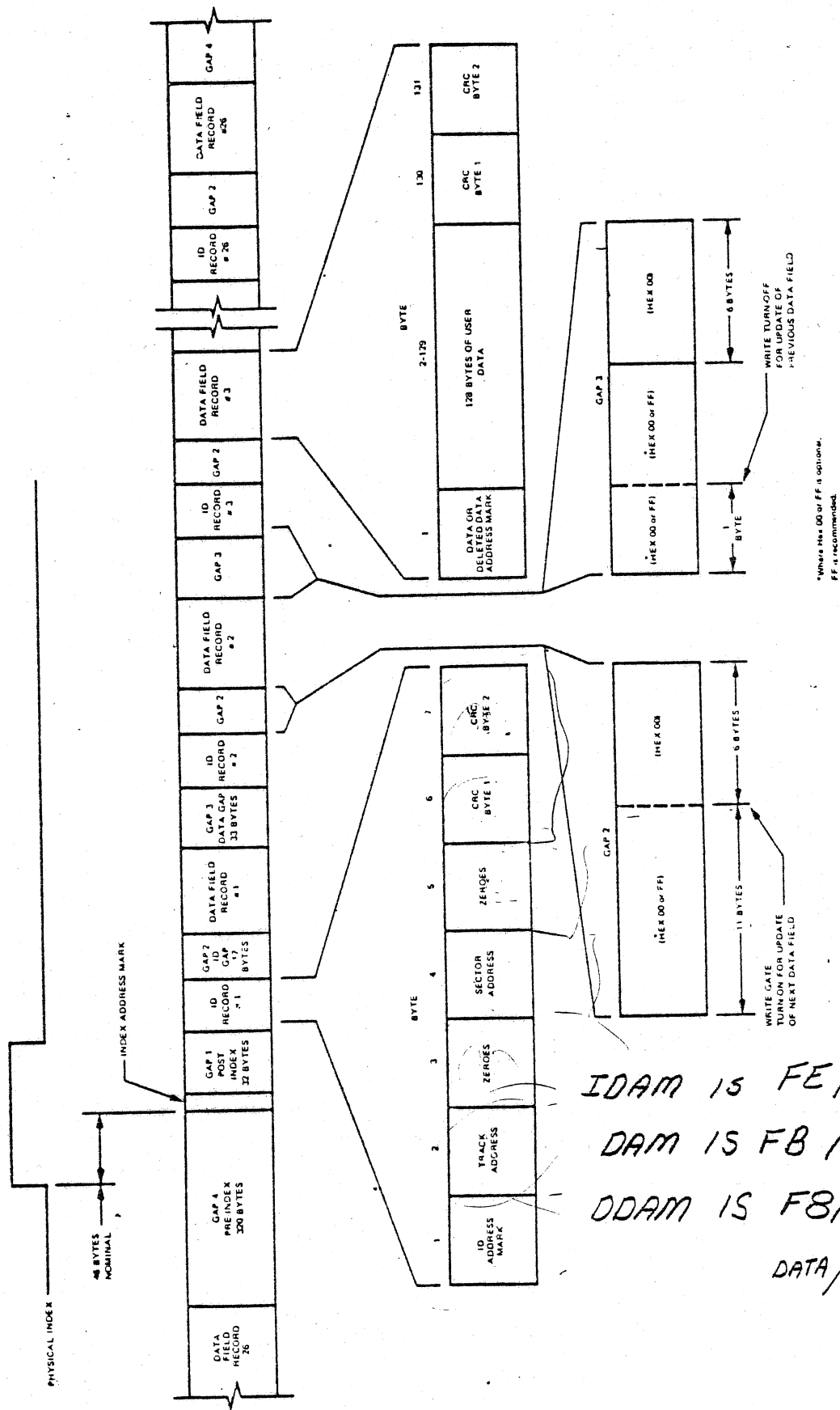


FIGURE 1-3 SINGLE DENSITY TRACK FORMAT

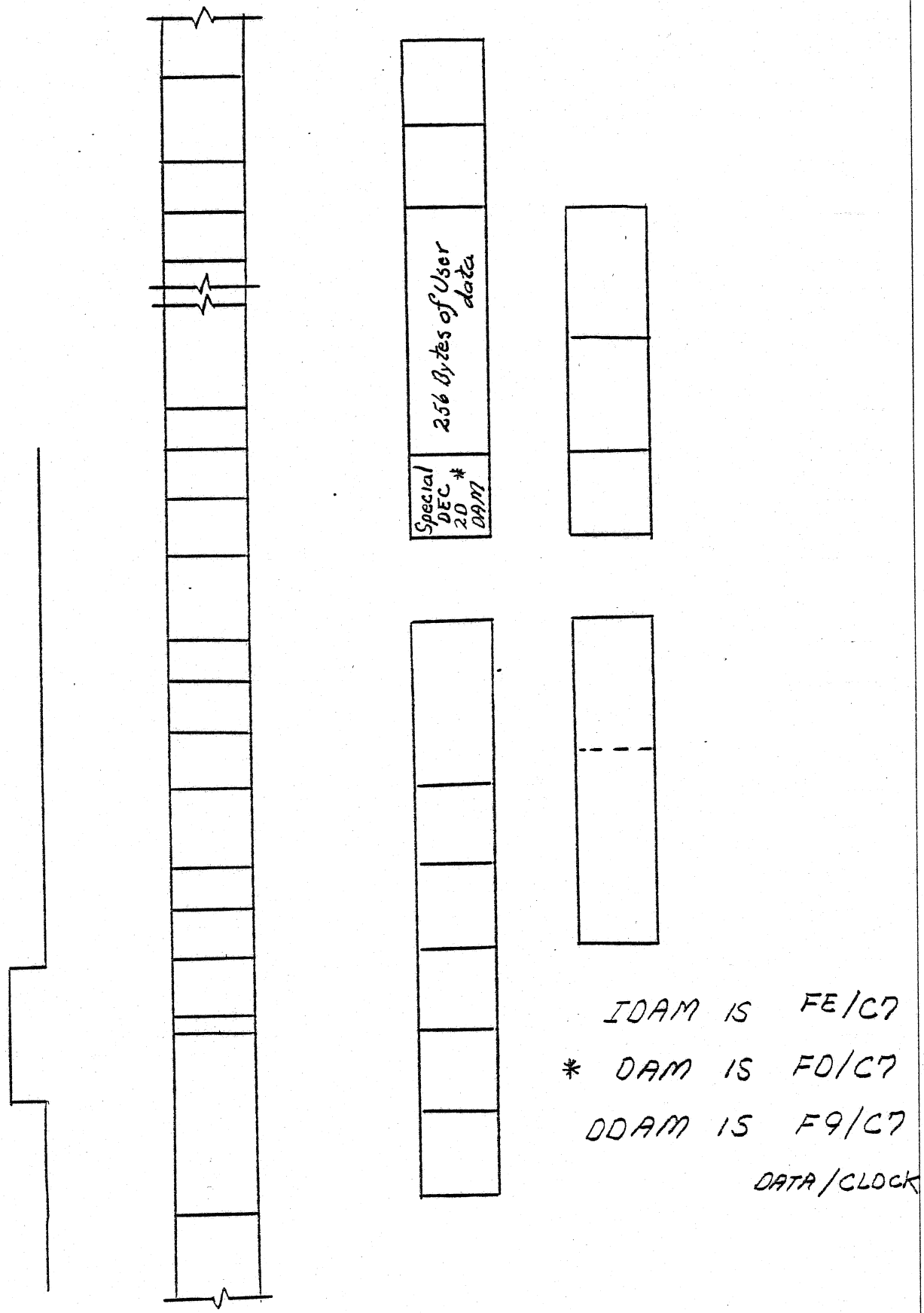


FIGURE 1-4 DEC DOUBLE DENSITY TRACK FORMAT

## 1-4 FEATURES

### \* HARDWARE COMPATIBLE

The DSD 440 can be used on any DEC PDP-11 (UNIBUS), LSI-11 (Q-BUS), or PDP-8 (OMNIBUS) computer when ordered with the appropriate interface module.

### \* SOFTWARE COMPATIBLE

All DEC software intended for either the RX01 or the RX02 will run on the DSD 440 without modification. No special device handlers or drivers are required.

### \* MEDIA AND FORMAT COMPATIBLE

The DSD 440 can read and write diskettes in the industry standard formats. This means diskettes can be freely interchanged between the DSD 440, DEC RX01 and RX02.

### \* INCREASED STORAGE CAPACITY

A two drive configuration is capable of one megabyte of data storage.

### \* DISKETTE FORMATTING CAPABILITY

The DSD 440 permits write-formatting of diskettes. The physical sector sequence written on the diskette can be determined by the programmer so that hard sector interleaving is possible. Diskettes with "blown" headers can usually be recovered by reformatting.

### \* LOW PROFILE 5 1/4" HIGH CHASSIS

The DSD 440 takes up half as much rack space as the DEC RX02.

580

580

580



1-5 DSD 440 FLOPPY DISK MEMORY SYSTEM  
SPECIFICATIONS

\* 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

SINGLE DENSITY FORMAT:	IBM 3740 FORMAT
DOUBLE DENSITY FORMAT:	COMPATIBLE WITH DEC DOUBLE DENSITY DEVICES

SINGLE DENSITY RECORDING TECHNIQUE:	DOUBLE FREQUENCY
DOUBLE DENSITY RECORDING TECHNIQUE:	DEC-MODIFIED MFM
FLUX TRANSITION DENSITY MAXIMUM:	3200 FCI (INNER TRACK)
TRACK DENSITY:	48 TRACKS PER INCH
TRACK-TO-TRACK SPACING:	.529 MM (.021 INCHES)
TRACK WIDTH:	.3048 MM (.012 INCHES)

\* SPEEDS

DOUBLE DENSITY

DISKETTE TO CONTROLLER BIT RATE: 500 KHZ

CONTROLLER TO CPU MEMORY TRANSFER RATE: 27 MICROSECONDS PER WORD  
PLUS ANY DMA OVERHEAD

"REAL" THROUGHPUT:

CASE 1 - ASSUMES ALL DATA COMING FROM THE SAME TRACK, AND A 2-WAY INTERLEAVE WHICH MEANS 26 SECTORS CAN BE TRANSFERRED IN TWO REVS. 20,000 BYTES PER SECOND

CASE 2 - ASSUMES SEEKING WITH A 7 SECTOR SKEW BETWEEN SEQUENTIAL TRACKS. 16,000 BYTES PER SECOND

SINGLE DENSITY

DISKETTE TO CONTROLLER BIT RATE: 250 KHZ

CONTROLLER TO CPU MEMORY TRANSFER RATE IN RX02 COMPATIBLE MODE: 27 MICROSECONDS PER WORD PLUS ANY DMA OVERHEAD

CONTROLLER TO CPU TRANSFER RATE WHILE IN RX01 COMPATIBLE MODE: FUNCTION OF THE PROGRAM LOOP EXECUTION TIME

"REAL" THROUGHPUT:

CASE 1 - SAME AS DESCRIBED ABOVE 10,000 BYTES PER SECOND

CASE 2 - SAME AS DESCRIBED ABOVE . 8,000 BYTES PER SECOND

DISKETTE ROTATION: 360 RPM +/- 2%

HEAD STEP RATE: 8 MSEC TRACK-TO-TRACK (SAB00)

HEAD LOAD TIME: 35 MSEC

AVERAGE ACCESS TIME: 296 MSEC

MAXIMUM ACCESS TIME: 645 MSEC

\* INTERFACE CHARACTERISTICS

INTERFACE MODULE BACKPLANE REQUIREMENTS

PDP-8 (OMNIBUS) 1 QUAD SLOT

✓ LSI-11 (Q-BUS) 1 HALF-QUAD SLOT

PDP-11 (UNIBUS) 1 QUAD SPC SLOT

INTERFACE MODULE POWER CONSUMPTION (+5 VOLTS)

	NOMINAL	MAXIMUM
DSD 440-2131 (PDP-8)	1.37 AMPS	2.35 AMPS
DSD 440-4432 (LSI-11)	1.44 AMPS	2.59 AMPS
DSD 440-4430 (PDP-11)	1.44 AMPS	2.17 AMPS

STANDARD DEVICE ADDRESSES OR CODES

DSD 440-2131 (PDP-8) 6750 - 6757

✓ DSD 440-4432 (LSI-11) 777170 - 777172

DSD 440-4430 (PDP-11) 777170 - 777172

176500 300,304 0

\* CHASSIS POWER CONSUMPTION

MASTER CONTROLLER MODULE,  
CURRENT FROM 5 VOLT SUPPLY: 4 AMPS NOMINAL, 5.6 AMPS MAXIMUM  
SINGLE DRIVE CHASSIS: 150 WATTS IDLE, 232 WATTS BUSY  
DUAL DRIVE CHASSIS: 248 WATTS IDLE, 330 WATTS BUSY

SELECTABLE INPUT VOLTAGES: 100 VAC OR 120 VAC RMS +/- 10%  
220 VAC OR 240 VAC RMS +/- 10%

INPUT FREQUENCIES: 50 HZ +/- 1 HZ  
60 HZ +/- 1 HZ

FUSE RATINGS:  
SINGLE DRIVE; 115 VAC 2.5 AMP SLOW-BLOW  
DUAL DRIVE; 115 VAC 3 AMP SLOW-BLOW  
SINGLE DRIVE; 220 VAC 1.25 AMP SLOW-BLOW  
DUAL DRIVE; 220 VAC 2 AMP SLOW-BLOW

\* HEAT DISSIPATION (IN BTU'S PER HOUR)

	NOMINAL	MAXIMUM
ALL DSD 440 INTERFACE CARDS:	24	44
MASTER CONTROLLER CARD:	85	95
SINGLE DRIVE CHASSIS: (IDLE)	290	512
(BUSY)	503	791
DUAL DRIVE CHASSIS: (IDLE)	468	846
(BUSY)	681	1125

\* ENVIRONMENT

U.L. LISTING: EDP EQUIPMENT, UL 478 STANDARD

OPERATING TEMPERATURES

INTERFACE MODULES:	0 C TO 50 C (32 F TO 122 F)
MASTER CONTROLLER CARD:	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 150.8 F)
MASTER CONTROLLER CARD:	-40 C TO 66 C (-40 F TO 150.8 F)
CHASSIS:	-40 C TO 66 C (-40 F TO 150.8 F)
DISKETTES:	-40 C TO 52 C (-40 F TO 125 F)

\* HUMIDITY

INTERFACE MODULES, CHASSIS, AND MASTER CONTROLLER CARD: 10% TO 95% (NON-CONDENSATING)

DISKETTES: 8% TO 80% WITH A MAXIMUM WET BULB TEMP. OF 29.4 C (85 F)

\* SIZES

CHASSIS:	5.25" H X 17.6" W X 21.0" D
SHIPPING CARTON:	30.0" H X 24.5" W X 12.5" D
MASTER CONTROLLER CARD:	17.1" H X 4.6" W X 1.0" D
QUAD INTERFACE MODULES:	9.0" H X 10.5" W X 0.5" D
DUAL-HEIGHT INTERFACE MODULE:	9.0" H X 5.2" W X 0.5" D

\* WEIGHT

CHASSIS:	50 POUNDS
SYSTEM, PACKAGED FOR SHIPPING:	74 POUNDS

\* SHOCK AND VIBRATION

OPERATING SHOCK:	1G FOR 10-20 MILLISECONDS
NON-OPERATING SHOCK:	15G FOR 10-20 MILLISECONDS
VIBRATION:	5 - 25 HZ @ .0014 INCHES
	25 - 55 HZ @ .0007 INCHES
	55 - 300 HZ @ .3G

## CHAPTER 2 DSD 440 OPERATING MODES AND SYSTEM CONFIGURATION

### 2-1 OPERATIONAL MODES

The DSD 440 has two different operating modes. These two modes differ from each other in the way data can be stored on a diskette, and the way the programs access the disk system. Chapter 5 will first describe the Mode 1 programmers' interface for PDP-11 family computers and then the Mode 2 programmers' interface. The programmers' interface for the PDP-8 family will be described for both modes simultaneously since the number of differences is far fewer and no interface module changes are required when switching modes.

#### MODE 1:

When operated in Mode 1, the DSD 440 emulates the DSD 210 and the DEC RX01 programmed I/O single density flexible disk systems. This means that programs written for the DSD 210 or the DEC RX01 will run on the DSD 440 without modification. Mode 1 operation does not permit double-density data to be read or written. The user should configure his system in Mode 1 if his operating system software has only a first generation floppy disk device handler and if double-density capability is not immediately needed. The DSD 440 is normally shipped configured to run in Mode 2. Complete instructions for changing the operating mode can be found in chapter 3.

#### MODE 2:

Mode 2 operation allows the DSD 440 to access double density diskettes and to transfer data across DEC 11 family processors' I/O bus via direct memory access (DMA). The DSD 440 has been designed to be program compatible with the DEC RX02. Data transfer is faster in Mode 2 because direct memory access is used to move data between main memory and the floppy disk system and because the bit transfer rate off the diskette is twice that in single density recording. The programmer can set the density of a given function using a bit in the control and status register. This means that the programmer is free to decide when to employ double density and single density while all the time remaining configured in Mode 2. A diskette should never contain data of mixed densities. The controller determines the density of a diskette by sampling the Data Address Mark on an arbitrary track and sector. The controller then assumes that the entire diskette has been recorded in that same density.

## 2-2 DSD 440 SYSTEM CONFIGURATION

A complete DSD 440 flexible disk data storage system consists of a computer interface module, a controller/drive sub-system, and an interface bus cable to connect the two system elements. This section will briefly describe each of the system elements, how it connects to the rest of the system, and how certain minor adjustments are performed. A familiarity with the purpose and physical location of the system elements will be of use during both system installation and maintenance. Figure 2-1 shows a top view of a DSD 440 system.

### 2-2.1 FLOPPY DISK DRIVES:

Up to two Shugart model 800R drives are mounted in the front of the main chassis. Since the bottom side of the chassis does not open, the drives must be removed from the chassis should any maintenance be required. Each drive is fastened by four screws accessible from the underside of the chassis. A 50 conductor flat cable forms the drive bus. This bus connects the master controller PCB assembly to the two drives. Each drive is connected to the power distribution PCB assembly by two cables. One cable supplies the 115 (or 220) VAC for the spindle motor and the other cable supplies the DC voltages for the electronics and solenoids. A Drive maintenance manual published by Shugart Associates is included as an appendix to this User's Manual.

### 2-2.2 MASTER CONTROLLER PCB:

The master controller board contains a microprogrammed read/write controller and a conventional 8 bit microprocessor. Connected to this board are the floppy disk drives, the interface bus connector, and a cable supplying power from the power distribution assembly. Located near the top of the board is a row of 8 LED indicators. LED 1 is colored green for easy identification. The interpretation of these indicators is explained in chapter 4. Near the indicators is a DIP-SWITCH. The eight switches in this assembly are used to establish different system configurations and specify the self contained "hyper-diagnostics" used during maintenance operations. Chapter 4 completely explains how to use the DIP-SWITCH. Figure 2-2 is a top view of the circuit board which shows the locations of the important parts. The normally installed jumpers are also shown in this figure. The master controller board requires a single power supply voltage of 5 VDC. Under normal operating conditions, the board draws approximately 3.5 Amps of current.

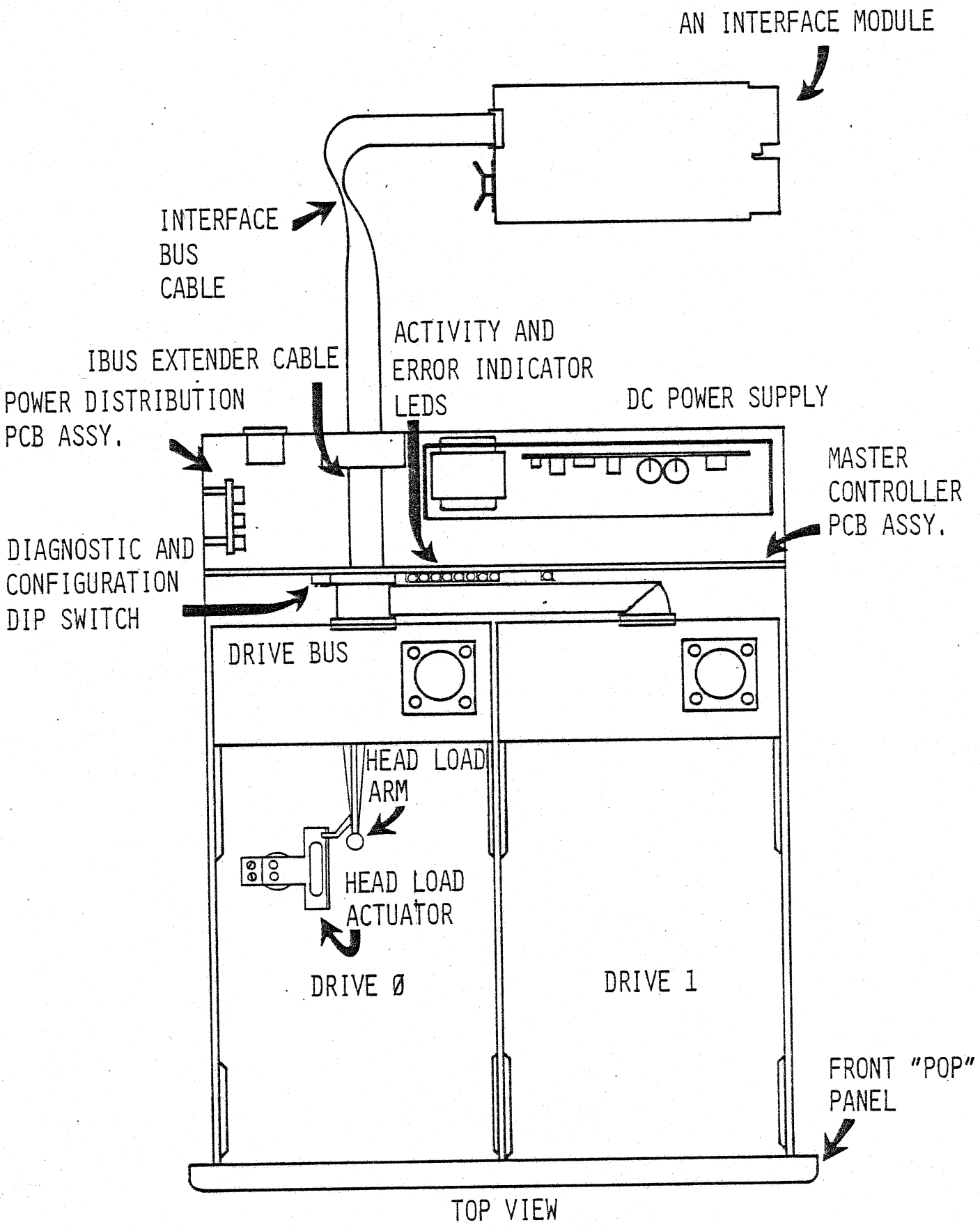
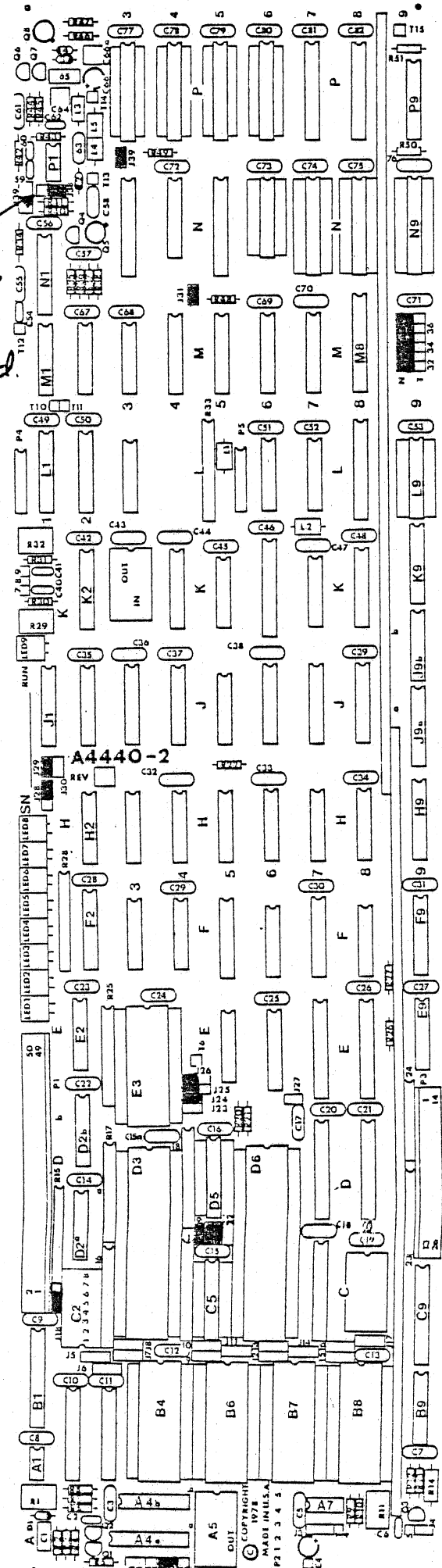


FIGURE 2-1 THE DSD 440 SYSTEM

shift this jumper down for a "-3" or later revision board.



19 JUMPERS ARE INSTALLED FOR NORMAL OPERATION. THEY ARE SHOWN DARKENED-IN ON THE DIAGRAM.

MASTER CONTROLLER NORMALLY INSTALLED JUMPERS



### 2-2.3 DC POWER SUPPLY:

The DC power supply is located in the rear of the chassis. It is a conventional open-frame supply using linear regulators. DC output voltages include: 5 volts, 24 volts, and unregulated -12 volts. Figure 2-3 shows points of interest on the power supply. Note the two trimmer potentiometers which can be used to adjust the +5 and +24 volt outputs. A schematic, parts list, list of specifications, and trouble shooting guide for the power supply are included as an appendix to this Users' Guide.

### 2-2.4 POWER DISTRIBUTION PCB ASSEMBLY:

The power distribution assembly is mounted on the left side of the chassis. This board is used to distribute both AC and DC voltages within the chassis. Any sub-system element can be rapidly removed from the chassis since all electrical connections are made using cables with a connector on at least one end.

### 2-2.5 AC POWER SWITCH:

An AC power switch is mounted just below the CORCOM connector. When the DSD 440 is installed in a computer system with a central AC power controller, this switch would normally be left ON. When the "hyper-diagnostic" test and utility programs are being executed (see Chapter 4), the AC power switch is a convenient way to start and stop these programs.

### 2-2.6 FAN:

The fan draws air in through the chassis vents and blows it out the small fan grill on the rear of the chassis. No filters are used in this cooling system, so no periodic filter changes are needed.

## 2-2.7 CORCOM CONNECTOR:

The CORCOM connector is mounted through the rear panel of the chassis next to the fan. This connector assembly contains a line filter, fuse, and a small PC board which can be moved to rapidly modify the way transformer primaries are wired to the AC line. A molded line cord mates with the CORCOM connector on the outside of the chassis. The following fuses should be installed in the CORCOM connector:

SINGLE DRIVE SYSTEM;	100/120 VAC	2.5 AMP SLOW-BLOW
DUAL DRIVE SYSTEM;	100/120 VAC	3 AMP SLOW-BLOW
SINGLE DRIVE SYSTEM;	220/240 VAC	1.25 AMP SLOW-BLOW
DUAL DRIVE SYSTEM;	220/240 VAC	2 AMP SLOW-BLOW

The small PC board can be inserted into the CORCOM connector in any one of four ways. When the PC board is fully inserted in the CORCOM connector, the AC voltage currently selected can be read directly off the PC board. Note that the connector is constructed in such a way that only one of the four voltage labels etched on the PC board can be seen when the board is fully inserted in the connector. It is possible to make rapid conversions between the two low line voltages (100 VAC, 120 VAC) simply by pulling the PC board out, changing its orientation, and re-inserting it. Using a similar procedure, a rapid conversion between the two high line voltages (220 VAC, 240 VAC) can also be made.

### CAUTION:

The procedure required in order 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 PC board. This conversion requires changing the fan, the two AC spindle motors in the floppy disk drives, the motor capacitors associated with the spindle motors, and the fuse value. For this reason, NEVER change the position of the PC board from the low line voltages to the high line voltages without changing the AC motors. If you do, there is a good chance the motors will burn up.

## 2-2.8 INTERFACE CABLE:

The interface cable is a 26 conductor flat cable with female 3M type connectors on each end. This cable is keyed to help prevent backward installation. The hole in the connectors corresponding to pin 23 has been plugged on both ends. Similarly, the pin that would normally mate with these holes has been clipped on both the main chassis connector and the interface module connector. No AC or DC power is transmitted across the interface bus (IBUS) cable. Half of the conductors are signals and the other half serve as grounds. Data Systems ships a ten foot interface cable with DSD 440 systems. It is recommended that users requiring a longer cable build one from a 26 conductor twisted pair/flat cable. Spectra-strip Corporation is a supplier of this type of cable.

## 2-2.9 INTERFACE MODULE:

The interface module is a printed circuit board that has been designed to meet the I/O bus interface specifications and the physical form factor of the host computer. Data Systems manufactures interface modules for the DEC PDP-11, PDP-8, and LSI-11 computers. In addition, Data Systems can supply a complete specification of the signals and protocols on the IBUS. Customers desiring to interface the DSD 440 to other computers can design their own interface modules based on the information contained in this "IBUS" specification.

The DSD 440-11 and DSD 440-L11 interface modules contain a built in hardware bootstrap. The hardware bootstrap circuit consists of address decoding circuits and a PROM with a PDP-11 program which can be executed by the host computer. The program which resides in this PROM speeds the loading of an operating system from diskette to the host CPU's memory. Also in this PROM are some simple CPU and memory diagnostic programs. More details about the bootstrap program can be found in chapter 3. An assembly listing of the program is included as an appendix to this users' manual.



## CHAPTER 3 SYSTEM INSTALLATION AND ACCEPTANCE

### 3-1 ENVIRONMENTAL CONSIDERATIONS

All floppy disk systems manufactured by Data Systems Design perform efficiently in a normal computer room environment. Temperature, humidity, and cleanliness are three environmental parameters that can impair reliable usage of diskettes if not kept within specified limits.

Diskettes are specified to operate within an ambient temperature range of 10 \*C to 51 \*C (50 \*F to 125 \*F). The maximum thermal gradient should not exceed 15 \*F per hour. The DSD 440 chassis should be installed where the ambient temperature does not exceed these limits while the system is in operation.

Humidity control is necessary for the efficient operation of diskette memory systems. At very low humidity (dry air) static electricity can be generated as a result of contact between the read/write head and the diskette. When the electrical potential becomes high enough to cause a discharge, soft data errors may occur. At very high humidity, mylar diskettes can start to swell as they absorb moisture from the air. This can have the effect of moving the centerline of a previously recorded track away from the centerline of the read/write head. Again, the noticeable effect will be an increase in the soft error rate. The operating relative humidity range is 8 to 80% with a maximum wet bulb temperature of 29.4 \*C (85 \*F).

Cleanliness is important wherever diskettes are going to be used, handled, and stored. Unlike most rigid disk cartridges, floppy disks are not sealed units. If the DSD 440 is operated in an environment which has a high concentration of abrasive airborne particles, the useable life of the diskettes is likely to be reduced and the soft error rate will increase. Care must be taken, while handling diskettes, never to touch the magnetic media.

Chapter 1 of this manual includes specifications on heat dissipation associated with of DSD 440 system. This data can be used to determine if the cooling capacity of your complete computer system will accommodate the DSD 440 system.

### 3-2 UNPACKING THE SYSTEM

No special tools or equipment are required to install the DSD 440 floppy disk system. It is recommended that all packing materials be saved in case the system requires shipment at some future date. Please adhere to the following list of steps:

- 1) Inspect the shipping carton for damage caused in shipment. Report any damage to the shipper before opening the carton.
- 2) Open the top of the outer carton and then the top of the inner carton.
- 3) Remove the foam blocks from the doors of the disk drives.
- 4) Pull the system from the inner box by inserting fingers inside the drive doors and pulling the chassis out of the box (2 person operation).
- 5) Inspect the unit for any obvious damage. Immediately report any damage to Data Systems Design.
- 6) Remove the other system parts from the carton.

Except where noted, all DSD 440 systems will be shipped with the following materials. Be sure to inform Data Systems Design immediately if any materials are missing or damaged.

#### LIST OF MATERIALS:

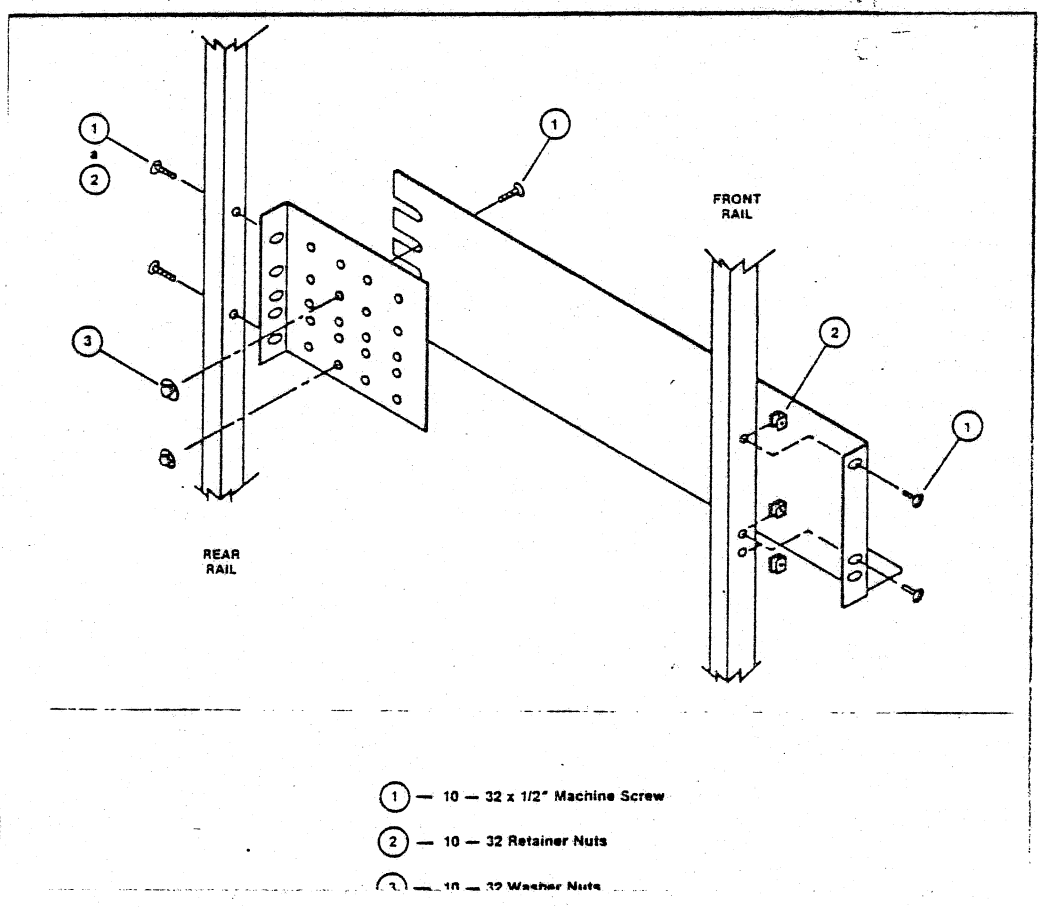
- 1) Chassis
- 2) Computer interface module
- 3) Documentation binder including:
  - a. System Users Manual
  - b. DSD bootable diagnostic diskette
- 4) AC power cord
- 5) Interface bus connecting cable
- 6) Chassis mounts (if ordered)
- 7) Bag with spare shunts and jumpers

### 3-3 MOUNTING THE DSD 440 CHASSIS

The DSD 440 chassis must be installed sufficiently close to the ultimate location of the interface module so that the 10 foot interconnecting cable will reach. Another consideration concerns the frequency of diskette changes. If the computer operator is likely to be changing diskettes often, it would be desirable to install the chassis as close to the console terminal as possible.

The following discussion pertains to installing the DSD 440 chassis in a 19" rack. Because of the width of the Shugart floppy disk drives and their horizontal mounting in the DSD 440, a wider than usual chassis is needed. The DSD 440 chassis can be mounted in a standard 19" RETMA rack if special slim-line chassis mounts are used. Figure 3-1 illustrates how to attach the 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. After the mounts are securely fastened to the rack, 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. (See Figure 3-2) Remove the molded front pop panel from the chassis by pulling the top of the panel out from the chassis. Secure the chassis in the rack by bolting the front flange of the chassis to the front rails of the rack. (See Figure 3-3) Replace the pop panel by pushing it straight back on to the two "head locks". Before actually bolting the chassis in the rack, it is advisable that the remainder of this chapter be read first. This is because some system configurations will require modification of a DIP-SWITCH on the controller board inside the chassis. It is generally difficult to modify the settings of this DIP-SWITCH once the chassis has been secured in the rack.

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



- ① — 10 — 32 x 1/2" Machine Screw
- ② — 10 — 32 Retainer Nuts
- ③ — 10 — 32 Washer Nuts

FIGURE 3-1 CHASSIS SLIDES MOUNTING



42-381 50 SHEETS 3 SQUARE  
42-382 100 SHEETS 3 SQUARE  
42-383 200 SHEETS 3 SQUARE  
NATIONAL  
MADE IN U.S.A.

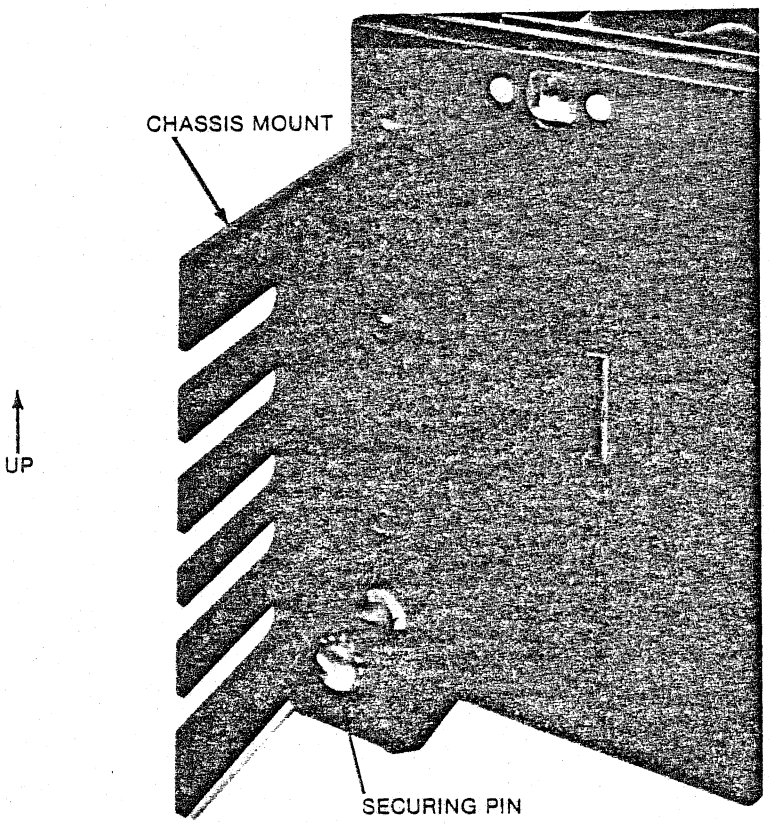


Figure 3-2. Bullet Entering Hole in Rear of Unit

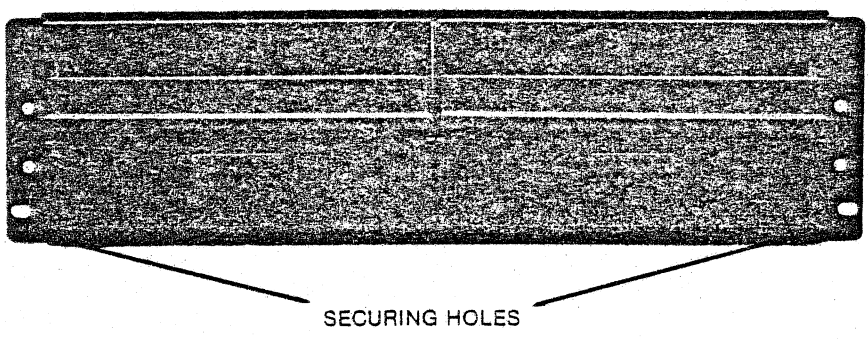


Figure 3-3. Front View of Chassis with Front Panel Removed to Indicate Securing Holes

### 3-4 INSTALLING THE INTERFACE MODULE AND CABLE

Ensure that all system 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 should be marked P/N 4432 on the component side of the board. The user can select from one of four device register addresses, one of four bootstrap PROM starting addresses, and a 7-bit interrupt vector address. There is a separate jumper which, when installed, disables the bootstrap PROM. (See Figure 3-4) Data Systems Design ships this module so that the device register address is at 177170, the bootstrap PROM is enabled at address 173000, and the interrupt vector is 264. The Mode 1 jumper is removed when this module is shipped, therefore the module is initially configured for Mode 2 (RX02 compatible) operation. Note that 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. Please check your module against the tables and Figure 3-4 to insure that it has been configured consistent with your needs. 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 generally have to be made in the software. Also, be sure to read the explanation of the bootstrap and diagnostic programs carefully if non-standard addresses are used.

STARTING REGISTER ADDRESS	POSITION 1	POSITION 2
✓ 177170 (NORMAL)	CLOSED	CLOSED
177160	OPEN	CLOSED
177140	CLOSED	OPEN
177150	OPEN	OPEN

STARTING BOOT PROM ADDRESS	POSITION 3	POSITION 4
173000 (NORMAL)	CLOSED	CLOSED
✓ 171000	OPEN	CLOSED
175000	CLOSED	OPEN
166000	OPEN	OPEN

When you are sure that the jumpers on the interface module are configured correctly, plug one end of the ten foot interface cable into the interface module such that pin 1 (the striped side) is closest to the edge of the board. Also confirm that the pin position of the clipped pin on the module connector matches the position of the plugged up hole on the cable connector. After verifying that power is off, plug the module into the lowest numbered available Q-bus slot.

**CAUTION:**

There must be no open Q-bus slots in between the processor and the DSD 440-L11 interface module. Since this module uses both interrupts and direct memory access, a break in either of the grant propagation chains would prevent the interface module from obtaining control of the Q-bus. Figure 3-5 shows how Q-bus slots are numbered in some of the more common backplanes available from DEC.

Installation information for LSI-11 based systems continues in section 3-5.

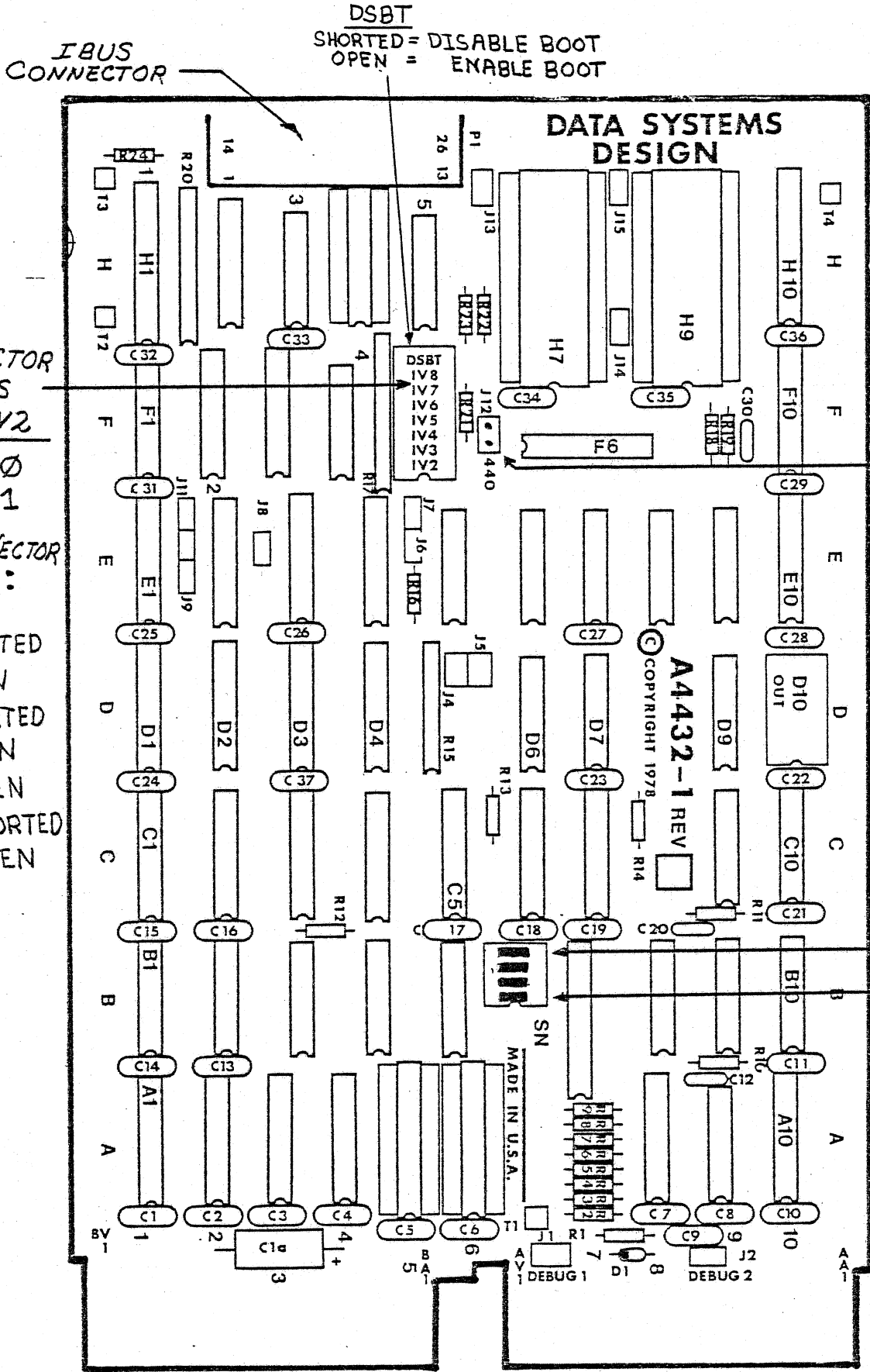
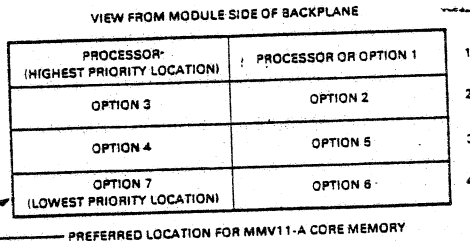


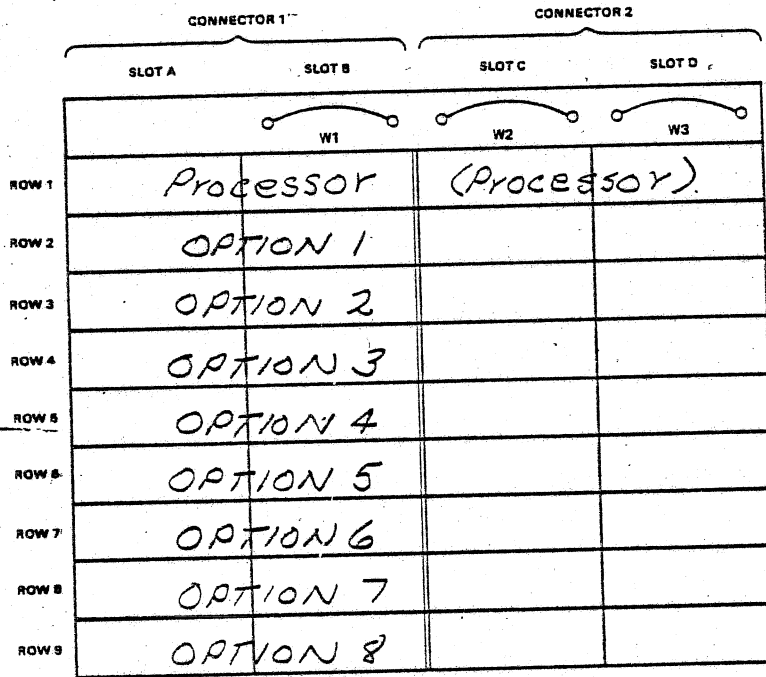
FIGURE 3-4 LSI-11 INTERFACE MODULE

H9270



42.381 20 SHEETS 5 SQUARE  
42.382 100 SHEETS 5 SQUARE  
42.389 200 SHEETS 5 SQUARE  
NATIONAL

H9273-A



VIEW IS FROM MODULE SIDE OF CONNECTORS.

MA-0740

DDVII-B

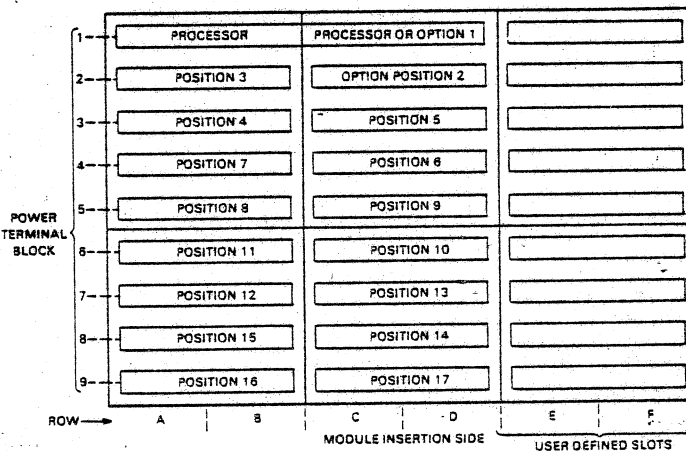


FIGURE 3-5 OPTION PRIORITY IN DEC BACKPLANES FOR LSI-11

## PDP-11 BASED SYSTEMS:

The DSD 440 interface for PDP-11 based systems is a quad module marked 4430 on the component side. Data Systems Design ships this interface module configured as follows:

REGISTER ADDRESS: 777170  
BOOTSTRAP PROM: ENABLED AT 771000  
INTERRUPT VECTOR: 264  
INTERRUPT PRIORITY: BR5  
OPERATING MODE: MODE 2 (RX02 COMPATIBLE)

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 3-6 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 (NORMAL)	OPEN	CLOSED
775000	CLOSED	OPEN
766000	CLOSED	CLOSED

The bootstrap program contained on the interface module will occupy 256 words of memory space, starting at one of the four selectable addresses shown above. If the user does not want the bootstrap program to respond to any addresses, the bootstrap disable jumper (shown in Figure 3-6) should be installed.

Shunt positions 3 thru 12 correspond to address bits A3 thru 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 this interface module is shipped, it is configured to respond to a base register address of 777170 (octal). This works out to having shunt positions 7 and 8 left closed, and positions 3, 4, 5, 6, 9, 10, 11, and 12 punched open.

There is an eight position shunt located at coordinates B-12 which is used to configure the interrupt vector address. Figure 3-6 shows how the 8 shunt positions are numbered. Position 1 is not used. Positions 2 thru 8 correspond to interrupt vector address bits IV2 thru 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 works out to having shunt positions 3, 6, and 8 left closed, and positions 2, 4, 5, and 7 punched open.

In the rare cases where the interrupt priority level must be changed, you will have to cut and jumper the circuit board so that it looks like the diagram corresponding to the desired interrupt priority level. (see Figure 3-7). If the priority levels are going to be changed often, it is suggested that the six permanent traces be cut, and four 8-pin IC sockets be installed in the positions outlined on the board. Either four-position shunts or dip-switches can then be placed in the sockets to facilitate rapid 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.

If the system is going to be operated in the RX01 compatible mode, then the EN RX01 jumper located near coordinates A-12 must be installed. This jumper is removed for RX02 compatible operation.

When you are sure the jumpers on the interface module are configured correctly, plug one end of the ten foot interface cable into the interface module such that pin 1 (striped side) is closest to the module handle. Also confirm that the pin position of the clipped pin on the module connector matches the position of the plugged up hole on the cable connector. After verifying that power is off, plug the module into a convenient small peripheral controller (SPC) slot. Make sure that there is grant continuity between the processor and the interface module. If there are any open SPC slots between the processor and the interface module, there should be a grant continuity card placed in slot D.

#### IMPORTANT !!!

Since the DSD 440-11 module uses direct memory access (DMA) you must ensure that there is no backplane jumper or foil trace between backplane pins CA1 and CB1 of the SPC slot you select. These two pins normally connect NPG 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 an interface module configured for RX02 compatible operation is installed, the whole system will hang. Remember to replace the jumper any time the 4430 module is removed. If you forget, DMA devices further out on the UNIBUS will never receive NPG and the UNIBUS will hang each time a DMA cycle is attempted by one of these devices.

BOOTSTRAP  
DISABLE  
JUMPER

12 POSITION SHUNT USED FOR  
SELECTING BOOTSTRAP ADDRESS  
AND DEVICE REGISTER ADDRESSES

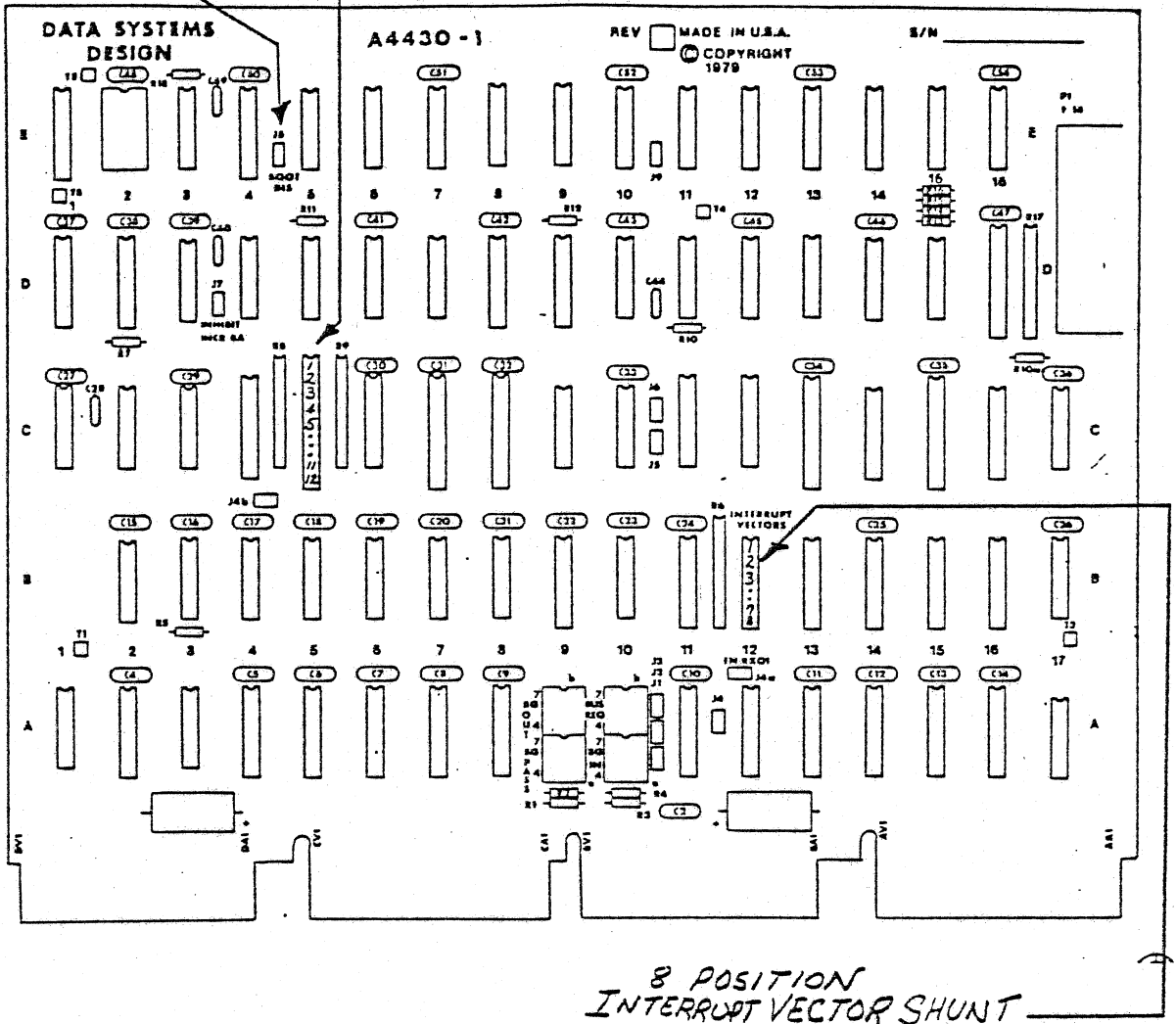


Figure 3-6 PDP-11 Interface Module



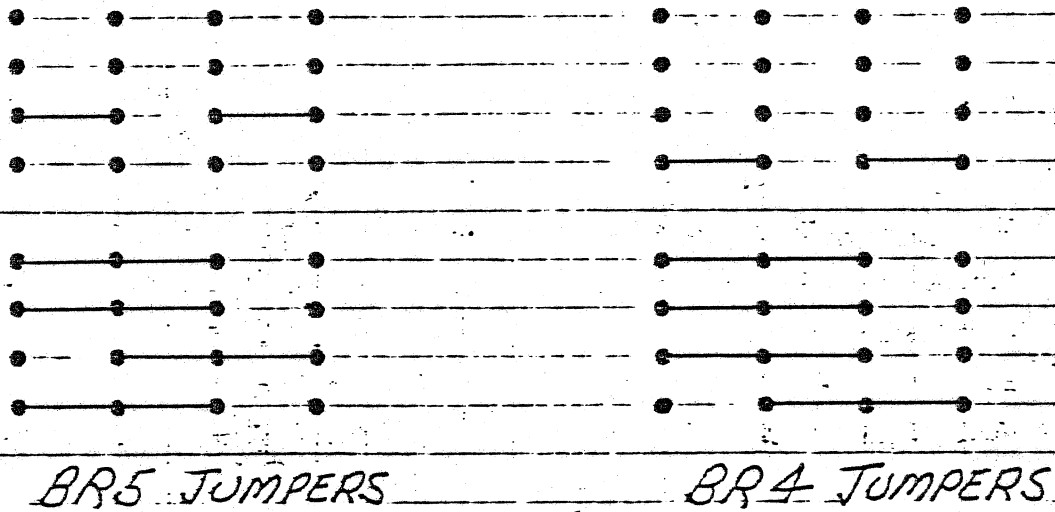
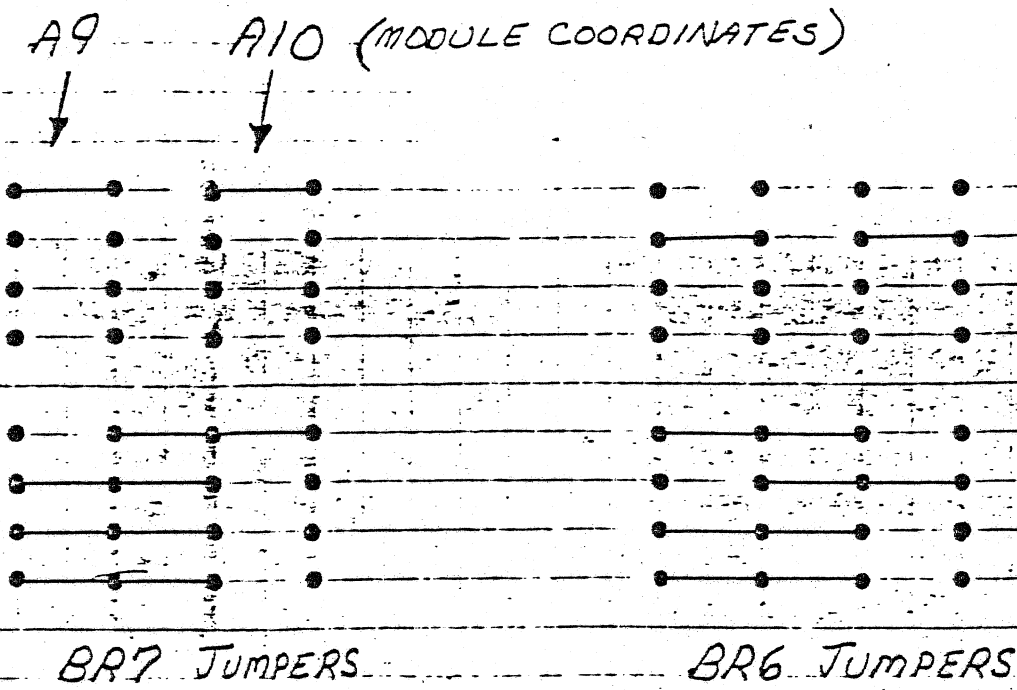


FIGURE 3-7 INTERRUPT PRIORITY LEVELS

## PDP-8 BASED SYSTEMS:

THE DSD 440 interface module for PDP-8 based systems should be marked DSD 2131 on the component side opposite the interface cable connector. 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 actually desired. There are no additional jumpers or adjustments on the 2131 interface module.

Locate the ten foot interface cable. Plug one end of the cable into the 2131 module such that the pin position of the clipped pin on the module connector matches the position of the plugged up hole on the cable connector. Plug the opposite end of the interface cable into the connector mounted on the rear panel of the chassis. Again, observe that there is only one correct way to insert the cable. After verifying that power is off, plug the module into an available Omnibus slot.

Note that the PDP-8 interface module does not contain a bootstrap. This module is not capable of direct memory access (DMA) either.

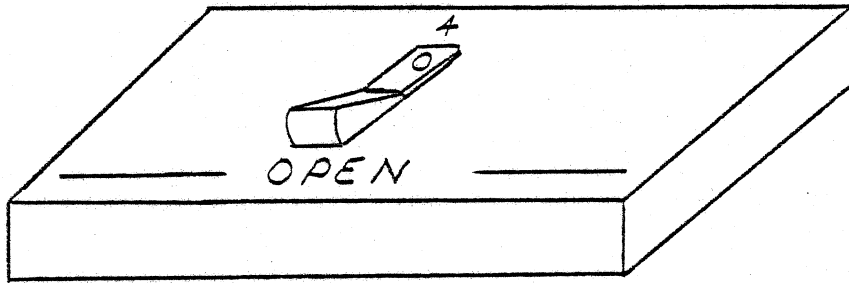
### 3-5 INPUT POWER CONSIDERATIONS

The DSD 440 can be ordered so as to be initially configured with any combination of 50 or 60 HZ and 120 or 240 VAC. Optional conversion kits allow users to change the system between 50 and 60 HZ operation. The system cannot be easily changed between 120 and 240 VAC in the field.

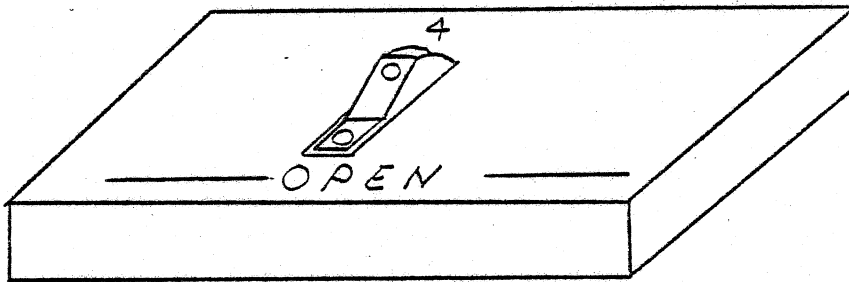
In geographic locations where the line voltage deviates more than 10% from 120 VAC or 240 VAC, an easy adjustment can be made to modify the way the power transformer primaries are wired to the AC line. The CORCOM filter which is mounted through the rear panel of the chassis contains a small PC board which can be removed from its slot and reinserted in different orientations. The different orientations correspond to various line voltages. Although this PC card allows for both + and - variations on both 120 VAC and 240 VAC systems, one should NEVER attempt to use this card to change a 120 VAC system into a 240 VAC system or visa-versa. It simply does not work that way! Section 2-2.7 discusses the CORCOM connector in more detail.

### 3-6 CHANGING THE OPERATING MODE

As mentioned in chapter 2, the DSD 440 can be configured to operate in either RX01 compatible mode or RX02 compatible mode. There is a switch that selects the operating mode located on the large circuit board assembly inside the chassis. Data Systems ships systems configured in RX02 compatible mode (MODE 2). If this is how you plan to use your DSD 440, then there is no need to open up the chassis at this time. If you desire the RX01 compatible mode (MODE 1), then the state of this switch and possibly a jumper on the interface module will have to be changed. Figure 3-9 shows the location of the switch and which position is which. If you are going to operate in RX01 mode, you must install a jumper on the DSD 440-11 or DSD 440-L11 interface modules. Figures 3-4 and 3-6 show the location of that jumper on each of the two modules. Obviously, the jumper should be removed when converting back to Mode 2 operation. No changes are required on the PDP-8 interface module when changing operating mode.



SWITCH SHOWN IN THE "MODE 2"  
RX02 COMPATIBLE POSITION



SWITCH SHOWN IN THE "MODE 1"  
RX01 COMPATIBLE POSITION

FIGURE 3-9 MASTER CONTROLLER BOARD DIP-SWITCH

### 3-7 FINAL INSTALLATION:

Locate the power cord and plug the female end into the connector on the back of the chassis and the other end into a suitable AC receptacle. Route the free end of the interface bus cable over to the rear of the chassis and plug it into the 26 pin connector such that the striped side of the cable is toward the middle of the chassis. Also confirm that the pin position of the clipped pin on the connector on the rear of the chassis matches the position of the plugged up hole in the cable connector.

### 3-8 POWERING UP

The DSD 440 chassis gets power from its own internal power supply. The interface module uses only +5 VDC, and it receives this from the computer backplane. The DSD 440 chassis and the interface module can be powered up in either order without the risk of any adverse affects. There is no danger of writing on diskettes loaded in the drives during power up or power down cycles.

### 3-9 INITIALIZATION RESPONSE CHECK

An initialization response should occur when the DSD 440 is powered up. If the system has been connected to the host computer correctly, an initialization response can be forced as a result of some of the following operator console actions:

#### LSI-11 BASED SYSTEMS:

- (1) Flip the INIT switch (if there is one).
- (2) Using ODT, use the "Q" command at any arbitrary starting address.
- (3) Using ODT, write the number 40000 into the DSD 440 RXCS register, normally at address 177170.

#### PDP-11 BASED SYSTEMS:

- (1) Generate a UNIBUS INIT by depressing the START switch or button.
- (2) Using the console, deposit 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.

An initialization response will only be observed on floppy disk drives if the door is closed. Each time you generate an INIT, you should hear a brief noise come from the flexible disk drives as the controller homes the head positioners. The activity LEDs 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 LED will remain on slightly longer as the controller reads track 1/sector 1 of the diskette into the sector buffer. Be sure that you insert diskettes into the drives as shown in Figure 3-10. Never use hard sectored diskettes in this system. If you did not observe the results described here, please confirm the following:

- 1) You have applied power to both the computer mainframe and the DSD 440 chassis.
- 2) You have connected both ends of the DSD 440 interface bus cable in the proper orientation as directed.
- 3) You are being successful at generating a system initialize or device initialize signal by one of the above methods, and that the signal is reaching the DSD 440 interface.
- 4) The drive doors are not open.

If you are unable to force an initialization response by any of the above mentioned methods, see the maintenance section of this manual or call the Data Systems Design Customer Service Department for assistance.

*Not No. 0,  
(It is No. 1.  
Drive 0 may  
be dead.*

*→ Drive 1*

42,381 50 SHEETS 5 SQUARE  
42,382 100 SHEETS 5 SQUARE  
42,383 200 SHEETS 5 SQUARE  
NATIONAL  
MADE IN U.S.A.

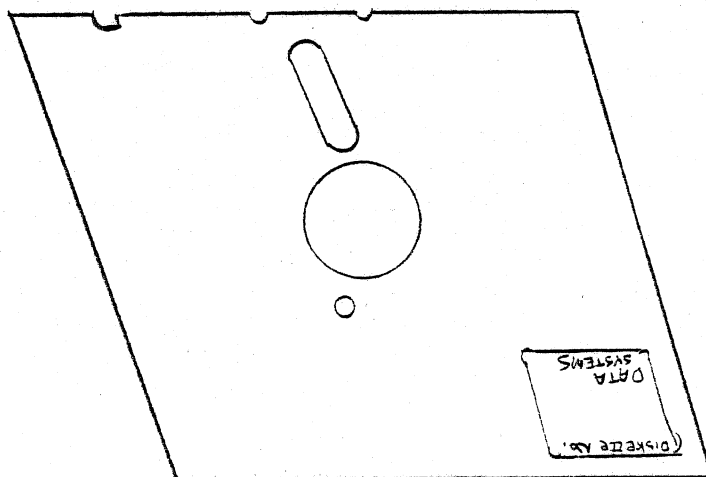
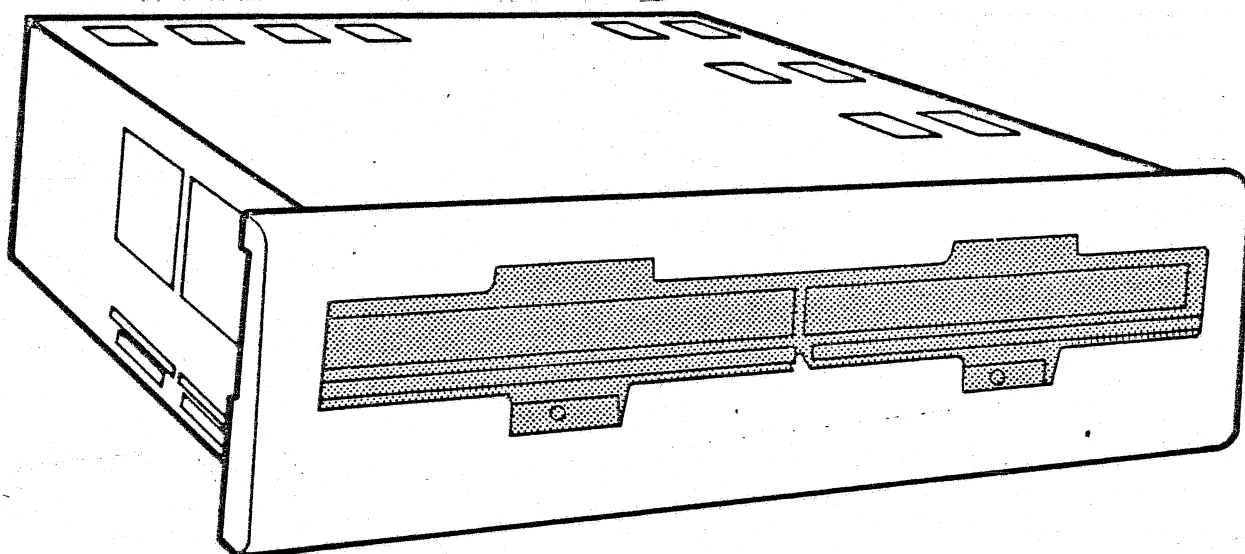


FIGURE 3-10 INSERTING A DISKETTE

### 3-10 SYSTEM BOOTSTRAPPING

Bootstrapping is a term which generally refers to the act of reading in a block of code from some mass storage media, and then having the processor jump to that code. Executing that code, the C.P.U. continues to read the system monitor into memory so that the user can interact with the system through the console keyboard.

#### BOOTSTRAP PROGRAM ON LSI-11 AND PDP-11 INTERFACE MODULES

In addition to bootstrapping the DSD 440, this program executes a number of system diagnostics. Among these are:

- (1) A limited C.P.U. instruction set test
- (2) A test for stuck address and data bits throughout all available memory
- (3) A bit-latch test of the DSD 440 interface registers
- (4) A DSD 440 fill/empty buffer test

Should a malfunction be detected during the execution of any of these tests, the processor will either HALT or hang. You can assume that the processor is "hung" during execution of the bootstrap if the floppy drives are quiet and nothing has been output to the console terminal. In this case, you will have to manually halt the processor to determine the address at which the program was hung. Once the "hang" or "halt" address is known, refer to the bootstrap program listing in the appendix to find out which test failed.

After successful completion of the system diagnostics, the bootstrap program will read track 1/sector 1 of drive 0 into the controller sector buffer. Should this operation induce a density error, the density bit is changed and the command is issued again. If any other error results, the processor will halt leaving the drive number in R0, the memory address of the extended status information in R4, and the definitive error code in R6. If the READ SECTOR operation is successful, the bootstrap program determines the present operational mode of the DSD 440. If the system is configured for RX02 compatible operation, a DMA empty buffer cycle takes place. A programmed I/O cycle takes place if the system is in RX01 compatible mode. At this point, the first word of data transferred to memory (at address 000000) is examined. If that word is a NOP 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 in the operating system. If the bootstrap program does not find a NOP instruction in address 0, it will switch to the other drive and try to bootstrap the diskette it contains.

One normally starts execution of the bootstrap program by loading the program counter with the floppy disk bootstrap program base address. This address is determined by the



position of switches or jumpers on the interface modules (see section 3-4). After loading the starting address, simply start the CPU.

## BOOTSTRAPPING SYSTEMS WITH NON-STANDARD DEVICE ADDRESSES

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

Let us consider a few specific cases to illustrate the different bootstrap starting procedures. Assume that the shunts on the interface module have been set up so that the bootstrap program base address is 173000 and the RXCS = 177170 (standard address). Under these circumstances the system is bootstrapped by starting the computer at the bootstrap program base address, which would be 173000 in this case. If now the interface module is modified so that the RKCS = 177150, the system could be bootstrapped by starting the computer at the bootstrap program base address plus 10 (octal), which would be 173010 in this case. If now the interface module is modified so that the device address is any legal address other than 177150 or 177170, the following steps are required to bootstrap the system: First, write the device address (assume 177160 in this case) into memory address 000000. Second, write the number 000002 into CPU register R1. Finally, start the computer at the bootstrap program base address plus 26 (octal), which would be 173026 in this case.

In order to successfully bootstrap any operating system, the system device handler software (on the operating system diskette) must have been specially adapted to accommodate the non-standard device address for which the floppy disk system hardware has been configured.

### 3-11 ACCEPTANCE TESTING ON PDP-11 AND LSI-11

The ACCEPTANCE test should be performed when the DSD 440 system is first installed or when a fault condition is suspected. To run the test program, locate the diagnostic diskette that was shipped with your system. The diskette should be labeled:

DSD 440-11 DIAGNOSTIC DISKETTE VERSION 6 (OR GREATER).

Insert the diskette in drive 0 and perform the bootstrap procedure described in the previous section. If the system is able to load the diagnostic program into memory, a short paragraph will be typed on the console terminal. Included will be a system memory map for your information. The diagnostic program indicates that it is waiting for a command by typing the prompt:

MODE:

When this word appears on the terminal, remove the diagnostic diskette and insert two scratch diskettes into the drives. These diskettes should not be write protected, as the ACCEPTANCE program will be writing on them. The scratch diskettes can have either single or double density data recorded on them. To start the acceptance test, simply type the character "A".

MODE: ACCEPTANCE

The proper operation of the DSD 440 is tested by running five passes of the acceptance test. Each time a pass is completed, an asterisk will be printed on the console terminal. If there are any errors, they will also be printed on the console terminal. If any errors occur, contact the Data Systems Design Customer Service Department at (408) 249-9353.

More detailed information about the diagnostic program FRD440 on the diagnostic diskette can be found in chapter 6.

### 3-12 ACCEPTANCE TESTING ON PDP-8

## CHAPTER 4 MAINTENANCE FEATURES AND ERROR ANALYSIS

### 4-1 OVERVIEW OF MAINTENANCE FEATURES

The DSD 440 has many built-in maintenance related features. These features were incorporated into the product so that the need for a specially trained field service force would be substantially reduced. If the product should fail in the field, Data Systems expects that service can be rapidly restored using a combination of the diagnostics which run on the host C.P.U. and the hyper-diagnostics which have been built directly into the DSD 440. Data Systems also maintains a telephone "HOT-LINE" to help customers solve problems related to DSD equipment.

An 8-bit microprocessor is the "brains" of the DSD 440 master controller. In addition to performing all of the standard floppy disk functions described in the programmers' interface chapter, this microprocessor executes a great deal of code designed to simplify the job of maintenance. Following every power-up or initialization cycle, a series of hardware "self-test" routines are executed. Also, by changing the position of the switches on the controller, a user can instruct the microprocessor to execute one of several system hyper-diagnostics. These hyper-diagnostics are unique in that they permit the user to thoroughly verify the integrity of a large part of his system without requiring any connection to a host C.P.U. The remainder of this chapter will explain in detail the maintenance features and how they are used.

#### 4-2 NORMAL VS. HYPER-DIAGNOSTIC MODE

The DSD 440 system is said to be in "NORMAL" mode when it is connected to a host C.P.U. and is being used to read and write data on diskettes. Most of the exposure a user has to the DSD 440 system will be through the application software running on the host central processor.

The system is said to be in "HYPER-DIAGNOSTIC" mode when the user has removed the DSD 440 chassis cover and has initiated a particular hyper-diagnostic through the 8 position DIP-SWITCH on the master controller circuit board assembly. The interface bus cable should be disconnected from the rear of the DSD 440 chassis. The user interface to the DSD 440 system is through 9 LED indicators and the DIP-SWITCH on the master controller. Hyper-diagnostic mode is very useful for demonstrating the DSD 440 in the field, mass formatting of diskettes, and verifying the proper operation of parts of the DSD 440 memory system which DO NOT involve the host C.P.U. interface. The individual hyper-diagnostic tests are started and stopped by cycling the main AC power switch located on the rear of the chassis.

#### 4-3 INDICATOR LEDS, DRIVE ACTIVITY LEDS, AND DIP-SWITCH

Figure 4-1 shows the relative location of the 9 indicator LEDs and the DIP-SWITCH on the master controller circuit board assembly. 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 2-8 are all red, and are located adjacent to LED 1. The meanings of LEDs 1-8 will vary according to whether the system is in "NORMAL" or "HYPER-DIAGNOSTIC" mode; and whether the microprocessor is running or halted. LED 9, which also is green, is ON when the microprocessor is running. Conversely, it is OFF when the microprocessor is halted. LED 9 will sometimes be referred to as the "RUN" LED.

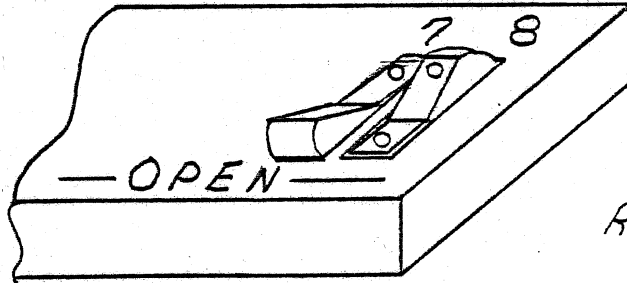
Note: If there is ever doubt as to whether a particular LED indicator is ON or OFF, this discrepancy can usually be eliminated by viewing the indicator from directly above. This is especially true of the green indicators, since they are not quite as bright as the red ones.

The drive activity LEDs are mounted in the diskette eject button on the front of each disk drive. These LEDs are mostly used to indicate when the head is loaded against the media and the drive door should not be opened. When the system is operating in "NORMAL" mode, these LEDs may be flashed on and off at about a 1 Hz. rate to indicate an error condition. This flashing will continue until an INIT occurs or two minutes have elapsed.

The 8 position DIP-SWITCH is the principal mechanism by which the user communicates with the microprocessor when the normal programmers' interface is not available. Figure 4-1 shows exactly how the switches are numbered, and which physical position of a switch corresponds to a binary "1" and which position corresponds to a "0". It is very easy to become confused about the switch position conventions defined by this illustration, so it would be worth your while to spend some time studying Figure 4-1.

SWITCH 7 IS CLOSED = 0  
SWITCH 8 IS OPEN = 1

### SWITCH INTERPRETATION



RUN LED ; GREEN

LED 1 ; GREEN  
LEDS 2-8 ; RED

DRIVE BUS CONNECTOR

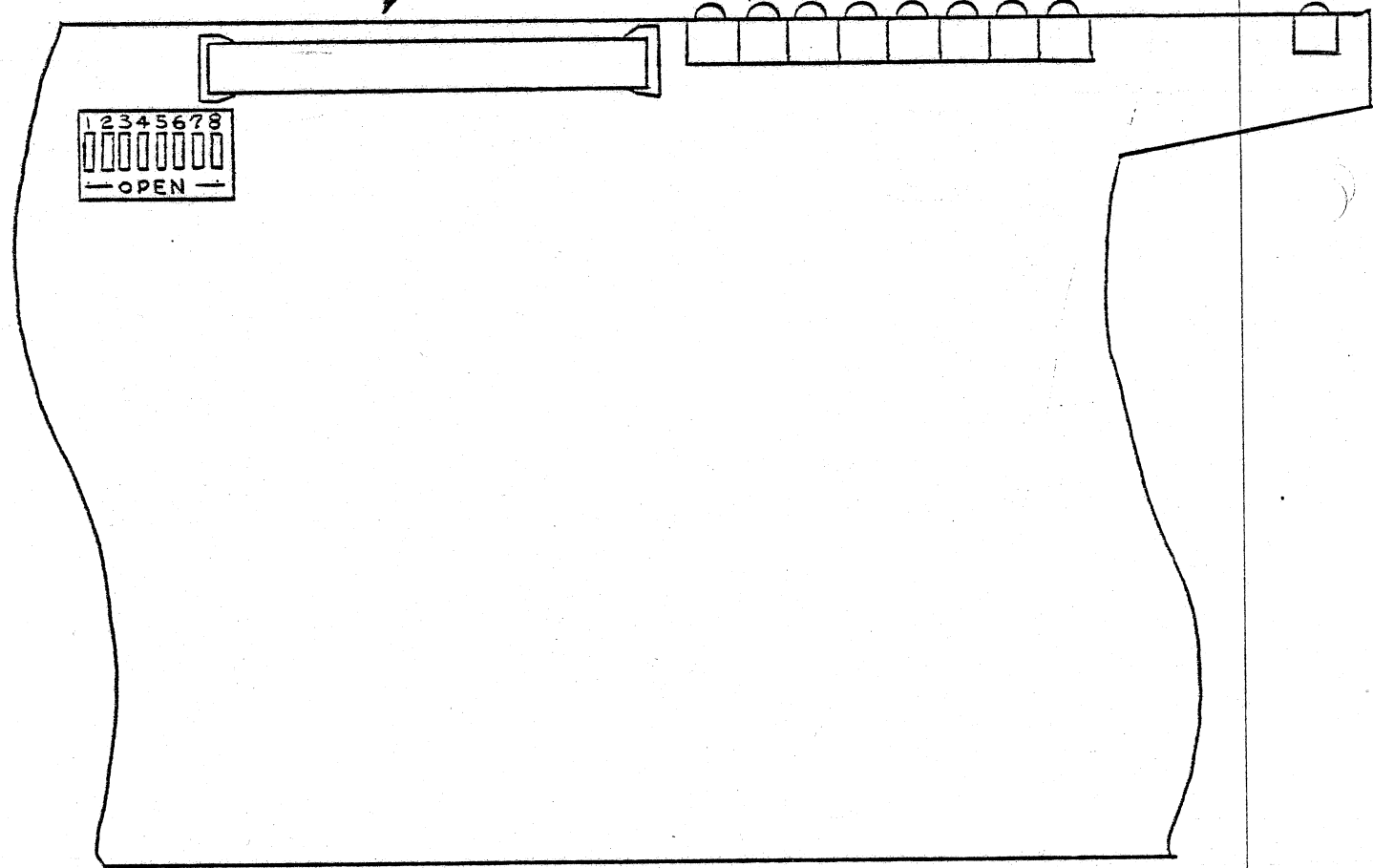
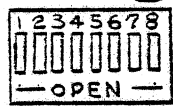


FIGURE 4-1 SWITCH AND LED ORIENTATION

42,381 50 SHEETS 5 SQUARE  
42,382 100 SHEETS 5 SQUARE  
42,383 200 SHEETS 5 SQUARE  
NATIONAL

#### 4-4 LED MEANINGS DURING "NORMAL" MODE

This section gives the meanings of LED 1 - LED 8 when the DSD 440 system is under control of the host computer ("NORMAL" mode). Naturally, the chassis cover will have to be removed in order to see these LEDs.

LED 1, when on, indicates that the DSD 440 system is currently operating in "NORMAL" mode. LED 1 is green. See section 4-2 for a review of what is meant by "NORMAL" mode.

LED 2, when on, indicates that the controller microprocessor is currently waiting for the host C.P.U. to issue a new command, write a parameter to the data buffer register, or read/write a data byte from/to the data buffer register.

LED 3, when on, indicates that the controller is currently in the process of writing on a diskette.

LED 4, when on, indicates that the controller is currently in the process of reading from a diskette.

LED 5 - LED 8 are used to display an "error class" code. When a LED is on, this corresponds to a binary 1, and when it is off, this 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 C.P.U. on command. The errors were grouped into 16 "classes" so that all possible errors could be visually coded using only four LED indicators. See Table 5-1 for a more detailed explanation of the ERREG codes referenced in Table 4-1.

110

BINARY  
5678

MEANING

*See page 110 about ERREG*

0000	NO ERRORS HAVE OCCURRED SINCE POWER ON
0001	PROGRAMMING ERROR-WRITE PROTECT (ERREG = 100)
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	DRIVE READ SIGNAL LOST (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, 300 OR 310)
1110	INTERFACE PARITY ERROR (ERREG = 210)
1111	AC POWER LOW ABORT OF WRITE OR FORMAT (ERREG = 370)

TABLE 4-1 ERROR CLASS CODES IN "NORMAL" MODE (BOTH GREEN LEDS ON)

The error class code will be displayed in the LEDs as soon as the error is detected. The code will be reset to zero if the power is cycled or 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 will start flashing the drive activity LED of the drive associated with the error at roughly a 1 HZ rate. This flashing will stop when either a system initialize is forced by the host C.P.U., or roughly two minutes passes.



#### 4-5 DIP-SWITCH SETTINGS DURING "NORMAL" MODE

The 8 rocker switches in the "DIP-SWITCH" have the following meanings when the system is being operated in "NORMAL" mode.

Switch 1, switch 2, and switch 3 must all be 0. When these three switches are configured this way, the microprocessor knows that the system is to operate in "NORMAL" mode.

Switch 4 tells the microprocessor which operational mode is to be used when communicating with the interface module. Chapter 2 explains operational mode and section 3-6 discusses changing it. When switch 4 is a 0, the system is configured in Mode 2 (RX02 compatible). When switch 4 is a 1, the system is configured in Mode 1 (RX01 compatible).

Switch 5 is called the drive mapping switch. 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 floppy disk drive to drive 0 in the event that the normal drive 0 fails.

Right: 0  
Left: 1

Switches 6 and 7 are not currently used to indicate anything in "NORMAL" mode.

Switch 8 is used to encode the number of floppy disk drives that are connected to the controller. When switch 8 is a 0, this indicates one drive. When switch 8 is a 1, this indicates 2 drives.

Data Systems ships the standard DSD 440 system with all switches in the 0 position EXCEPT switch 8. This means: NORMAL mode, Mode 2 (RX02 compatible), normal drive mapping, 2 floppy disk drives. (See Figure 4-2)

42-381 50 SHEETS 5 SQUARE  
42-382 100 SHEETS 3 SQUARE  
42-383 100 SHEETS 3 SQUARE  
NATIONAL

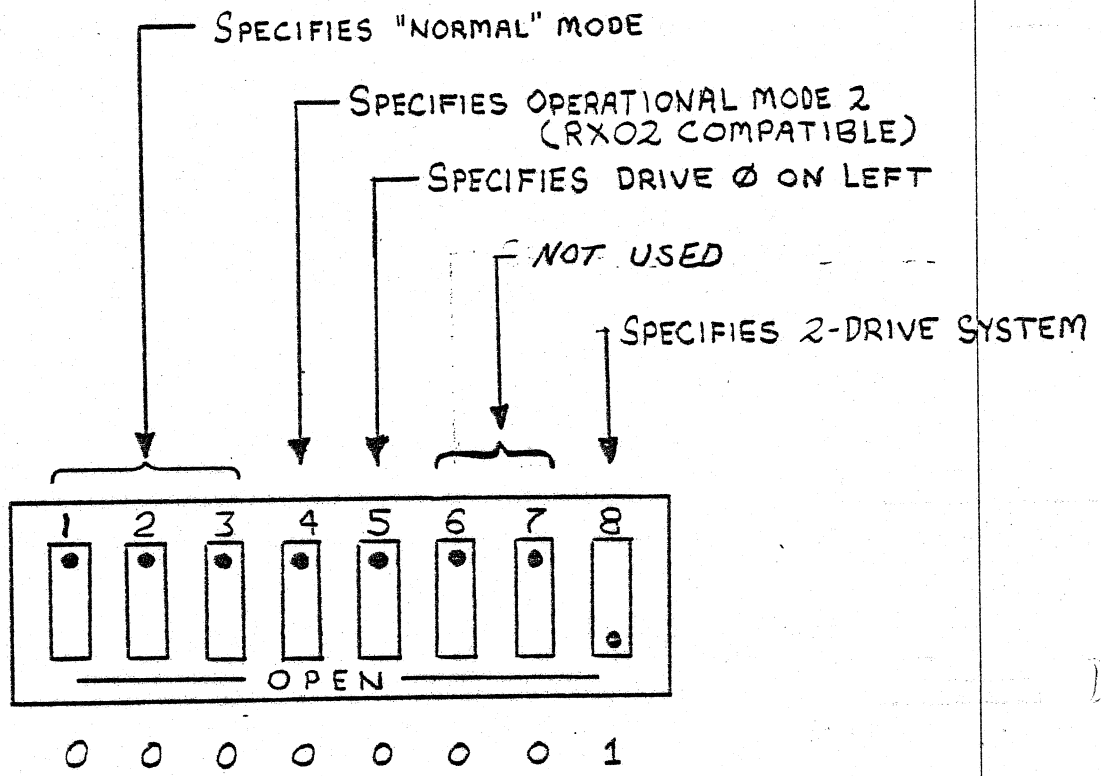


FIGURE 4-2 DIP-SWITCH SETTING WHEN PRODUCT SHIPPED

*Black dot is on depressed side of switch*

#### 4-6 HARDWARE SELF-TEST ROUTINES

The microprocessor always executes a number of system hardware tests following power-up or an initialization. This is true even when the system is being used in "HYPER-DIAGNOSTIC" mode. These tests are called "PASSIVE" tests because there is no way that the user can inhibit them from executing, and there is no way that the user can operate the system should one of these tests detect a malfunction. Just before each hardware test routine is executed, the microprocessor writes the error code associated with the failure of that particular test in LEDs 5-8. In the event the test detects a malfunction, the microprocessor does a hard HALT leaving the error code displayed. The codes and their interpretation are shown in Table 4-3. The user knows that the code being displayed in the LEDs is from Table 4-3 if the green RUN LED is OFF and LED 2 is ON. Also, LEDs 1, 3, and 4 should all be OFF.

BINARY 5678	MEANING
0000	UNASSIGNED
0001	THIS ERROR CODE NOT CURRENTLY ASSIGNED
0010	THIS ERROR CODE NOT CURRENTLY ASSIGNED
0011	MICROPROCESSOR (P)ROM CHECKSUM ERROR
0100	THIS ERROR CODE NOT CURRENTLY ASSIGNED
0101	ERROR DURING TEST OF THE INTERFACE SHIFT REGISTER
0110	PROGRAMMABLE I/O PORT FAILURE IN 8155 CHIP
0111	PARITY LOGIC/LATCHED INIT LOGIC TEST FAILURE
1000	RAM TEST FAILED IN THE 2111 CHIP(S)
1001	RAM TEST FAILED IN THE 8155 CHIP
* 1010	PHASE-LOCKED-LOOP TEST FAILURE
* 1011	READ/WRITE CONTROLLER TEST FAILURE
* 1100	CRC/SERIAL DATA PATH TEST FAILURE
1101	COUNTER/TIMER FAILURE IN 8155 CHIP
1110	INVALID SWITCH SETTING (POSSIBLY BAD HYPER-DIAGNOSTIC CODE)
1111	8085 CPU TEST FAILURE

TABLE 4-3 SELF-TEST ERROR CODES

\* These three error codes can only occur following an INIT if the system is configured for "NORMAL" operation. If the system is configured for running hyper-diagnostics, these codes can only appear if the corresponding hyper-diagnostic test has been selected. The hardware self-test loop hyper-diagnostic test will never test these functions and thus cannot produce these error codes. (See section 4-10)

Should a passive test error occur, and the solution to the problem not be obvious, try cycling the main power several times. If the error is consistent and the solution is not evident, call the Data Systems Design Customer Service Department for assistance.

#### 4-7 HYPER-DIAGNOSTIC MODE

As mentioned earlier, hyper-diagnostic mode is used when the user wants to adjust, exercise, or test his controller and/or drives independent of a host computer system and the associated software. The DSD 440 chassis need only be connected to the AC power in the wall to run the hyper-diagnostics. The user selects particular tests and particular floppy disk drives using the DIP-SWITCH on the controller board. Test results are observed through a combination of the 9 indicator LEDs on the controller board and perhaps an oscilloscope. After the switch and LED conventions are explained, the details of each of the hyper-diagnostic routines will be discussed.

#### NOTE:

With the exception of the cable orientation test, all of the hyper-diagnostic tests are to be executed with the interface bus (IBUS) cable disconnected from the DSD 440 chassis.

#### 4-8 DIP-SWITCH SETTINGS DURING "HYPER-DIAGNOSTIC" MODE

Switch 1 through switch 5 are used to encode the desired test. Since 5 individual switches constitutes 5 bits, one would assume that 32 individual tests could be encoded in these switches. This is not true however because any time switches 1, 2, and 3 are all zeros, the microprocessor assumes "NORMAL" mode operation, as previously explained in section 4-5. Switch 1 is the MSB and switch 5 is the LSB.

Switches 6 and 7 are not used to indicate anything during hyper-diagnostic mode, except during the DIP-SWITCH / LED test.

Switch 8 is used to encode the particular floppy disk drive that is to be used for a given test. When the switch is a 0, drive 0 is selected, when it is a 1, drive 1 is selected. Note that not all of the tests involve a drive, so in some cases the position of switch 8 will be irrelevant. The general exerciser tests are capable of exercising more than one drive. These two tests (switch codes 11110 and 11111) interpret switch 8 as the number of drives to be exercised. In other words, 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 that there is no drive mapping function available in hyper-diagnostic mode. This function is only available when the system is being operated in "NORMAL" mode.

To run a particular hyper-diagnostic, one first powers down the controller/drive subsystem possibly using the AC switch conveniently mounted on the rear of the chassis. Next, set the 8 switches to reflect the desired test, and in some cases, the desired drive. To start the test, simply turn the power back on. One should not play with the DIP-SWITCH settings while power is on EXCEPT where explicitly directed to do so in the explanation of a particular test. An example hyper-diagnostic DIP-SWITCH configuration is shown in Figure 4-3.

After extended use, the reliability of the DIP-SWITCH may become questionable. A switch that "appears" to be in the shorted state may in fact be in the open state. If you ever have reason to believe that the microprocessor is mis-interpreting the byte encoded in the DIP-SWITCH, always use a pointed object (such as a ball point pen) to depress the rocker switches. Another technique that can be used to confirm your switch settings is the DIP-SWITCH/LED hyper-diagnostic. The code for this test is (10000). Once the microprocessor recognizes this test code, all it does is read the DIP-SWITCH and echo 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 hyper-diagnostic test indicated by the switches can be executed by simply powering the unit down, and then up again.

#### 4-9 LED MEANINGS DURING "HYPER-DIAGNOSTIC" MODE

Except in the tests where noted otherwise, the LEDs will work as follows. LED 1 and LED 2 will be off to indicate that the microprocessor is in hyper-diagnostic mode and is not currently executing any of the hardware self-test routines described in section 4-6. LEDs 3 and 4 work the same way they did in "NORMAL" mode. When LED 3 is on, the system is currently writing on a diskette. When LED 4 is on, the system is currently reading from a diskette. These two LEDs can be very useful when trying to decide when to stop a particular hyper-diagnostic by turning off the AC power. In general, it is not a good idea to turn off the power while the WRITE LED is still on.

Just as in the hardware self-tests, the microprocessor will halt whenever it detects an error. The user knows when the microprocessor is halted by observing the green "RUN" LED shown in Figure 4-1. An error code is displayed in LEDs 5-8 when the microprocessor detects an error and halts. The error codes are similar to the error class codes discussed in section 4-4. Errors that involve a host computer interface, such as non-existent memory and parity errors, could never occur during execution of any of the hyper-diagnostics. Table 4-5 shows the code interpretations. The activity LED of the drive selected when the failure was detected will be left on.

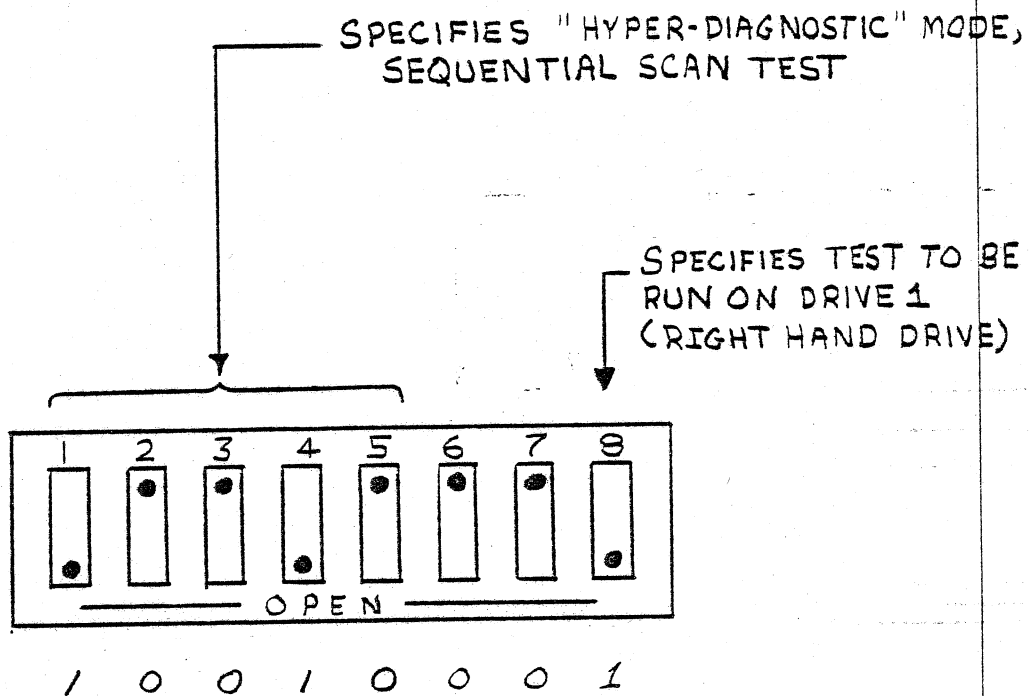


FIGURE 4-3 EXAMPLE HYPER-DIAGNOSTIC  
DIP-SWITCH CONFIGURATION

BINARY 5678	MEANING
0000	NO ERRORS HAVE OCCURRED SINCE POWER ON OR LAST INIT
0001	OPERATOR ERROR - WRITE PROTECT VIOLATION (ERREG = 100)
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
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 CONTR. FAILURE (ERREG = 220, 320, OR 330)
1100	MASTER CONTROLLER FAILURE (ERREG = 340)
1101	DRIVE FAILURE (ERREG = 010, 020, 030, 050, 300 OR 310)
1110	DATA PATTERN READ NOT THE SAME AS PATTERN WRITTEN
1111	AC POWER LOW ABORT OF WRITE OR FORMAT (ERREG = 370)

TABLE 4-5 HYPER-DIAGNOSTIC ERROR CODE INTERPRETATIONS



The following note of caution should be heeded here: The hardware self-tests and the hyper-diagnostics report errors by writing an error code in LEDs 5-8 and then halting. The microprocessor will execute the hardware self-tests before it gets to the hyper-diagnostic routine encoded in the switches. The only way to tell if the error code displayed in LEDs 5-8 is from Table 4-5 or Table 4-3 is by looking at LED 2. LED 2 will be ON following an error caused by the hardware self-test routines. LED 2 is OFF following an error caused by most of the hyper-diagnostic routines. The exceptions are noted in the test explanations.

LEDs 5-8 have a different meaning when the microprocessor is executing a hyper-diagnostic routine but has not yet detected any error. One knows that this is the condition by observing that the "RUN" LED is still ON. At this time, LEDs 5 and 6 encode density, and LEDs 7 and 8 encode selected drive. Coding is as follows:

LED 5	LED 6	DENSITY
OFF	OFF	DENSITY UNKNOWN (DRIVE PROBABLY NOT READY)
OFF	ON	IBM 3740 SINGLE DENSITY
ON	OFF	DEC DOUBLE DENSITY

TABLE 4-6 INTERPRETATION OF LEDS 5 AND 6 DURING EXECUTION OF HYPER-DIAGNOSTIC ROUTINES (EXCEPT WHERE OTHERWISE NOTED)

LED 7	LED 8	SELECTED DRIVE
OFF	OFF	0
OFF	ON	1

TABLE 4-7 INTERPRETATION OF LEDS 7 AND 8 DURING EXECUTION OF HYPER-DIAGNOSTIC ROUTINES (EXCEPT WHERE OTHERWISE NOTED)

#### 4-10 DESCRIPTION OF INDIVIDUAL HYPER-DIAGNOSTIC TESTS

Table 4-8 shows the DIP-SWITCH settings for all of the hyper-diagnostic routines implemented in the DSD 440.

SWITCHES    HYPER-DIAGNOSTIC NAME  
12345

---

00100	HEAD LOAD TIMING ADJUSTMENT ROUTINE
00101	TRACK 00 DETECTOR ASSEMBLY ADJUSTMENT ROUTINE
00110	SEEK TRACK 01 AND LOAD HEAD
00111	SEEK TRACK 02 AND LOAD HEAD
01000	SEEK TRACK 38 AND LOAD HEAD
01001	SEEK TRACK 76 AND LOAD HEAD
01010	CODE NOT ASSIGNED
01011	CODE NOT ASSIGNED
01100	TEST READ/WRITE CONTROLLER
01101	TEST PHASE LOCKED LOOP/CRC GENERATOR
01110	TEST CABLES
01111	LOOP ON HARDWARE SELF-TEST ROUTINES
10000	DIP-SWITCH / LED TEST
10001	BUTTERFLY SEEK TEST
10010	SEQUENTIAL SCAN TEST
10011	BUTTERFLY SCAN TEST
10100	CODE NOT ASSIGNED
10101	CODE NOT ASSIGNED
10110	CODE NOT ASSIGNED
10111	CODE NOT ASSIGNED
11000	SEQUENTIAL WRITE/READ TEST
11001	WRITE-FORMAT SINGLE DENSITY DISKETTE
11010	SET MEDIA DENSITY (SINGLE DENSITY)
11011	SET MEDIA DENSITY (DOUBLE DENSITY)
11100	CODE NOT ASSIGNED
11101	CODE NOT ASSIGNED
11110	GENERAL EXERCISER STARTING WITH WRITE-FORMAT SINGLE DENSITY
11111	GENERAL EXERCISER

TABLE 4-8 DIP-SWITCH CODES FOR HYPER-DIAGNOSTIC ROUTINES

## FLOPPY DISK DRIVE ALIGNMENT ROUTINES (00100-01001)

The first six test routines about to be described are intended for use with a special alignment diskette (PART NUMBER SA120-1) available from Shugart Associates. To perform some of the alignment procedures, it will be necessary to remove several screws in the bottom of the DSD 440 chassis so that the drives can be propped up on their side. Many of the adjustment screws and oscilloscope test points are located on the underside of the drives. This section will only describe what the six floppy disk alignment routines built into the DSD 440 do. The actual alignment procedures can be found in a document published by Shugart Associates, which was included in your DSD 440 documentation binder. The document is called the SAS800/801 DISKETTE STORAGE DRIVE MAINTENANCE MANUAL.

### HEAD LOAD ACTUATOR ADJUSTMENT ROUTINE (00100)

This routine starts by "homing" the selected drive to track 00. Once there, the head is loaded and unloaded at roughly a 5 Hz. rate. The head is loaded for 100 milliseconds, and then unloaded for 100 milliseconds before the cycle is repeated. As mentioned earlier, the routine is terminated by disconnecting AC power from the chassis. Paragraph 4.5.3 in the Shugart manual calls for a routine such as this one.

### TRACK 00 DETECTOR ASSEMBLY ADJUSTMENT ROUTINE (00101)

This routine starts by "homing" the selected drive to track 00. The head is then alternately moved between track 01 and track 02 about once every 70 msec. The head is loaded during this test. The routine is terminated by disconnecting AC power. Paragraph 4.11.8 in the Shugart manual calls for a routine such as this one.

### SEEK TRACK 01 AND LOAD HEAD (00110)

This routine starts by "homing" the selected drive 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 the Shugart manual calls for a routine such as this one.

### SEEK TRACK 02 AND LOAD HEAD (00111)

This routine starts by "homing" the selected drive 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 the Shugart manual calls for a routine such as this one.

#### SEEK TRACK 38 AND LOAD HEAD (01000)

This routine starts by "homing" the selected drive 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 the Shugart manual.

#### SEEK TRACK 76 AND LOAD HEAD (01001)

This routine starts by "homing" the selected drive 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 paragraph 4.11.9 of the Shugart manual.

#### TEST READ/WRITE CONTROLLER (01100)

This routine causes the read/write controller state machine to be continuously cycled through its internal self-test microcode. This test should be run if there is reason to believe that the read/write controller state machine is not performing reliably. Should this test generate an error, the code is shown in Table 4-3 as a 1011. Note that LED 2 will also be on if this error is reported.

#### TEST PLL/CRC GENERATOR (01101)

This routine checks the operation of the phase locked loop circuitry by counting the number of PLL VCO cycles that occur during a 50 millisecond interval. This test should be run to determine if a READ problem is being caused by the PLL circuitry. If it is, the error code will be shown in Table 4-3 as a 1010. Note that 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 is functioning properly. If this test detects a malfunction, the error code is shown in Table 4-3 as a 1100. LED 2 will also be on.

#### TEST CABLE ORIENTATION

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 not installed backwards. Note that 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 hyper-diagnostic tests EXCEPT this one. If there is an error, Table 4-5 indicates which cable is causing the problem.

#### HARDWARE SELF-TEST LOOP (01111)

The DSD 440 microprocessor executes the hardware self-test once after powering up. When this hyper-diagnostic routine is selected, the microprocessor will keep executing the hardware self-test over and over again indefinitely until either power is removed or an error is detected. LEDs 5, 7, and 8 should be flashing 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 correctly read all 8 switches in the DIP-SWITCH, and correctly illuminate LED's 1-8. The routine will simply read the DIP-SWITCH and write that byte to the LED bank. As an example, if switch 2 were in the 1 position, then LED 2 should be on. If switch 2 were in the 0 position, then LED 2 should be off. Since the microprocessor is executing this loop continuously, the state of a LED should appear to change very shortly following a switch position change. The test is terminated by removing power.

#### BUTTERFLY SEEK TEST (10001)

This routine starts by "homing" the selected drive to track 00. The head positioner is then moved back and forth in what has been called 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 seek the positioner to track 00. If the track 00 signal is not asserted, the error code is reported in the LEDs and the microprocessor halts. If the track 00 signal is asserted, the test is repeated. The head is not loaded at any time during this test so no reading or writing is done.

#### SEQUENTIAL SCAN TEST (10010)

This routine starts by "homing" the selected drive to track 00. The head is then loaded and the media density is determined and displayed in the LEDs (see Table 4-6). The controller then sequentially reads every sector of every track. If no errors occur, this scan routine will cycle over and over until power is removed. If an error does occur, the processor will halt and the error code will be displayed in LEDs 5-8. The meanings associated with the 16 possible error codes are shown in Table 4-5. Remember, the code displayed in LEDs 5-8 is not an error code UNLESS the green CPU run LED is out and the head positioner is not moving. The meaning of LEDs 5-8 are shown in Tables 4-6 and 4-7 when the test is still executing.

#### BUTTERFLY SCAN TEST (10011)

This routine is similar to the sequential scan test, except that the sequence of tracks read is 76, 01, 75, 02, 74 ..... This test will take much 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 will detect problems associated with seeking and/or reading.

#### SEQUENTIAL WRITE/READ TEST (11000)

This routine starts by "homing" the selected drive to track 00. Next, the density of the 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. Any error encountered will be reflected in the LEDs when it occurs and the machine will halt. 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 can be used to determine if the routine has completed the write cycle.

#### WRITE-FORMAT DISKETTE IN 3740 S. D. FORMAT (11001)

This routine starts by "homing" the selected drive to track 00. Next, the entire 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 error or power disconnect. LEDs 3 and 4 can be used to determine when the write cycle has been completed.

#### SET MEDIA DENSITY (TO SINGLE DENSITY) (11010)

This routine starts by "homing" the selected drive to track 00. Next, every sector on the diskette is written with a single density data address mark, 128 bytes of 0's, and 2 CRC bytes. Unlike the previous write-format routine, this routine does not modify the sector headers. Control is transferred to the sequential scan test as soon as all sectors have been written. LEDs 3 and 4 can be used to determine when the writing has stopped and the reading has begun.

## SET MEDIA DENSITY (TO DOUBLE DENSITY) (11011)

This routine starts by "homing" the selected drive to track 00. Next, every sector on the diskette is written with a double density data address mark, 256 bytes of DEC MMFM 0's, and 2 CRC bytes. This routine does not modify the sector headers. Control is transferred to the sequential scan test as soon as all sectors have been written. LEDs 3 and 4 can be used to determine when the writing has stopped and the reading has begun.

## GENERAL EXERCISER STARTING WITH WRITE-FORMAT S.D. (11110)

This test is designed to exercise all parts of a DSD 440 system as thoroughly as possible. Unlike the other hyper-diagnostic routines, this routine and the following routine can operate on multiple drives. Switch 8 is used to specify the drives 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 that are to be exercised be 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 3740 S.D. 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 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); otherwise go to step (3)

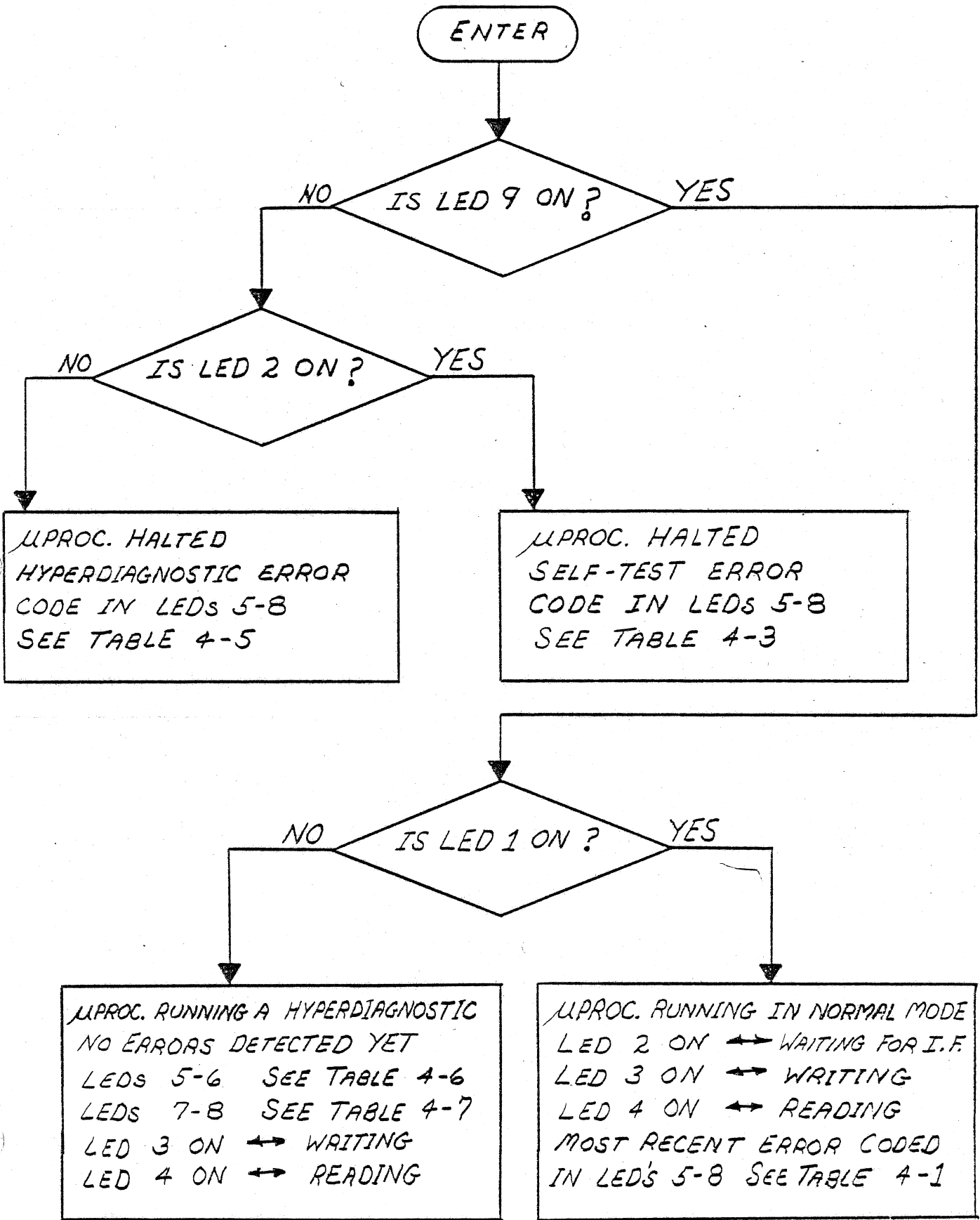
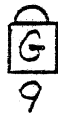
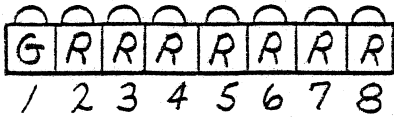
## GENERAL EXERCISER (11111)

This test is the same as the previous one (11110), but minus the single density write-format routine indicated in step (2).

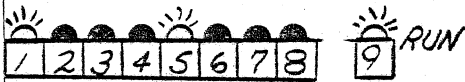
#### 4-11 HOW TO READ THE INDICATOR LEDS

The nine indicator LEDs on the master controller circuit board assembly encode different information at different times. Although all of the encoding algorithms have been explained in previous sections of this chapter, they will be reviewed in this section using a "flow-chart" technique. By following the steps of this flow-chart, a minimum amount of time is required to determine the information encoded in the LEDs at any given time. Four examples are given in Figure 4-4. To test your understanding of the flow-chart, try covering up the right side of Figure 4-4 and interpreting each of the four LED patterns. When you are done, verify that your interpretation agrees with the ones listed on the right side of Figure 4-4.

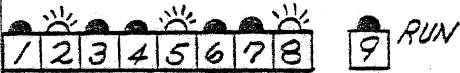




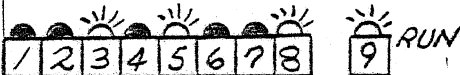
## LED INTERPRETATION EXAMPLES:



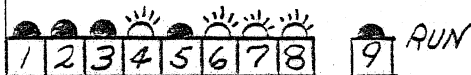
"NORMAL" MODE; NOT WAITING, WRITING, OR READING; LAST ERROR WAS DATA CRC ERROR (200<sub>B</sub>); CONTROLLER  $\mu$ PROC. STILL RUNNING



SELF-TEST FAILURE; BAD 2111 RAM CHIP DETECTED;  $\mu$ PROC. HALTED



"HYPER-DIAGNOSTIC" MODE; WAITING; DEC 2D FORMAT; DRIVE 1 SELECTED;  $\mu$ PROC. STILL RUNNING



"HYPER-DIAGNOSTIC" MODE; WAS READING; HEADER CRC ERROR DETECTED;  $\mu$ PROC. HALTED

FIGURE 4-4

42 301 50 SHEETS 5 SQUARE  
 42 302 100 SHEETS 5 SQUARE  
 42 303 100 SHEETS 3 SQUARE  
 MADE IN U.S.A.  
 BENTON & BOWLES  
 ADVERTISING AGENCY

## SECTION 5 DSD 440 PROGRAMMING INFORMATION

The programmers' interface to the DSD 440 varies depending on which host computer family (such as the DEC 8's and 11's) and in which operational mode the system is configured. The characteristics of each programmers' interface are determined by the controller. The controller can utilize five separate IBUS protocols to communicate with the interface module and in turn the host computer program. This section will be organized by computer family and operational mode. MODES 1 and 2 of the PDP-11 and LSI-11 will be discussed separately. Since the PDP-8 is not capable of DMA, even in MODE 2, the programmers' interface does not differ substantially between MODE 1 operation and MODE 2 operation. For this reason, both operational modes of the PDP-8 interface will be discussed together, with the differences being emphasized.

### 5-1 DEC 11 FAMILY PROGRAMMERS' INTERFACE

For a person primarily interested in programming, there is little difference between the PDP-11 and the LSI-11. The programmers' interface to the DSD 440 flexible disk system is the same for both computers. This section will present an overview of that programmer interface.

Data is transferred to and from the flexible disk or "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 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 operation positions the read/write head of a flexible disk drive over the specified portion of the diskette, and then 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 memory. The CONTROL and STATUS REGISTER is normally located at address 777170. The DATA BUFFER REGISTER is normally located at address 777172. There are a total of seven "logical registers" that will be referred to 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. Associated with each task is a specific "protocol". A protocol is defined as a set of rules which determine the parameters or data the computer should be passing through the data buffer register at any time during the execution of a task. As an example, operations which cause the read/write head in the flexible disk drive to move will 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 operational modes which were discussed in chapter 2 influence the protocol that is associated with the various flexible disk tasks. The main differences in these modes center in two areas. First, in Mode 1 programmed I/O is used exclusively for the transfer of both data and parameters between computer and controller. In Mode 2, programmed I/O is used to transfer parameters, but DMA is used to transfer data between controller and main memory. Second, in Mode 1 data is recorded on diskette in single density only. In Mode 2, data can be recorded on diskette in either single or double density. This chapter on programming has been divided up by operational mode. It is suggested that the programmer determine the operational mode he is going to be using and then read the corresponding section of this chapter.

## 5-1.1 MODE 1 OPERATION

The system assumes MODE 1 operation when the "RX01" switch (located on the main controller board) is placed in the "1" position. (see section 2-1,4-5) Any program that runs successfully with the DEC RX-11 (or RXV-11) will run equally well on a DSD440 system configured to operated in this mode.

### 5-1.1.1 PERIPHERAL DEVICE REGISTERS

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

COMMAND AND STATUS REGISTER (RXCS = 777170)  
DATA BUFFER REGISTER (RXDB = 777172)

Peripheral device registers reside in the top 4K words of the 11-family computer's 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 these registers non-standard bus addresses, see section 3-4.

#### 5-1.1.1.1 COMMAND AND STATUS REGISTER

Writing the bits of this register controls the DSD 440. The format for this register is shown in figure 5-1. The RXCS register also provides important status information and error indications when read by the program.

INSERT FIGURE 5-1 ON THIS PAGE

Format for RXCS Register (RXCS = 177170)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ER	IN							TR	IE	UN2 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 110. 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 this 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 to the controller. Read only.
6	IE — INTERRUPT ENABLE causes 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 <ul style="list-style-type: none"> <li>0 = FILL SECTOR BUFFER from memory</li> <li>1 = READ SECTOR BUFFER into memory</li> <li>2 = WRITE SECTOR BUFFER to disk</li> <li>3 = READ SECTOR from disk to SECTOR BUFFER</li> <li>4 = Not used</li> <li>5 = READ STATUS (RXDB — RXES)</li> <li>6 = Write sector with deleted data address mark</li> <li>7 = READ ERROR REGISTER (RXER — RXDB)</li> </ul> <p>Function select bits are write only.</p>
0	EX — Execute, when set, causes the function coded in RXCS bits 3-1 to be executed.

FIGURE 5-1 COMMAND AND STATUS REGISTER; MODE 1

42 SHEETS 5 SQUARE  
 43 SHEETS 100 SHEETS 5 SQUARE  
 44 SHEETS 200 SHEETS 5 SQUARE  
 NATIONAL

#### 5-1.1.1.2 DATA REGISTER

This register provides the general purpose communication link between the host processor and the DSD 440 system. The information passed through this register is based upon a predetermined protocol as defined in the Section 5-1.1.2.

If the DSD 440 is not in the process of executing a command, the RXDB can be written without the risk of any adverse affects. 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 adhered to. Only RXDB bits 0-7 will be accepted by the controller. Bits 8-15 will be ignored.

The following descriptions explain the various formats of the Data Buffer Register. (RXDB)

#### 5-1.1.1.3 DATA BUFFER REGISTER

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

#### 5-1.1.1.4 FLOPPY 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 TRACK ADDRESS REGISTER. (RXTA = 777172) Track numbers from 0-76 (decimal) are valid.

#### 5-1.1.1.5 FLOPPY 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 SECTOR ADDRESS REGISTER. (RXSA = 777172) Sector addresses from 1-26 (decimal) are valid.

#### 5-1.1.1.6 SYSTEM ERROR AND STATUS REGISTER

The RXES register provides status and error information about the drive that has been selected in bits 4 and 5 of the RXCS register. At the completion of a command, the controller will place the RXES register into the data buffer register (RXDB = 777172) so that the host processor can check the status of the most recent operation. See figure 5-2 for detailed definitions of the individual bits.





## 5-1.1.2 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 I/O register. 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 may result in lost or incorrect data.

### 5-1.1.2.1 FILL SECTOR BUFFER (000)

The FILL SECTOR BUFFER command is used to fill a storage buffer inside the DSD 440 with 128 8-bit bytes of data from the host processor. Other functions can later be used to either write that data to 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 8 bits of the RXDB register by the host processor. When the processor writes a byte into the RXDB register, the TRANSFER REQUEST flag will be cleared. When the TRANSFER REQUEST flag is again set by the controller, another byte of data should be 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 Section 5-1.1.2.7 READ ERROR REGISTER).
- 3) The controller will ignore all data sent after the 128th byte.
- 4) Since the FILL BUFFER command is not associated with any one drive, RXCS bits 4 and 5 do not affect this function.
- 5) Interrupts are generated by the logical AND of DONE and interrupt enable. If the DONE bit is already set the first time the programmer sets the interrupt enable bit, a spurious interrupt should be expected.

#### 5-1.1.2.2 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. When the host computer reads the byte, the TRANSFER REQUEST flag is cleared. The TRANSFER REQUEST flag will again be 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.

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

### 5-1.1.2.3 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 RXES register and the DONE flag. Next, the controller sets the TRANSFER REQUEST flag, RXCS bit 7, to request a sector address. The program should respond 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 should respond 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 will cause the selected drive to seek to the desired track and locate the desired sector. TRANSFER REQUEST will stay unasserted for the remainder of the function. If the correct track and sector are found, the controller will write the 128 bytes of data from the sector buffer, plus two bytes of CRC onto the diskette. When this is finished, the controller completes the function by writing the RXES to the data buffer register and setting the DONE flag. As in all functions, an interrupt request will be 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(8). If the specified sector cannot be found within two diskette revolutions, the RXES is set to a 70(8). 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. 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 the sector number written into the RXSA is 152 (octal) the WRITE SECTOR function becomes a WRITE FORMAT TRACK function. Track formatting is explained in section 5-1.1.2.9 .

#### 5-1.1.2.4 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 should respond 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. This clears the TRANSFER REQUEST flag. After receiving the track address, the controller will cause the selected drive to seek to the desired track and locate the desired sector. TRANSFER REQUEST will be left reset for the remainder of this function. If the correct track and sector are located, the controller will start looking 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 will terminate the function by writing the RXES to the data buffer register and set the DONE flag. An interrupt request will be generated if the interrupt enable bit, RXCS bit 6, was set when DONE was asserted.

If the deleted data address mark (see Section 5-1.1.2.6) was detected, the controller will set the deleted data flag. This flag appears in the ERROR/STATUS register (RXES bit 6). If a CRC error is detected, the controller will set RXES bit 0 and the ERROR flag (RXCS bit 15) to indicate that fact. Seek errors and missing sector errors are reported just as in the WRITE SECTOR function.

#### 5-1.1.2.5 READ STATUS (101)

The READ STATUS command is used to determine the current status of the drive selected by RXCS bits 4 and 5. The status information passed back is: 1) Is the drive ready? 2) Is the diskette write-protected? \*

When the command is issued, the DONE flag is cleared. The controller then checks to see that the door of the selected drive 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) of the RXES. If the drive detects an uncovered write protect notch on the diskette, the controller senses this and sets bit 3 (WRITE PROTECT) of the RXES. The remaining bits of the RXES will reflect the result of the last command before READ STATUS. The controller terminates the 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.

\* If the interface module being used is either the 4430 or 4432, bit 3 of the RXES, when asserted, means that AC power is low in the controller/drive subsystem. On the other LSI-11 (2132,2133) and PDP-11 (2130) interface modules, bit 3 corresponds to the drive write protect status as stated above.

#### 5-1.1.2.6 WRITE DELETED DATA SECTOR (011)

This function performs the same task as WRITE SECTOR. The difference between the commands 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 will be set.

### 5-1.1.2.7 READ ERROR REGISTER (111)

When a command terminates because of an error condition, RXCS bit 15 will be 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.

When the READ ERROR REGISTER command is initiated, the DONE flag is cleared. The controller then moves the RXER into the RXDB and signals the completion of the transfer by asserting the DONE flag. Note that this is the only command that DOES NOT terminate with the RXES sitting 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 will clear the RXER.

OCTAL CODE	MEANING
00	NO ERROR
10	NO DRIVE 0 OR DRIVE 0 FAILED TO FIND TRACK 0 ON INIT
20	NO DRIVE 1 WHEN DIP SWITCH INDICATES THERE SHOULD BE A DRIVE 1 OR DRIVE 1 FAILED TO FIND TRACK 0 ON INIT
30	TRACK 0 FOUND WHILE STEPPING IN ON INITIALIZE
40	TRACK ADDRESS PASSED TO CONTROLLER WAS INVALID (>76)
50	TRACK 0 FOUND BEFORE DESIRED TRACK WHILE STEPPING
70	REQUESTED SECTOR NOT FOUND IN TWO REVOLUTIONS
100	WRITE PROTECT VIOLATION
110	DRIVE READ SIGNAL LOST
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
240	DENSITY ERROR
260	INDETERMINATE DENSITY
300	DRIVE 2 FAILED TO HOME ON INITIALIZE
310	DRIVE 3 FAILED TO HOME ON INITIALIZE
320	READ/WRITE CONTROLLER WRITE CIRCUITS FAILURE
330	READ/WRITE CONTROLLER TIMED OUT ON RESET
340	MASTER CONTROLLER OUT OF SYNC WITH RD/WR CONTROLLER
370	AC POWER LOW CAUSED ABORT OF WRITE ACTIVITY

TABLE 5-1 ERROR REGISTER CODES

#### 5-1.1.2.8 POWER FAIL OR INIT COMMAND (40000 -> RXCS)

When a power failure occurs or DC power to the DSD 440 is interrupted, the controller gradually drains the filter capacitors and dies. Just prior to any diskette writing activity, the controller checks the power. This insures that the sector being written will be written entirely before power is lost.

When power is returned, the DSD 440 controller will initiate 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.
- 6) Bit 2 of RXES (INITIALIZE DONE) is set.
- 7) Bits 7 (DRIVE READY) and 3 (WRITE PROTECT) of RXES are updated according to the status of drive 0. \*
- 8) RXCS bit 5 (DONE) is set.

\* Bit 3 of RXES corresponds to AC LO if the system is configured with either the 4430 or 4432 interface modules. Bit 3 corresponds to the write protect status of the selected drive when any of the other interface modules available for PDP-11 and LSI-11 are being used.

## 5-1.1.2.9 DISKETTE FORMATTING

The DSD 440 floppy disk system can be used to write-format diskettes. This involves re-writing all of the header and data fields on a specified track. The entire track is always written. Figure 1-3 shows a detailed diagram of the single density track format. The protocol necessary to write-format a track is as follows:

The user program issues the WRITE SECTOR function code (010) to the controller. When the controller requests a sector address by setting the TRANSFER REQUEST flag, the number 152(8) is written into the data buffer register. When the controller sees this special number, it branches to a special section of microcode which handles track formatting. The next time the controller sets the TRANSFER REQUEST flag, the program specifies the track to be formatted by writing a valid track address (RXTA) into the data buffer register. At this point, the controller will raise the TRANSFER REQUEST flag 26 more times. Each time the user program sees the TRANSFER REQUEST flag, another valid and unique sector address should be written to the data buffer register. Note that the controller does NOT check these sector addresses for uniqueness or for being in the range 1-26. After receiving the 26th sector address, the controller will seek the heads to the specified track and await the index pulse. Starting at the index mark, the controller will write the entire track. The sector addresses written in the sector headers will be in the same order that they were passed to the controller. This enables all types of hard sector interleaving techniques to be easily implemented by the programmer. When the track has been fully written, the DONE flag will be set and an interrupt will be generated if the interrupt enable bit was set.



### 5-1.1.3 TYPICAL SEQUENCES OF OPERATIONS

The programming examples shown in figures 5-4 thru 5-5 are intended to illustrate how to write routines which will successfully manipulate the DSD 440 data storage system.

#### 5-1.1.3.1 READ/WRITE BUFFER

#### 5-1.1.3.2 READ/WRITE/WRITE D. D.

#### 5-1.1.3.3 STATUS READ

Status information is usually needed to determine what the status of a drive is, or what the cause of an error was. To determine drive related status (DRIVE READY, WRITE PROTECT) the READ STATUS command (Section 5-1.1.2.5) should be used. When the ERROR flag, RXCS bit 15, is set following a function, the RXES should be read first. Remember that the RXES is left in the RXDB following all functions EXCEPT the READ ERROR REGISTER function. As shown in figure 5-2, the RXES has error bits for: CRC ERROR, PARITY ERROR, 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 5-1.

#### 5-1.1.3.4 COMMON PROGRAMMING PITFALLS AND SUGGESTIONS

This chapter will point out some of the more common programming mistakes that can cause data loss and/or error indications.

- 1) Illegal track or sector address sent to 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 an indication of power failure. A short power outage will cause 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 looked at by the controller during FILL BUFFER and EMPTY BUFFER functions.
- 5) It is recommended that a two-sector interleave be used.
- 6) Typically a FILL BUFFER command will precede a WRITE SECTOR command. Similarly, a READ SECTOR command will precede an EMPTY BUFFER command.

#### 5-1.1.4 INTERRUPTS

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

## 5-1.2 MODE 2 OPERATION

The system assumes MODE 2 operation when the "RX01" switch is placed in the 0 position (see sections 2-1, 3-6, 4-5). The system will only operate according to MODE 2 protocol when connected to an interface module which is capable of DMA. The programmers' interface for MODE 2 is identical to that of the DEC RX02 based systems. Any program that runs successfully with the DEC RX211 (or RXV211) will run equally well on a DSD 440 system configured to operated in MODE 2.

### 5-1.2.1 PERIPHERAL DEVICE REGISTERS

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

COMMAND AND STATUS REGISTER	(RX2CS = 777170)
DATA BUFFER REGISTER	(RX2DB = 777172)

Peripheral device registers reside in the top 4K words of the 11-family computer's 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 these registers non-standard bus addresses, see section 3-4.

## 5-1.2.1.1 COMMAND AND STATUS REGISTER

Writing the bits of this register controls the DSD 440. The format for this register is shown in figure 5-6. The RX2CS register also provides important status information and error indications when read by the 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 will be negated, the controller will reset some internal flags and variables, and then execute the self-test microcode. All of the floppy disk drives will be homed to track 0. If the controller is configured in "NORMAL" mode (as opposed to "HYPER-DIAGNOSTIC" mode), the controller will now attempt to read track 1 sector 1 of the diskette in drive 0. When the READ SECTOR function has been attempted, the INITIALIZE DONE bit in the error/status register is set. Assuming there actually was a readable diskette in drive 0, the DRIVE READY bit will also be set. If the diskette happened to have been double density, then the drive density bit will be 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 13 - A17 Extended address bit 17. This write only bit will be 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 will be 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 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 is used to specify 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 any magnetic media, a valid density bit is important so that the controller can evaluate the validity of the word count parameter.

BIT 7 - TR TRANSFER REQUEST flag. This read only bit indicates to the program that the DATA BUFFER REGISTER has been emptied and needs loading, or is loaded and needs emptying to the controller.

BIT 6 - IE INTERRUPT ENABLE bit. This read/write bit, when set, will allow 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 5 serves a different purpose when it is written.

BIT 4 - UN1 Drive unit select bit. The binary encoding of this bit selects drive 0-1. Drive selection only occurs if a drive related function is executed.

BITS 3-1 - FN FUNCTION SELECT

- 0 = FILL BUFFER
- 1 = EMPTY BUFFER
- 2 = WRITE SECTOR
- 3 = READ SECTOR
- 4 = SET MEDIA DENSITY
- 5 = READ STATUS
- 6 = WRITE DELETED DATA SECTOR
- 7 = READ ERROR CODE

Function select bits are write only.

BIT 0 - EX Execute the function encoded in bits 3-1 of this register.

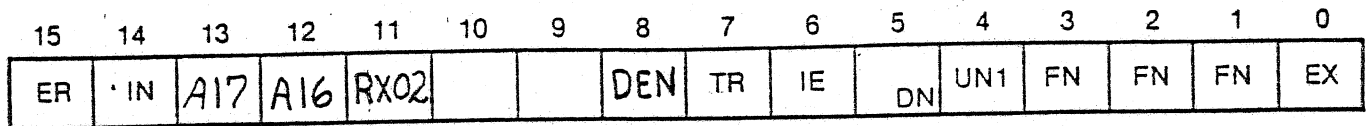


FIGURE 5-6

BIT 7 - TR TRANSFER REQUEST flag. This read only bit indicates to the program that the DATA BUFFER REGISTER has been emptied and needs loading, or is loaded and needs emptying to the controller.

BIT 6 - IE INTERRUPT ENABLE bit. This read/write bit, when set, will allow 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 5 serves a different purpose when it is written.

BITS 5-4 - UN2 UN1 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.

BITS 3-1 - FN FUNCTION SELECT

- 0 = FILL BUFFER
- 1 = EMPTY BUFFER
- 2 = WRITE SECTOR
- 3 = READ SECTOR
- 4 = SET MEDIA DENSITY
- 5 = READ STATUS
- 6 = WRITE DELETED DATA SECTOR
- 7 = READ ERROR CODE

Function select bits are write only.

BIT 0 - EX Execute the function encoded in bits 3-1 of this register.

### 5-1.2.1.2 DATA REGISTER

This register provides the general purpose communication link between the host processor and the DSD 440 system. The information passed through this register is based upon a predetermined protocol as defined in the Section 5-1.2.2 .

If the DSD 440 is not in the process of executing a command, the RX2DB can be written without the risk of any adverse affects. However, during the execution of an instruction, the RX2DB register will only provide or accept information (according to the RX2DB protocol) when the TRANSFER REQUEST flag is set.

#### NOTE:

Data may be lost if the correct protocol is not adhered to.

The following descriptions explain the various formats of the Data Register. (RX2DB)

### 5-1.2.1.3 FLOPPY 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 TRACK ADDRESS REGISTER. (RX2TA = 777172) Track numbers from 0-76 (decimal) are valid.

### 5-1.2.1.4 FLOPPY 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 SECTOR ADDRESS REGISTER. (RX2SA = 777172) Sector addresses from 1-26 (decimal) are valid.

### 5-1.2.1.5 WORD COUNT REGISTER

The WORD COUNT REGISTER is used to specify 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 one half this amount (128 bytes). The programmer loads the actual count into this register, NOT the 2's complement of the count.

#### 5-1.2.1.6 BUS ADDRESS REGISTER

This register specifies the bus address to which data is to be transferred during any DMA operation. It is in fact a 16 bit counter that 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. Do not try to load bit 00 with a 1 (it is ignored).

#### 5-1.2.1.7 SYSTEM ERROR AND STATUS REGISTER

The RX2ES register provides status and error information about the drive that is selected by bits 4 and 5 of the RX2CS register. At the completion of a command, the controller will place the RX2ES register into the data buffer register (RX2DB = 777172) so that the host processor can check the most recent operation. See figure 5-7 for detailed definitions of the individual bits.



## 5-1.2.2 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. Because of this, the controller must identify parameters based on the order in which they are transmitted across the data 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 may result in lost or incorrect data.

### 5-1.2.2.1 FILL SECTOR BUFFER (000)

The FILL SECTOR BUFFER command is used to fill a storage buffer inside the DSD 440 with up to 128 or 256 8-bit bytes of data from computer memory. Other functions can later be used to either write that data to diskette, or transfer it back to memory.

When the FILL SECTOR BUFFER command is given, the DSD 440 responds by clearing the DONE flag, RXCS 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 has been 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 thus ending the operation. An interrupt request will be 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:

Bits 4 and 5 of the RX2CS do not affect this function since no floppy disk drives need be selected.

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.

#### 5-1.2.2.2 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 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 again asserted, the program should respond by loading the starting memory address into the data buffer register. When this is done, the controller will clear the TRANSFER REQUEST flag and it will remain clear for the rest of the operation. The data in the sector buffer will be transferred to memory one word at a time until the word count is decremented to zero. When the data has been transferred, the controller will place the RX2ES into the data buffer register and set the DONE flag. If the interrupt enable bit is set, an interrupt request will be initiated when DONE becomes true.

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

### 5-1.2.2.3 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 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. 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 should respond 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 will cause the selected drive to seek the desired track. TRANSFER REQUEST will be left reset for the remainder of the function. At this time, the heads of the selected drive are positioned over the specified track and are loaded against the media. If the controller does not already know the density and format of the media, this information will be determined by reading a random sector on the diskette. If the density of the media does not agree with the command density (RX2CS bit 8) the operation is terminated. Bit 4 of the RX2ES register is set to indicate a density error. If the densities agree, the controller checks the track address and starts looking for the specified sector address. If the correct track and sector are found, the controller will write 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(8). If the specified sector cannot be found within two diskette revolutions, the RX2ES will be set to a 70(8). 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 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 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 (octal) is written to the RXSA, the WRITE SECTOR function turns into a WRITE FORMAT TRACK function. The protocol for formatting is explained in section 5-1.2.2.10 .

#### 5-1.2.2.4 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 should respond by writing the desired track address (RX2TA) into the data buffer register. This clears the TRANSFER REQUEST flag. After receiving the track address, the controller will cause the selected drive to seek the desired track. TRANSFER REQUEST will be left reset for the remainder of this function. The controller loads the heads against the media and tries to determine the density of the media (if it is not already known). If the density of the diskette does not agree with the command density (RX2CS bit 8), a density error is reported and the function is terminated. If the densities agree, the controller starts looking for the specified sector. When the right sector is located, the controller starts looking 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 will terminate the function by writing the RX2ES to the data buffer register and setting the DONE flag. An interrupt request will be generated if the interrupt enable bit was set when DONE became true.

If the deleted data address mark (see Section 5-1.2.2.7) 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) to indicate that fact. Seek errors and missing sector errors are reported just as in the WRITE SECTOR function.

### 5-1.2.2.5 SET MEDIA DENSITY (100)

This command function is used to initialize an entire diskette to some specified density. When the SET MEDIA DENSITY command is executed, the controller attempts to write zeros 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 RX2ES register. Next, the controller sets the TRANSFER REQUEST flag. The program should respond by writing a "key byte" into the RX2DB. If the key byte is an ASCII "I" (111)8, the SET MEDIA DENSITY function will be executed. If the byte written into the RX2DB at this time is not an "I", the DONE and ERROR flags will be set and the operation will terminate. The error register will be loaded with a (250)8 to indicate an invalid key. The purpose of the key is to make it difficult for a programmer to accidentally 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 zeros 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 26 sectors on track 0, the controller seeks to track 1,2,... writing all 26 sectors on each track. This process continues until either every sector has been written through track 76 sector 26 or a bad header is found. The ERROR and DONE flags will be 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 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 re-written again. If the SET MEDIA DENSITY function will not complete because of an unreadable header, the TRACK FORMAT procedure can be used to re-write the blown header. ( See Section 5-1.2.2.10 )

### 5-1.2.2.6 READ STATUS (101)

The READ STATUS command is used to determine the current status of the drive selected by RX2CS bits 4 and 5. The status information passed back is: 1) Is the drive ready? and 2) What is the density of the diskette currently in the drive?

When the command is issued, the DONE flag is cleared. The controller then checks to see that the door of the selected drive is closed, a diskette is inserted, and that 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) of 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 remaining bits of the RX2ES will reflect the result of the last command before READ STATUS. The controller terminates the function by shifting the RX2ES over to the RX2DB and setting the DONE flag. An interrupt request will be generated if the interrupt enable bit, RX2CS bit 6, was set when done became true.

#### 5-1.2.2.7 WRITE DELETED DATA SECTOR (011)

This function performs the same task as WRITE SECTOR. The difference between the commands 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 RX2ES will be set. The density bit associated with this function (RX2CS bit 8) determines whether a single or double density deleted data address mark will be written.

#### 5-1.2.2.8 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 will go false. The controller will then assert the TRANSFER REQUEST flag. The program should then load a starting memory address into the RX2DB. The controller will then transfer 4 words directly to memory. When the words are in memory, the controller will assert DONE and this will generate 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
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

OCTAL CODE	MEANING
00	NO ERROR
10	NO DRIVE 0 OR DRIVE 0 FAILED TO FIND TRACK 0 ON INIT
20	NO DRIVE 1 WHEN DIP SWITCH INDICATES THERE SHOULD BE A DRIVE 1 OR DRIVE 1 FAILED TO FIND TRACK 0 ON INIT
30	TRACK 0 FOUND WHILE STEPPING IN ON INITIALIZE
40	TRACK ADDRESS PASSED TO CONTROLLER WAS INVALID (>76)
50	TRACK 0 FOUND BEFORE DESIRED TRACK WHILE STEPPING
70	REQUESTED SECTOR NOT FOUND IN TWO REVOLUTIONS
100	WRITE PROTECT VIOLATION
110	DRIVE READ SIGNAL LOST
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
300	DRIVE 2 FAILED TO HOME ON INITIALIZE
310	DRIVE 3 FAILED TO HOME ON INITIALIZE
320	READ/WRITE CONTROLLER DETECTED WRITE CIRCUIT FAILURE
330	READ/WRITE CONTROLLER TIME OUT ON RESET
340	MASTER CONTROLLER OUT OF SYNC WITH RD/WR CONTROLLER
✓ 350	NON-EXISTANT MEMORY ERROR DURING DMA
360	DRIVE NOT READY DURING FORMAT COMMAND
370	AC POWER LOW CAUSED ABORT OF WRITE ACTIVITY

TABLE 5-2 ERROR REGISTER CODES



#### 5-1.2.2.9 POWER FAIL

When a power failure occurs or DC power to the DSD 440 is interrupted the controller gradually drains the filter capacitors and dies. Just prior to any diskette writing activity, the controller checks the power. This insures that the sector being written will be valid.

When power is returned, and the controller dip switch is configured for "NORMAL" mode, the DSD 440 controller will initiate the following sequence of events:

- 1) DONE is cleared.
- 2) Controller executes the hardware self-tests.
- 3) All drives homed 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) Bits 7 (DRIVE READY) and 5 (DRIVE DENSITY) of RX2ES are updated according to the status of drive 0.
- 8) RX2ES bit 5 (DONE) is set.

## 5-1.2.2.10 DISKETTE FORMATTING IN MULTIPLE DENSITIES

When configured in operational Mode 2, the DSD 440 floppy disk system can be made to write-format diskettes in three unique formats. The DEC RX02 will not support the command protocol about to be described. In this sense, the DSD 440's commands are a super-set of those of the RX02.

Each time the write-format command protocol is executed, one entire track is re-written. 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, will be 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(8) into the data buffer register. If the track format desired is DEC double density, the user program writes the number 153(8) into the data buffer register. When the controller sees these "strange" 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 should respond by writing a valid track address into the data buffer register. Next, the controller enters a 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. (See Figures 1-3, 1-4) 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 all types of hard sector interleaving techniques to be easily implemented by the programmer to improve the effective throughput of the disk system.

### 5-1.2.3 TYPICAL SEQUENCES OF OPERATIONS

The programming examples shown in figures 5-8 and 5-9 are intended to illustrate how to write routines which will successfully manipulate the DSD 440 data storage system.

#### 5-1.2.3.1 READ/WRITE BUFFER

#### 5-1.2.3.2 READ/WRITE/WRITE D. D.

#### 5-1.2.3.3 STATUS READ

Status information is usually needed to determine what the status of a drive is, or what the cause of an error was. To determine drive related status (DRIVE READY, DRIVE DENSITY) the READ STATUS command (Section 5-1.2.2.6) 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. As shown in figure 5-7, the RX2ES has error bits for: CRC ERROR 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. The code interpretations are shown in table 5-2.

; PROGRAMMING EXAMPLE  
; FILL / EMPTY RX02 SECTOR BUFFER

	177170			RXCS=177170	; CONTROL AND STATUS REGISTER
	177172			RXDB=177172	; DATA BUFFER REGISTER
					; EMPTY RX02 SECTOR BUFFER
000000	012700	000003		EMPBUF: MOV #3,RO	; BUILD EMPBUF COMMAND IN RO
000004	000402			BR ENCENT	
					; FILL RX02 SECTOR BUFFER ROUTINE
000006	012700	000001		FILBUF: MOV #1,RO	; BUILD FILBUF COMMAND IN RO
000012	005737	000154'		FNCENT: TST UNIT	; UNIT 0 OR UNIT 1?
000016	001402			BEG 1*	
000020	052700	000020		BIS #20,RO	; SET UNIT
000024	005737	000146'	1*:	TST DEN	; HIGH OR LOW DENSITY?
000030	001402			BEG 2*	
000032	052700	000400		BIS #400,RO	; SPECIFY HIGH DENSITY
000036	010037	177170	2*:	MOV RO,@#RXCS	; ISSUE COMMAND TO CONTROLLER
000042	105737	177170	3*:	TSTB @#RXCS	; WAIT FOR TRANSFER REQUEST
000046	100375			BPL 3*	
000050	013737	000150'	177172	MOV WRDCNT,@#RXDB	; PASS WORDCOUNT TO CONTROLLER
000056	105737	177170	4*:	TSTB @#RXCS	; WAIT FOR TRANSFER REQUEST
000062	100375			BPL 4*	
000064	013737	000152'	177172	MOV BUFADR,@#RXDB	; PASS BUS ADDRESS TO CONTROLLER
000072	032737	100040	177170	5*:	BIT #100040,@#RXCS ; TEST FOR DONE AND ERROR
000100	001774			BEG 5*	
000102	100401			BMI ERFIN	; ERROR BIT SET?
000104	000207			RTS PC	
000106	012737	000017	177170	ERFIN: MOV #17,@#RXCS	; GET DEFINITIVE STATUS
000114	105737	177170	6*:	TSTB @#RXCS	; WAIT FOR TRANSFER REQUEST
000120	100375			BPL 6*	
000122	012737	000156'	177172	MOV #ERBUF,@#RXDB	; SEND ERROR BUFFER ADDRESS
000130	032737	000040	177170	7*:	BIT #40,@#RXCS ; WAIT FOR DONE BIT
000136	001774			BEG 7*	
000140	113700	000156'		MOVB @#ERBUF,RO	; LEAVE ERROR REGISTER IN RO
000144	000000			HALT	
000146	000000			DEN: .WORD 0	; DENSITY - 0=SINGLE 1=DOUBLE
000150	000100			WRDCNT: .WORD 100	; WORDCOUNT - FULL SD BUFFER IS 100 FULL DD BUFFER IS 200
000152	002000			BUFADR: .WORD 2000	; BUFFER ADDRESS VARIABLE
000154	000000			UNIT: .WORD 0	; UNIT - 0=DRIVE 0 1=DRIVE 1
000156	000000			ERBUF: .WORD 0	; ERROR BUFFER
	000164'			. = +4	
	000001			. END	

FIGURE 5-8

PROGRAMMING EXAMPLE  
WRITE / READ RX02 SECTOR

```

177170          RXCS=177170      ; CONTROL AND STATUS REGISTER
177172          RXDB=177172      ; DATA BUFFER REGISTER

; READ RX02 SECTOR
000000 012700 000007      READ:  MOV #7,RO      ; BUILD READ SECTOR COMMAND IN RO
000004 000402              BR SYNTAX

; WRITE RX02 SECTOR ROUTINE
000006 012700 000005      WRITE: MOV #5,RO      ; BUILD WRITE SECTOR COMMAND IN RO
000012 005737 000154      SYNTAX: TST UNIT      ; UNIT 0 OR UNIT 1?
000016 001402              BEG 1$
000020 052700 000020              BIS #20,RO      ; SET UNIT
000024 005737 000146      1$:  TST DEN      ; HIGH OR LOW DENSITY?
000030 001402              BEG 2$
000032 052700 000400              BIS #400,RO    ; SPECIFY HIGH DENSITY
000036 010037 177170      2$:  MOV RO,@#RXCS    ; ISSUE COMMAND TO CONTROLLER
000042 105737 177170      3$:  TSTB @#RXCS     ; WAIT FOR TRANSFER REQUEST
000046 100375              BPL 3$
000050 013737 000150      4$:  MOV SECTOR,@#RXDB ; PASS SECTOR TO CONTROLLER
000054 105737 177170      4$:  TSTB @#RXCS     ; WAIT FOR TRANSFER REQUEST
000062 100375              BPL 4$
000064 013737 000152      5$:  MOV TRACK,@#RXDB  ; PASS TRACK NUMBER TO CONTROLLER
000072 032737 100040 177170 5$:  BIT #100040,@#RXCS ; TEST FOR DONE AND ERROR
000100 001774              BEG 5$
000102 100401              BMI ERFIN      ; ERROR BIT SET?
000104 000207              RTS PC

;
000106 012737 000017 177170 ERFIN: MOV #17,@#RXCS ; GET DEFINITIVE STATUS
000114 105737 177170      6$:  TSTB @#RXCS     ; WAIT FOR TRANSFER REQUEST
000120 100375              BPL 6$
000122 012737 000156      7$:  MOV #ERBUF,@#RXDB ; SEND ERROR BUFFER ADDRESS
000130 032737 000040 177170 7$:  BIT #40,@#RXCS     ; WAIT FOR DONE BIT
000136 001774              BEG 7$
000140 113700 000156      MOVB @#ERBUF,RO ; LEAVE ERROR REGISTER IN RO
000144 000000              HALT

;
000146 000000      DEN:      WORD 0      ; DENSITY - 0=SINGLE 1=DOUBLE
000150 000001      SECTOR:  WORD 1      ; DESIRED SECTOR ADDRESS
000152 000001      TRACK:   WORD 1      ; DESIRED TRACK ADDRESS
000154 000000      UNIT:    WORD 0      ; UNIT - 0=DRIVE 0 1=DRIVE 1
000156 000000      ERBUF:   WORD 0      ; ERROR BUFFER
000164          =. +4
000001          END

```

FIGURE 5-9

#### 5-1.2.3.4 COMMON PROGRAMMING PITFALLS AND SUGGESTIONS

This chapter will point out some of the more common programming mistakes that can cause data loss and/or error indications.

- 1) Illegal track or sector address sent to 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, 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 even though invalid data may have been read or written.
- 4) The drive select bits, RX2CS bits 4 and 5, are not looked at by the controller during FILL BUFFER and EMPTY BUFFER functions.
- 5) It is recommended that a two-sector interleave be used.
- 6) Typically a FILL BUFFER command will precede a WRITE SECTOR command. Similarly, a READ SECTOR command will precede an EMPTY BUFFER command.

#### 5-1.2.4 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.

## 5-2 DEC PDP-8 FAMILY PROGRAMMERS' INTERFACE

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

### 5-2.1 INSTRUCTION SET

The 8 IOT instructions used to program the DSD 440 are shown in Table 5-3. These same IOT instructions are used in both Mode 1 and Mode 2 operation.

IOT	MNEMONIC	DESCRIPTION
67x0		No Operation
67x1	LCD	Load Command, Clear Accumulator
67x2	XDR	Transfer Data Register
67x3	STR	Skip on Transfer Request, Clear Flag
67x4	SER	Skip on Error Flag, Clear Flag
67x5	SDN	Skip on Done Flag, Clear Flag
67x6	INTR	Enable or Disable Disk Interrupts
67x7	INIT	Initialize Controller and Interface

TABLE 5-3

#### 5-2.1.1 THE LOAD COMMAND (LCD) INSTRUCTION

This instruction transfers the contents of the accumulator to the interface register and then clears the accumulator. The command word is laid out as shown in Figure 5-10 when the system is configured in Mode 1, and as shown in Figure 5-11 when the system is configured in Mode 2 and programmed for 12 bit word length. The format shown in Figure 5-12 is used in Mode 2 when 8 bit word length is selected. 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.



### 5-2.1.2 THE TRANSFER DATA REGISTER (XDR) INSTRUCTION

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 things to the DSD 440: 1) The PDP-8 has accepted the last word supplied by the controller, or, 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 in twelve-bit mode, and eight-bit or'ed transfers in eight-bit mode.

### 5-2.1.3 THE SKIP ON TRANSFER REQUEST (STR) INSTRUCTION

This instruction will cause 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/from the controller.

### 5-2.1.4 THE SKIP ON ERROR (SER) INSTRUCTION

This instruction will cause 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 visa-versa). A typical program would execute an SDN followed by a SER. (see examples)

### 5-2.1.5 THE SKIP ON DONE (SDN) INSTRUCTION

This instruction will cause 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 will generate an interrupt when it is first asserted by the controller.

42.381 50 SHEETS 5 SQUARE  
 42.382 100 SHEETS 5 SQUARE  
 42.383 200 SHEETS 5 SQUARE  
 NATIONAL

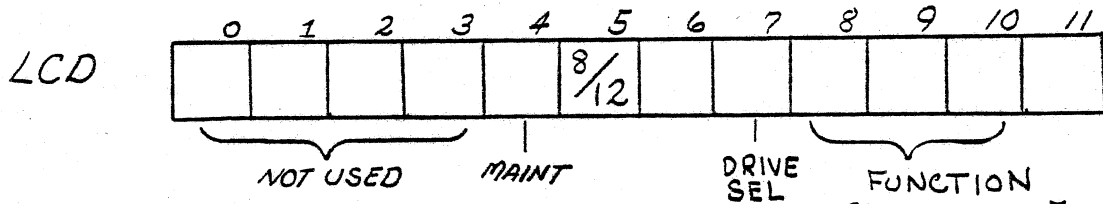


FIGURE 5-10 MODE 1 COMMAND REGISTER FORMAT

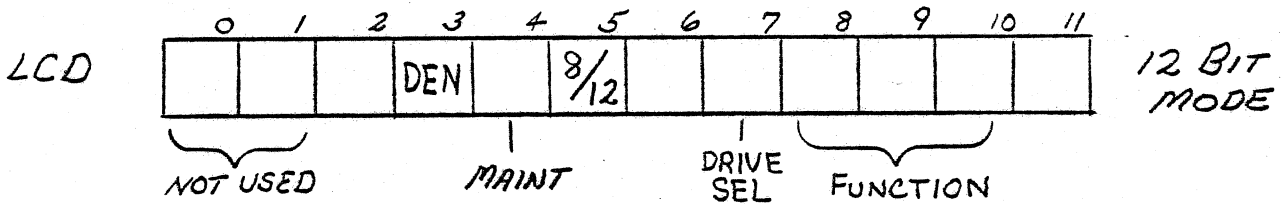


FIGURE 5-11 MODE 2, 12 BIT MODE, COMMAND REG. FORMAT

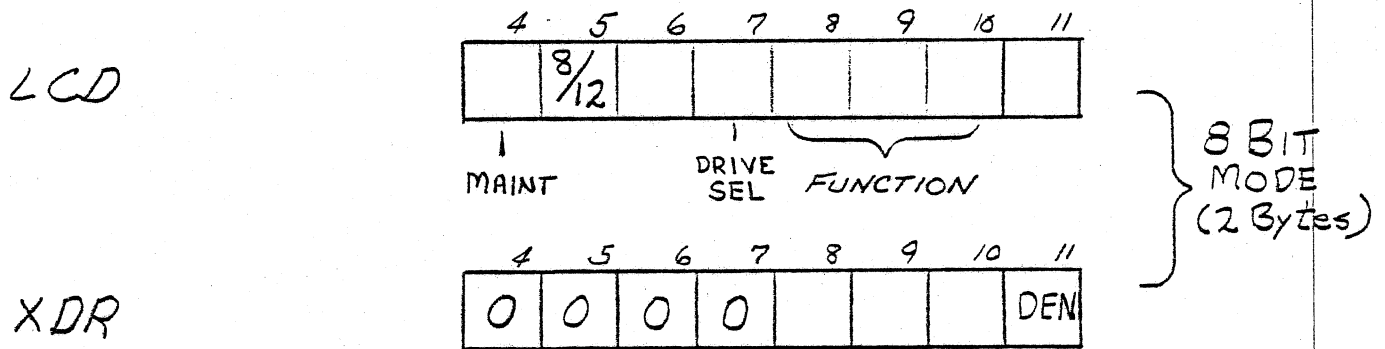


FIGURE 5-12 MODE 2, 8 BIT MODE COMMAND REGISTER FORMAT/SEQUENCE

#### 5-2.1.6 THE INTERRUPT ENABLE/DISABLE (INTR) INSTRUCTION

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 will be disabled by a CLEAR ALL FLAGS instruction, a bus initialize, or a programmed initialize.

#### 5-2.1.7 THE INITIALIZE (INIT) INSTRUCTION

This instruction initializes the DSD 440 system by "homing" all of the floppy disk drives to track 0. The controller microprocessor runs all of the hardware self-test routines (Section 4-6). 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 (RXES bit 9) is set. This initialize sequence can take as long as 1.8 seconds to complete. Either this instruction or a CLEAR ALL FLAGS instruction is capable of starting this sequence of events.

## 5-2.2 REGISTER DESCRIPTIONS

There is only one physical register in the DSD 440 interface hardware for the PDP-8. That register, referred to as the interface register, can represent any one of six controller registers depending on the protocol of the function in progress. Each of the possible register formats will now be described.

### 5-2.2.1 COMMAND REGISTER

The command register has the format shown in figure 5-10 when the controller is configured in Operational Mode 1 (RX01 compatible). 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 instruction regardless of the 8/12 bit mode bit. 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 makes the word length only eight bits. Bit 7 specifies the floppy disk 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. A density error will be 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 5-4 lists the codes:

(The function codes are discussed in section 5-2.3)

CODE BITS 8 9 10	MODE 1 FUNCTION	MODE 2 FUNCTION
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	NO-OP(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

TABLE 5-4

### 5-2.2.2 TRACK ADDRESS REGISTER

This register is written to the controller interface register with an XDR instruction when a particular function requires a track address. 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 5-13 shows a diagram of this register.

### 5-2.2.3 SECTOR ADDRESS REGISTER

This register is written to the controller interface register with an XDR 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 5-14 shows a diagram of this register. This register works the same in both Mode 1 and Mode 2 system operation.

### 5-2.2.4 DATA BUFFER REGISTER

All data transferred to and from the floppy disk 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 and all 12 bits of the register contain valid data. If bit 5 was 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 5-15 shows a diagram of this register.



## 5-2.2.5 ERROR AND STATUS REGISTER (MODE 1 OPERATION)

Figure 5-16 shows the bit assignments of the error and status register when the controller is configured for Mode 1 (RX01 compatible) 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 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 DRIVE READY - WHEN ASSERTED, THIS BIT INDICATES THAT THE SELECTED DRIVE EXISTS, HAS POWER, HAS A DISKETTE PROPERLY INSTALLED AND UP TO SPEED. THIS BIT CAN ONLY BE ASSUMED VALID WHEN IT WAS RETRIEVED USING THE READ STATUS FUNCTION OR IMMEDIATELY FOLLOWING AN INIT.

BIT 5 DELETED DATA - WHEN ASSERTED, THIS BIT INDICATES THAT A DELETED DATA ADDRESS MARK WAS FOUND WHILE THE LAST SECTOR WAS BEING READ. THIS BIT WILL ALSO BE ASSERTED IF THE LAST FUNCTION EXECUTED WAS WRITE DELETED DATA SECTOR.

BIT 6-8 NOT USED IN MODE 1

BIT 9 INITIALIZE DONE - THIS BIT INDICATES THAT THE SERIES OF TASKS PERFORMED BY THE CONTROLLER FOLLOWING A POWER-UP OR A PROGRAMMED INIT HAVE JUST BEEN COMPLETED.

BIT 10 PARITY ERROR - WHEN SET, THIS BIT INDICATES THAT THE CONTROLLER RECEIVED THE WRONG PARITY CHECK BIT APPENDED TO A STRING OF COMMAND OR PARAMETER BITS BEING SHIFTED FROM THE INTERFACE MODULE TO THE MASTER CONTROLLER.

BIT 11 CRC ERROR - WHEN SET, THIS BIT INDICATES THAT A CYCLIC REDUNDANCY CHECK ERROR WAS DETECTED AS A RESULT OF TRYING TO READ THE DATA CONTAINED IN A DISKETTE SECTOR. THE DATA CONTAINED IN THE CONTROLLER BUFFER MUST BE CONSIDERED INVALID. THE SECTOR THAT PRODUCED THIS ERROR CAN BE READ SEVERAL TIMES IN THE HOPE THAT THE DATA ERROR IS "SOFT".

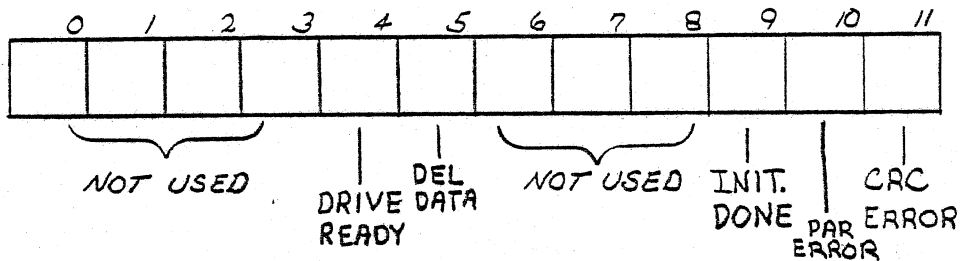


FIGURE 5-16 ERROR/STATUS REGISTER IN MODE 1

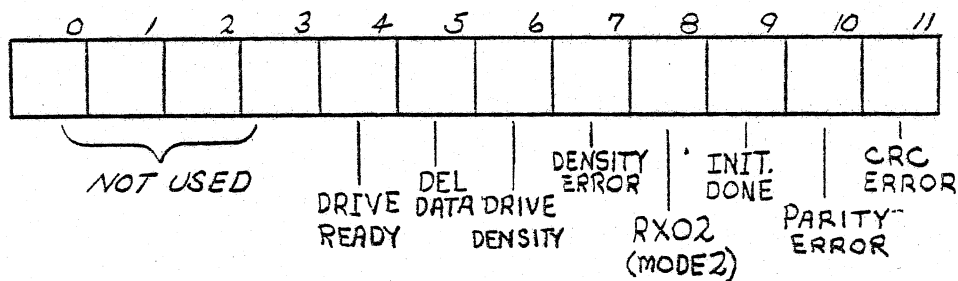


FIGURE 5-17 ERROR/STATUS REGISTER IN MODE 2



## 5-2.2.6 ERROR AND STATUS REGISTER (MODE 2 OPERATION)

Figure 5-17 shows the bit assignments of the error and status register when the controller is configured for Mode 2 (RX02 compatible) 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 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 DRIVE READY - WHEN ASSERTED, THIS BIT INDICATES THAT THE SELECTED DRIVE EXISTS, HAS POWER, HAS A DISKETTE PROPERLY INSTALLED AND UP TO SPEED. THIS BIT CAN ONLY BE ASSUMED VALID WHEN IT HAS BEEN RETRIEVED USING THE READ STATUS FUNCTION.
- BIT 5 DELETED DATA - WHEN ASSERTED, THIS BIT INDICATES THAT A DELETED DATA ADDRESS MARK WAS FOUND WHILE THE LAST SECTOR WAS BEING READ. THIS BIT WILL ALSO BE ASSERTED IF THE LAST FUNCTION EXECUTED WAS WRITE DELETED DATA SECTOR.
- BIT 6 DRIVE DENSITY - THIS BIT INDICATES THE DENSITY OF THE DISKETTE IN THE SELECTED DRIVE. A ONE INDICATES DOUBLE DENSITY AND A ZERO INDICATES SINGLE DENSITY.
- BIT 7 DENSITY ERROR - WHEN SET, THIS BIT INDICATES THAT THE DENSITY OF THE DISKETTE IN THE SELECTED DRIVE DID NOT AGREE WITH THE FUNCTION DENSITY SPECIFIED IN THE COMMAND REGISTER. UPON DETECTION OF THIS ERROR THE FUNCTION IS TERMINATED AND BOTH ERROR AND DONE FLAGS ARE ASSERTED.
- BIT 8 RX02 (MODE 2) - THIS BIT INDICATES THAT THE INTERFACE MODULE IS PRESENTLY CONNECTED TO A CONTROLLER/DRIVE SUBSYSTEM WHICH IS CAPABLE OF BOTH SINGLE AND DOUBLE DENSITY OPERATION.
- BIT 9 INITIALIZE DONE - THIS BIT INDICATES THAT THE SERIES OF TASKS PERFORMED BY THE CONTROLLER FOLLOWING A POWER-UP OR A PROGRAMMED INIT HAVE JUST BEEN COMPLETED.
- BIT 10 PARITY ERROR - WHEN SET, THIS BIT INDICATES THAT THE CONTROLLER RECEIVED THE WRONG PARITY CHECK BIT APPENDED TO A STRING OF COMMAND OR PARAMETER BITS BEING SHIFTED FROM THE INTERFACE MODULE TO THE MASTER CONTROLLER.
- BIT 11 CRC ERROR - WHEN SET, THIS BIT INDICATES THAT A CYCLIC REDUNDANCY CHECK ERROR WAS DETECTED AS A RESULT OF TRYING TO READ THE DATA CONTAINED IN A DISKETTE SECTOR. THE DATA CONTAINED IN THE CONTROLLER BUFFER MUST BE CONSIDERED INVALID. THE SECTOR THAT PRODUCED THIS ERROR CAN BE READ SEVERAL TIMES IN THE HOPE THAT THE DATA ERROR IS "SOFT".

### 5-2.3 FUNCTION CODE DESCRIPTIONS

A floppy disk function is initiated by writing one of the function codes shown in Table 5-4 to the command register using the LCD instruction. The Done flag should always be tested and cleared with the SDN instruction prior to issuing the command instruction to verify that the controller is really in the Done state. The protocol associated with each of the functions shown in Table 5-4 will now be described in detail.

#### 5-2.3.1 FILL BUFFER (000)

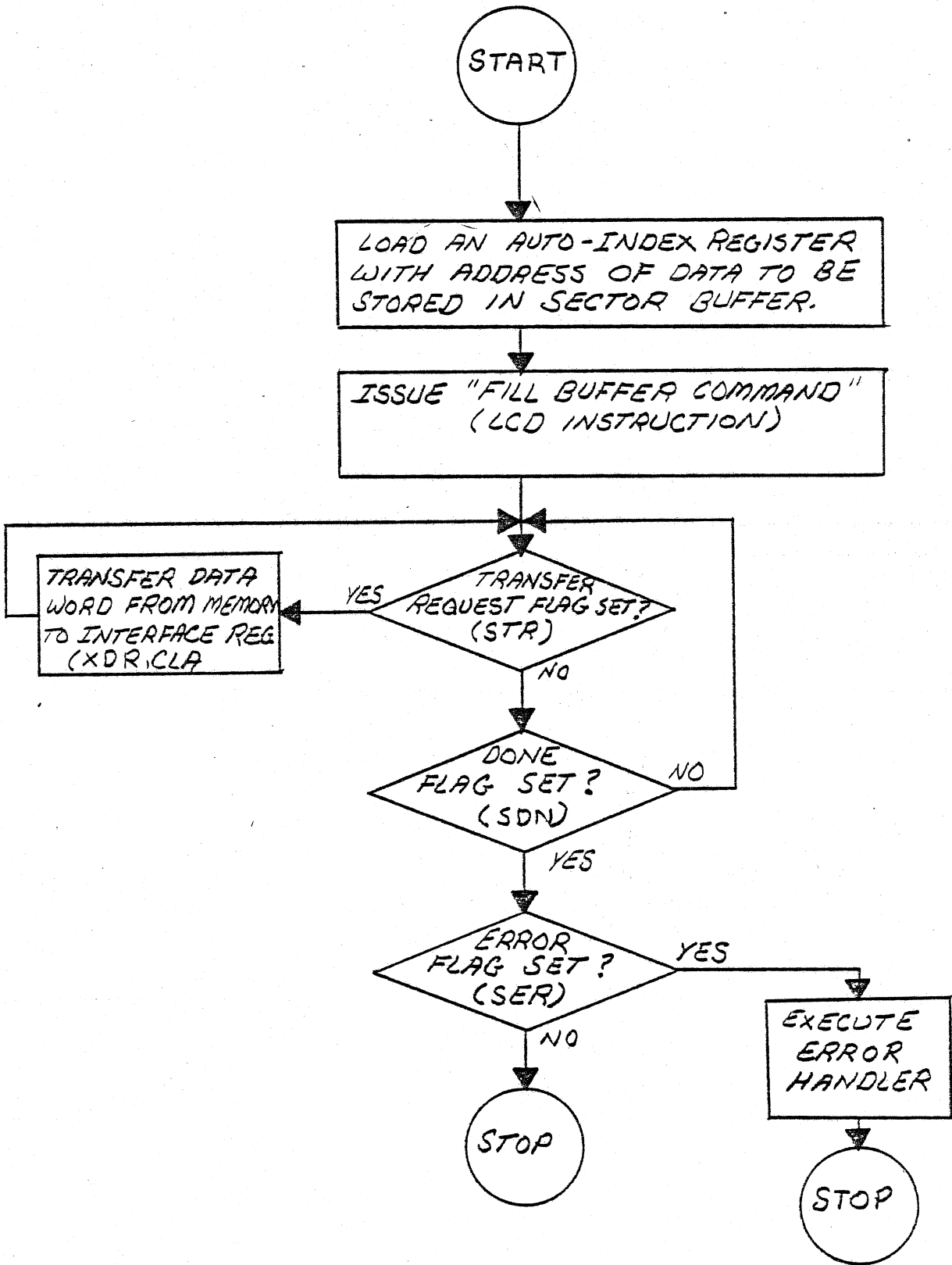
This function is used to load the controller sector buffer with the data that is to be written into a sector or later transferred back to the host computer. The amount and format of the data is determined by the operating mode of the controller (Mode 1 or Mode 2) and the word length bit of the command register. After the command is issued to the controller with the LCD instruction, the controller will assert the Transfer Request flag once for each 8 or 12 bit word that is needed from the host computer to place in the sector buffer. The host computer should test and clear the Transfer Request flag with the STR instruction and then transfer a data word to the controller with the XDR instruction. When the controller has determined that the appropriate number of words has been transferred in this manner, the Done flag will be asserted and an interrupt will occur assuming interrupts have been enabled. If an XDR instruction is executed after Done is set, it will have the affect of loading the error/status register into the accumulator. The chart below shows the relationship between the number of XDR cycles needed to fill the buffer, the operating mode, and the word length:

OPERATING MODE	DATA WORD LENGTH	NUMBER OF CYCLES
1 (RX01 COMP.)	8 BITS	128
1 (RX01 COMP.)	12 BITS	64
2 (RX02 COMP.)	8 BITS	256 DD, 128 SD
2 (RX02 COMP.)	12 BITS	128 DD, 64 SD

TABLE 5-5

FIGURE 5 - 18

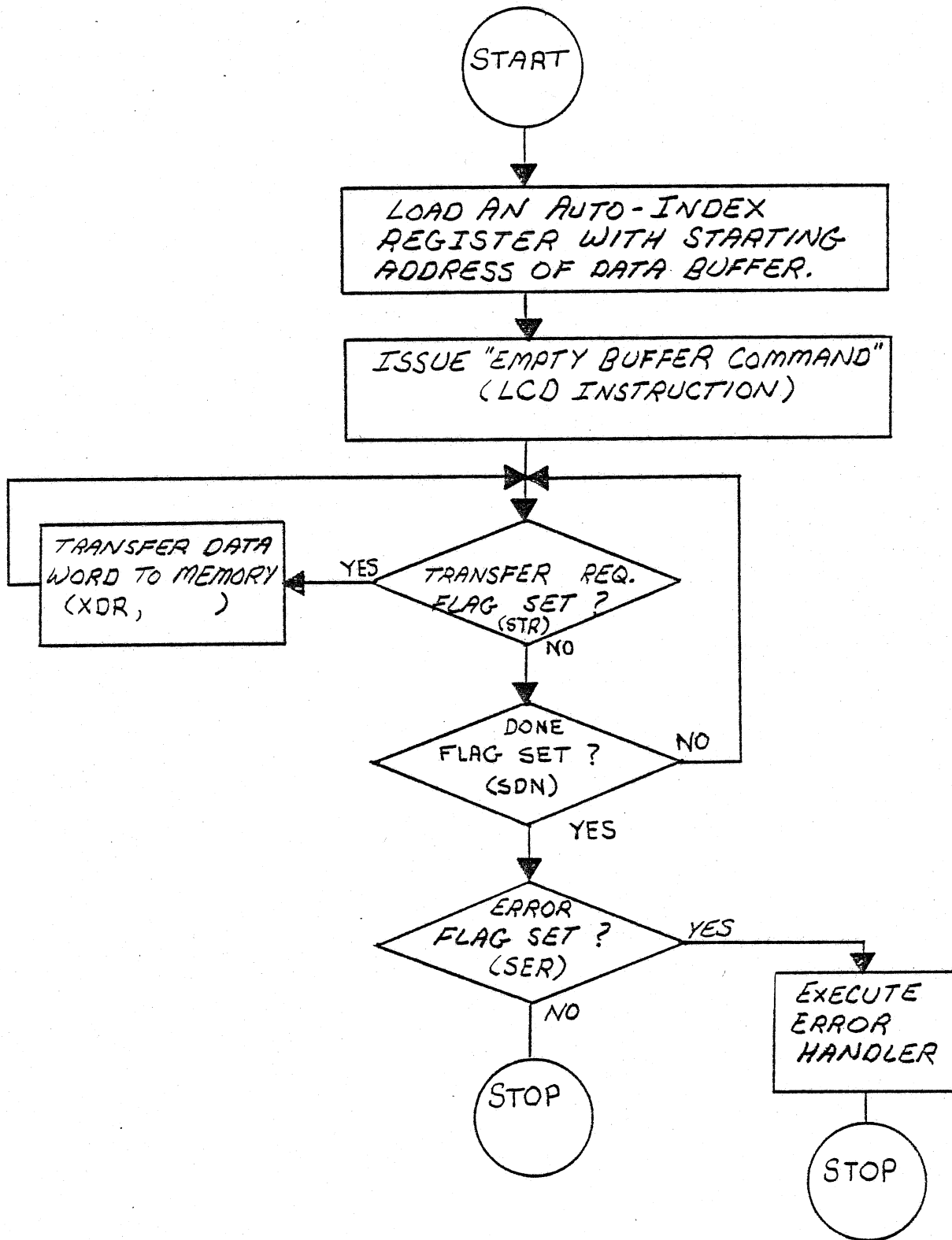
FILL BUFFER FLOW CHART



43 300 50 SHEETS 3 SQUARE  
43 388 100 SHEETS 3 SQUARE  
NATIONAL

FIGURE 5-19

EMPTY BUFFER FLOW CHART



47,381 50 SHEETS 5 SQUARE  
47,382 100 SHEETS 5 SQUARE  
47,383 200 SHEETS 5 SQUARE  
NATIONAL

### 5-2.3.2 EMPTY BUFFER (001)

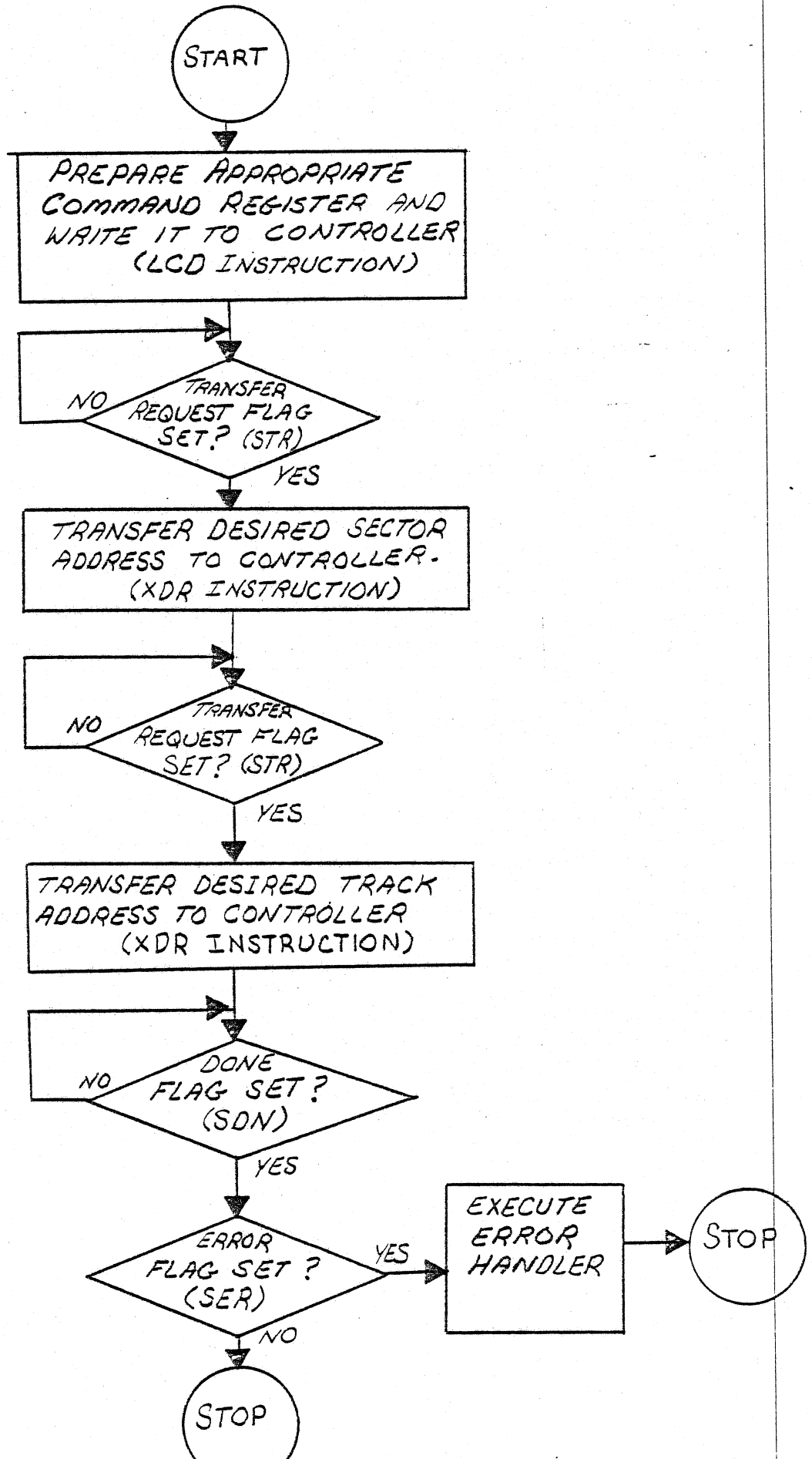
This function is very similar to the FILL BUFFER function except that words are moved from the interface register to the accumulator every time the XDR instruction is executed. Table 5-5 indicates the number of XDR cycles as a function of the word length and operating mode. Execution of this function does NOT modify the contents of the controller sector buffer. When the controller asserts the Transfer Request flag, this indicates that a word is currently in the interface register. The program must test the Transfer Request flag with the STR instruction before moving the word to the accumulator with the XDR instruction. When the buffer has been fully emptied, the controller will assert the Done flag and place the error/status register into the interface register. An XDR instruction executed after Done has been set will move the error/status register into the accumulator. An interrupt will occur when the controller sets the Done flag, assuming interrupts have previously been enabled.

### 5-2.3.3 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 instruction, which will also clear 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 will set the Transfer Request flag a second time. The program must test the flag with the STR instruction, which clears the flag, and then transfer a valid track address register to the controller using the XDR instruction. 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. Assuming compatible densities, the controller will now attempt 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 contents of the data buffer is not modified regardless of whether it is written successfully on the diskette or not. When the function has been completed, the error/status register is loaded into the interface register and the Done flag is set. An interrupt will be generated when Done is set assuming interrupts have been enabled.

FIGURE 5-20

WRITE / READ / WRITE D.D. FLOW CHART



42-384 100 SHEETS 3 SQUARE  
42-385 200 SHEETS 3 SQUARE  
NATIONAL

#### 5-2.3.4 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 instruction. When the flag becomes true, the skip will take place and Transfer Request will be cleared automatically. At this point, the program loads the accumulator with the sector address register and executes the XDR instruction to transfer the AC to the controller. The controller will assert the Transfer Request flag as soon as it is ready for a track address. The program should test Transfer Request and 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 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 will be terminated. If the densities are compatible, the controller starts looking 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 that mark to synchronize with the data. 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 check bytes produce a zero residue when sequentially passed through the CRC checker hardware circuits. If a CRC error is detected, the controller will set bit 11 of the error/status register, load an octal 200 into the error register, and then set 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 will produce an interrupt if interrupts have been enabled. If the data address mark found happened to have been 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.

### 5-2.3.5 SET MEDIA DENSITY (100)

If the controller is configured in Mode 1 (RX01 compatible), a command code of (100) does nothing except put the error/status register in the interface register and set 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 to make sure that it is configured in Mode 2. 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)8. If this particular key byte is not transferred to the controller at this time, the Error and Done error flags will be set and the function will terminate. The error register will be loaded with a (250)8 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 was a 0, a single density Data Address Mark and 128 8-bit FM zeros are written followed by 2 CRC bytes. If the density bit was a 1, a double density Data Address Mark and 256 8-bit DEC MMFM zeros are written followed by 2 CRC bytes. After writing all 26 sectors on track 0, the controller 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 flag will be 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 completely re-written again. If the SET MEDIA DENSITY function can not be made to complete normally because of an unreadable header, the WRITE-FORMAT option can be used to re-write the bad header. The SET MEDIA DENSITY function only writes the data fields, not the headers.



### 5-2.3.6 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 5-16 and 5-17 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 currently installed in the selected drive. The controller determines the density by loading the head, wherever it happens to be, and trying to read the first sector that passes by. If the density of the diskette happens to be different from the density indicated by the density bit in the command register, then the DENSITY ERROR bit of the error/status register will be set.

### 5-2.3.7 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. See paragraph 5-2.3.3 for a description of the WRITE SECTOR function. When the WRITE DELETED DATA SECTOR function is executed, the DELETED DATA bit will be set in the error/status register. When a sector which was written using this function is read at a later time, the DELETED DATA bit will be set in the error/status register.

### 5-2.3.8 READ ERROR REGISTER (111)

This function is used to obtain explicit error information after the error flag has been detected using the SER 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 instruction, and then transfer the error code from the interface register to the accumulator using the XDR 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 5-6. Some of these codes are only possible if the system is configured in Mode 2 (RX02 compatible).

OCTAL CODE	MEANING
00	NO ERROR
10	NO DRIVE 0 OR DRIVE 0 FAILED TO FIND TRACK 0 ON INIT
20	NO DRIVE 1 WHEN DIP SWITCH INDICATES THERE SHOULD BE A DRIVE 1 OR DRIVE 1 FAILED TO FIND TRACK 0 ON INIT
30	TRACK 0 FOUND WHILE STEPPING IN ON INITIALIZE
40	TRACK ADDRESS PASSED TO CONTROLLER WAS INVALID (>76)
50	TRACK 0 FOUND BEFORE DESIRED TRACK WHILE STEPPING
70	REQUESTED SECTOR NOT FOUND IN TWO REVOLUTIONS
100	WRITE PROTECT VIOLATION
110	DRIVE READ SIGNAL LOST
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
240	DENSITY ERROR
250	WRONG KEY FOR SET MEDIA DENSITY OR FORMAT COMMAND
260	INDETERMINATE DENSITY
300	DRIVE 2 FAILED TO HOME ON INITIALIZE
310	DRIVE 3 FAILED TO HOME ON INITIALIZE
320	READ/WRITE CONTROLLER DETECTED WRITE CIRCUIT FAILURE
330	READ/WRITE CONTROLLER TIME OUT ON RESET
340	MASTER CONTROLLER OUT OF SYNC WITH RD/WR CONTROLLER
360	DRIVE NOT READY DURING FORMAT COMMAND
370	AC POWER LOW CAUSED ABORT OF WRITE ACTIVITY

TABLE 5-6 ERROR REGISTER CODES

### 5-2.3.9 POWER FAIL

When a power failure occurs or DC power to the DSD 440 is interrupted, the controller gradually drains the filter capacitors in the power supply and ceases to function. Just prior to any diskette writing activity, the controller microprocessor checks the power level. If the voltage is below a certain level, the microprocessor will not begin writing. This feature can insure that complete sectors will always be written if the energy stored in the filter capacitors is sufficient to keep the controller functioning properly for at least one sector time (4.5 msec).

When power is returned, the controller will perform the following sequence of events:

- 1) Done flag is cleared.
- 2) Controller executes the hardware self-tests.
- 3) All drives are homed 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.
- 8) The Done flag is set.

### 5-2.3.10 DISKETTE FORMATTING IN MULTIPLE DENSITIES

Each time the write-format command protocol is executed, one entire track is re-written. The protocol starts when the user program sends the WRITE SECTOR function code (010) to the controller. The state of the density bit, whether it is transmitted in bit 3 or bit 11, will end up being 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 instruction, which will also clear the flag. Instead of a valid sector address, the user program specifies a single density track format operation by transferring a 152(8) to the controller with the xdr instruction. If the track format desired was DEC double density, then the number 153(8) is transferred. When the controller sees these "strange" 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 should respond by writing a valid track address into the data buffer register. Next, the controller enters a 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. (See figures 1-3, 1-4) 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 all types of hard sector interleaving to be easily implemented by the programmer to improve the effective throughput of the disk system.



## SECTION 6 HIGH LEVEL SOFTWARE AND THE DSD 440

This section of the manual discusses various aspects of using the DSD 440 with specific high level software. For PDP-11 and LSI-11 users, the software to be discussed will include the RT-11 operating system and the FRD440 system diagnostic program. Most RT-11 users who integrate a double density floppy disk into their system will find the procedures in this section very useful. The FRD440 program is shipped with the DSD 440 on a bootable diskette. The capabilities and use of this program are discussed.

### 6-1 GENERATING AN RX02 COMPATIBLE SYSTEM DISKETTE

In order to use the DSD 440 memory system while it is configured in the RX02 compatible mode, either the DYMNSJ.SYS or the DYMNFB.SYS monitor program must be installed on the system diskette. These programs are already installed in RT11-V03B on the QJ013-CX and QJ013-AX distributions of the operating system. If the version of RT11 distributed to you was intended to be booted on an RX01 flexible disk, RK05 hard disk, or some device other than the RX02, the procedure about to be described should be followed in order to generate an RX02 compatible system.

For simplicity, let us make the assumption that you have a version of RT11 intended for the RX01 floppy system. An RX01 type system will be required since your distribution of RT11-VO3B has a "DX-monitor". Your DSD 440 can be configured to look like an RX01, however it is shipped in RX02 compatible mode. If another RX01 compatible floppy disk system is not conveniently available, instructions describing how to reconfigure the DSD 440 for RX01 compatible operation can be found in section 3-6 of this manual. Basically, these instructions tell you to move switch 4 on the master controller circuit board to the "open" side, and install a jumper on the interface module (J12 on the LSI-11 interface and J4a on the PDP-11 interface).

The next step is to bootstrap your RX01 compatible distribution of RT11-VO3B on the system just located. Once the operating system is loaded, you must 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 some other diskette included in your RT11 distribution kit. When you have located the "DY-monitor" file, put it aside for the time being. Put a write-enabled blank diskette in Drive 1 and initialize the directory of that diskette by typing the following command to the RT11 monitor:

```
.INIT DX1:/NOQ
```

Next, copy all of files on the Drive 0 diskette to the Drive 1 diskette that you are going to want on your new RX02 compatible system diskette. If the file DYMNX.X.SYS, (where XX represents either SJ or FB), which you were just asked to locate, was NOT located on the Drive 0 diskette, then you must be sure that you leave at least 63 blocks of space on the Drive 1 diskette to accomodate this file. (Leave 74 blocks if you are using the FB monitor.) A good command, which will query you about each file is shown below:

```
.COPY/SYS/QU DX0: DX1:
```

As each file name is typed by the computer, type a "Y" if you want that particular file to be copied to the diskette in Drive 1. Be sure that the file DXMNSJ.SYS or DXMNFB.SYS is included in the files that you copy over. The "DY-monitor" file should also be included if it was found on your Drive 0 diskette.



Copy the DX bootstrap onto the new Drive 1 diskette by typing the following command to the RT11 monitor:

```
. COPY/BOOT DX1:DXMNSJ.SYS DX1:A
```

(Substitute DXMNFB.SYS if you are using the foreground/background monitor instead of the single job monitor.)

Remove the 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 you have not already copied the "DY-monitor" file onto the Drive 0 diskette, place the diskette which contained that file in Drive 1. Copy the file onto the Drive 0 diskette using the following command:

```
. COPY/SYS DX1:DYMNSJ.SYS DX0:DYMNSJ.SYS
```

The last step is to change the bootstrap on the diskette you have installed in Drive 0 so that when you bootstrap, the "DY-monitor" will be loaded instead of the "DX-monitor". To do this, type the following command:

```
. COPY/BOOT DX0:DYMNSJ.SYS DX0:A
```

You now have an RX02 compatible bootable diskette which should boot without any trouble on the DSD 440 system when it is configured in mode 2 (RX02 compatible). When you have reached this point, your new diskette will no longer boot on RX01 compatible hardware. If you changed your DSD 440 to be RX01 compatible to perform the steps just described, now is the time to change it back to RX02 compatible mode. After you have transferred this new diskette over to RX02 compatible hardware and successfully booted, it is safe to delete the "DX-monitor" which you are no longer using. This is done by typing the following command:

```
. DELETE/SYS/NOQ DXMNSJ.SYS
```

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

```
. SQ/NOQ DY0:
```

## 6-2 GENERATING DOUBLE DENSITY DISKETTES

Before you can generate a double density system (or file storage) diskette, diskettes with double density data address marks must be generated. These data address marks are what the controller uses to distinguish single density diskettes from double density diskettes. Ordinary IBM 3740 type single density diskettes can be very easily turned into double density diskettes if you have an RT11 utility program called FORMAT.SYS. If this file is not presently on your system diskette, locate it on one of the diskettes included in your RT11-VO3B distribution kit. Run the program by typing the command:

```
.R FORMAT
```

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

```
.RUN DY1:FORMAT
```

When this utility program is successfully loaded in memory and is ready to accept a command, it will type an \*. At this point, remove any diskette in Drive 1 and replace it with the blank diskette which you want to make into a double density diskette. The write enable tab should obviously be in place. To run the program, type the following command to the \* prompt:

```
*DY1:/Y
```

When the \* prompt returns, you can insert another diskette in Drive 1 and repeat the process if you want to generate some additional double density diskettes. If you want to return to the monitor, simply type a CNTRL C.

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

```
. INIT/NOQ DY1:
```

You are now ready to transfer files to your new double density diskettes. If the bootable system diskette you are currently using in Drive O is single density, it would probably be a good idea to generate a double density version of that diskette. To do that, type the following commands to the monitor:

```
. COPY/SYS DY0: *.* DY1:
```

```
. COPY/BOOT DY1: DYMNSJ.SYS DY1: A
```

When these commands are completed, you should place your new double density system diskette in Drive O and reboot the system. You should notice a lot of added room (indicated by the number of free blocks) on your system diskette.

## 6-3 FRD440 - FLOPPY DISK SYSTEM DIAGNOSTIC PROGRAM

All DSD 440 systems intended for connection to either PDP-11 or LSI-11 computers are shipped with a diskette containing a comprehensive diagnostic program called "FRD440". This particular section of chapter 6 will explain what is required to run this program and how to run the program.

### 6-3.1 PROGRAM LOADING AND MONITOR PROTOCOL

FRD440 will require at least 12K words of memory to run. When provided on diskette, the file FRD440.SAV and its sources can be located via the RT11 compatible directory. This assumes that the RT11 operating system has been bootstrapped off of either another diskette or some other mass storage peripheral. From within the RT11 monitor, the diagnostic program can be loaded into memory and started by typing:

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

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

In addition, the diagnostic diskette can be bootstrapped directly on a wide variety of system configurations. The bootstrap program previously described in section 3-10 will successfully boot the diagnostic diskette on floppy disk controllers that are configured to respond to addresses 777170-777172 or 777150-777152. The bootstrap program determines whether the floppy disk controller is presently configured in Mode 1 (RX01 compatible) or Mode 2 (RX02 compatible). The file FRD440.SAV is automatically loaded into memory. Control is automatically transferred to this program. At this point, the diagnostic diskette should be removed from the floppy disk drive so it does not get accidentally wiped out. Note that all information recorded on the diagnostic diskette is recorded in single density format. This, together with the fact that both bootstrap and diagnostic programs will handle the RX01 and RX02 protocols, make this particular diagnostic diskette useable on a large variety of DEC compatible flexible disk systems.

After FRD440 is loaded into memory, a brief operational description is typed out on the terminal. The version number of the program is also indicated at this time. A memory map is typed out showing which ranges of the address space respond with SSYNC (or BRPLY) when accessed by the C.P.U.

FRD440 will type <CRLF># when starting, and then the program attempts an INIT. If the INIT cycle is successful, the program will type the prompt word:

" MODE: "

This prompt string informs the operator that he can input a command. Each of the possible commands will be described in detail.

Legal responses to "MODE:" are listed below. Only the characters enclosed in parenthesis need be typed by the operator. The program will fill in the remaining characters and then proceed to execute the function. The parenthesis should not be typed.

(A)CCEPTANCE TEST  
(H)ELP  
(MA)P ADDR  
(F)ILL-EMPTY  
(SEQW)R  
(SEGR)D  
(RN)D RW  
(RD) RANDOM  
(SC)AN  
(RA)NDOM  
(SK) RANGE  
(SA) 125  
(ST)ATUS  
(RES) STATUS  
(SV)-STATUS  
(REC)OVER STATUS  
(DUMPC)  
(SI)NGLE  
(T)AP  
(SET )UNIT  
(SET-)TRACK  
(SEC)TOR INCREMENT  
(SETW)COUNT  
(I)NTERRUPT STT  
(SETD)EVICE  
(SH)ORT  
(V)ERIFY  
(REE)NTER ACCEPTANCE TEST  
(SETM)EDIA DENSITY  
(X)FORMAT REALLY  
(DUP)LICATE  
(C)OMPARE  
(DUMPD)  
(DUMPB)  
(DUMPA)

## 6-3.2 FRD440 PROGRAM FUNCTIONAL MODES

This section describes each functional mode. What the test or function does, as well as any communication protocol with the operator is discussed. In the event that some specific detail about a given function is not described here, a well commented listing of the program can be generated from the source files included on the diagnostic diskette.

### \* ACCEPTANCE TEST

The ACCEPTANCE test is generally used to verify that a floppy disk system is functioning properly after first being installed. The cumulative error status is maintained (see section 6-3.3), and can be observed at any time by simply typing a <LF>. The ACCEPTANCE test consists of an ordered execution of many of the more specialized tests about to be described. It will run indefinitely unless the operator stops the test with CTRL R. Error information will be displayed on the console terminal as it is detected. How to interpret the error/status messages will be discussed in the next section.

### \* HELP

The HELP command will cause all of the valid "MODE:" responses to be displayed on the console terminal. The "MODE:" prompt is typed when this function is complete.

### \* MAP ADDR

The MAP ADDRESS command will cause a memory and device address map of your system to be displayed on the console terminal. This is the same map that was displayed when the FRD440 program was first loaded. The "MODE:" prompt is typed when this function is complete.

### \* FILL-EMPTY

The FILL-EMPTY test checks everything associated with the FILL BUFFER and EMPTY BUFFER controller commands. If the controller under test is configured in RX01 compatible mode, then the test is relatively simple since only programmed I/O is involved. If the controller is configured as an RX02, the controller does FILL/EMPTYIES into three different buffers so as to verify proper operation of all possible address bits. FILL/EMPTYIES are done in both densities covering all possible word counts. This test will run until the operator types a CTRL R.

\* SEQWR

The SEQUENTIAL WRITE READ test sequentially writes pseudo-random data on all selected drives. The test then reads all the data and checks it against what was written. 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 will continue running until the operator types CTRL R. This test will do 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 will be reported if this is not done.

\* SEQRD

The SEQUENTIAL READ test sequentially reads the data on all selected drives, comparing the data pattern against what was written. The program types "READING" at the beginning of each pass.

\* RND RW

The RANDOM READ/WRITE test selects a random sector on one of the selected drives and then reads or writes it, checking data when appropriate.

\* RD RANDOM

The READ RANDOM test will read randomly selected sectors. The data is checked following each read performed.

\* SCAN

The SCAN test determines the density of, and sequentially reads all sectors on all selected drives and checks for CRC errors. No direct data checking takes place in this test, only status is checked.

\* RANDOM

The RANDOM test reads randomly selected sectors on all selected drives. Only status is checked.

\* SK RANGE

The SEEK RANGE function is an extremely versatile drive test routine that does all possible seeks covering within the operator specified track and seek length boundaries.

\* SA 125

The SA 125 test is intended to be used in conjunction with a special test diskette available from Shugart Associates. The part number of this diskette is SA125. The test can determine the off-track margins and head alignment of a floppy drive without the use of any test equipment. These measurements can be made by reading one of two specially written tracks on the SA125 diskette. These tracks were formatted with the heads in alignment and then data was written with 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 being executed, a graphic display indicating which sectors have been read correctly, is repeatedly output 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 (- 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 test, the user should type "SA" in response to the "MODE:" prompt typed by the FRD440 program. Next, the FRD440 program will ask "UNIT:". The user should then insert a write protected SA125 test diskette into the floppy drive to be tested, and then 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), the user simply types a carriage return. To select the other test track, the user should 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. Be sure to remove the SA125 test diskette when it is not being used.

#### \* STATUS

The STATUS function will cause all of 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 RES STATUS. The "MODE:" prompt is typed when this function is complete.

#### \* RES STATUS

The RESET STATUS function will first display all of the currently available status counts. Next, the operator will be queried as to whether he really wants to reset all of the status counts. If the operator responds 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.

#### \* SV-STATUS

The SAVE STATUS command will cause all of the status counts associated with a particular drive to be written on track 0 sector 1 of the diskette installed 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 made use of by the acceptance test so that it can survive a loss of C.P.U. memory without any loss of cumulative error data. The "MODE:" prompt is typed when this function is complete.

#### \* RECOVER STATUS

The RECOVER STATUS function does exactly 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 typed when this function is complete.

#### \* DUMPC

The DUMPC function is used to display the circular output buffer associated with all console terminal output. This function would be useful on systems where the console terminal was a CRT of some kind. Messages that were previously output can be re-examined on the console.

#### \* SINGLE

The SINGLE function causes an operator specified drive, track, and sector address to be continuously read. This test would be useful in determining if a particular sector was prone to getting intermittent CRC errors. SINGLE could also be used to measure head/media wear because the head remains in contact with the media continuously.

#### \* TAP

The TAP function repeatedly reads an operator specified sector, as above, but in this test the head is unloaded after each read operation. TAP could be used to measure head/media wear.

## MODE SETTING COMMANDS:

### \* SET UNIT

This function enables the operator to specify which floppy disk drives are to be accessed by the various test functions. The default drives are units 0 and 1. This function prompts with "UNIT:", expecting a number <0-3>. Unit numbers are accepted, as long as they are valid. The "MODE:" prompt is issued as soon as an invalid character is entered.

### \* SET-TRACK

This function enables the operator to specify lower and upper track limits for all the other test functions. The default lower track limit is track 0 and upper track limit is track 76. The "MODE:" prompt is issued after the new limits have been entered by the operator.

### \* SECTOR INCREMENT

This function enables the operator to specify the sector increment value. This number is added to the present sector address to determine the next sector address in many of the test functions that read multiple sectors on a single track. If this number were simply 1, and the diskette did not have an interleaved format, an entire revolution would probably be required to read each sector. On PDP-11 processors the default increment value is 2. On LSI-11 processors, the default is 3. The "MODE:" prompt is issued after the new value has been entered by the operator.

### \* SETWCOUNT

The SET WORD COUNT function enables the operator to specify the number of words that will be transferred when the DSD 440 is instructed to perform a DMA FILL/EMPTY BUFFER operation. Only test functions which do data checking use this word count variable. The FILEMP test controls word count independently of this variable. The default word count stored in this variable is 64 words in SD and 128 words in DD.

### \* INTERRUPT STT

The SET INTERRUPT STATUS function enables the operator to test the floppy disk system with interrupts either enabled or disabled. If interrupts are enabled, the program makes sure that an interrupt does in fact occur whenever it is supposed to. The operator enters a 0 to disable interrupts and a 1 to enable interrupts.

\* 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 will ask the operator for device address, interrupt vector, and flag word. If a space is typed, the program will step past that field, leaving it intact. To return to "MODE:", type a <CR> 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	S50	DDN	MPX	US3	US2	US1	US0	MPN	MPN

Where, when set to a 1, the bit labeled:

- DMA indicates the device is capable of performing DMA.
- DBS indicates the device is double sided.
- S50 indicates that the device uses Shugart SA950 drives.
- DDN indicates double density operation.
- MPX indicates a multiplexed 110 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)

The normal flag variable is 4400 octal.



\* SHORT

This function changes some of the variables used by the ACCEPTANCE test such that only the first 9 tracks of each selected drive will be tested. It then starts the ACCEPTANCE test.

\* VERIFY

The VERIFY test does a short ACCEPTANCE test, as described above, and then it resets the limit variables back to the normal default values. It then induces an automatic CTRL P to inhibit all but error printout and jumps to the ACCEPTANCE test.

\* REENTER ACCEPT

This function re-enters the ACCEPTANCE test, but after the seek tests have been performed.

FORMAT INITIALIZATION COMMANDS:

\* SET MEDIA DENSITY

This function enables the operator to initialize a diskette to single density or double density format. The function will prompt for function conformation, unit, and desired density. Note that this function will cause any status that was saved on track 0 sector 1 to be erased.

\* XFORMAT

This function is used to re-write diskette headers, as well as all the other data on a particular diskette. The function will prompt the operator for confirmation, unit, and standard or interleaved format.

Standard track header format is 01, 02, 03, ... 24, 25, 26 .. RT-11 maps block numbers 0,1... into the sector sequence 1,3,5,7, 9,11,13,15, ... to achieve what is called a "two-way interleave". This is done in order to provide enough time to process each sector before the next sector comes around on the disk. 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 function, a net three way system interleave is achieved. This has been shown to greatly improve system throughput when there is heavy I/O overhead, as often occurs under the Foreground/Background monitor.

## DUMP AND COPY UTILITY COMMANDS

### \* DUPL

The DUPLICATE utility command enables the operator to make a duplicate copy of a diskette. The function will prompt for a source drive unit number and then a destination drive unit number. For each possible sector address, the function will do 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 will prompt for: SOURCE UNIT, STARTING TRACK, STARTING SECTOR, NUMBER OF SECTORS, and DESTINATION UNIT.

### \* DUMPO

This utility command enables the operator to cause an octal dump of specified sectors to the console terminal. The function will prompt for: UNIT, STARTING TRACK, STARTING SECTOR, and NUMBER OF SECTORS.

### \* DUMPB

This utility command enables the operator to cause a binary dump of specified sectors to the console terminal. The function will prompt for: UNIT, STARTING TRACK, STARTING SECTOR, and NUMBER OF SECTORS.

### \* DUMPA

This utility command enables the operator to cause an ASCII dump of specified sectors to the console terminal. The function will prompt for: UNIT, STARTING TRACK, STARTING SECTOR, and NUMBER OF SECTORS.

There are several control characters which FRD440 will respond to at any time. These characters, and the responses which they invoke, are listed below:

CTRL R RESTARTS THE PROGRAM AT THE "MODE:" PROMPT  
CTRL S HANG OUTPUT TO TERMINAL UNTIL ANOTHER CHAR TYPED  
CTRL O THROWS AWAY OUTPUT UNTIL ANOTHER CHAR TYPED  
CTRL P THROWS AWAY ALL OUTPUT EXCEPT FOR ERRORS UNTIL  
ANOTHER CHAR TYPED  
<LF> TYPES CURRENT TRACK/SECTOR AND STATUS COUNTS  
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,  
CTRL D 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 WHERE YOU CAME FROM.  
CTRL W RETURNS PROGRAM CONTROL TO JUST PAST WHERE A TRAP WAS CAUSED

### 6-3.3 FRD440 PROGRAM STATUS AND ERROR PRINTOUTS

FRD440 will type 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 the operator desires to examine some error printouts that are no longer on the face of the CRT screen. The DUMPC function is used to examine messages in the circular buffer. The remainder of this section is an explanation of each of the status variables that might appear on the console terminal.

DEV<XXX> IS PRINTED ONLY WHEN RUNNING MULTIPLE CONTROLLERS. XXX IS 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 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 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.

ERREG: <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 OF THE USERS MANUAL.



```

; TITLE DSD 440 BOOTSTRAP PROM
; BOT440.MAC 21-MAR-79
; BOOTSTRAP FOR DSD440 FLOPPY DISK CONTROLLER
; BOOTS EITHER SINGLE OR DOUBLE DENSITY FLOPPIES
; ALSO WORKS WITHOUT CHANGE IN RX01 - DSD-210 MODE.
; NOTE - THE DISKETTE BEING BOOTED MUST HAVE THE CORRECT MONITOR
; FOR THE EXISTING HARDWARE CONFIGURATION.
; ** NOTE ON BOOTING WHILE REAL TIME CLOCK IS ENABLED. **
; THIS BOOT CAN BE STARTED WITH A RUNNING REAL TIME CLOCK IN 2 WAYS.
; 1) ENSURING THAT THE STACK IS POINTING TO NON-EXISTANT MEMORY THUS
; FORCING A DOUBLE BUS ERROR ON ANY INTERRUPT AND TYPING
; "173000G" AND TYPING "P" IF HALTS OCCUR DUE TO ATTEMPTED INTERRUPTS.
; 2) BY SETTING THE PSW AHEAD OF TIME TO DISABLE INTERRUPTS BY TYPING
; "%S/ 340<CR>" AND "R7/ 173000<CR>" AND HITTING "P".
; IF A 173000G IS TRIED AND A CLOCK INTERRUPT OCCURS AFTER THE
; FIRST INSTRUCTION AND BEFORE THE THIRD INSTRUCTION THEN TYPE "P"
; UNTIL THE CLOCK IS DISABLED.

```

; THE BOOTSTRAP PROCEEDS IN 4 STEPS

```

; 1) SELECT DEVICE DETERMINES DEVICE TO BE BOOTED
; 2) RAM TEST CHECKS ALL AVAILABLE MEMORY FOR STUCK BITS
; ON BOTH DATA AND ADDRESS LINES. <0-30K>
; DOES BOTH DATA = ADDRESS AND PATTERN TESTS
; 1) CLEARS MEMORY TO 0'S AND SIZES MEMORY
; 2) LOADS MEMORY = ADDRESS AND CHECKS
; 3) LOADS MEMORY = ADDRESS COMPLEMENT, CHECKS
; 4) LOADS MEMORY WITH THE REPEATING PATTERN OF
; 131617, 154707, 166343, 173161, 175470
; 3) FILL-EMPTY CHECKS DSD440 - PROCESSOR DATA PATH FOR
; SYNTAX AND DATA ERRORS. ALSO INSURE'S ALL
; AVAILABLE ADDRESS LINES TOGGLE UNDER DMA.
; CHECKS FILL-EMPTY WITH BUFFERS AT 774,
; 17700, 37676, 77704, 137700 IF MEMORY EXISTS.
; 4) BOOTSTRAP READS IN BLOCK 0 FROM DISKETTE IN EITHER
; RX01 OR RX02 MODE AND STARTS AT LOC 0
; ALSO SELECTS CORRECT DENSITY IN RX02 MODE.

```

; ERROR HALTS OR HANG UP LOOPS (ADDRESSES RELATIVE TO BOOT BASE ADDR)

```

; 156 HALT MEMORY ERROR AT LOC -2(R4), READ R0, EXPECT ZERO
; 204 HALT MEMORY ERROR AT -2(R4), READ R0, EXPECT 0
; 252 HALT 1) FILL-EMPTY ERROR IF R5=BOOT+522, SP=5000
; 2) MEMORY ERR IF R5=BOOT+112, SP=5002
; 314 LOOP DEVICE ADDRESS SELECTED FOR BOOTING DOESN'T RESPOND
; 324 HALT ERROR FLAG IN RXCS SET AFTER INIT
; 342 HALT RXCS INTERFACE REGISTER STUCK BIT PROBLEM
; 364 HALT RXDB INTERFACE LATCH PROBLEM, NOTE C(RXDB)
; 400-402 LOOP DSD440 TRANSFER REQUEST HANGUP (FILL-EMPTY)
; 414-416, 452-454 TRANSFER REQUEST HANGUP (FILL-EMPTY)
; 576-600, 604-606 TRANSFER REQUEST HANGUP (BOOTSTRAP)
; 652-654, 666-670 TRANSFER REQUEST HANGUP (BOOTSTRAP)
; 742-746 LOOP DSD440 FLAG WAIT ROUTINE HANGUP
; 774 HALT FLOPPY READ ERROR, PROCEED TO TRY NEXT DRIVE
; C(SP) = DEFINITIVE ERROR STATUS
; C(R5) = SECTOR # WITH PROBLEM
; C(R0) = DRIVE # WITH ERROR

```

THIS USUALLY HAPPENS WITH A BAD DISKETTE AND MAY OCCUR  
IF AN UN-BOOTABLE DISKETTE IS IN DRIVE 0. A "PROCEDE"  
FROM HERE RESULTS IN ATTEMPTING TO BOOT THE OTHER DRIVE.

```

; START ADDRESSES
; BOOT+0 (TYPICALLY 173000) BOOTS DEVICE WITH RXCS AT 177170
; BOOT+20 (TYPICALLY 173020) BOOTS DEVICE WITH RXCS AT 177150
; BOOT+40 (TYPICALLY 173040) GENERAL DEVICE ENTRANCE - USER
;
; SET'S R0=340, R1=2, LOC 0 = DESIRED RXCS
; IF REAL TIME CLOCK MUST BE LEFT ON THEN SET
; $S/ 340<CR> AND R7/ 173040<CR> AND PROCEED
;
; A "BOOT" ON AN 11/04 OR 11/34 PRINTS R0, R4, SP, R7 ON THE TERMINAL.
; IF AN ERROR HALT OCCURS AT BOOT+774 WHILE BOOTING THEN
; BOOTING AGAIN ON AN 11/04 OR /34 PRINTS OUT THE FOLLOWING.
;
; R0 = CURRENT DRIVE # BEING BOOTED FROM.
; R4 = LOAD ADDRESS WHERE ERROR OCCURRED
; SP = DEFINITIVE STATUS OF ERROR
; R7 = ERROR HALT ADDR+2
;
; NOTE - A HALT OR HANGUP OCCURRING BETWEEN 742-746 THAT WILL NOT
; RESPOND TO BREAK OR HALT IS GENERALLY DUE TO LACK OF DMA GRANT
; CONTINUITY ON THE BUS. USER SHOULD PUT DSD440 INTERFACE CARD
; CLOSER TO THE PROCESSOR AND ENSURE GRANT CONTINUITY.

```

; DSD440 - RX02 REGISTER SYNTAX DEFS

RXCS=177170

ERR	INI	XM	XM	X02	??	SID	DEN	TRQ	IEN	DON	UNI	FUN	FUN	FUN	GO
.ERR=	100000														
	40000														
	30000														
.DBDMA=	4000														
	400														
	200														
	16														

RXDB=RXCS+2

; RXES ERROR BIT LAYOUT

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

```

; R0 UNIT # BOOTED FROM (0, 1)
XCS= %1 ; R1 POINTER TO RXCS
XDB= %2 ; R2 POINTER TO RXDB
; R3 READ COMMAND VAL WITH DENSITY BIT
LDP= %4 ; R4 LOAD POINTER
SCT= %5 ; R5 CURRENT SECTOR # (1, 3, 5, 7)
; (SP) WORD COUNT FOR CURRENT DENSITY

```

; START HERE FOR DEVICE 177170 BOOT

```

000000 012706 BOT170::MOV #-1, SP ; INHIBIT INTERRUPTS IN ONE INSTRUCTION
177777
000004 012700 MOV #340, R0 ; SET PROCESSOR STATUS WORD
000340
000010 106400 MTPS R0 ; FROM REG SINCE READ-MODIFY-WRITE
; CYCLE INTO PROM CAUSES TIMOUT
; ABOVE 5 WORDS BECOME / MOV #340, R0 / MOV R0, @#177776 /
; / NOP / IN PDP-11 BOOT
000012 012710 MOV #177170, (R0) ; SET DEVICE ADDRESS
177170
000016 000406 BR BOTCOM
000020 012700 BOT150: MOV #340, R0 ; 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-EXISTANT MEMORY TIMEDUT
; SETS CARRY AND RETURNS ON NON-EXISTANT 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     R0, (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?
      170000
; STOP AT 160000 IF PDP-11

000104 103773      BLD     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

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*9816.>         ; 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$: MOV R4, RO ; KILL Z FLAG <MOV R4, (R4)+>
000142 010024 MOV RO, (R4)+ ; LOAD CONTENTS = ADDRESS
000144 020402 CMP R4, R2 ; AT END OF TEST?
000146 103774 BLO 2$
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
131617
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), RO ; 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 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 000426
000314 103777 BCS ; LOOP IF NO BUS RESPONSE
000316 032711 BIT #.ERR!.DBDMA, (R1) ; ERROR SET OR RX02?
000400 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
000334 001420
000334 022711 CMP #5460, (XCS) ; DID THEY LATCH OK?
0005460
000340 001401 BEQ .+4
000342 000000 HALT ; STUCK BITS IN RXCS
000344 022712 CMP #1420, (XDB) ; LATCHED OK IN RXDB?
000420 001420
000350 001005 BNE RXHALT ; NO - BAD INTERFACE.

000352 012712 RXDBTS: MOV #173767, (XDB) ; CHECK RXDB LATCH
000356 173767
000356 022712 CMP #173767, (XDB) ; DID THEY LATCH
000356 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 000200
000374 012722 MOV #401, (XDB)+ ; DO FILL COMMAND
000401 000401
000400 105711 TSTB (XCS) ; WAIT FOR TRREQ
000402 100376 BPL .-2
000404 032711 BIT #.DBDMA, (XCS) ; RX02 STYLE FILL?
000400 004000
000410 001404 BEQ FILX01 ; NO - DO RX01 STYLE PROG XFER
000412 011612 MOV (SP), (XDB) ; WORDCOUNT (=200)
000414 105711 TSTB (XCS) ; WAIT FOR TRREQ
000416 100376 BPL .-2
000420 010412 MOV R4, (XDB) ; BUFFER ADDR
000422 004767 FILX01: CALL WTFLAG ; WAIT FOR DONE, ERROR, OR TRREQ
000314 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 000100
000440 000770 BR FILX01 ; CHECK FOR ANOTHER BYTE

```

## ; 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 IT 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     WTFLAG      ; 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      BR      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
000518 177514
000522 000662      BR      FILEMP      ; DO NEXT FILL-EMPTY

```

```

; BOOT THE DEVICE IN LOC 0, REGISTERS USED AS INDICATED BELOW
000001 XCS= %1 ; R0 UNIT # BOOTED FROM (0, 1)
000002 XDB= %2 ; R1 POINTER TO RXCS
; 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 #.
177776
000544 004367 JSR R3, UNTDEC ; GEN A POINTER INTO RDTBL
000002
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
000100
000562 012705 MOV #1, SCT ; INIT SECTOR TO READ
000001

000566 004767 RDLP: CALL WTFLAG ; WAIT FOR DONE FLAG SET?
000150
000572 010102 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
000001
000614 004767 CALL WTFLAG ; WAIT FOR DONE
000122
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?
000020
000630 001450 BEQ DEFNST ; NO- DO DEFINITIVE STATUS
000632 052703 BIS #400, R3 ; SET COMMAND TO DOUBLE DENSITY
000400
000636 012716 MOV #200, (SP) ; SET TO D.D. WORD COUNT
000200
000642 000751 BR RDLP ; AND TRY READING AGAIN

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

```





SYMBOL TABLE

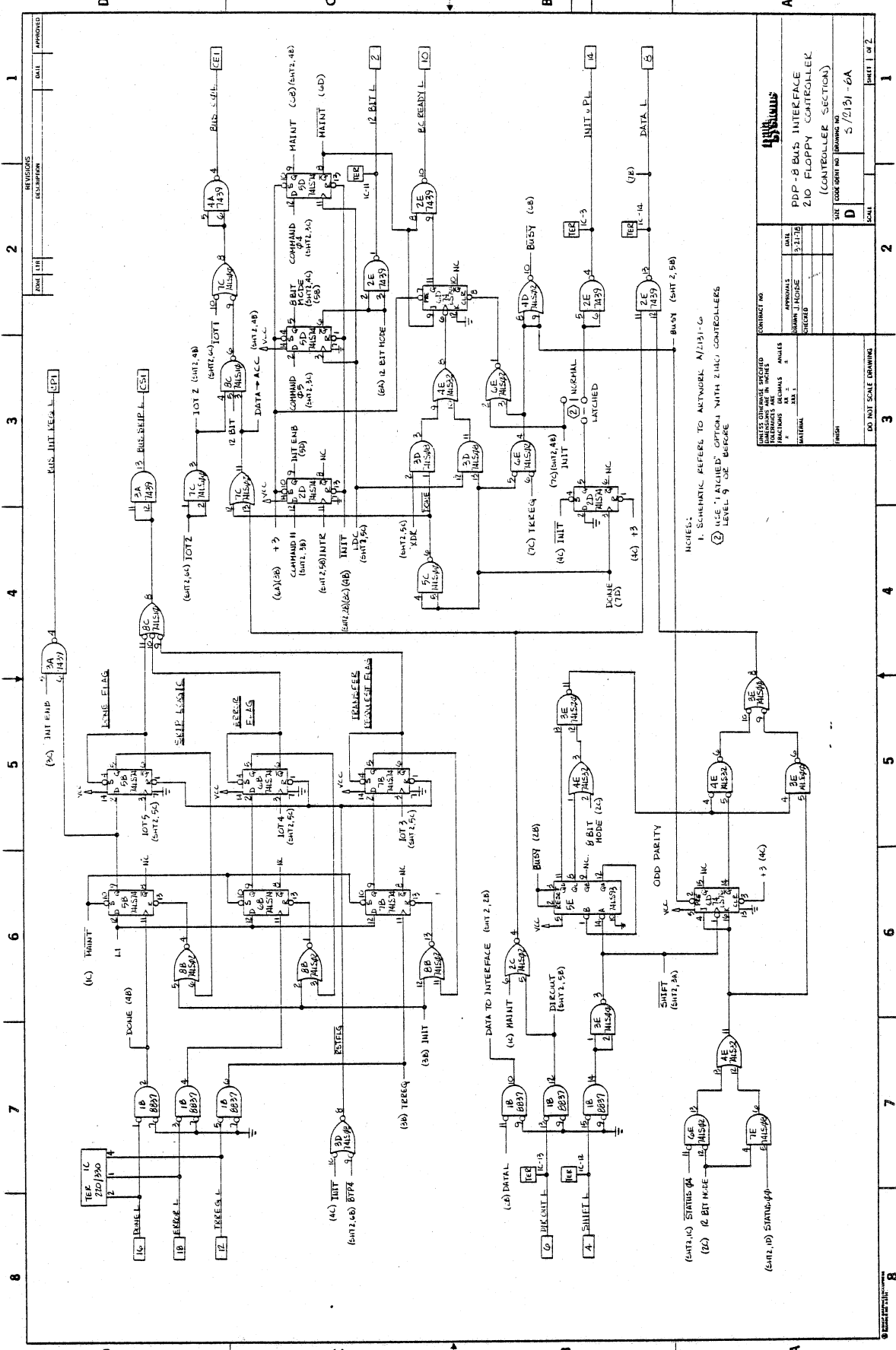
BOOTR1	000534R	002	EMPBUF	000644R	002	RXDB	= 177172	
BOTCOM	000034R	002	EMPDON	000714R	002	RXDBTS	000352R	002
BOTGEN	000040RG	002	EMPX01	000474R	002	RXFIEM	000366R	002
BOTLST	001000RG	002	FILBUF	000302R	002	RXHALT	000364R	002
BOT150	000020R	002	FILEMP	000270R	002	SCT	=%000005	
BOT170	000000RG	002	FILEXT	000266R	002	SETCOM	000164R	002
BOT440	000524R	002	FILX01	000422R	002	SETPAT	000214R	002
CHKADP	000150R	002	LDP	=%000004		TRAP4	000052R	002
CHKCDM	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 50 PAGES  
 BOT440,BOT440<BOTLLX







REVISED DESCRIPTION DATE APPROVED

1 2 3 4 5 6 7 8

CONTRACT NO.		PDP-8 BUS INTERFACE	
DRAWING NO.		Z10 FLOPPY CONTROLLER	
DATE		(CONTROLLER SECTION)	
APPROVALS		DATE	
DESIGNED BY		5/2/75	
CHECKED BY		S/2/75 - 6A	
DRAWN BY		DATE	
MATERIAL		DO NOT SCALE DRAWING	
FINISH		1	

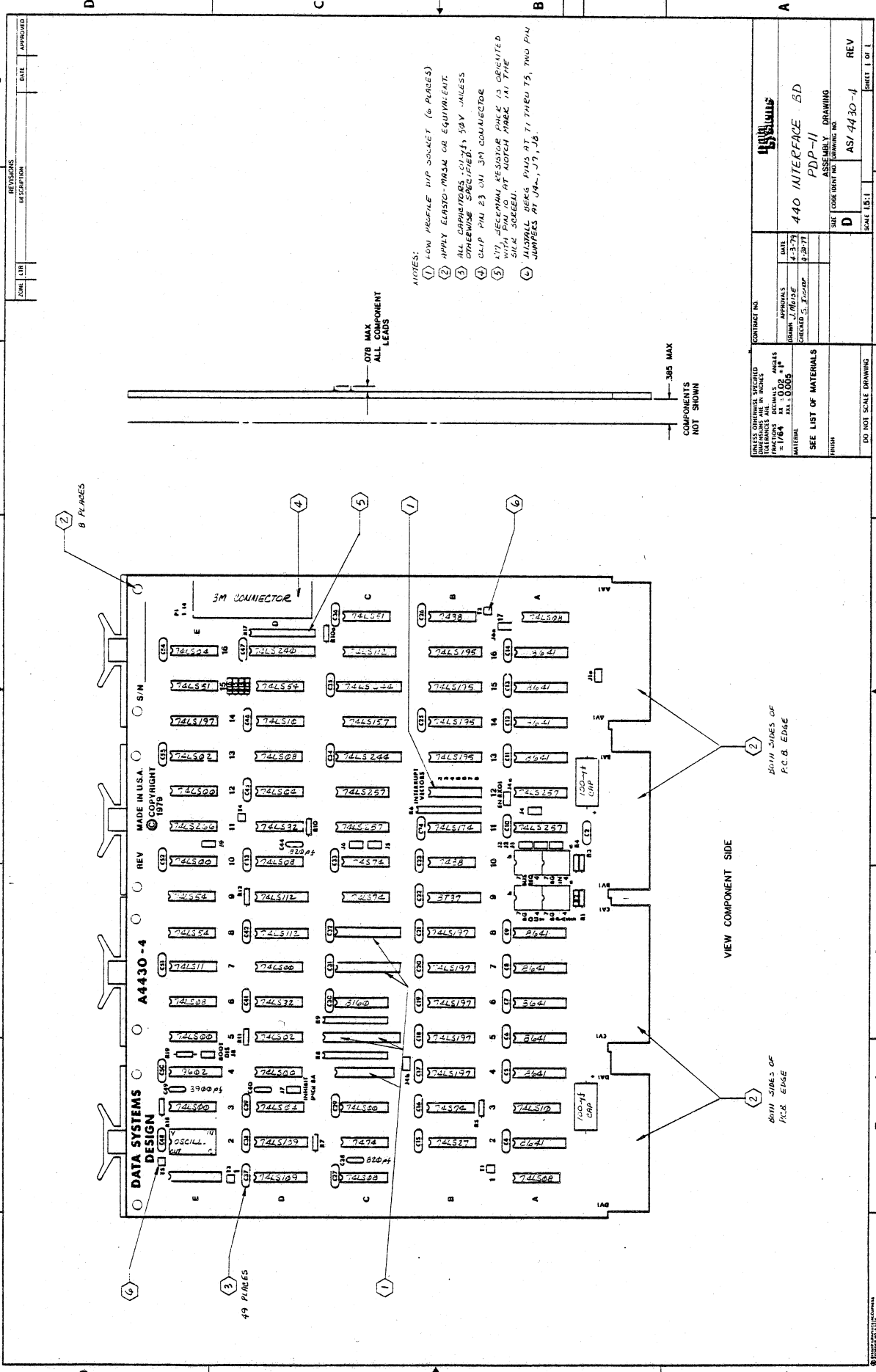
NOTES:

1. SCHEMATIC REFERS TO NETWORK A/2131-6

2. USE 'LATCHED' OPTION WITH Z10 CONTROLLERS

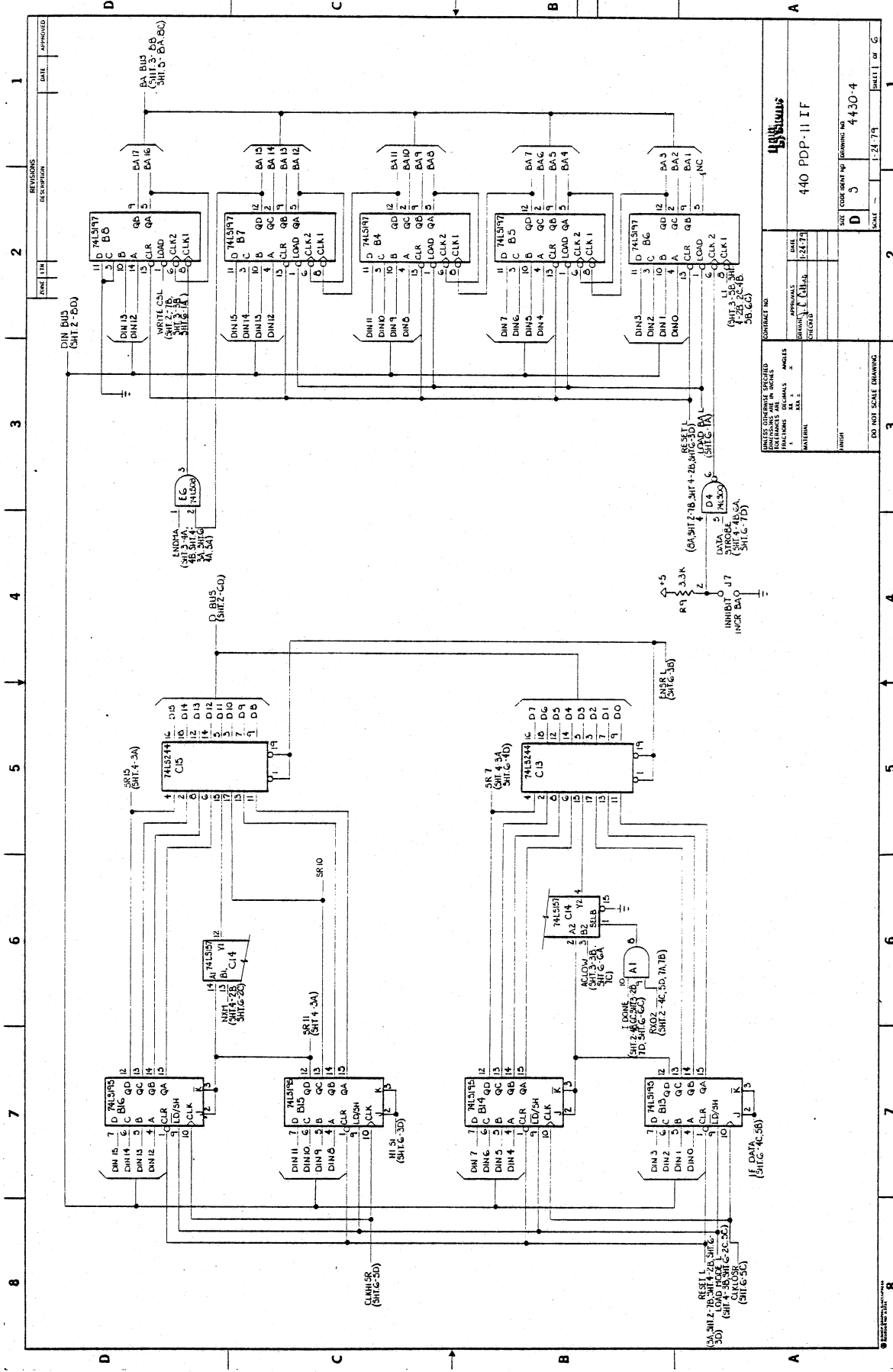
3. LEVEL 9 USE BURGER





- NOTES:
- LOW PROFILE DIP SOCKET (6 PLACES)
  - APPLY EASO-MASK OR EQUIVALENT
  - ALL CAPACITORS MUST BE 50V UNLESS OTHERWISE SPECIFIED
  - CLIP PIN 23 ON 3M CONNECTOR
  - 100-ohm RESISTOR PACK IS ORIENTED WITH PIN NO. AT RIGHT HAND SIDE SILK SCREEN.
  - 113 SMALL DIERS PINS AT T1 THRU T5, TWO PIN JUMPEES AT J4, J7, J8.

CONTRACT NO.		DATE	
APPROVALS		DATE	
DESIGNER		DATE	
CHECKER		DATE	
DRAWING NO.		DATE	
MATERIAL		DATE	
FINISH		DATE	
SEE LIST OF MATERIALS		DATE	
DO NOT SCALE DRAWING		DATE	
SCALE 1:1		DATE	
SHEET 1 OF 1		DATE	



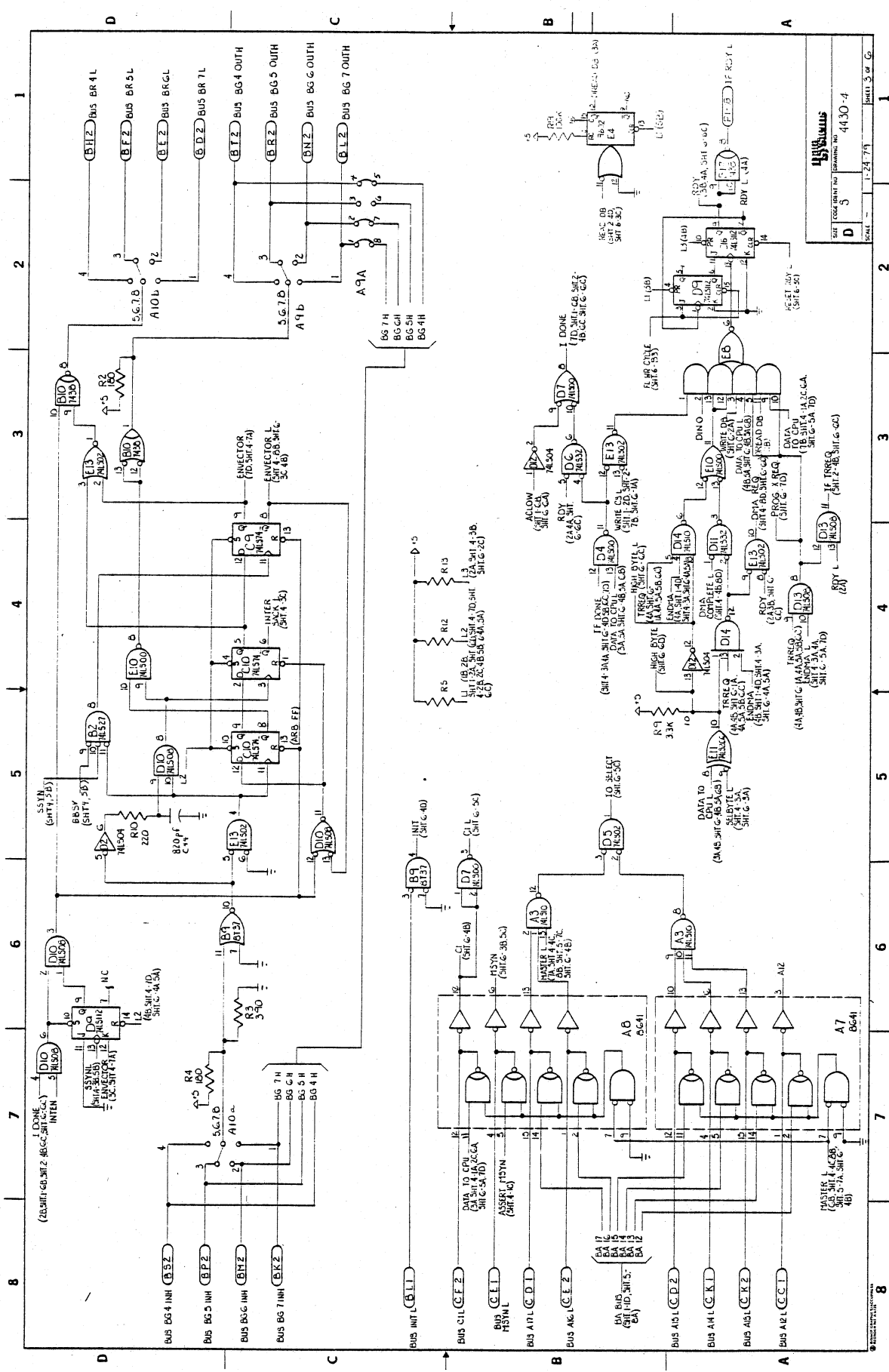
REV	DATE	DESCRIPTION	APPROVED
1			

CONTRACT NO.	440 PDP-11 IF
DATE	1-24-79
SCALE	1:1
SHEET NO.	4430-4
TOTAL SHEETS	4

UNLESS OTHERWISE SPECIFIED	ALL DIMENSIONS ARE IN INCHES
FRACTIONS	DECIMALS
1/8	0.125
1/4	0.250
3/8	0.375
1/2	0.500
3/4	0.750
1	1.000
1 1/8	1.125
1 1/4	1.250
1 3/8	1.375
1 1/2	1.500
1 5/8	1.625
1 3/4	1.750
1 7/8	1.875
2	2.000
2 1/8	2.125
2 1/4	2.250
2 3/8	2.375
2 1/2	2.500
2 5/8	2.625
2 3/4	2.750
2 7/8	2.875
3	3.000
3 1/8	3.125
3 1/4	3.250
3 3/8	3.375
3 1/2	3.500
3 5/8	3.625
3 3/4	3.750
3 7/8	3.875
4	4.000
4 1/8	4.125
4 1/4	4.250
4 3/8	4.375
4 1/2	4.500
4 5/8	4.625
4 3/4	4.750
4 7/8	4.875
5	5.000
5 1/8	5.125
5 1/4	5.250
5 3/8	5.375
5 1/2	5.500
5 5/8	5.625
5 3/4	5.750
5 7/8	5.875
6	6.000
6 1/8	6.125
6 1/4	6.250
6 3/8	6.375
6 1/2	6.500
6 5/8	6.625
6 3/4	6.750
6 7/8	6.875
7	7.000
7 1/8	7.125
7 1/4	7.250
7 3/8	7.375
7 1/2	7.500
7 5/8	7.625
7 3/4	7.750
7 7/8	7.875
8	8.000
8 1/8	8.125
8 1/4	8.250
8 3/8	8.375
8 1/2	8.500
8 5/8	8.625
8 3/4	8.750
8 7/8	8.875
9	9.000
9 1/8	9.125
9 1/4	9.250
9 3/8	9.375
9 1/2	9.500
9 5/8	9.625
9 3/4	9.750
9 7/8	9.875
10	10.000
10 1/8	10.125
10 1/4	10.250
10 3/8	10.375
10 1/2	10.500
10 5/8	10.625
10 3/4	10.750
10 7/8	10.875
11	11.000
11 1/8	11.125
11 1/4	11.250
11 3/8	11.375
11 1/2	11.500
11 5/8	11.625
11 3/4	11.750
11 7/8	11.875
12	12.000
12 1/8	12.125
12 1/4	12.250
12 3/8	12.375
12 1/2	12.500
12 5/8	12.625
12 3/4	12.750
12 7/8	12.875
13	13.000
13 1/8	13.125
13 1/4	13.250
13 3/8	13.375
13 1/2	13.500
13 5/8	13.625
13 3/4	13.750
13 7/8	13.875
14	14.000
14 1/8	14.125
14 1/4	14.250
14 3/8	14.375
14 1/2	14.500
14 5/8	14.625
14 3/4	14.750
14 7/8	14.875
15	15.000
15 1/8	15.125
15 1/4	15.250
15 3/8	15.375
15 1/2	15.500
15 5/8	15.625
15 3/4	15.750
15 7/8	15.875
16	16.000
16 1/8	16.125
16 1/4	16.250
16 3/8	16.375
16 1/2	16.500
16 5/8	16.625
16 3/4	16.750
16 7/8	16.875
17	17.000
17 1/8	17.125
17 1/4	17.250
17 3/8	17.375
17 1/2	17.500
17 5/8	17.625
17 3/4	17.750
17 7/8	17.875
18	18.000
18 1/8	18.125
18 1/4	18.250
18 3/8	18.375
18 1/2	18.500
18 5/8	18.625
18 3/4	18.750
18 7/8	18.875
19	19.000
19 1/8	19.125
19 1/4	19.250
19 3/8	19.375
19 1/2	19.500
19 5/8	19.625
19 3/4	19.750
19 7/8	19.875
20	20.000
20 1/8	20.125
20 1/4	20.250
20 3/8	20.375
20 1/2	20.500
20 5/8	20.625
20 3/4	20.750
20 7/8	20.875
21	21.000
21 1/8	21.125
21 1/4	21.250
21 3/8	21.375
21 1/2	21.500
21 5/8	21.625
21 3/4	21.750
21 7/8	21.875
22	22.000
22 1/8	22.125
22 1/4	22.250
22 3/8	22.375
22 1/2	22.500
22 5/8	22.625
22 3/4	22.750
22 7/8	22.875
23	23.000
23 1/8	23.125
23 1/4	23.250
23 3/8	23.375
23 1/2	23.500
23 5/8	23.625
23 3/4	23.750
23 7/8	23.875
24	24.000
24 1/8	24.125
24 1/4	24.250
24 3/8	24.375
24 1/2	24.500
24 5/8	24.625
24 3/4	24.750
24 7/8	24.875
25	25.000
25 1/8	25.125
25 1/4	25.250
25 3/8	25.375
25 1/2	25.500
25 5/8	25.625
25 3/4	25.750
25 7/8	25.875
26	26.000
26 1/8	26.125
26 1/4	26.250
26 3/8	26.375
26 1/2	26.500
26 5/8	26.625
26 3/4	26.750
26 7/8	26.875
27	27.000
27 1/8	27.125
27 1/4	27.250
27 3/8	27.375
27 1/2	27.500
27 5/8	27.625
27 3/4	27.750
27 7/8	27.875
28	28.000
28 1/8	28.125
28 1/4	28.250
28 3/8	28.375
28 1/2	28.500
28 5/8	28.625
28 3/4	28.750
28 7/8	28.875
29	29.000
29 1/8	29.125
29 1/4	29.250
29 3/8	29.375
29 1/2	29.500
29 5/8	29.625
29 3/4	29.750
29 7/8	29.875
30	30.000
30 1/8	30.125
30 1/4	30.250
30 3/8	30.375
30 1/2	30.500
30 5/8	30.625
30 3/4	30.750
30 7/8	30.875
31	31.000
31 1/8	31.125
31 1/4	31.250
31 3/8	31.375
31 1/2	31.500
31 5/8	31.625
31 3/4	31.750
31 7/8	31.875
32	32.000
32 1/8	32.125
32 1/4	32.250
32 3/8	32.375
32 1/2	32.500
32 5/8	32.625
32 3/4	32.750
32 7/8	32.875
33	33.000
33 1/8	33.125
33 1/4	33.250
33 3/8	33.375
33 1/2	33.500
33 5/8	33.625
33 3/4	33.750
33 7/8	33.875
34	34.000
34 1/8	34.125
34 1/4	34.250
34 3/8	34.375
34 1/2	34.500
34 5/8	34.625
34 3/4	34.750
34 7/8	34.875
35	35.000
35 1/8	35.125
35 1/4	35.250
35 3/8	35.375
35 1/2	35.500
35 5/8	35.625
35 3/4	35.750
35 7/8	35.875
36	36.000
36 1/8	36.125
36 1/4	36.250
36 3/8	36.375
36 1/2	36.500
36 5/8	36.625
36 3/4	36.750
36 7/8	36.875
37	37.000
37 1/8	37.125
37 1/4	37.250
37 3/8	37.375
37 1/2	37.500
37 5/8	37.625
37 3/4	37.750
37 7/8	37.875
38	38.000
38 1/8	38.125
38 1/4	38.250
38 3/8	38.375
38 1/2	38.500
38 5/8	38.625
38 3/4	38.750
38 7/8	38.875
39	39.000
39 1/8	39.125
39 1/4	39.250
39 3/8	39.375
39 1/2	39.500
39 5/8	39.625
39 3/4	39.750
39 7/8	39.875
40	40.000
40 1/8	40.125
40 1/4	40.250
40 3/8	40.375
40 1/2	40.500
40 5/8	40.625
40 3/4	40.750
40 7/8	40.875
41	41.000
41 1/8	41.125
41 1/4	41.250
41 3/8	41.375
41 1/2	41.500
41 5/8	41.625
41 3/4	41.750
41 7/8	41.875
42	42.000
42 1/8	42.125
42 1/4	42.250
42 3/8	42.375
42 1/2	42.500
42 5/8	42.625
42 3/4	42.750
42 7/8	42.875
43	43.000
43 1/8	43.125
43 1/4	43.250
43 3/8	43.375
43 1/2	43.500
43 5/8	43.625
43 3/4	43.750
43 7/8	43.875
44	44.000
44 1/8	44.125
44 1/4	44.250
44 3/8	44.375
44 1/2	44.500
44 5/8	44.625
44 3/4	44.750
44 7/8	44.875
45	45.000
45 1/8	45.125
45 1/4	45.250
45 3/8	45.375
45 1/2	45.500
45 5/8	45.625
45 3/4	45.750
45 7/8	45.875
46	46.000
46 1/8	46.125
46 1/4	46.250
46 3/8	46.375
46 1/2	46.500
46 5/8	46.625
46 3/4	46.750
46 7/8	46.875
47	47.000
47 1/8	47.125
47 1/4	47.250
47 3/8	47.375
47 1/2	47.500
47 5/8	47.625
47 3/4	47.750
47 7/8	47.875
48	48.000
48 1/8	48.125
48	

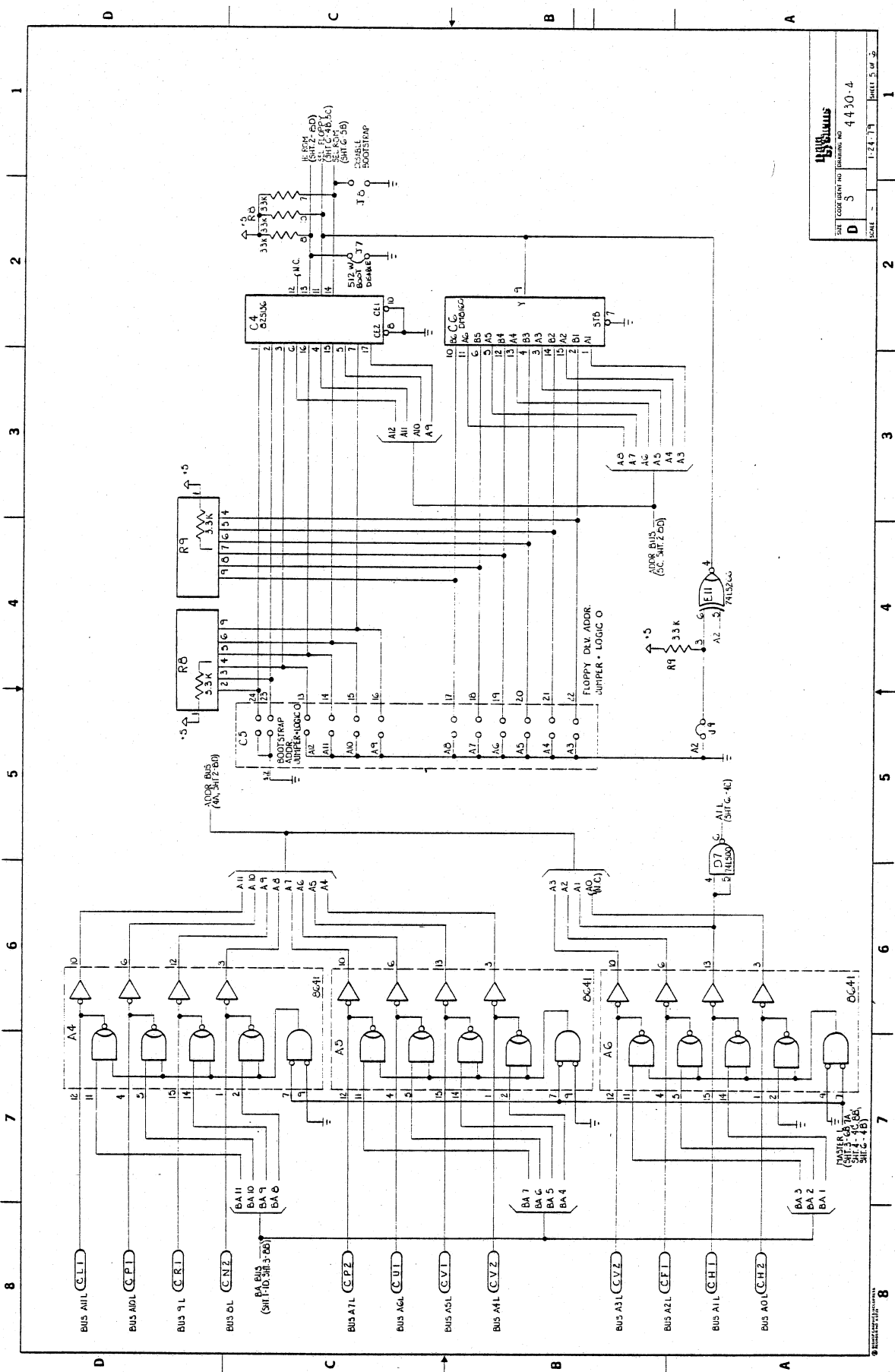






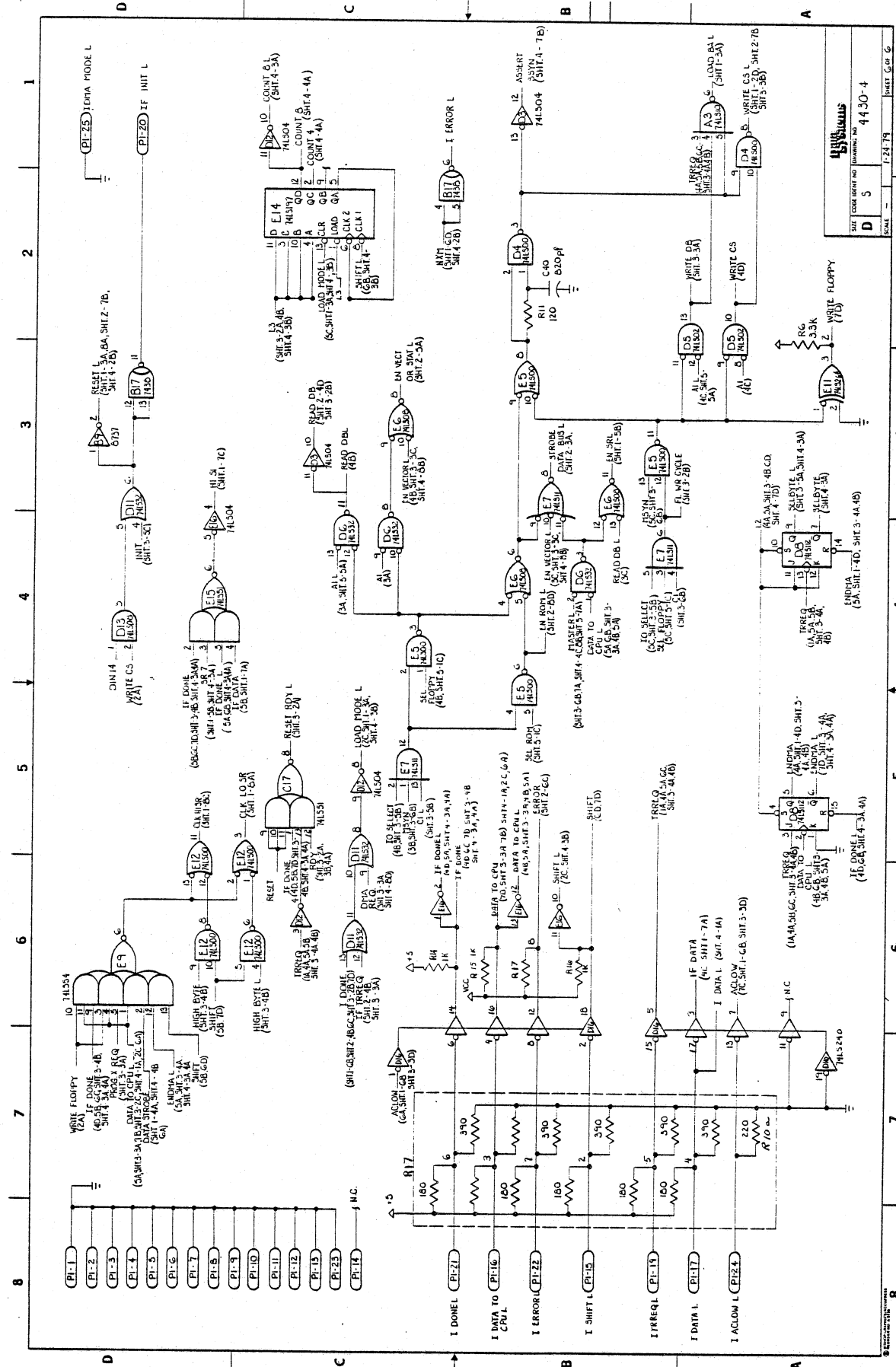
44484848  
 D 3  
 1-24-77  
 4430-4  
 SHEET 3 of 6





SUN COMPANY NO		DRAWING NO	
D		4430-4	
SCALE		SHEET 5 OF 5	
DATE		1-24-79	

BUS A11 (C.L.I.)  
 BUS A10 (C.P.I.)  
 BUS A9 (C.R.I.)  
 BUS A8 (C.N.Z.)  
 BUS A7 (C.P.Z.)  
 BUS A6 (C.U.T.)  
 BUS A5 (C.V.T.)  
 BUS A4 (C.V.Z.)  
 BUS A3 (C.V.Z.)  
 BUS A2 (C.F.T.)  
 BUS A1 (C.H.T.)  
 BUS A0 (C.H.Z.)



REV	5	DATE	1.24.79	INSTR	Gr 6
DOC	4430-4				

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

4430-4

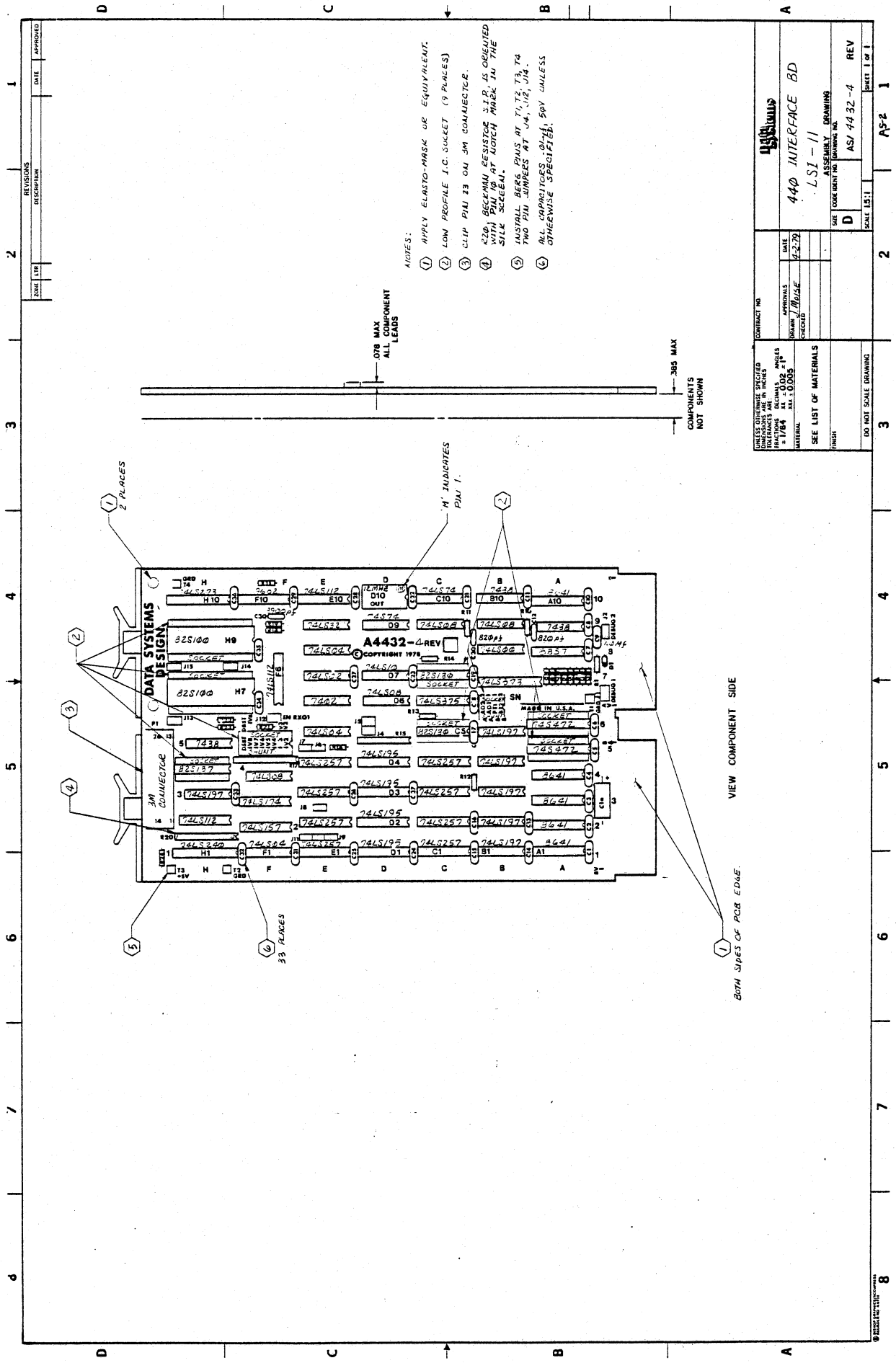
4430-4

4430-4

4430-4

4430-4

4430-4

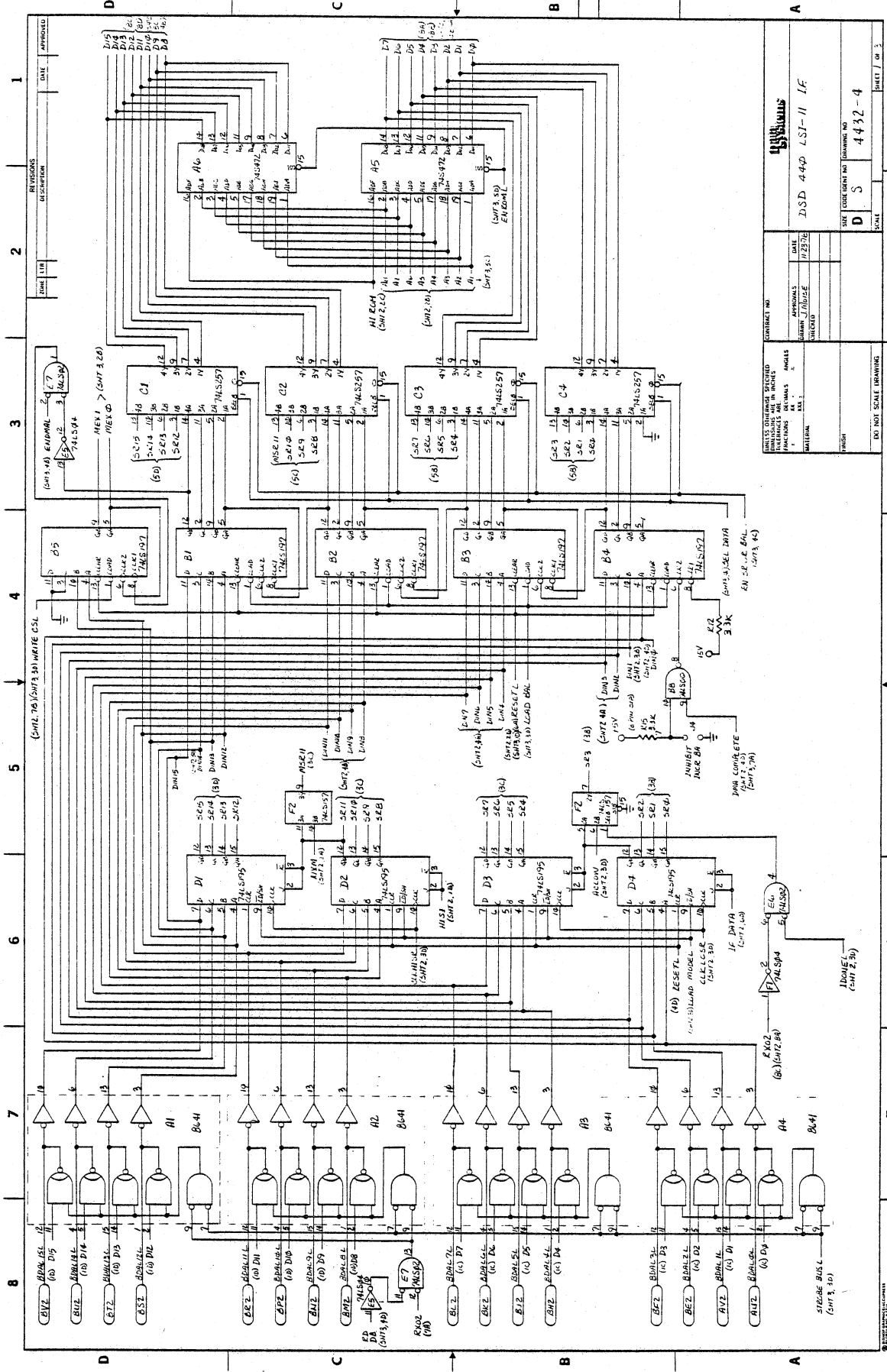


- NOTES:
- 1 APPLY ELASTO-MARK OR EQUIVALENT.
  - 2 LOW PROFILE I.C. SOCKET (9 PLACES)
  - 3 CLIP PIN 23 ON 3M CONNECTOR.
  - 4 420. BECKMAN RESISTOR 3.1P. IS OBTAINED WITH PIN 19 AT NOTCH MARK IN THE SILK SCREEN.
  - 5 INSTALL BERS. PINS AT T<sub>1</sub>, T<sub>2</sub>, T<sub>3</sub>, T<sub>4</sub> TWO PII JUMBERS AT U<sub>4</sub>, U<sub>5</sub>, U<sub>6</sub>.
  - 6 ALL CAPACITORS 0.01μF, 50V UNLESS OTHERWISE SPECIFIED.

<b>DATE</b> 2-2-79		<b>APPROVALS</b> DRAWN: J. ROUSE CHECKED:	
<b>CONTRACT NO.</b> 440 INTERFACE BOARD		<b>ASSEMBLY DRAWING</b> ASI 4432-4	
<b>SCALE</b> 1:1		<b>SHEET</b> 1 OF 1	
<b>UNLESS OTHERWISE SPECIFIED</b> DIMENSIONS ARE IN INCHES DECIMALS ARE TO 0.02" ± 0.005 FRACTIONS ARE TO 1/16" ± 0.005 MATERIAL: SEE LIST OF MATERIALS		<b>DO NOT SCALE DRAWING</b>	

VIEW COMPONENT SIDE

BOTH SIDES OF PCB EDGE



REVISIONS		DATE	APPROVED
1			
2			
3			
4			
5			
6			
7			
8			

DATE	LTR	DESCRIPTION
1/15/77	2, 1, 7	(SMT 3, 20) ENHANCEMENT
1/15/77	3, 4, 7, 8	(SMT 2, 0) ENHANCEMENT
1/15/77	5, 6, 7, 8	(SMT 3, 30) ENHANCEMENT

PARTS LIST		QUANTITY	DESCRIPTION
74LS04	1	1	HEX INVERTER
74LS10	4	4	3-INPUT NAND
74LS16	4	4	HEX INVERTER
74LS20	4	4	4-INPUT NAND
74LS27	4	4	7-SEGMENT DRIVER
74LS30	4	4	10-INPUT NAND
74LS54	4	4	DECADECOUNTER
74LS55	4	4	MONOSTABLE MULTIVIBRATOR
74LS57	4	4	MONOSTABLE MULTIVIBRATOR
74LS92	4	4	DIVIDER
74LS93	4	4	DIVIDER
74LS95	4	4	DIVIDER
74LS97	4	4	DIVIDER
74LS99	4	4	DIVIDER
74LS101	4	4	DIVIDER
74LS102	4	4	DIVIDER
74LS103	4	4	DIVIDER
74LS104	4	4	DIVIDER
74LS105	4	4	DIVIDER
74LS106	4	4	DIVIDER
74LS107	4	4	DIVIDER
74LS108	4	4	DIVIDER
74LS109	4	4	DIVIDER
74LS110	4	4	DIVIDER
74LS111	4	4	DIVIDER
74LS112	4	4	DIVIDER
74LS113	4	4	DIVIDER
74LS114	4	4	DIVIDER
74LS115	4	4	DIVIDER
74LS116	4	4	DIVIDER
74LS117	4	4	DIVIDER
74LS118	4	4	DIVIDER
74LS119	4	4	DIVIDER
74LS120	4	4	DIVIDER
74LS121	4	4	DIVIDER
74LS122	4	4	DIVIDER
74LS123	4	4	DIVIDER
74LS124	4	4	DIVIDER
74LS125	4	4	DIVIDER
74LS126	4	4	DIVIDER
74LS127	4	4	DIVIDER
74LS128	4	4	DIVIDER
74LS129	4	4	DIVIDER
74LS130	4	4	DIVIDER
74LS131	4	4	DIVIDER
74LS132	4	4	DIVIDER
74LS133	4	4	DIVIDER
74LS134	4	4	DIVIDER
74LS135	4	4	DIVIDER
74LS136	4	4	DIVIDER
74LS137	4	4	DIVIDER
74LS138	4	4	DIVIDER
74LS139	4	4	DIVIDER
74LS14	4	4	HEX INVERTER
74LS15	4	4	HEX INVERTER
74LS16	4	4	HEX INVERTER
74LS17	4	4	HEX INVERTER
74LS18	4	4	HEX INVERTER
74LS19	4	4	HEX INVERTER
74LS20	4	4	4-INPUT NAND
74LS21	4	4	4-INPUT NAND
74LS22	4	4	4-INPUT NAND
74LS23	4	4	4-INPUT NAND
74LS24	4	4	4-INPUT NAND
74LS25	4	4	4-INPUT NAND
74LS26	4	4	4-INPUT NAND
74LS27	4	4	7-SEGMENT DRIVER
74LS28	4	4	7-SEGMENT DRIVER
74LS29	4	4	7-SEGMENT DRIVER
74LS30	4	4	10-INPUT NAND
74LS31	4	4	10-INPUT NAND
74LS32	4	4	10-INPUT NAND
74LS33	4	4	10-INPUT NAND
74LS34	4	4	10-INPUT NAND
74LS35	4	4	10-INPUT NAND
74LS36	4	4	10-INPUT NAND
74LS37	4	4	10-INPUT NAND
74LS38	4	4	10-INPUT NAND
74LS39	4	4	10-INPUT NAND
74LS40	4	4	HEX DECADECOUNTER
74LS41	4	4	HEX DECADECOUNTER
74LS42	4	4	HEX DECADECOUNTER
74LS43	4	4	HEX DECADECOUNTER
74LS44	4	4	HEX DECADECOUNTER
74LS45	4	4	HEX DECADECOUNTER
74LS46	4	4	HEX DECADECOUNTER
74LS47	4	4	HEX DECADECOUNTER
74LS48	4	4	HEX DECADECOUNTER
74LS49	4	4	HEX DECADECOUNTER
74LS50	4	4	HEX DECADECOUNTER
74LS51	4	4	HEX DECADECOUNTER
74LS52	4	4	HEX DECADECOUNTER
74LS53	4	4	HEX DECADECOUNTER
74LS54	4	4	DECADECOUNTER
74LS55	4	4	MONOSTABLE MULTIVIBRATOR
74LS56	4	4	MONOSTABLE MULTIVIBRATOR
74LS57	4	4	MONOSTABLE MULTIVIBRATOR
74LS58	4	4	MONOSTABLE MULTIVIBRATOR
74LS59	4	4	MONOSTABLE MULTIVIBRATOR
74LS60	4	4	DIVIDER
74LS61	4	4	DIVIDER
74LS62	4	4	DIVIDER
74LS63	4	4	DIVIDER
74LS64	4	4	DIVIDER
74LS65	4	4	DIVIDER
74LS66	4	4	DIVIDER
74LS67	4	4	DIVIDER
74LS68	4	4	DIVIDER
74LS69	4	4	DIVIDER
74LS70	4	4	DIVIDER
74LS71	4	4	DIVIDER
74LS72	4	4	DIVIDER
74LS73	4	4	DIVIDER
74LS74	4	4	DIVIDER
74LS75	4	4	DIVIDER
74LS76	4	4	DIVIDER
74LS77	4	4	DIVIDER
74LS78	4	4	DIVIDER
74LS79	4	4	DIVIDER
74LS80	4	4	DIVIDER
74LS81	4	4	DIVIDER
74LS82	4	4	DIVIDER
74LS83	4	4	DIVIDER
74LS84	4	4	DIVIDER
74LS85	4	4	DIVIDER
74LS86	4	4	DIVIDER
74LS87	4	4	DIVIDER
74LS88	4	4	DIVIDER
74LS89	4	4	DIVIDER
74LS90	4	4	DIVIDER
74LS91	4	4	DIVIDER
74LS92	4	4	DIVIDER
74LS93	4	4	DIVIDER
74LS94	4	4	DIVIDER
74LS95	4	4	DIVIDER
74LS96	4	4	DIVIDER
74LS97	4	4	DIVIDER
74LS98	4	4	DIVIDER
74LS99	4	4	DIVIDER

DATE	DESCRIPTION
1/15/77	(SMT 3, 20) ENHANCEMENT
1/15/77	(SMT 2, 0) ENHANCEMENT
1/15/77	(SMT 3, 30) ENHANCEMENT

DATE	DESCRIPTION
1/15/77	(SMT 3, 20) ENHANCEMENT
1/15/77	(SMT 2, 0) ENHANCEMENT
1/15/77	(SMT 3, 30) ENHANCEMENT

DATE	DESCRIPTION
1/15/77	(SMT 3, 20) ENHANCEMENT
1/15/77	(SMT 2, 0) ENHANCEMENT
1/15/77	(SMT 3, 30) ENHANCEMENT

DATE	DESCRIPTION
1/15/77	(SMT 3, 20) ENHANCEMENT
1/15/77	(SMT 2, 0) ENHANCEMENT
1/15/77	(SMT 3, 30) ENHANCEMENT

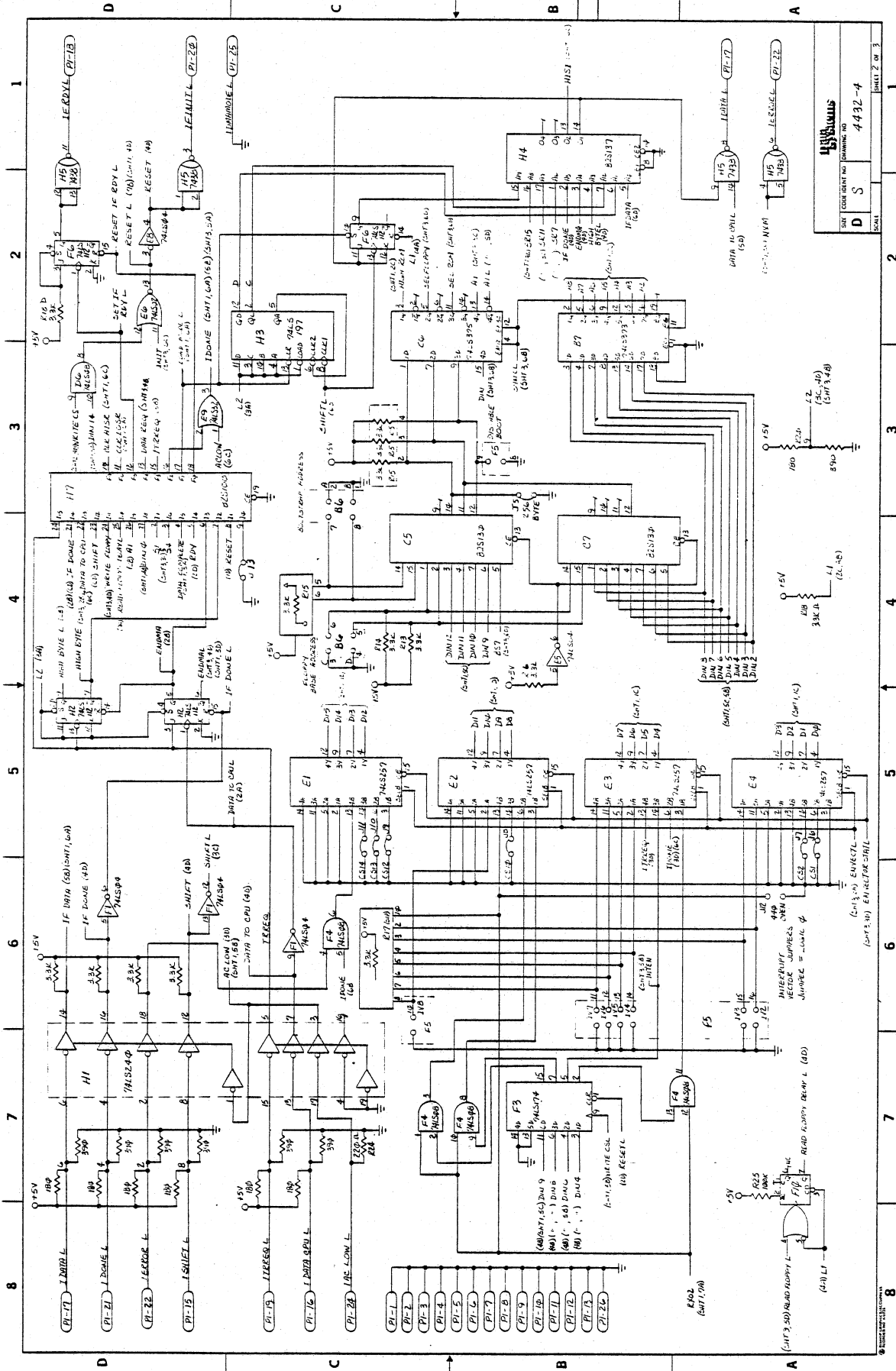
DATE	DESCRIPTION
1/15/77	(SMT 3, 20) ENHANCEMENT
1/15/77	(SMT 2, 0) ENHANCEMENT
1/15/77	(SMT 3, 30) ENHANCEMENT

CONTRACT NO.	DATE	REVISIONS
4432-4	1/15/77	1
4432-4	1/15/77	2
4432-4	1/15/77	3
4432-4	1/15/77	4
4432-4	1/15/77	5
4432-4	1/15/77	6
4432-4	1/15/77	7
4432-4	1/15/77	8

CONTRACT NO.	DATE	REVISIONS
4432-4	1/15/77	1
4432-4	1/15/77	2
4432-4	1/15/77	3
4432-4	1/15/77	4
4432-4	1/15/77	5
4432-4	1/15/77	6
4432-4	1/15/77	7
4432-4	1/15/77	8

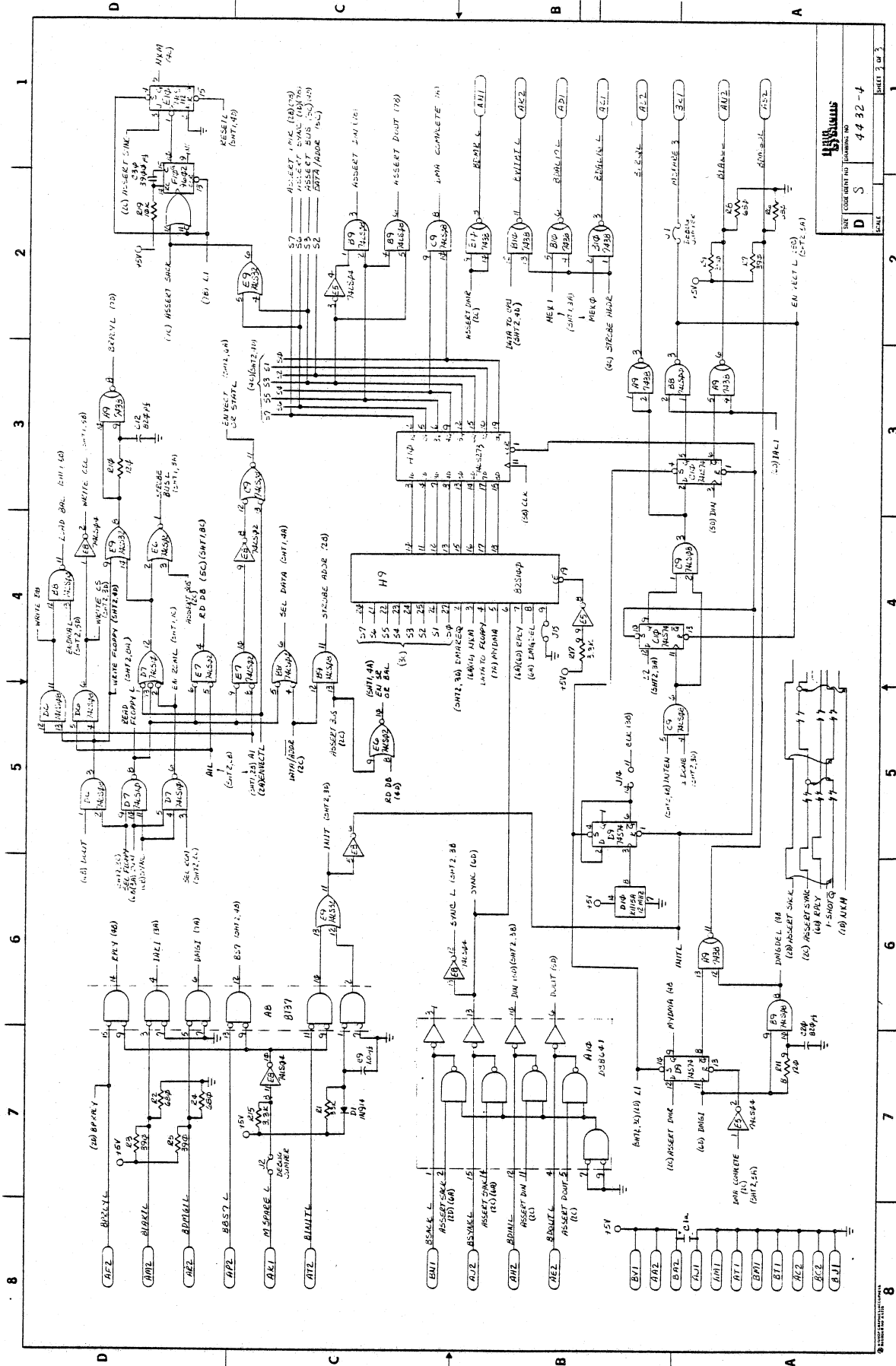
CONTRACT NO.	DATE	REVISIONS
4432-4	1/15/77	1
4432-4	1/15/77	2
4432-4	1/15/77	3
4432-4	1/15/77	4
4432-4	1/15/77	5
4432-4	1/15/77	6
4432-4	1/15/77	7
4432-4	1/15/77	8

CONTRACT NO.	DATE	REVISIONS
4432-4	1/15/77	1
4432-4	1/15/77	2
4432-4	1/15/77	3
4432-4	1/15/77	4
4432-4	1/15/77	5
4432-4	1/15/77	6
4432-4	1/15/77	7
4432-4	1/15/77	8



CONTROL UNIT (CONTINUED)  
 DRAWING NO. 4432-4  
 SCALE 1:1  
 SHEET 2 OF 3

HYPERON

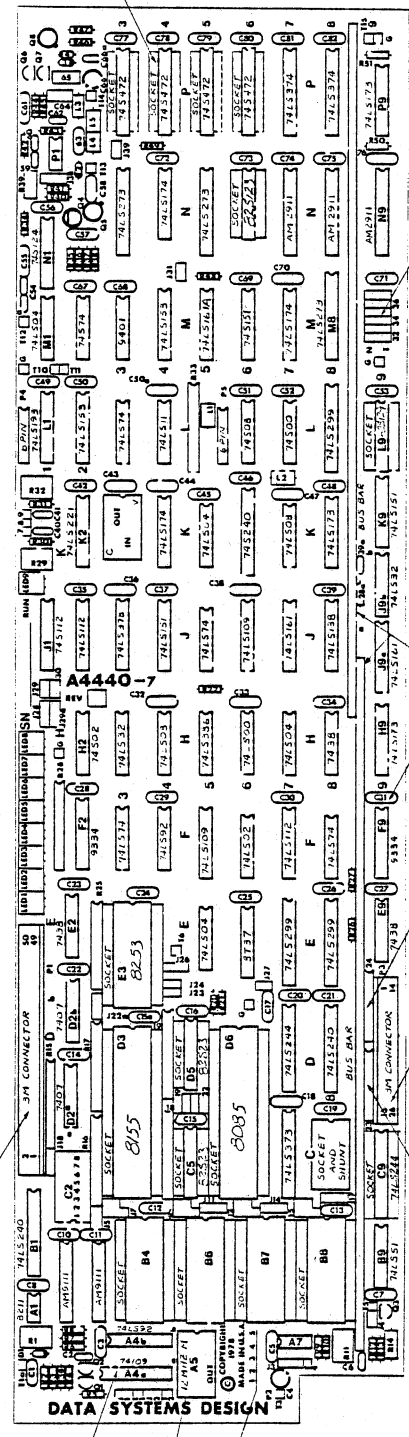


**Raytheon**  
 SMT CONTROL UNIT DRAWING NO  
**D S 4432-1**  
 SCALE



1 2 3 4 5 6 7 8

- ① APPLY ELASTO-MASK OR EQUIVALENT.
- ② LOW PROFILE I.C. SOCKET (16 PLACES)
- ③ REMOVE PIN 23 FROM THE 50 PIN 3M CONNECTOR.
- ④ R23, R24 BECKMAN RESISTOR S.I.P. IS ORIENTED WITH PIN 8 AT NOTCH MARK IN THE SILK SCREEN. (ALL RESISTOR S.I.P.'S HAVE A DOT OR A GOOVE ON PACKAGE TO INDICATE PIN 1)
- 5.
- ⑥ ADJOINING ROWS OF BERG WELD POSTS HAVE TO BE ASSEMBLED WITH MATING TRACKS.
- ⑦ TRIM I.C. LEADS TO P.C.B. AFTER FLOW SOLDER AT A46, A41.
- ⑧ HAND SOLDER OSCILLATOR AND BUS BARS AFTER BOARDS HAVE BEEN WASHED AND DRIED.

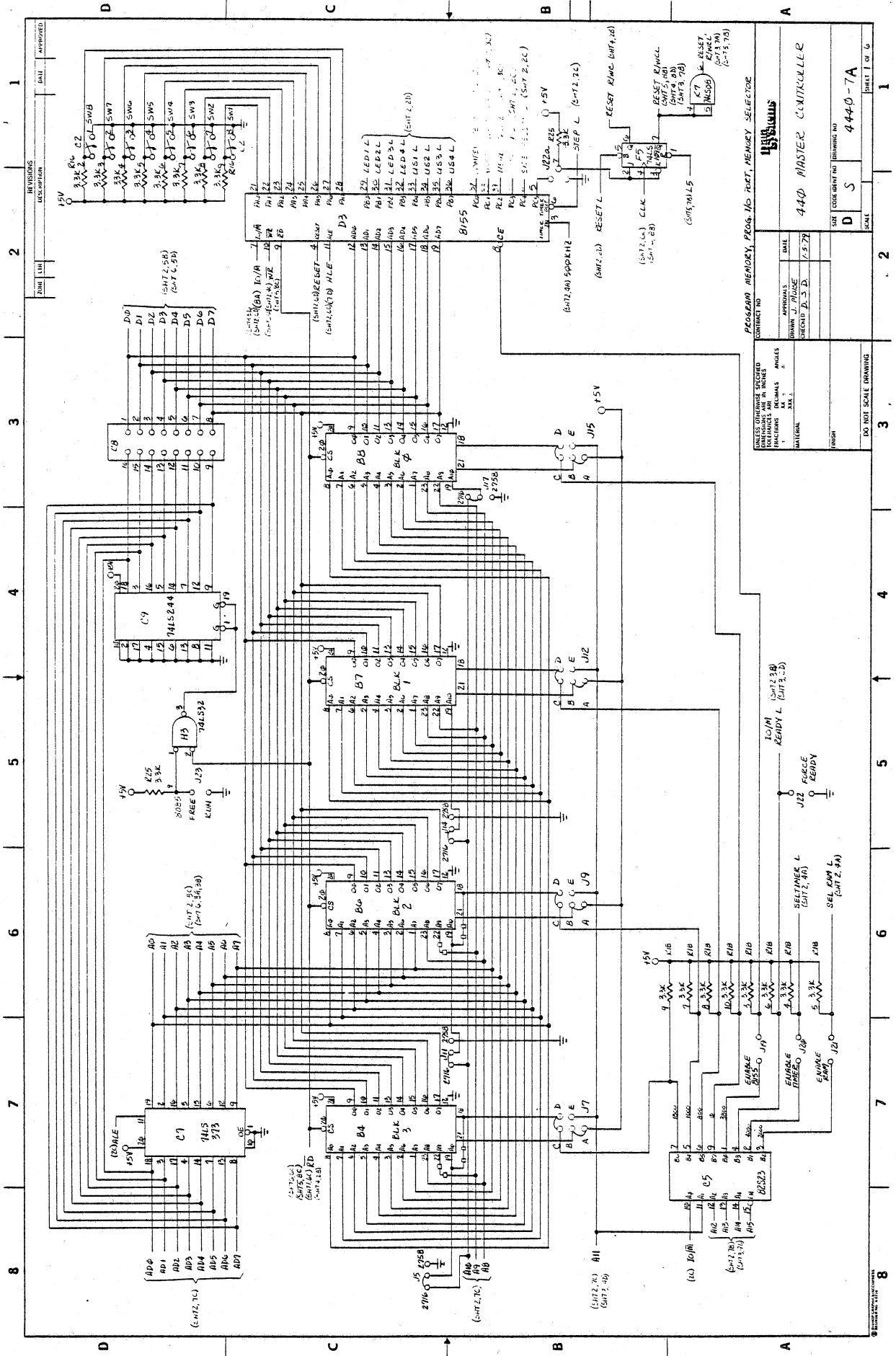


CONTRACT NO.		DATE		APPROVAL	
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		DATE		APPROVAL	
DIMENSIONS IN PARENTHESES ARE IN MILLIMETERS		DATE		APPROVAL	
MATERIAL		DATE		APPROVAL	
SEE LIST OF MATERIALS		DATE		APPROVAL	
FINISH		DATE		APPROVAL	
DO NOT SCALE DRAWING		DATE		APPROVAL	
SCALE 15:1		DATE		APPROVAL	
AS-1		DATE		APPROVAL	
SHEET 1 OF 1		DATE		APPROVAL	

440 MASTER CONTROLLER BOARD

AS/440-7 REVA

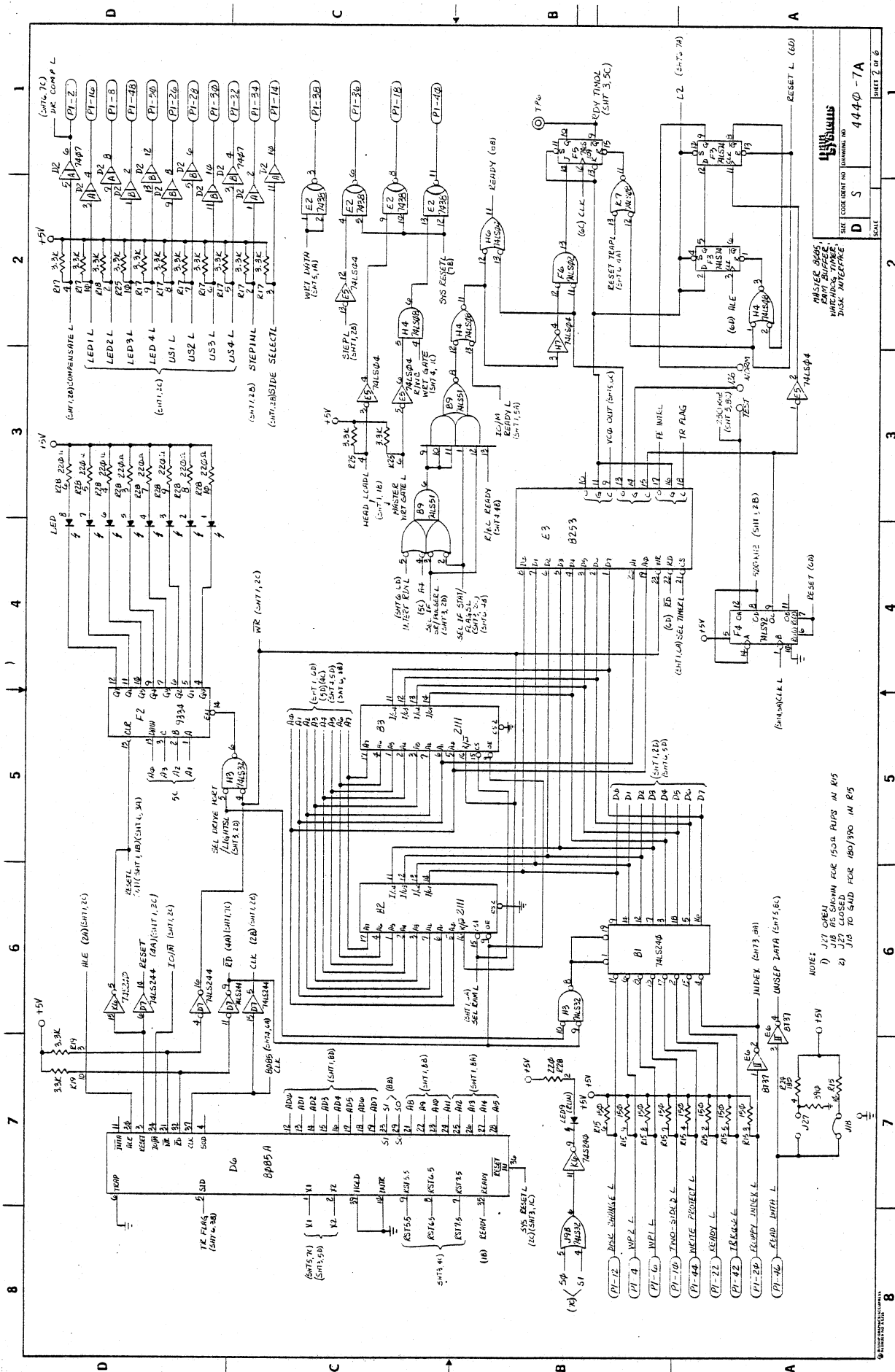
ASSEMBLY DRAWING



REV	DATE	DESCRIPTION	APPROVED
1			
2			
3			
4			
5			
6			
7			
8			

**4400 MASTER CONTROLLER**  
 CONTRACT NO. \_\_\_\_\_  
 PROGRAM MEMORY, PROG. I/O PROT. MEMORY SELECTOR  
 APPROVALS:  
 DESIGNER: J. J. MOSE  
 CHECKED: D. S. D.  
 DATE: 1/3/78  
 DRAWING NO.: 4440-7A  
 SCALE: D S  
 SHEET 1 OF 6

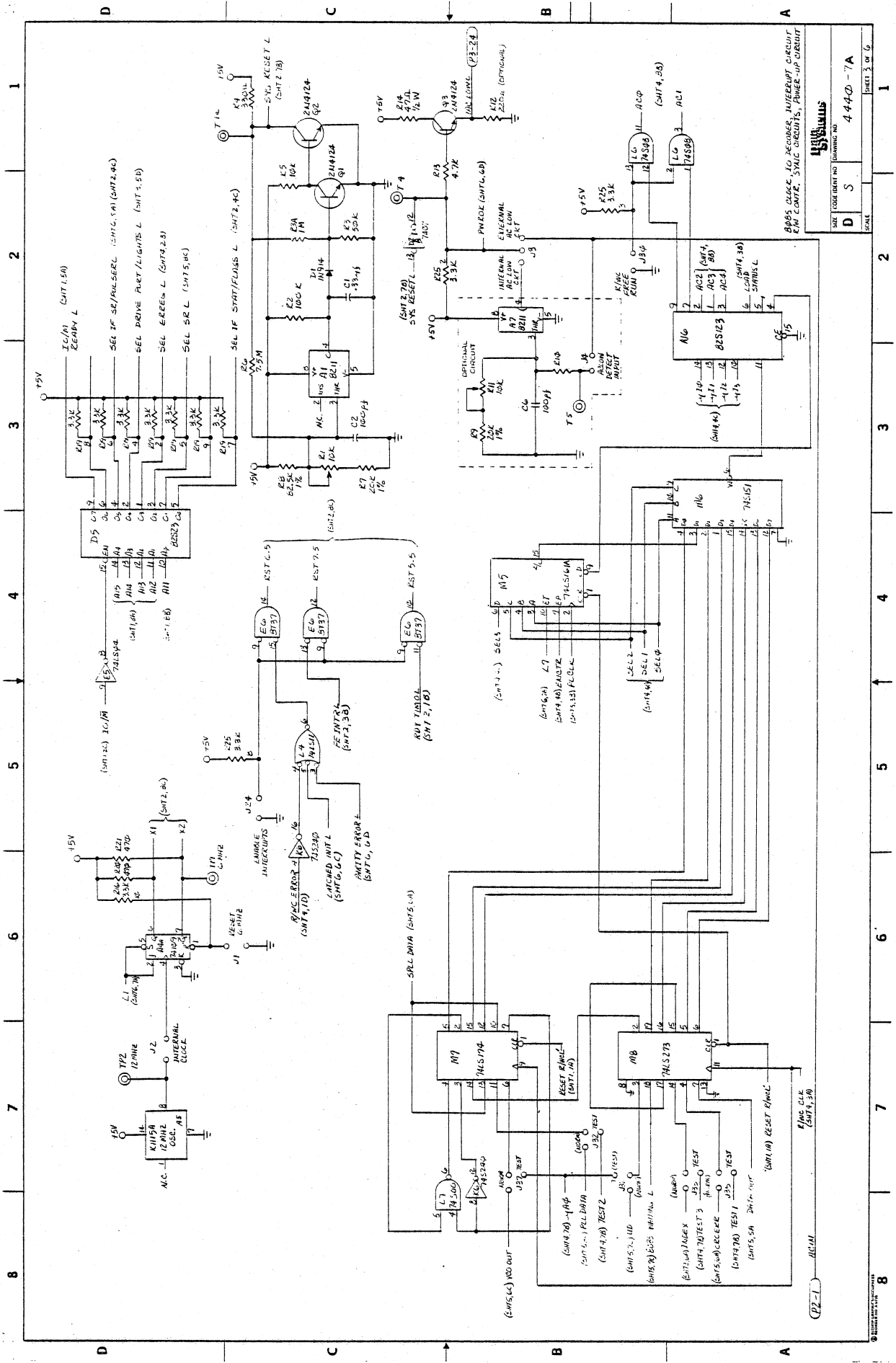
UNLESS OTHERWISE SPECIFIED  
 DIMENSIONS ARE IN INCHES  
 DECIMALS ARE TO BE USED  
 MATERIAL: \_\_\_\_\_  
 FINISH: \_\_\_\_\_  
 DO NOT SCALE DRAWING



PROJECT NO. 4440-7A  
 SHEET CODE IDENT NO. 4440-7A  
 DISK INTERFACE  
 SCALE  
 SHEET 2 OF 6

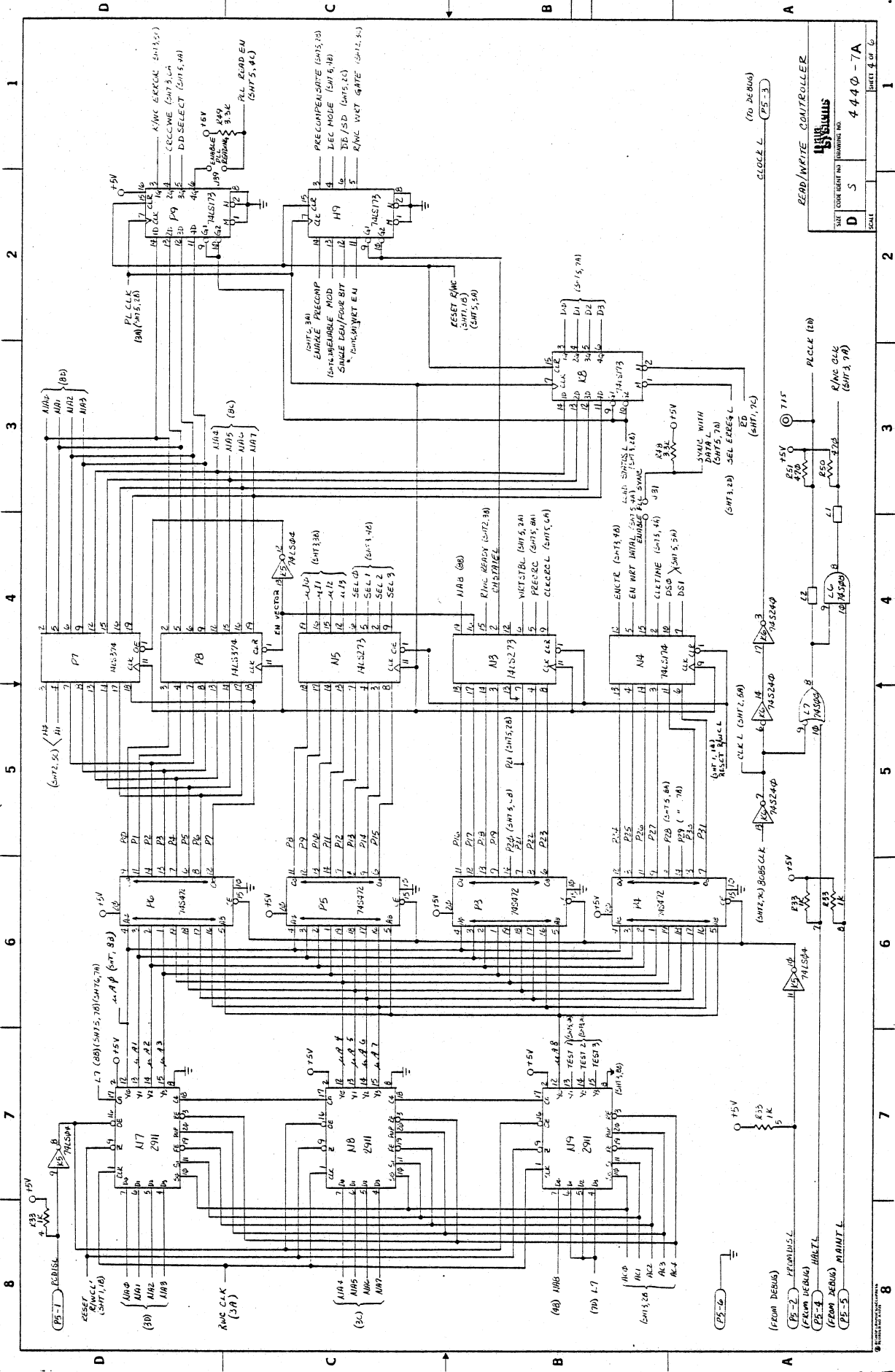
NOTE:  
 1) J27 OPEN FOR 1504 AIRS IN R15  
 2) J27 CLOSED FOR 1601/190 AIRS IN R15

J27 OPEN FOR 1504 AIRS IN R15  
 J27 CLOSED FOR 1601/190 AIRS IN R15



8085 CLOCK LOG. RECORDER, INTERRUPT CIRCUIT  
 2.4M. CONTROL, SYNC CIRCUITS, POWER-UP CIRCUIT

USE FORM 100 (REV. 10-67) (INCLUDING NO.)  
 D S 4440-7A  
 SCALE SHEET 3 OF 6



**4440-7A**  
 DATE: 00000000  
 DRAWING NO: 4440-7A  
 SCALE: 1:1  
 SHEET 4 OF 6

**READ/WRITE CONTROLLER**  
 PS-3  
 CLOCK L (TO DEBUG)

PS-1  
 RESET  
 PS-2  
 R/W CLK  
 PS-3  
 R/W CLK  
 PS-5  
 R/W CLK

PS-10  
 FROM DEBUG  
 PS-2  
 FROM DEBUG  
 PS-3  
 FROM DEBUG  
 PS-5  
 FROM DEBUG

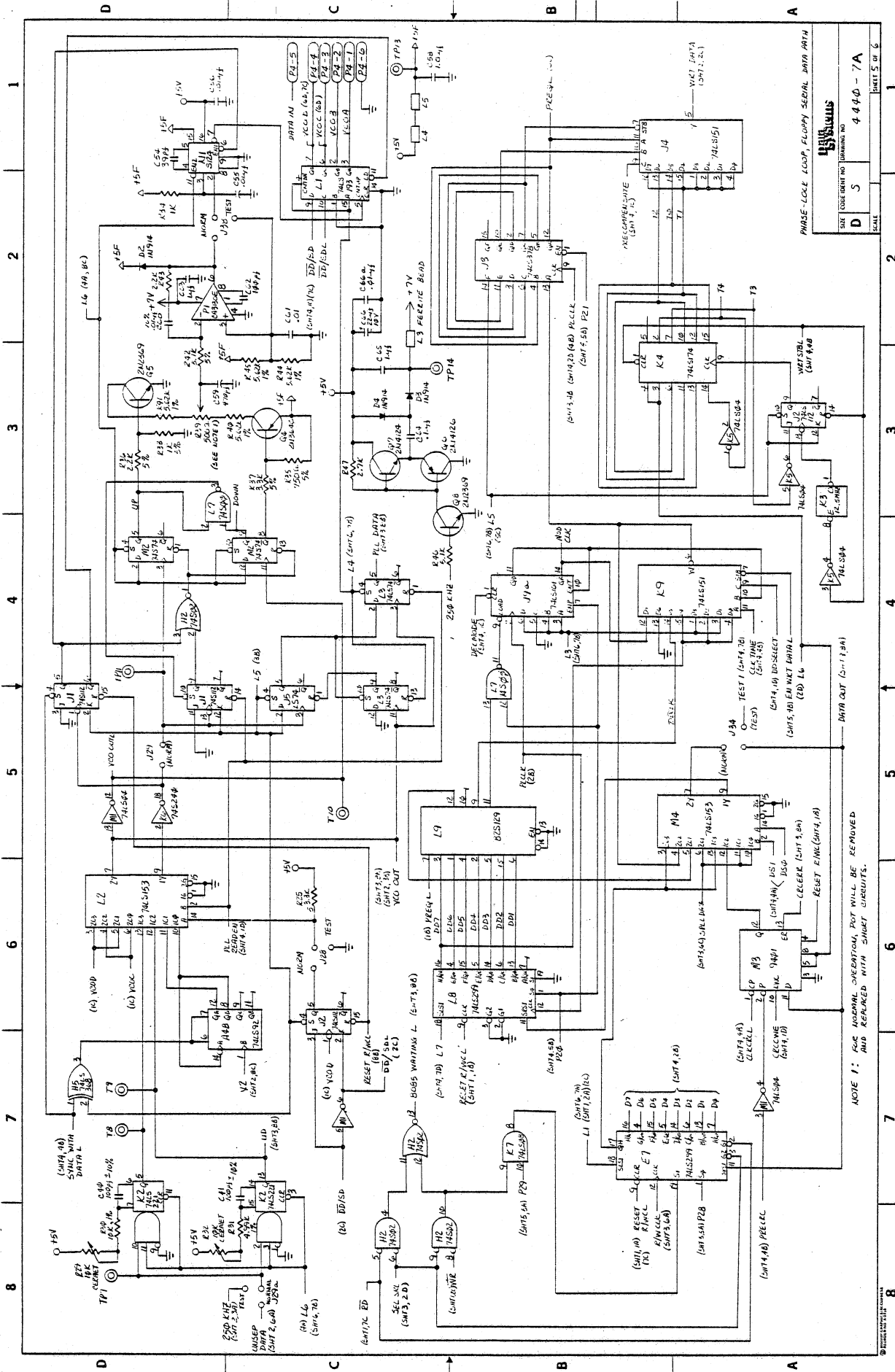
PS-1  
 RESET  
 PS-2  
 R/W CLK  
 PS-3  
 R/W CLK  
 PS-5  
 R/W CLK

PS-10  
 FROM DEBUG  
 PS-2  
 FROM DEBUG  
 PS-3  
 FROM DEBUG  
 PS-5  
 FROM DEBUG

PS-1  
 RESET  
 PS-2  
 R/W CLK  
 PS-3  
 R/W CLK  
 PS-5  
 R/W CLK

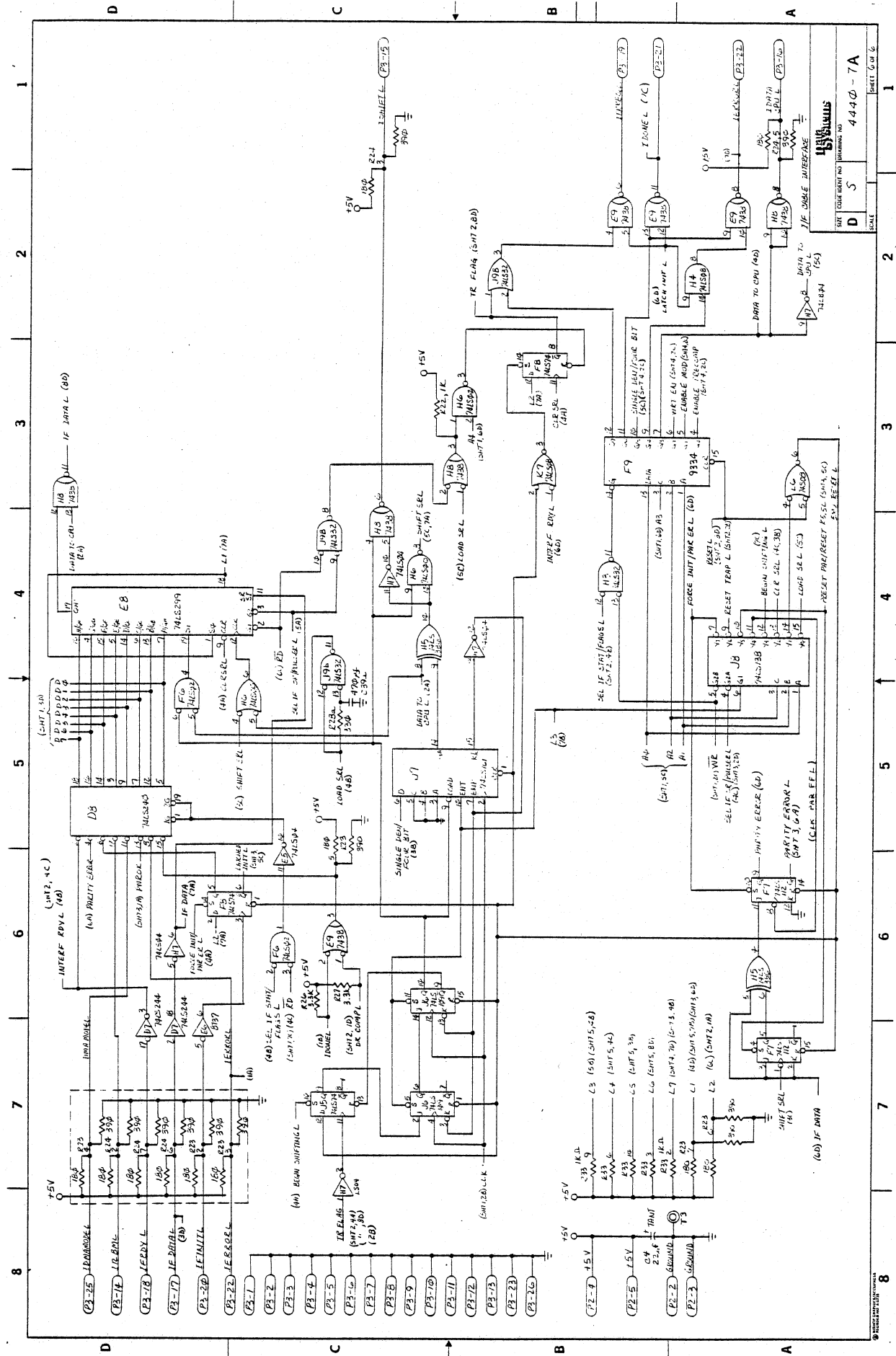
PS-10  
 FROM DEBUG  
 PS-2  
 FROM DEBUG  
 PS-3  
 FROM DEBUG  
 PS-5  
 FROM DEBUG

PS-1  
 RESET  
 PS-2  
 R/W CLK  
 PS-3  
 R/W CLK  
 PS-5  
 R/W CLK



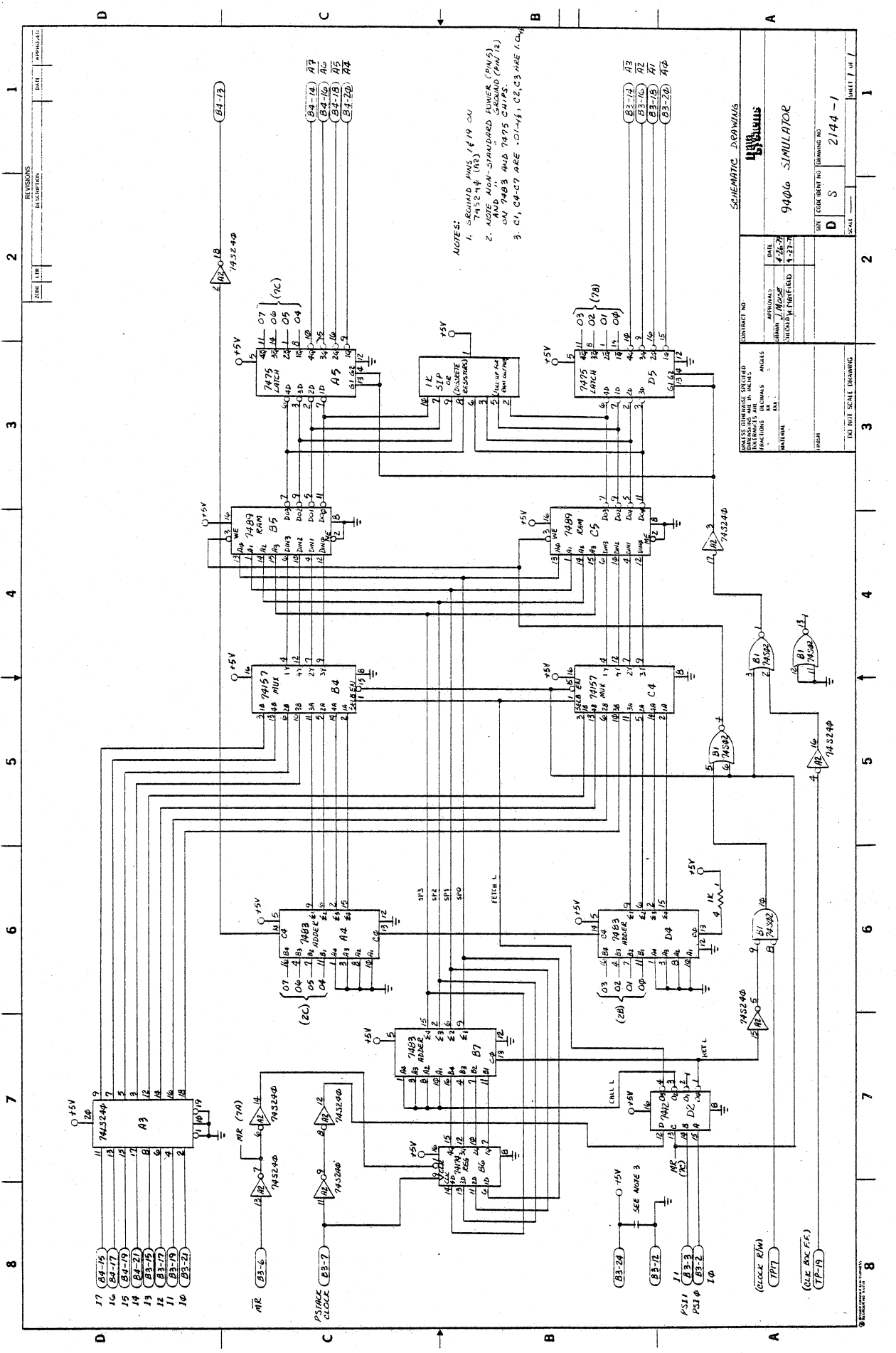
PHASE-LOCK LOOP, FLOPPY SERIAL DATA PATH  
 SHEET NO. DRAWING NO. 4440-7A  
 SCALE S

NOTE 1: FOR NORMAL OPERATION, POT WILL BE REMOVED AND REPAIRED WITH SHUNT CIRCUITS.



4440-7A  
 4440-7A  
 4440-7A  
 4440-7A

4440-7A  
 4440-7A  
 4440-7A



- NOTES:
1. SECOND WMS 1/19 CW
  2. 74254 AND 74255 SIMILAR TO 74254 (P14.5) ON 7483 AND 7475 CHIPS.
  3. C1, C4-C7 ARE .01-UF, C2, C3 ARE 1.0-UF.

CONTRACT NO. **9406**

DATE: **4-26-78**

APPROVED: **J. MOSE**

DESIGNED BY: **J. MOSE**

CHECKED BY: **J. MOSE**

SCALE: **D S**

FIGURE NO: **2144-1**

DO NOT SCALE DRAWING

UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN INCHES.

CONTRACT NO. **9406**

DATE: **4-26-78**

APPROVED: **J. MOSE**

DESIGNED BY: **J. MOSE**

CHECKED BY: **J. MOSE**

SCALE: **D S**

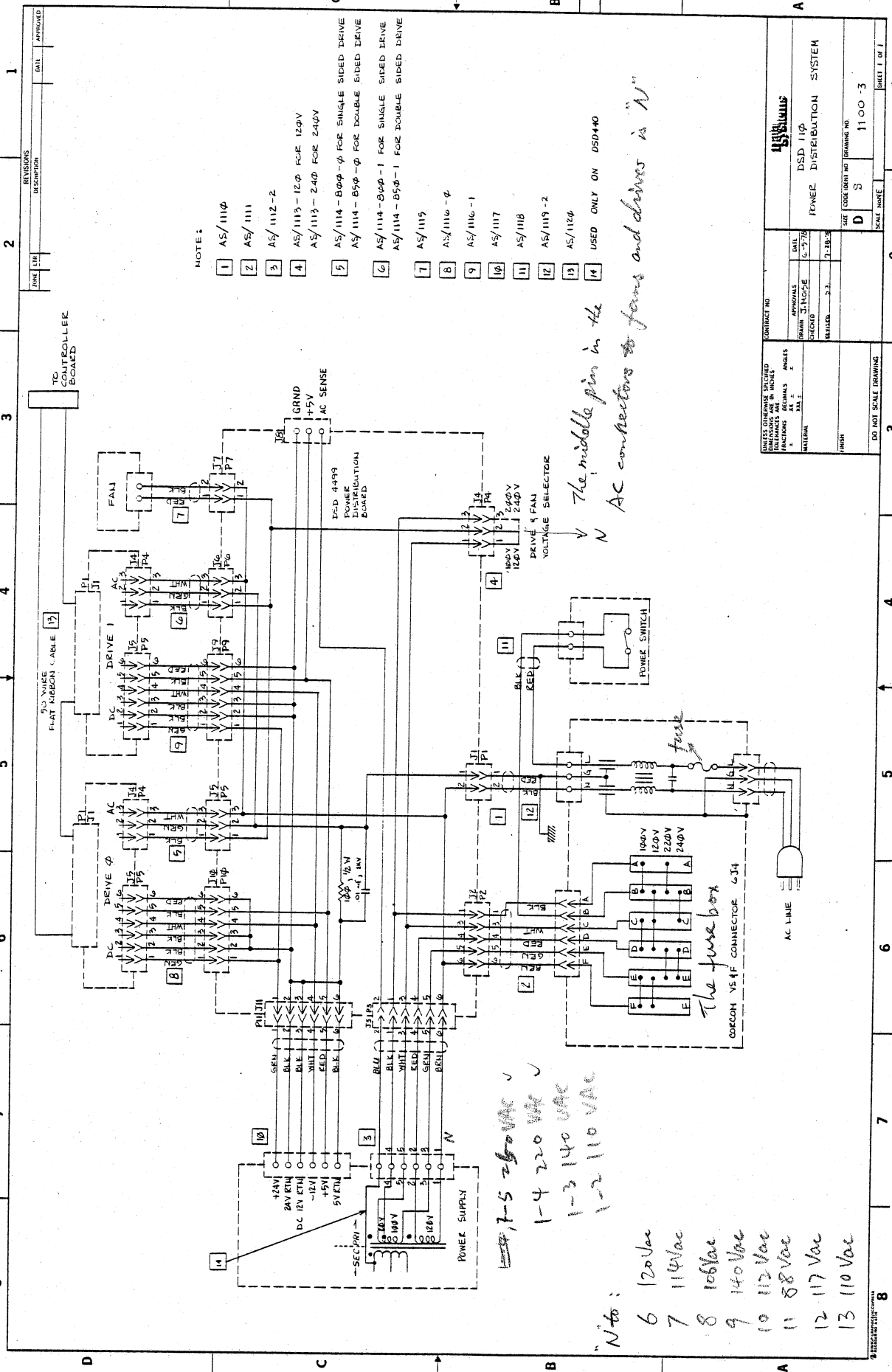
FIGURE NO: **2144-1**

DO NOT SCALE DRAWING

REVISIONS

NO.	DATE	DESCRIPTION
1		
2		
3		
4		
5		
6		
7		
8		





NOTE:

- 1 AS/1110
- 2 AS/1111
- 3 AS/1112-2
- 4 AS/1113-120V FOR 120V
- 5 AS/1113-240V FOR 240V
- 6 AS/1114-000-0 FOR SINGLE SIDED DRIVE
- 7 AS/1114-000-1 FOR DOUBLE SIDED DRIVE
- 8 AS/1115
- 9 AS/1116-0
- 10 AS/1117
- 11 AS/1118
- 12 AS/1119-2
- 13 AS/1120
- 14 USED ONLY ON DSD440

The middle pin in the AC connectors to fans and drives is "N"

7-5 260VAC  
 1-4 220VAC  
 1-3 140VAC  
 1-2 110VAC

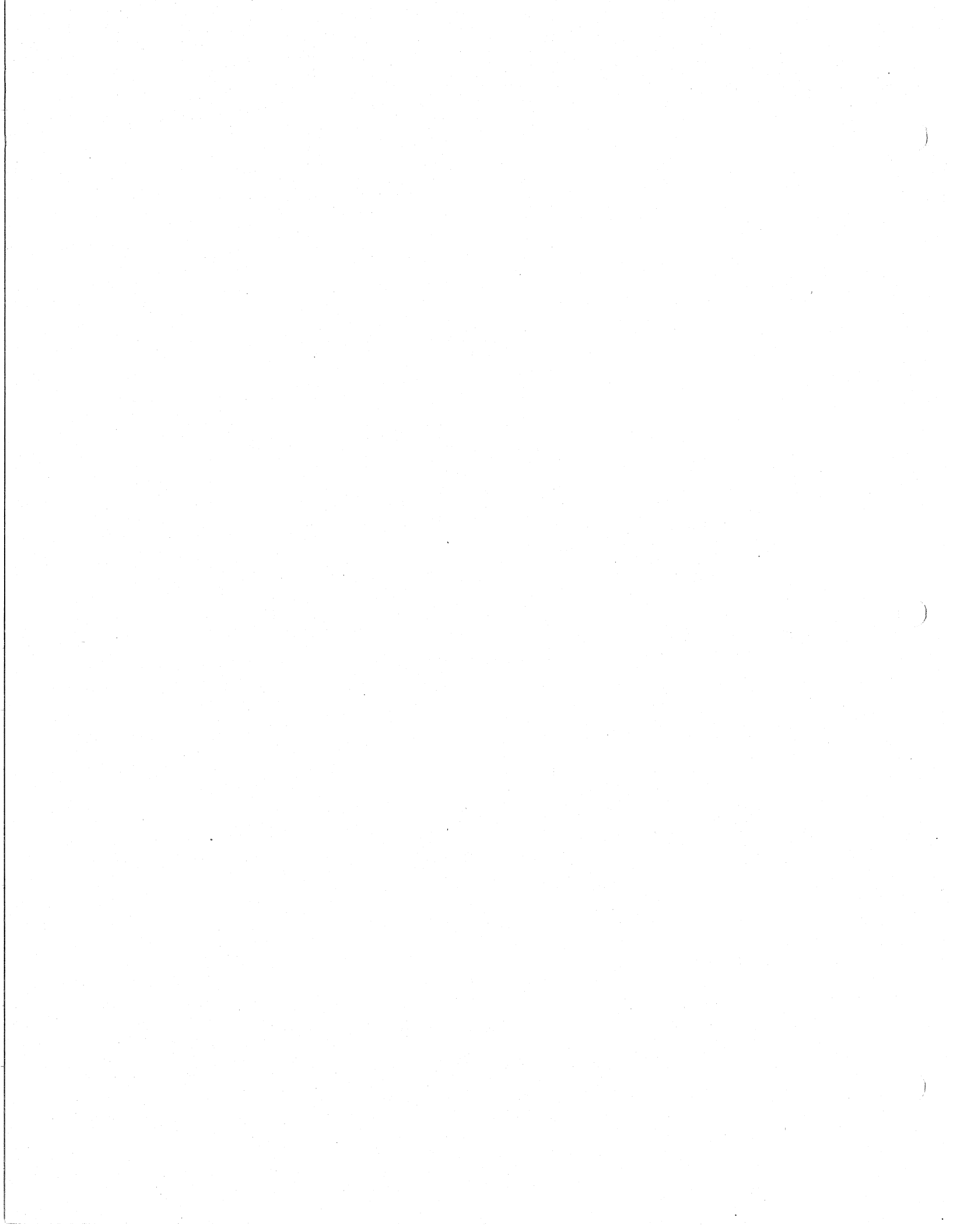
Note:  
 6 120VAC  
 7 110VAC  
 8 106VAC  
 9 140VAC  
 10 113VAC  
 11 88VAC  
 12 117VAC  
 13 110VAC

CONTRACT NO.		APPROVALS	
DSD 110		DATE	INITIALS
POWER DISTRIBUTION SYSTEM		CHECKED	DATE
D S 1100-3		DESIGNED	DATE
SCALE NAME		SCALE	
DO NOT SCALE DRAWING		SHEET 1 OF 1	

REV.	DATE	DESCRIPTION
1		CONTROL BOARD

1 2 3 4 5 6 7 8

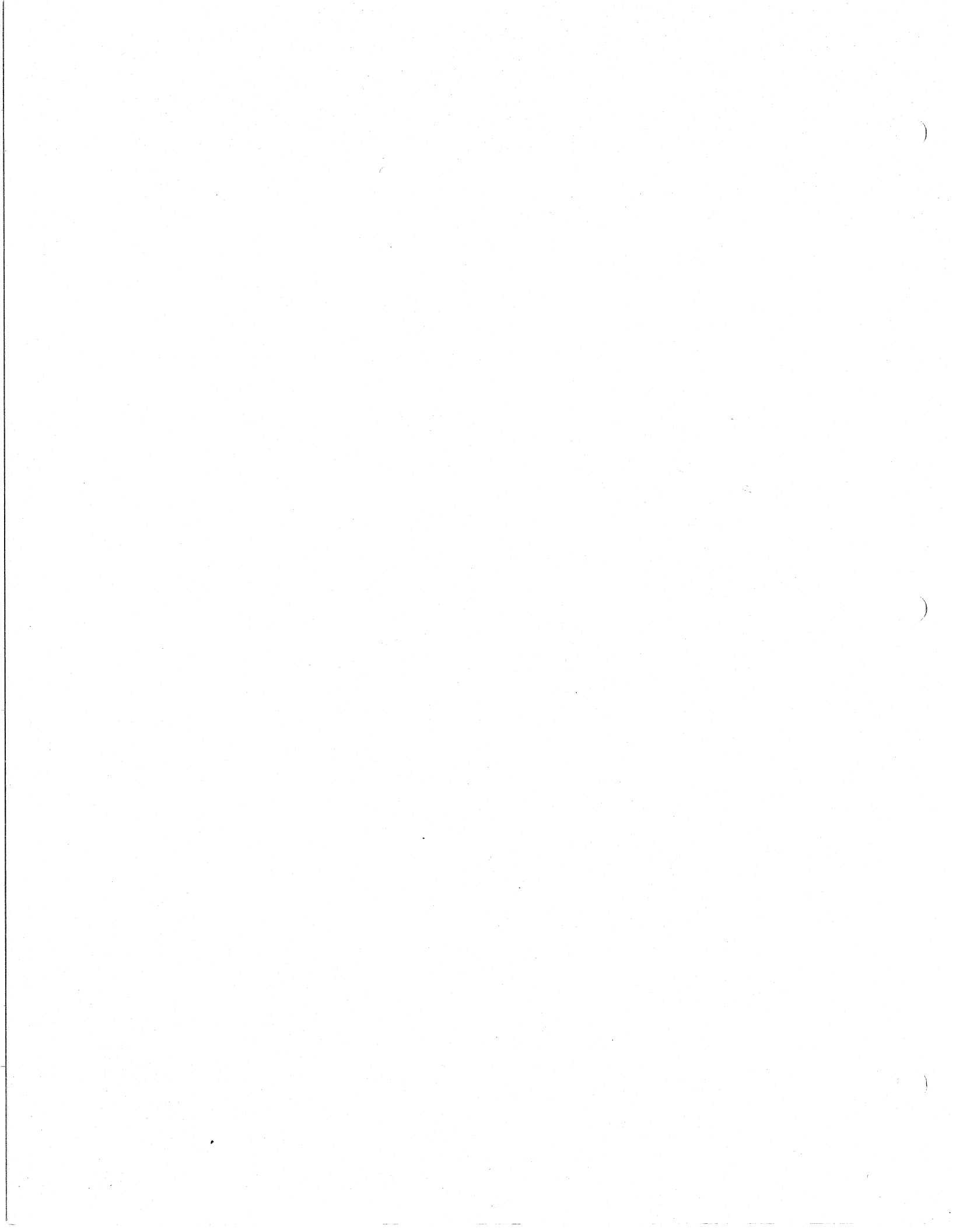
A B C D



## SPECIFICATIONS

### CP127 POWER SUPPLY

AC INPUT:	100/ 120/ 220/ 240 VAC $\pm$ 10% 47 - 63 Hz
DC OUTPUT:	+5 VDC @ 10A OVP at $6.2 \pm .4$ VDC +24 VDC @ 1.7A -12 VDC @ .5A (at 115 VAC INPUT)
LINE REGULATION:	$\pm$ .05% for a 10% line change
LOAD REGULATION:	$\pm$ .05% for a 50% load change
OUTPUT RIPPLE:	5 mv peak to peak maximum
OVER VOLTAGE PROTECTION:	Built-in on 5V output
TRANSIENT RESPONSE:	30 $\mu$ s for a 50% load change
SHORT CIRCUIT AND OVERLOAD PROTECTION:	Automatic current limit / foldback
STABILITY:	$\pm$ 5% for a 24-hour warm-up
TEMPERATURE RATING:	0°C to 50°C ambient at full load, derate linearly to 40% at 70°C
TEMP-COEFFICIENT:	$\pm$ .03% / °C maximum
EFFICIENCY:	Approximately 55% combined efficiency with a full load on all outputs at 115VAC line voltage



CP127 POWER SUPPLY TROUBLE SHOOTING GUIDE

SYMPTOM

POSSIBLE PROBLEM

UNIT OVER-HEATING

1. OUTPUT 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, 3 or 4 OPEN (CR5, 6, 7)\*
4. C1, 2, or 3 OPEN (C7, 6)\*
5. Q1, 2 or 3 OPEN (Q5, 5)\*
- × 6. R2 or R7 OPEN (R14)\*
7. SCR1 SHORTED OR OVP TRIGGERED

724V

723C

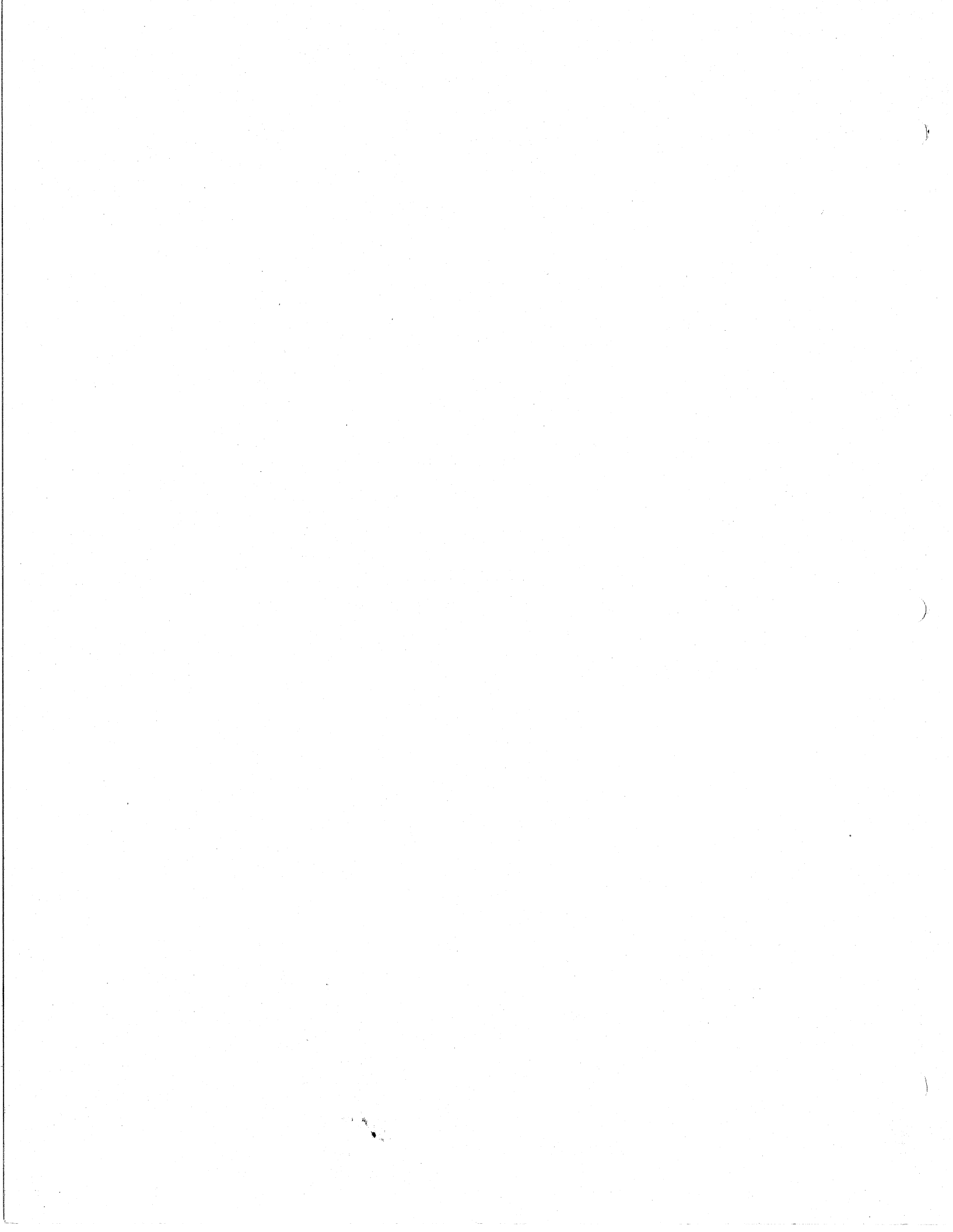
HIGH OUTPUT VOLTAGE  
AND RIPPLE, POOR  
REGULATION

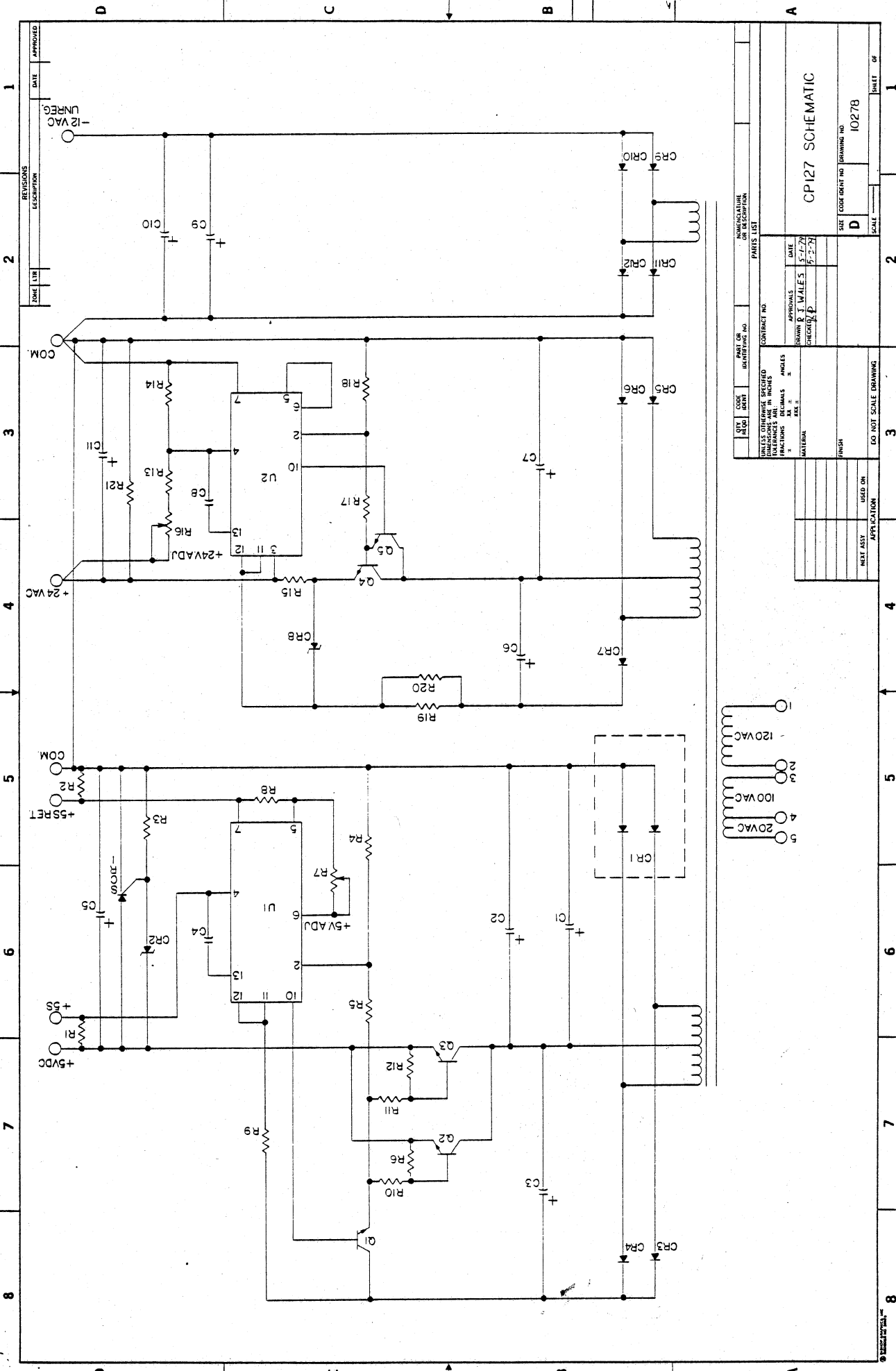
1. Q1, 2 or 3 SHORTED (Q4 or Q5)\*
2. U1 FAULTY (U2)\*
3. R1 OPEN (R13 or R16)\*

HIGH INPUT CURRENT  
BLOWS FUSE

1. IMPROPER INPUT VOLTAGE OR FREQUENCY
2. C1, C2 or C3 SHORTED (C7 or C6)\*
3. CR1, CR3 or CR4 SHORTED (CR5, CR6, CR7)\*
4. CR9 - 12 SHORTED
5. C9, 10 SHORTED

\*DENOTES 24V COMPONENTS





REVISIONS	DATE	APPROVED
1		

ZONE	LIB	DESCRIPTION

DATE	DESCRIPTION

PARTS LIST	
QTY	DESCRIPTION

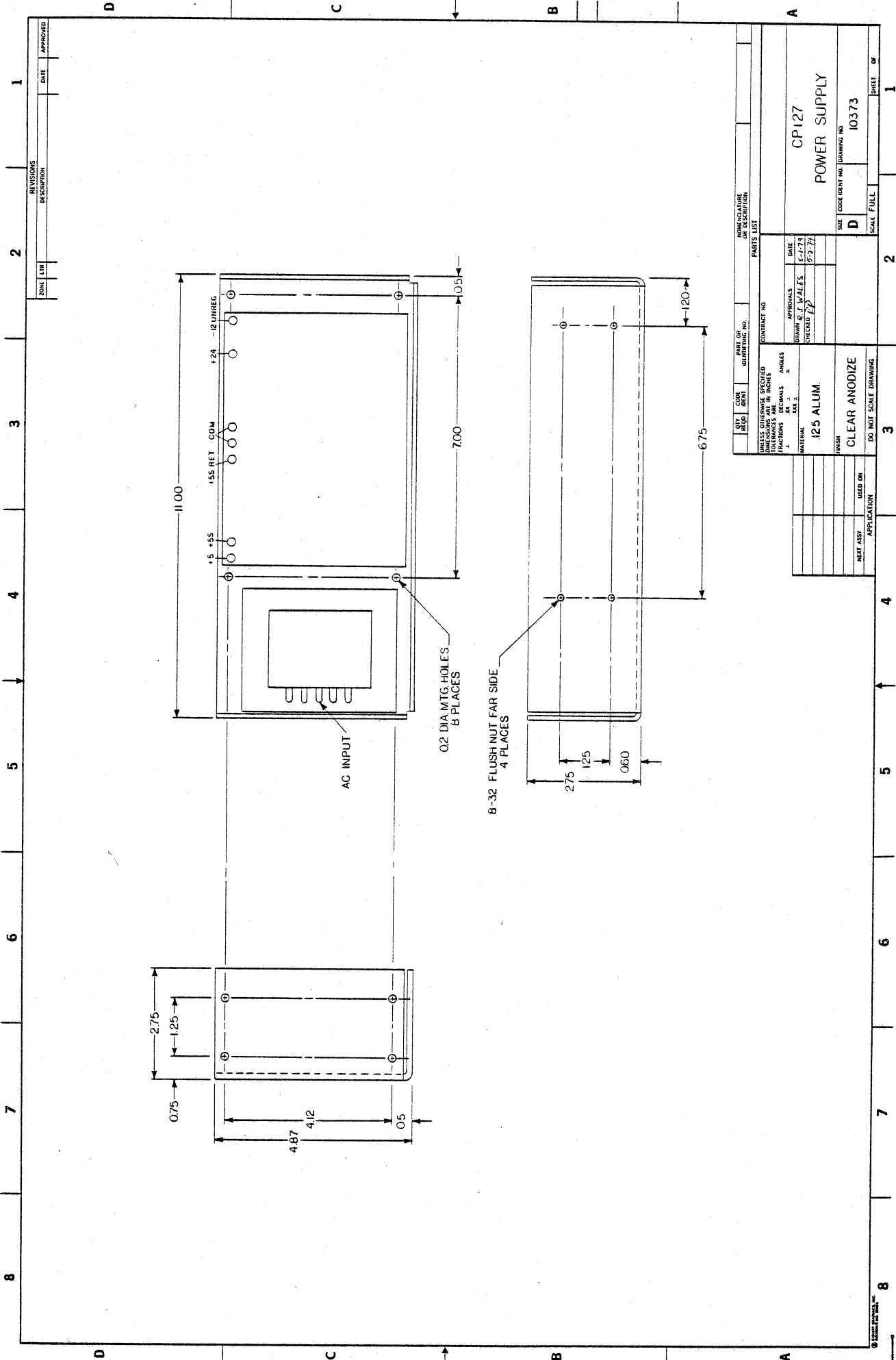
CONTRACT NO.	
DRAWN BY	
CHECKED BY	
DATE	

MATERIALS	
FINISH	
DO NOT SCALE DRAWING	
USED ON	
APPLICATION	

UNLESS OTHERWISE SPECIFIED	
DIMENSIONS ARE IN INCHES	
TOLERANCES ARE:	
FRACTIONS	
DECIMALS	
MATERIAL	

CPI27 SCHEMATIC	
SIZE	CODE IDENT NO
SCALE	DRAWING NO
D	10278

NO	OF
1	1



REV	DATE	DESCRIPTION	APPROVED

QTY	CODE	UNIT	DESCRIPTION	CONTRACT NO.	DATE
UNLESS OTHERWISE SPECIFIED			NOMENCLATURE		
DIMENSIONS ARE IN INCHES			PARTS LIST		
TOLERANCES ARE			DRAWN BY		
FRACTIONS 2			CHECKED BY		
DECIMALS 2			DATE		
ANGLES 2			5-7-74		
MATERIAL			SCALE FULL		
125 ALUM			DRAWING NO.		
FINISH			10373		
CLEAR ANODIZE			SHEET OF		
DO NOT SCALE DRAWING			1		

CP127  
POWER SUPPLY



**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 Select		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)	X
Z	In Use from Drive Select	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 of wire wrap pins which are shorted together.

*INCORRECT.*



CP127 PARTS LIST

<u>REF. DES.</u>	<u>CP127</u>	<u>DESCRIPTION</u>
C1, 2	16,000/15	CAPACITOR ALUM ELEC
C3	100/16	" " "
C5	470/16	" " "
C6, 11	47/50	" " "
C7	2,100/50	" " "
C9, 10	470/35	" " "
C4, 8	.001/100	CAPACITOR MYLAR FILM
CR 1	R 711 A	RECTIFIER 30A 100V TO -3
CR2, 8	IN 752 A	DIODE ZENER 300MW
CR3, 4, 7, 9-12	IN 4003	RECTIFIER 1A 200V
CR5, 6	MR 501	RECTIFIER 3A 100V
Q1-4	2N 3055	TRANSISTOR NPN POWER
Q5	TIP 31 A	TRANSISTOR NPN POWER
R1, 2, 6, 12, 9	6.8	RESISTOR $\frac{1}{2}$ W CF 5%
R3	47	" " " "
R4	2.2K	" " " "
R5, 14	750	" " " "
R8	1.5K	" " " "
R10, 11	3.9	" " " "
R13, 21	1.1K	" " " "
R17	330	" " " "
R18, 19, 20	10K	" " " "
R7, 16	1500	POTENTIOMETER ZW WW
R15	.12	RESISTOR ZW WW 10%
SCR 1	S0308LS3	SCR 8A 30V
U1, 2	723C	I.C. VOLTAGE REGULATOR





# Data Systems